ICIC Express Letters
Part B: Applications
Volume **7**, Number **8**, August **2016**
ICIC International ©2016 ISSN 2185-2766
pp. 1651–1656

OTSU MULTI-THRESHOLDING BASED ON DCG3A FOR MEDICAL IMAGE

Zheng Fang[1,2], Xuanjing Shen[1,2], Haipeng Chen[1,2,*] and Yuncong Feng[1,2]

[1]College of Computer Science and Technology
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education
Jilin University
No. 2699, Qianjin Ave., Changchun 130012, P. R. China
[*]Corresponding author: chenhp@jlu.edu.cn

Received January 2016; accepted April 2016

Abstract. *In order to solve the problem that multi-thresholding segmentation spends too much time finding the optimal solution in medical image segmentation, Otsu multi-thresholding based on dynamic combination of genetic algorithm and ant algorithm (abbreviated as DCG3A-OM) was proposed in this paper. Firstly, the algorithm uses genetic algorithm to generate preliminary partition results, and then converts them into the initial information of ant colony. Finally ant algorithm will approach to optimal values efficiently. While running genetic algorithm, the conditions of combination will be analyzed dynamically to avoid stopping genetic algorithm early or late. Besides, it increases the probability of variation and decreases the numbers of individuals which are calculated for the evolution rate of colony. The experimental results demonstrate that DCG3A-OM gets the optimal solution more quickly and efficiently than the others, and it can always get the optimal values mostly.*
Keywords: Multi-thresholding segmentation, Medical image segmentation, Dynamic combination

1. **Introduction.** Segmentation of medical image is the first, and toughest, challenge to medical science. The quality of segmentation will affect the operations of subsequent image registration and fusion. In the process of clinical diagnosis, surgery and radiotherapy, technology of medical image segmentation exhibits increasingly important clinical value [1]. Otsu's method has been the most popular and basic segmentation technology because of the stable capability and simple implementation [2-4]. With the increasing number of thresholds, the execution time of the algorithm will show an exponential tendency, so the application of multi-thresholding segmentation for medical image is still scarce.

It is common for genetic algorithm (GA) to converge prematurely and the capability of GA is poor as it is searching in certain scope [5,6]. Xiong et al. compared GA with ant colony optimization (ACO) in the aspect of constringency speed, and then proposed a new method named dynamic combination of genetic algorithm and ant algorithm (DCG3A) [7]. In this paper, we proposed a new method combining DCG3A with Otsu multi-thresholding firstly. It will initialize the colony by the global search capability of GA stochastically and approach to optimal solution to narrow the search scope. Then it will find optimal solution with the local search capability and positive feedback mechanism of ACO. The experimental results demonstrate that DCG3A-OM runs more quickly than the others, and it can always get the optimal values, because it can do global and local searching quickly.

The rest of the paper is organized as follows. Section 2 introduces Otsu method. Section 3 discusses strategy of DCG3A. Section 4 describes the steps of algorithm. Section 5 reveals the experimental results followed by concluding remarks.

2. **Otsu.** Implementation of Otsu's method in a multi-level framework requires an exhaustive search for determining the optimal set of thresholds while maximizing the between-class variance [4]. ω_i is the proportion of class i, μ_i is the average value of class i, μ is the average value of all pixels, and the between-class variance will be:

$$\sigma^2 = \omega_0 (\mu - \mu_0)^2 + \omega_1 (\mu - \mu_1)^2 + \cdots + \omega_{m-1} (\mu - \mu_{m-1})^2 \tag{1}$$

If the values of k_1, k_2,…,k_{m-1} are as the following:

$$k_1, k_2, \ldots, k_{m-1} = Arg \left\{ \max \left[\delta^2(k_1, k_2, \ldots, k_{m-1}) \right] \right\}, \tag{2}$$

then k_1, k_2,…,k_{m-1} will be a group of optimal values.

3. **Strategy of DCG3A.** GA is a kind of random search algorithm which benefits from the choice and genetics of nature [5,6]. ACO is proposed by M. Dorigo to solve the travelling salesman problem originally, and Wang is the first one that combined ACO with Otsu's method, and gave an example about single threshold [8,9]. In the work of [7], Xiong et al. did some experiments to compare the speed of ACO with GA. The result is shown in Figure 1.

