Ant Colony Optimization Variants in Image Edge Detection

Susmita Koner and Sriyankar Acharyya

***Abstract*—Edges in an image are the curves consisting of pixels wherein both side contains pixels with non-uniform intensity. Edge detection is a part of low level image processing, much needed in various fields. Though edge detection can be done by various derivative techniques but it can also be detected well using meta-heuristic approximation algorithms. Ant Colony Optimization (ACO) is such a meta-heuristic technique to solve it. In basic ACO which comprises five phases: Initialization, Construction, Updation, Decision and Visualization, we have proposed and implemented total eight variations in this paper by modifying initialization and construction phase. In the initialization phase we have given a constraint in one variant that ants will be initialized near to edge to eliminate useless construction steps and unwanted edge detection where the other variant is without this constraint which may generate unnecessary edges in the resulting image. We have taken other two variations in selecting the next pixel in the construction phase: in one Greedy method is used, in another Roulette wheel selection method is used. Apart from these, in this phase two more variations have been done depending on memory size of ants i.e. applying tabu list memory of ants and ants without memory. Hence on the basis of two types of selection method used, two types of memory size of ants and two types of initialization phase, we have implemented eight variations individually in this paper. We observe that the variant, with roulette wheel selection, incorporated with the tabu list memory of ants, and with the new initialization condition outperforms others.**

***Index Terms*— Ant Colony Optimization, Edge detection, greedy selection method, metaheuristic, pheromone, roulette wheel selection method, tabu list.**

I. Introduction

EDGES, in an image, are the set of pixels that have high intensity contrast with respect to their neighbor pixels. Detection of edges is a preliminary step of image processing. Several years ago this job was done by using the pattern fitting approach or by applying some derivative operators like Sobel operators, Roberts cross operator, Canny's edge detector etc. Then different metaheuristic soft computing approaches were proposed and they are proved to detect edges better than those.

F. Susmita Koner, PG Student, West Bengal University of Technology, West Bengal, India (e-mail: susmita.koner@gmail.com).

S. Sriyankar Acharyya, West Bengal University of Technology, West Bengal, India.(e-mail: srikalpa8@gmail.com).

978-1-4799-3358-7/14/$31.00 ©2014 IEEE

Ant Colony Optimization (ACO), a meta-heuristic type optimization technique, proposed first by Marco Dorigo [1], which is based on the behavior of real natural ants that find the shortest path from source to destination, comprises five phases: initialization, construction, updation, decision and visualization phase.

Many research have been done on edge detection by using ant colony optimization with variants applied [2]-[5]. Likewise in this paper we have applied two variations in the selection method of the construction phase where one uses Roulette wheel method and another uses Greedy method for selecting next pixel to move. Besides this, another two variations depending on ant's memorizing power of already visited pixels has been proposed and implemented. In one of them ants have tabu list memory i.e. it will not select the already visited pixel again. In the other the ants are memory less. Apart from these two other variants propositions are made, that are with constraint imposition that ants will be initialized near to edge and without applying the constraint at the time of initialization of ants in the initialization phase. The imposition of the condition eliminates unnecessary edges in the resulting image. Therefore on the basis of selection method used, the two variants are implemented taking two types of memory size of ants. In this way four variants are achieved. Now all these four variants are also implemented by taking two different initialization phase i.e with and without the constraint imposition. Hence eight variants in total are implemented individually in this paper. The clarity of the result images lead us to conclude that the variant consisting of Roulette wheel selection method with ants having tabu list memory and with imposed constraint during the period of initialization of ant's position outperforms others.

The paper is organized as follow: section II describes the problem to solve, and then in section III after mentioning some existing methods of edge detection we have briefly explained our variations. Experimental results are discussed with figures and table in section IV and section V concludes and discusses some future scopes of work.

II. Problem Description

Digital image is a two dimensional function g(a,b), where co-ordinates(a,b) and amplitudes at (a,b), i.e intensity are all finite in number. Each particular co-ordinate location is called a pixel [6]. Fig. 1 shows how an image is represented by pixels that contains the grey level values for those pixels.

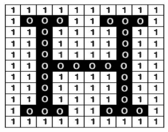


Fig. 1 Representation of image in matrix that contains the intensity values for pixels [7].

