An Application of Continuous Ant Colony Optimization for Multi-level Thresholding

Yun-Chia Liang†, Yueh-Chuan Yin‡, Chi-Chun Lao

Department of Industrial Engineering and Management, Yuan Ze University, Chungli, Taiwan 320, R.O.C.
Email: ycliang@saturn.yzu.edu.tw
s939512@mail.yzu.edu.tw
s975446@mail.yzu.edu.tw

Abstract. Ant colony optimization (ACO) was first proposed by Dorigo in 1992 as a multi-agent approach to solve difficult combinatorial problems. Many variants have been presented and employed to many different applications. Socha and Dorigo proposed an extension of ACO to continuous domains named ACO_R (K. Socha and M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research, 185(3), 1155-1173, 2008.) In this paper, we present a fast estimation algorithm based on ACO_R and the expectation-maximization (EM) algorithm for solving multi-level thresholding problems in image segmentation. The distribution of image intensity is modeled as random variables, which are approximated by a mixture of Gaussian distributions. The solution quality provided by the iterative EM algorithm is intensely affected by the initial solution fed into the EM algorithm. Therefore, we apply the ACO_R to provide a good initial solution for the EM algorithm. The EM algorithm is then used as a local search procedure to improve each new solution generated by ACO_R while fitting the mixture Gaussian model. The preliminary experiment results successfully show the effectiveness and efficiency of the ACO_R–EM algorithm.

Keywords: Ant Colony Optimization, Multi-level Thresholding, Expectation Maximization.

1. INTRODUCTION

Dorigo (1992) first introduced the Ant System (AS), the earliest version of the Ant Colony Optimization (ACO), in his dissertation. The behavior of artificial ants simulate the real ants. However, they differ in two important aspects. First, the artificial ants are not blind, i.e., they can “see” information regarding their environment; second, artificial ants have memory, such as a tabu list in a traveling salesman problem (TSP) application. ACO approaches have been successfully applied to different optimization problems such as the traveling salesman problem (TSP), the quadratic assignment problem (QAP), the vehicle routing problem (VRP), telecommunication networks, graph coloring, scheduling and so on (Dorigo et al., 1999).

In recent years, ACO approaches have become one of the most novel and promising metaheuristics to the discrete optimization problems.

Continuous optimization is hardly a new research field so there exist numerous algorithms developed for tackling this type of problems. Since the emergence of ACO as a combinatorial optimization tool, attempts have been made to use it for solving continuous problems. However, applying ACO algorithms to continuous domains was not an easy task, and the methods proposed often took inspiration from ACO, but did not follow ACO’s characteristics exactly (Socha and Dorigo, 2008).

Bilchev and Parmee (1995) first proposed an ant-inspired algorithm, Continuous Ant Colony Optimization (CACO), for the continuous optimization problems. In CACO, the ants start from a point, called a nest. It should be a point in the search space which seems promising for fine local search exploitation. The good solutions found are stored as a set of vectors, which originate in the nest. A search radius R will be defined, which determines the extent of the subspace to be considered in each generation (cycle). Then CACO sends ants in various directions at a radius not greater than R. Monmarché et al. (2000) presented another ant-inspired algorithm for continuous optimization problems, named API (after Pachycondyla APIcalis). API employed the foraging strategy of the Pachycondyla apicalis ponerin ants. These ants use relatively simple principles to search their preys, both from global and local viewpoint. Dréo and Siarry (2002)
proposed another ant-inspired algorithm, Continuous Interacting Ant Colony (CIAC). CIAC used two types of communication between ants: stigmergic information (spots of pheromone deposited in the search space) and direct communication between ants. The ants move through the search space being attracted by pheromone laid in spots, and guided by some direct communication between individuals. Note that these three ant-inspired algorithms did not follow the two core ideas of ACO, indirect communication by pheromone and incremental construction of solutions. Scoha and Dorigo (2008) presented a way to effectively apply ACO to continuous optimization problems without the need to make any major conceptual change to its structure and denoted the algorithm by ACO\textsubscript{R}. In this paper, we present a fast estimation algorithm based on ACO\textsubscript{R} and the expectation-maximization (EM) algorithm for solving multi-level thresholding problems in image segmentation.