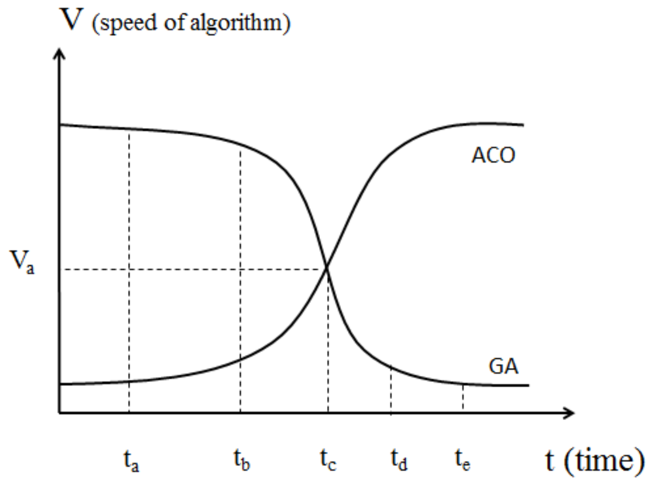


Figure 1. Speed-time curves of genetic algorithm and ant algorithm

As shown in Figure 1, the speed of GA is higher than ACO from t_b to t_a, because GA has stronger global search capability and the pheromones of ACO are not enough. From t_a to t_e, the speed of ACO is higher than GA oppositely, because GA has poor local search capability with much iteration, but ACO has enough pheromones to search. We can see that t_a is the best time to merge GA with ACO. What we should do is to find out t_a and execute GA before t_a. In this paper, the conditions of combining GA with ACO are as follows.

1) Maximum execution generation of GA. (*genmax*)

2) Evolution rate of colony. (*rate*)

3) The maximum number of continuous generations is *gen* when evolution rate is lower than *rate*. (*gen*)

4) When the number of current generation is higher than *genmax*, or *gen* is higher than the one we suppose, GA will stop.

4. **DCG3A-OM.** DCG3A-OM executes GA firstly but it is easier to do variation for GA at the end of evolution in that we increase the probability of variation to expand the variety and randomness. It will stop to execute ACO when it meets the conditions. Apart from this, it will decrease the calculation by changing the number of individuals from N

to M ($M < N$). Those M individuals are all the biggest of current colony. In this paper, we will show the steps of DCG3A-OM about two thresholds.

GA OF DCG3A-OM

1) Coding: it uses binary numbers. There are 256 gray levels in each image, $256 = 2^8$. For that reason, 8 bits will represent a pixel and 16 bits will be used as two thresholds.

2) Decoding: the first half and end half of the 16 numbers will be converted into two numbers (from 0 to 255) as two parameters of Equation (1).

3) Individual evaluation: it is the same as Equation (1).

4) Initialization: it initializes the colony with random numbers which consists of 0 and 1.

5) Evolution

(1) Selection: elite strategy (maintaining the optimal individual in the next generation) and comparison strategy (comparing each one to others) are the two important strategies.

(2) Crossover: it exchanges numbers with the others from one point to another point.

(3) Variation: it does variation at basic bit (changing 0 to 1 or 1 to 0). It will increase the probability of variation at the end of evolution.

6) Analyzing termination conditions 1)-4): it analyzes *gen* and *rate*. Compare current generation with *genmax*. While it meets the condition 4), GA will stop.

ACO OF DCG3A-OM

1) Initialization: it converts the biggest M individuals of GA into ACO's colony. Thresholds will be stored in two arrays named $ant1[M]$ and $ant2[M]$. The pheromones, stored in an array named $T[i]$, are equal to the result, calculated with Equation (1).

2) Individual evaluation: it is the same with GA.