The edge, set of pixels, denotes the local feature that separate two regions in each of which the gray level is more or less uniform, therefore it is useful for segmentation, registration and identification of objects of an image. Moreover it is useful to get the information from an image.

Edge detection is a technique that comprises three steps: Filtering, Enhancement and Detection, identifies the edges from an image. Fig. 2 shows an example of an image with the result image of edge detection. Though it can be solved by applying different operators like Prewitt, Sobel, and Canny etc but can be detected well by applying approximation algorithms that gives a near optimal solution. Some of them are Genetic Algorithm, Fuzzy Technique, Simulated Annealing, Neural network and Ant colony optimization etc.

III. SOLUTION METHOD

Detection of edges in an image is an important part in the field of image processing. As said in the previous section, basic ways to detect edges are by applying two main techniques such as: Derivative approach, and Pattern fitting approach. For example Sobel edge detector, Prewitt operator, canny detector etc. But many optimization based approaches have been proposed gradually to extract edges from an image that performs better than these operators.

A. *Derivative approaches*

Edges are detected by taking the derivatives of intensity values of pixels followed by a pre given threshold value.

Ordinary operator, Robert Cross operator, Prewitt operator, Sobel operator and Canny operator are the different derivative operators to detect edges.

B. *Pattern Fitting Approach*

In this approach the main purpose is to fit a given pattern over the neighbors of a particular pixel and decide if that pixel belongs to an edge or not. Hueckel operators, Best-plane Fit Gradient (bpf) fall in this category.

C. *Meta-heuristic Algorithm Based Approach*

As said previously that edges are well detected by applying approximation algorithms. Some such examples of previous works where meta-heuristic algorithms are applied to edge detection are:

(a) (b)

Fig. 2 (a) Shows an input image. (b) The edges after edge detection [11].

In the year 1994 L.Caponetti, M. Abbattista, G. Carapella proposed edge detection using Genetic approach [8]. Before that in 1989 H.L.Tan, S. B. Gelfand, E. J. Delp proposed a cost minimization approach to edge detection using Simulated Annealing [9]. In the year 2008 Jing tian, W.Yu, S. Xie have proposed a different Ant Colony Optimization for edge detection [2]. Then in 2010 O.P.Verma, M. Hanamandlu, A.K. Sultania, Dhruv proposed a novel Fuzzy ant System for edge detection [4] and Mahdi Setayesh, M. Zhang, M. Johnston proposed improving edge detection using Particle Swarm optimization [10] etc.

1) Basic idea of Ant Colony Optimization (ACO):

Ant colony optimization (ACO) is a population-based metaheuristic technique that can be used to find near optimal solutions to optimization problems. This algorithm is based on the shortest path finding behavior of the real natural ants. When an ant begins to search for food it primarily moves in a random way and to mark its way they deposit a certain amount of pheromone in it. The follower ants sense the pheromone and it will select the path that contains more pheromone. As overtime pheromone trails evaporate, so longer path will not contain enough pheromone to be sensed by the follower ants so gradually after sometime the shortest path will be discovered. Fig. 3 describes the given behavior of natural ants. Applying this strategy Marco Dorigo proposed the ACO algorithm [1] and its variants like Ant Colony System, Max Min Ant Colony, and Rank Based Ant Colony etc.

2) Edge detection Technique by using Ant Colony Optimization:

In general the proposed approach contains five phases: initialization, construction, updating, decision and visualization phase, each described below.

a) Initialization Phase:

K numbers of ants are randomly assigned initially on an image I, each pixel of which can be viewed as a position where the ants can move. The initial value in the pheromone matrix $t^{(0)}$ for each pixel is set to be a constant value $t^{(init)}$ and heuristic value for all the pixels will be calculated by (1) [2].

$$\eta_{i,j} = \frac{1}{Z}\left(V_c\left(I_{i,j}\right)\right) \tag{1}$$

Where $I_{(i,j)}$ is the intensity value of the pixel (i, j) of the image I, the function $Vc(I_{i,j})$ is performed over the c numbers of neighbors of (i,j) and it is calculated by(2) [2]:

$$V_c\left(I_{i,j}\right) = f\left(\left|I_{i-1,j-1} - I_{i+1,j+1}\right| + \left|I_{i-1,j} - I_{i+1,j}\right| + Ii-1,j+1-Ii+1,j-1+Ii,j-1-Ii,j+1\right. \tag{2}$$