The remainder of the paper is organized as follows: Section 2 describes the proposed ACO\textsubscript{R}-EM algorithm and their key features in detail. Section 3 presented the computational results on test images. Finally, the concluding remarks are provided in Section 4.

2. THE HYBRID ACO\textsubscript{R}-EM ALGORITHM

In practice, the frequency data observed from complex images intensity can be modeled as a random variable, which could be approximated by a mixture Gaussian model. In this paper, we hybridize ACO\textsubscript{R} with EM algorithm to develop an efficient and effective estimation algorithm for the multi-level thresholding problems. Since the performance of the EM algorithm is strongly affected by the initial solution, the ACO\textsubscript{R} algorithm will be employed as a global search procedure to provide a good initial solution for the EM algorithm. The EM algorithm is then used as a local search procedure to improve each new solution generated by ACO\textsubscript{R} while trying to seek the best-practice parameter estimates of the mixture Gaussian model. The general description of the proposed ACO\textsubscript{R}-EM algorithm is summarized in Fig. 1.

2.1 ACO\textsubscript{R} Algorithm

In ACO algorithms for discrete optimization problems solutions are constructed by sampling at each construction step a discrete probability distribution that is derived from the pheromone and heuristic information. The pheromone information represents the stored search experience of the algorithm and the heuristic information is usually determined by some simple rules. In contrast, ACO\textsubscript{R} utilizes a continuous probability density function (PDF). This density function is produced, for each solution construction, from a population of solutions that the algorithm maintains through the search process. Before the algorithm begins, the population is filled with random solutions, and this corresponds to the pheromone value initialization in discrete ACO algorithms. Then, at each iteration the set of generated solutions is added to the population and the same number of the worst solutions is removed from it. This action corresponds to the pheromone update in discrete ACO (Blum, 2005).

![Flow chart of the ACO\textsubscript{R}-EM algorithm.](APIEMS2009Dec. 14-16, Kitakyushu2160)

ACO\textsubscript{R} algorithm keeps track of a number of solutions in a solution archive \( T \). For each solution \( s_i \) to an \( n \)-dimensional problem, ACO\textsubscript{R} stores in \( T \) the values of its \( n \) variables and the corresponding objective function value \( f(s_i) \). The \( i^{th} \) variable of the \( j^{th} \) solution is hereby denoted by \( x_{i,j} \). The structure of the solution archive \( T \) is presented in Figure 2.
At each construction step \(i=1, \ldots, n\), an ant chooses a value for decision variable \(s_i\). In other words, if there are \(n\) dimensions for the given optimization problem, an ant chooses in each of \(n\) steps a value for exactly one of the dimensions. An ant uses a Gaussian kernel, which is a weighted superposition of several Gaussian functions, as PDF. The Gaussian kernel \(G^s\) is given as follows:

\[
G^s(s) = \sum_{i=1}^{k} \omega_i \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(s - \mu_i)^2}{2\sigma_i^2}\right), \quad \forall s \in \mathbb{R},
\]

where the \(i^{th}\) Gaussian function is derived from the \(l^{th}\) member of the population, whose cardinality is at all times \(k\). Note that \(\omega\) is the vector of weights, whereas \(\mu\) and \(\sigma\) are the vectors of means and standard deviations respectively. Each ant, before choosing a value for the first dimension, chooses exactly one of the Gaussians \(l\), which is then used for all \(n\) construction steps. The choice of this Gaussian function, denoted by \(l^*\), is performed with probability in equation (2)

\[
p_l = \frac{\omega_l}{\sum_{r=1}^{k} \omega_r}, \quad \forall l = 1, \ldots, k
\]

where \(\omega_l\) is the weight of Gaussian function \(l\), which is obtained by the following procedure. All solutions in the population are ranked according to their quality with the best solution on the top, i.e. with the rank of 1. Assuming the rank of the \(i^{th}\) solution in the population to be \(l\), the weight \(\omega_l\) of the \(i^{th}\) Gaussian function is calculated according to the following formula:

\[
\omega_l = \frac{1}{qk \sqrt{2\pi}} \exp\left(-\frac{(l-1)^2}{2q^2 k^2}\right)
\]

which defines the weight to be a value of the Gaussian function with argument \(l\), with a mean of 1.0, and a standard deviation of \(qk\), where \(q\) is a parameter of the algorithm. The value of \(q\) controls the preference of the solutions in the list. If the value of \(q\) is small, the best-ranked solutions are strongly preferred, and in case it is larger, the probability of choosing each solution becomes more uniform.

Before performing the sampling procedure, the mean and the standard deviation of the \(l^{th}\) Gaussian function must be specified. First, the value of the \(i^{th}\) decision variable in solution \(l^n\) is chosen as the mean, denoted by \(\mu_i\), of the Gaussian function. Second, the average distance of the other population members from the \(l^{th}\) solution multiplied by a parameter \(\rho\) is selected as the standard deviation, denoted by \(\sigma_i\), of the Gaussian function as follows:

\[
\sigma_i = \rho \sum_{l=1}^{k} \left|\frac{s_i - \mu_i}{k-l}\right|
\]
histogram into different groups. Then, the priori probability and standard deviation of each group can be calculated and saved as the initial solutions in the archive $T$.

2.2 Expectation-Maximization (EM) Algorithm

The EM was first introduced by Dempster et al. (1977) as a mean of fitting incomplete data and is sensitive to the initial values of the parameters estimated. Given a normalized multimodal histogram, $h(j)$ of an image, where $j$ represents a gray level over a given range $[0, L-1]$ and $L$ is the total number of gray levels, we address the problem of finding the optimal threshold(s) for use in separating the modes. The approach for finding the best threshold(s) of that image is to fit the histogram with the sum of $d$ probability density functions (PDFs). If the PDFs are Gaussian, the model has the following form (Synder and Bilbro, 1990):

$$h(j) = \sum_{i=1}^{d} \phi_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(j-\mu_i)^2}{2\sigma_i^2}\right]$$  \hspace{1cm} (5)

And

$$\sum_{i=1}^{d} \phi_i = 1$$  \hspace{1cm} (6)

where $\phi_i$ represents the a priori probability, $d$ is the number of thresholding levels, $\mu_i$ denotes the mean and $\sigma_i^2$ is the variance of mode $i$. A PDF model must be fitted to the histogram data, typically by the maximum likelihood or mean-square error approach in order to locate the optimal thresholds. Given the histogram data $p_j$ ( $p_j = n_j / N$ , where $n_j$ is the occurrence of pixels with gray level $j$ and $N$ denotes the total number of the pixels in the image), we wish to find the set of parameters, $\Omega = \{\phi_i, \mu_i, \sigma_i; i = 1, 2, ..., k\}$, defining the mixture Gaussian PDFs and the probabilities which minimizes the fitting error as follows:

$$\text{Minimize } HD = \sum_j \left[ p_j - h(j, \Omega) \right]^2$$  \hspace{1cm} (7)

where $HD$ is the objective function, i.e., Hamming Distance, to be minimized with respect to $\Omega$. The EM algorithm consists of expectation and maximization steps. The expectation step involves estimating a mixture distribution using current parameter values. The maximization step computes new parameter values that optimize the expected value of their data likelihood. This two-stage process is iterated to convergence (Leite and Hancock, 1997).

The EM algorithm is summarized as shown below:

**E-step:** Compute the probability of density $i$ that generates gray level $j$ by equation (8).

$$P_{ij}^{EM} = \frac{\phi_i^{(l)}h(j; \mu_i^{(l)}, \sigma_i^{(l)})}{\sum_{m=1}^{d} \phi_m^{(l)}h(j; \mu_m^{(l)}, \sigma_m^{(l)})}$$  \hspace{1cm} (8)

where $l$ is the iteration number.