3) Evolution

(1) Probability of diversion: it will calculate the probability of diversion (denoted by $ant\_prob[i]$) for each ant. Let optimal ant of current generation be *Max*, and

$$ant\_prob[i] = (Max - T[i])/Max \tag{3}$$

(2) Diverting location:

If ($ant\_prob[i] < P_0$), then

$$temp1 = ant1[i] + \min\_step * rd1; \quad temp2 = ant2[i] + \min\_step * rd2 \tag{4}$$

Else:

$$temp1 = ant1[i] + \max\_step * rd3; \quad temp2 = ant2[i] + \max\_step * rd4 \tag{5}$$

P_0 is the minimum probability to divert, *min_step* is the minimum local diversion length and *max_step* is the maximum global diversion length. Actually, $rd1$, $rd2$, $rd3$ and $rd4$ are all random numbers $[-0.5, 0.5]$, thus $temp1$ and $temp2$ are from 0 to 255.

(3) Updating pheromones: it compares the result, calculated with Equation (1) about $temp1$ and $temp2$, with $T[i]$. If ($COST(temp1, temp2) > T[i]$):

$$ant1[i] = temp1; \quad ant2[i] = temp2 \tag{6}$$

$$T[i] = (1 - p_1) * T[i] + COST(temp1, temp2) \tag{7}$$

$COST$ is calculated with Equation (1), P_1 is the evaporation rate of pheromone.

4) Analyzing termination condition: the maximum generation to run is *genmax_ant* (15). If the optimal ant does not change in continuous 5 generations, it will stop. Finally, $ant1[i]$ and $ant2[i]$, corresponding to $T[i]_{\max}$, are two thresholds.

5. **Experimental Result and Analysis.** The experiment environment is Intel (R) Celeron (R) cpu, E3400 @2.60GHz memory3G and C++ in Microsoft Visual Studio 2010. All the images come from a website (http://www.med.harvard.edu/aanlib/cases/caseNA /pb9.htm) and we take an image named SLICE 32 to show the results of Otsu, Otsu

based on GA and DCG3A-OM. Besides, there are some results about two thresholds and three thresholds.

We did it 30 times to get the values of *genmax*, *rate* and *gen*, and after that we got something shown in Figure 2. The value of *Total_Cost* is the total cost of colony.

$N = 12$. Current generation is denoted by *Generation*. In this experiment, the value of *Total_Cost* increases slowly after *Generation* is equal to a ($a < 15$). The steps as the following are what we did to get the value of *rate*:

$$Total\_Cost = c * 10^4 \tag{8}$$

$$\frac{a}{Total\_Cost} = d * 10^{-3} \tag{9}$$

The value of a, b, c are from 1 to 10 stochastically. Therefore, we supposed that: $rate = 0.002$, $genmax = 15$, and $gen = 3$. While ($Generation > genmax$ or $gen > 3$), GA will stop.

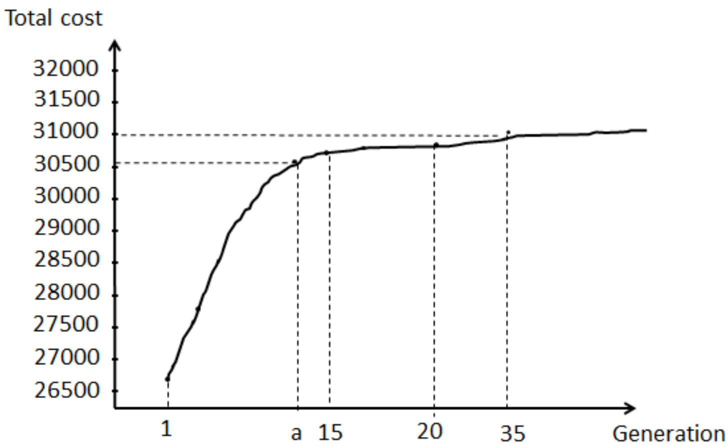
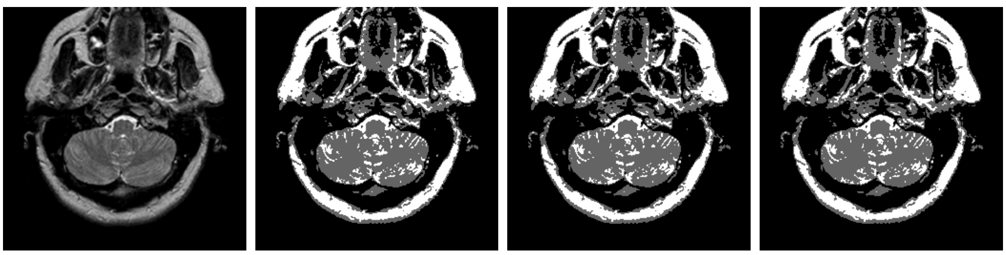
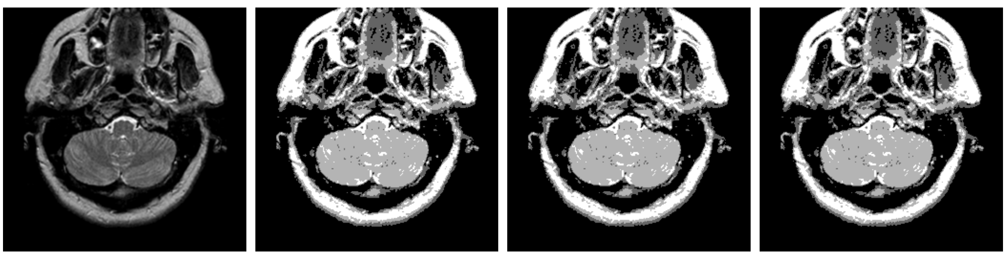