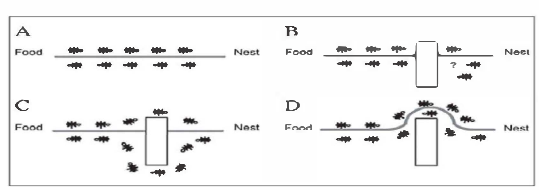


Fig. 3. Example of the behavior of natural ants[12].

To determine the function f (x) in (2), the following function (3) [2] is used in our paper where value of λ is set to 1;

$$f(x) = \lambda x, \qquad for x \geq 0, \tag{3}$$

b) Construction Process:

Total N number of construction step will be performed. At the nth construction-step, one ant is randomly selected, and this ant will consecutively perform L movement-steps. The ant move from the pixel (l,m) to its neighboring pixel (i, j) depending on the transition probability that is defined as

$$P_{i,j}{}^{(n)} = \frac{(\tau_{i,j}{}^{(n-1)})^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{j\in\Omega_i}\left((\tau_{i,j}{}^{(n-1)})^{\alpha}(\eta_{i,j})^{\beta}\right)}, if j \in \Omega_i$$

(4)

where $t_{i,j}{}^{(n-1)}$ is the pheromone value of the pixel i,j, Ωi are the neighborhood pixels of (l,m) (here eight neighbors), the constants α and β are the influence factor of pheromone information and heuristic information, respectively. η_(i,j) represents the heuristic information of pixel i,j calculated by (1) and depending on the value obtained from (4) [2] and selection method used, one neighbor will be selected as the current pixel i.e. i,j for the next movement step.

c) Update Process:

The proposed approach performs two updates operations for updating the pheromone matrix.

- The first update or local update is performed after the completion of all L movements of an ant within a particular construction-step. It is done for the purpose of deposition of pheromone in the visited pixels by an ant within that cycle. Each component of the pheromone matrix is updated according to the (5) [2] after n^{th} construction step.

$$\tau_{i,j}{}^{(n-1)} = \begin{cases} (1-\rho).\tau_{i,j}{}^{(n-1)} + \rho.\Delta_{i,j}{}^{(k)} & \text{if } (i,j) visited\ by\ Kth\ ant \\ \tau_{i,j}{}^{(n-1)}, & otherwise \end{cases} \quad (5$$

Where ρ is the pheromone evaporation rate $\tau_{i,j}{}^{(n-1)}$ is the pheromone value at the pixel i,j till n-1^{th} construction step and $\Delta_{i,j}{}^{(k)}$ is determined by the heuristic matrix given in (6) [2],

$$\Delta_{i,j}{}^{(k)} = \eta_{i,j} \tag{6}$$

- The second update is performed after the completion of movements of all ants within each construction-step by (7) [2].

$$\tau^{(n)} = (1-\psi).\tau^{(n-1)} + \psi.\tau^{(0)} \tag{7}$$

where ψ denotes the pheromone decay coefficient.

d) Decision Process:

In this step, a binary decision is made for each pixel location to determine whether it belongs to an edge or not, by applying a pre set threshold value T [mean value of the final pheromone matrix] on the final pheromone matrix $\tau^{(n)}$.

e) Visualize Process:

In this step, an image of the size of the original image but with the pixels detected as edge by the previous process is only made black is created. Then the output image is stored.

3)Our Proposed and Implemented Variants:

In this paper we have proposed two variants for selection method in construction phase. Each is also combined with two other variants where modification in the initialization phase is done. Apart from that with these four, other two variants are incorporated depending on the memory size of ants. Hence in total eight variants are proposed and implemented here. They are named as GZM, GTM, GIZM, GITM, RZM, RTM, RIZM, RITM based on the strategy used.