**M-step:** Update parameters $\phi_i$, $\mu_i$, $\sigma_i$ for the mixed Gaussian distribution.

$$\phi_i^{(l+1)} = \frac{1}{N} \sum_{j=0}^{L} P_{ij}^{EM} \cdot n_j$$  \hspace{1cm} (9)

$$\mu_i^{(l+1)} = \frac{\sum_{j=0}^{L} P_{ij}^{EM} \cdot j \cdot n_j}{\sum_{j=0}^{L} P_{ij}^{EM} \cdot n_j}$$  \hspace{1cm} (10)

$$\sigma_i^{(l+1)} = \sqrt{\frac{\sum_{j=0}^{L} P_{ij}^{EM} \cdot (j - \mu_i^{(l+1)})^2 \cdot n_j}{\sum_{j=0}^{L} P_{ij}^{EM} \cdot n_j}}$$  \hspace{1cm} (11)

where $N$ denotes the total number of the pixels in the image, $n_j$ is the occurrence of pixels with gray level $j$. According to the EM algorithm, the final parameter estimates are obtained staring from a set of initial values $\Omega^0 = \{\phi_i^0, \mu_i^0, \sigma_i^0; i = 1, 2, ..., k\}$ and then iterating the above equations until convergence (Bazi et al., 2007).

3. COMPUTATIONAL RESULTS AND ANALYSIS

In this section, the performance of the proposed ACO$_R$-EM algorithm has been evaluated. The ACO$_R$-EM algorithm was implemented on a personal computer with Intel(R) core(TM) 2 Quad CPU Q9400 2.66 GHz, 2.67 GHz, and 3.0 GB RAM. The program was compiled and run using Borland C++ Builder 6.0. Two test images
(Lena and Cameraman) were taken under natural room lighting without the support of any special light source and have been transformed into gray-scale images of 256×256 pixels. That is these images including a collection of pixels have been assigned values from 0 to 255 in this study. Table 1 shows the maximum and minimum values of gray-level in two test images, the number of peaks and their gray level values selected from the histogram of the two test images, respectively. Note that the values of peaks are arranged in the order of frequency, i.e., from the most to the least frequent among peaks. Figure 3(a), (c), (e), and (g) illustrate the original images and histograms of Lena and Cameraman images respectively.

To account for both efficiency and effectiveness, the best parameter combination for ACO\textsubscript{R} is set up from the preliminary experiments as follows: \( q = 0.0009, \rho = 0.67 \) for all the following experimental runs. The best population size \((k)\) of the ACO\textsubscript{R} is set at 35 and the number of ants for each iteration is set at 8. The stopping criterion for ACO\textsubscript{R}-EM algorithm is the maximum number of iterations equal to 50 and for EM algorithm is the absolute difference of the likelihood function value over two successive iterations less than \(10^{-6}\).

Table 1: Search ranges and the gray-level of peaks on two test images

<table>
<thead>
<tr>
<th>Images</th>
<th>Gray-level of pixels</th>
<th>Number of peaks</th>
<th>Gray-level of peaks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimum value</td>
<td>Maximum value</td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>26</td>
<td>242</td>
<td>12</td>
</tr>
<tr>
<td>Cameraman</td>
<td>7</td>
<td>253</td>
<td>16</td>
</tr>
</tbody>
</table>

| Lena       | 154,143,128,48,99,113,172,177,209,195,75,234 |
| Cameraman  | 14,162,181,150,130,113,27,43,35,64,58,74,213,238,225,233 |