FIGURE 2. Total_Cost – Generation



(a) SLICE 32 (b) Otsu (c) Otsu – GA (d) DCG3A-OM

FIGURE 3. Dual-threshold segmentation



(a) SLICE 32 (b) Otsu (c) Otsu – GA (d) DCG3A-OM

FIGURE 4. Three-threshold segmentation

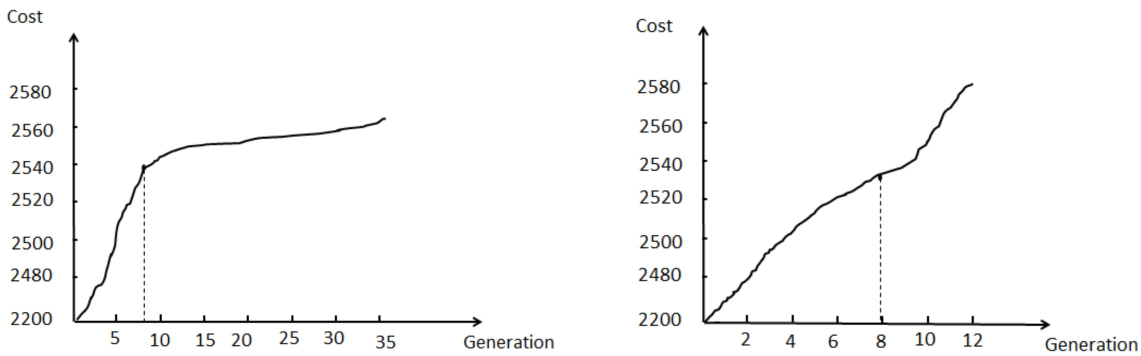
In Figure 3, it shows the results about dual-threshold, and the thresholds of them are all (43*112). In Figure 4, it shows the results about three-threshold, and the thresholds of them are all (23*69*122).

In the experiment, all of them are related to Otsu, and they got the same thresholds, so the segmentation images have no distinction. However, the speed of DCG3A-OM is the highest among them. We compared the speed of GA with DAG3A-OM about two thresholds. The value of GA's *genmax* is 50, and the parameters of DCG3A-OM are the same as before $N = 12$, $M = 6$. The result is shown in Figure 5. Especially, the value of *COST* is equal to the value of optimal individual in current colony.

In Figure 5(a), optimal individual in current colony varies as *Generation* increases. When *Generation* changes from 8 to 35, the value of *COST* gets a little increase (from 2539.7 to 2572.3). Especially, it does not get the optimal solution. The reason why it slows down is that GA has poor local search capability. In Figure 5(b), when *Generation* is 8, it tends to slow down. At the moment, it meets the condition to stop GA. Then the information of GA is converted into ACO's colony and executes ACO to find the optimal solution in the certain scope until the value of *Generation* is equal to 12. After that it will maintain the tendency. Therefore, DCG3A-OM has strong local and global search capability, and it can get optimal values quickly and correctly. The algorithm utilizes the advantages of the two algorithms and overcomes their disadvantages.

Due to randomness of GA and DCG3A-OM we did 30 times experiments and averaged each value with SLICE 32. The result is shown in Table 1.

As shown in Table 1, DCG3A-OM can get optimal solution quickly and exactly, and it will save much more time. When the number of thresholds changes from 2 to 3, the



(a) GA (b) DCG3A-OM

Figure 5. Speed-value curve of GA and DCG3A-OM

Table 1. Results of these algorithms

The number of thresholds	Test	Otsu	Otsu based on GA	DCG3A-OM
2	Optimal threshold	43,112	43,112	43,112
	Average cost	2586.65	2581.63	2585.47
	Running time(S)	0.128	0.043	0.009
3	Optimal threshold	43,112	43,112	43,112
	Average cost	2586.65	2586.65	2586.65
	Running time(S)	35.158	0.305	0.087

running time of Otsu changes from 0.128 to 35.158, but it is from 0.009 to 0.087 for DCG3A-OM. Besides, DCG3A-OM's average cost is better than GA and close to Otsu's mostly.

6. **Conclusions.** Image segmentation is a traditional and challenging problem. We use DCG3A to save more time for Otsu multi-thresholding. DCG3A-OM is better than both Otsu and Otsu based on GA. It can save much more time than Otsu and its optimal solution is close to Otsu's mostly. It can get the same optimal solution sometimes because of randomness. With increase of the number of thresholds, DCG3A-OM can still save much time and get what we want. Although it gets a good result, we intend to improve it. Because it only saves much time, we try to make it get a good segmentation. If the algorithm makes it, it will be applied more widely.

Acknowledgment. This research is supported by the National Natural Science Foundation of China for Youths (No. 61305046), Jilin Province Science Foundation for Youths (No. 20130522117JH), and the Natural Science Foundation of Jilin Province (No. 2014010 1193JC).

REFERENCES

[1] G.-P. Jiang, W.-J. Qin, S.-J. Zhou and C.-M. Wang, State-of-the-art in medical image segmentation, *Jisuanji Xuebao (Chinese Journal of Computers)*, vol.38, no.6, 2015.

[2] Z. Y. He, L. N. Sun and L. G. Chen, Fast computation of threshold based on Otsu criterion, *Dianzi Xuebao (Acta Electronica Sinica)*, vol.41, no.2, pp.267-272, 2013.

[3] C. D. Fan and Y. J. Zhang, Improved Otsu method based on histogram oblique segmentation for segmentation of rotary kiln flame image, *Journal of Image and Graphics*, vol.40, no.11, pp.2480-2489, 2014.

[4] Y. Wang and J. Chen, Improvement of two-dimensional Otsu algorithm and its application, *Computer Simulation*, no.4, 2008.

[5] Y. Ma and W. Yun, Research progress of genetic algorithm, *Application Research of Computers*, vol.29, no.4, 2012.

[6] J. Zhuang, Q. Yang, H. Du and D. Yu, High efficient complex system genetic algorithm, *Journal of Software*, vol.21, no.11, pp.2790-2801, 2010.

[7] Z. Xiong, S. Li and J. Chen, Hardware/software partitioning based on dynamic combination of genetic algorithm and ant algorithm, *Journal of Software*, vol.16, no.4, pp.503-512, 2005.

[8] S. Wang, Y. Huang and D. Li, Multilevel thresholding methods for image segmentation with improved-Otsu based on ant colony algorithm, *Microcomputer Applications*, vol.29, no.4, 2008.

[9] Y. M. Xia, B. Cheng, J. L. Chen et al., Optimizing services composition based on improved ant colony algorithm, *Jisuanji Xuebao (Chinese Journal of Computers)*, vol.35, no.2, pp.270-281, 2012.