Table 2: Best curve fitting results of Lena and Cameraman images

<table>
<thead>
<tr>
<th>Image</th>
<th>Lena</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Mixture Gaussian</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>5.53e-05</td>
<td>2.53e-04</td>
</tr>
<tr>
<td>Thresholds</td>
<td>63,88,115,115,124,126,140,166,175,183,191</td>
<td>20,27,46,54,60,79,87,109,150,153,172,197,209,220</td>
</tr>
<tr>
<td>CPU time (in seconds)</td>
<td>1.02</td>
<td>1.44</td>
</tr>
</tbody>
</table>
In the first experiment, we selected 12 peaks from the original histogram of Lena image with the initial solution generate procedure (see table 1) and performs the ACO$_R$-EM algorithm to form the thresholds from bi-level to 12-level. Figure 4 shows the convergence of objective value (i.e., Hamming distance) and the best fitting appeared at the case of 12-level. Figure 3(f) illustrates the experimental results of 12-level mixture Gaussian model curve fitting to Lena histogram. From Table 2, we can also find that the ACO$_R$-EM algorithm produces extremely accurate estimate result for Lena image with very small Hamming distance of merely 5.53e-05 and the CPU time needs just 1.02 seconds. Figure 3(b) shows the thresholding result of 12-level curve fitting, all the features of Lena image are perfectly identified and rehabilitated.

In the second experiment, we selected 16 peaks from the original histogram of Cameraman image with the initial solution generate procedure (see table 1) and performs the ACO$_R$-EM algorithm to find the results from bi-level to 16-level. Figure 5 shows the convergence of objective value and the best fitting appeared at 15-level curve fitting. As shown in Figure 5, the objective value drops quickly at tri-level fitting but the Hamming distance fluctuates between 2e-04 and 6e-04 and decreases slowly. Figure 3(h) illustrates the experimental result of 15-level mixture Gaussian model curve fitting to Cameraman histogram. From table 2, we can find the ACO$_R$-EM algorithm also produces accurate estimate result for Cameraman image with a small Hamming distance of merely 2.53e-04 and the CPU time is just 1.44 seconds. Figure 3(d) demonstrates the thresholding result of 15-level curve fitting, all the features of Cameraman image are also perfectly identified and rehabilitated.

Based on the experimental results of the pervious two test images, we can conclude that the proposed ACO$_R$-EM algorithm is an efficient and effective estimation algorithm.

4. CONCLUSIONS

In this paper, we present a fast estimation algorithm based on the hybridization of ACO$_R$ and the EM algorithm for solving multi-level thresholding problems in image segmentation. Since the initial values of the statistical parameters of classes strongly affect estimates obtained by the EM algorithm at both convergence and the thresholding results, it is important to adopt an accurate and robust initialization procedure, which can properly explore the space of solutions. In this study, we successfully introduced the ACO$_R$ as a global search procedure to provide a good initial solution efficiently for the consecutive EM algorithm. The EM algorithm is then used as a local search procedure to improve the new solutions generated by ACO$_R$ while fitting the mixture Gaussian model. Experimental results of the two test images show that the ACO$_R$-EM algorithm is an efficient and effective estimation algorithm; moreover, the ACO$_R$-EM algorithm can also be considered as a promising and valuable multi-level thresholding technique for the on-line application in practice.

ACKNOWLEDGMENT

This work was partially supported by National Science Council of the Republic of China (NSC 97-2221-E-155-043-MY2).

REFERENCES


Leite, J.A.F., and Hancock, E.R. (1997) Iterative curve organization with the EM algorithm, **Pattern Recognition** 18, 143-155.


**AUTHOR BIOGRAPHIES**

**Yun-Chia Liang** received his first M.S. (Mechanical Engineering, 1996) from Carnegie Mellon University, a second M.S. (Industrial Engineering, 1999) from University of Pittsburgh, and the Ph.D. in Industrial and Systems Engineering from Auburn University (2001). His research interests are in application of meta-heuristics to optimization problems. Dr. Liang has authored over 60 publications in books, refereed proceedings and journals including articles in *IEEE Transactions on Reliability, International Journal of Production Research, Computers and Operations Research, Computers and Industrial Engineering, European Journal of Operational Research*. Dr. Liang is currently an Associate Professor in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. He is a member of IEEE, CIIE and ORSTW. His email address is <ycliang@saturn.yzu.edu.tw>

**Yueh-Chuan Yin** is currently a Ph.D. student in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. His research interests include image processing, logistics management, metaheuristics and combinatorial optimization. His email address is <s939512@mail.yzu.edu.tw>

**Chi-Chun Lao** is currently a graduate student in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. His research interests include production scheduling, and combinatorial optimization. His email address is <s975446@mail.yzu.edu.tw>