GZM (Greedy selection with Zero Memory and without modified initialization phase): In this variant for selection of next pixel in the construction phase, greedy selection method is used without incorporating the checking condition in the initialization phase that the initial positions of ants should be near to an edge. This is done to eliminate the detection of unwanted edges. In GZM the ants are memory less i.e. an ant can select it's just previously visited pixel also.

GTM (Greedy selection with Tabu list Memory and without modified initialization phase): For this variant also greedy selection method is used without the modification of initial phase that ants will be set in a position that is near to an edge. But here the ants can remember its last N number of visited pixels and it will not select them or move to those pixel again in a particular construction cycle.

GIZM (Greedy selection with modified Initialization phase and Zero Memory): Like previous variants it also uses greedy selection method in the construction phase but this time the modification in the initialization phase is imposed. So it will check if the initial position of ant is near to an edge or not. If not then another random position will be selected for that ant. But like GZM here also the ants are memory less.

GITM (Greedy selection with modified Initialization phase and with Tabu list Memory): It is like GTM that applies greedy selection method and ants having tabu list memory but this variant applies the condition checking in initialization phase that ants should be initialized near to edge to avoid useless movements of ants.

RZM (Roulette wheel selection with Zero Memory and without modified initialization phase): This variant is superior to previous four. Here Roulette wheel selection method is used to the probability value of (4) to select next pixel in the construction phase but without the constraints given in the initialization phase and also the ants are memory less here so it can move to the just visited pixel again.

RTM (Roulette wheel selection with Tabu list Memory and without modified initialization phase): The variant RTM applies roulette wheel selection method without modifying the general initialization process but it is implemented by the ants that have N size memory to remember N numbers of already visited pixels.

RIZM (Roulette wheel selection with modified Initialization phase and Zero Memory): Like previous two variants it also uses roulette wheel selection method but it comprises the constraints at the time of initialization of ant's position that the ants should be initialized near to edge points. This is done to avoid unwanted edges in the result image and also to avoid unnecessary movements of ants in the background pixels. But here the ants are not adopted with memory and since just visited pixel can be visited again by an ant therefore cycle can occur in the path of an ant in a construction phase.

RITM (Roulette wheel selection with modified Initialization phase and with Tabu list memory): The last variant that outperforms all others uses roulette wheel selection method with the constraints imposed in the initialization phase and also the ants are having tabu list memory of size N i.e. in a construction cycle an ant will not visited its last N numbers of visited pixels so loop will not form and more edge points will be selected. These strategies are used to detect more true edge points than other variants.

IV. EXPERIMENTAL RESULT

We have applied the proposed variations to binary images. Since the intensity values of the pixels of a binary image can only be two types: black and white. Hence the modification in the initialization phase have been made to make the ACO algorithm more effective and optimized for binary images. Here we have applied two different selection processes with or without the constraint checking in the initialization phase and with different memory size of the ants. A global threshold at the final pheromone matrix is used to discard irrelevant edges.

The proposed variations of ACO-based edge detection algorithm is implemented using Matlab 2010 programming language and run on a PC with a Intel(R) Core(TM)2Duo CPU E8400 @ 3.00 GHz with 2 GB RAM and 32 bit operating system.

The appropriate values for some parameters as below are selected by trial and error process.

A= No. of ants ,computed as floor($\sqrt{(row*col)}$).

N= Main construction cycle number. (2, 3, 4).

L= Total movements step for each ants. (A).

α , β = The weight factor of the pheromone information and heuristic information in the probability function. The value taken for it is 1.0 and 0.05 respectively.

$t^{(0)} = t^{(init)}$ =Initial pheromone value of each pixels = 0 .0001.

Ω = No of neighbors considered is 8.

ρ = The pheromone evaporation rate is 0.1.

Ψ= The pheromone decay coefficient set to 0.05.

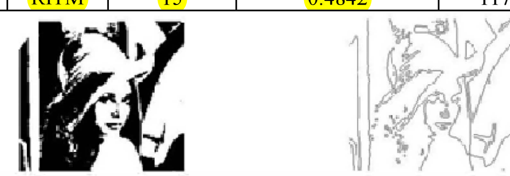
λ= The adjusting factor for calculating heuristic value = 1.

M= Size of ants memory for a construction step.

The original image and the result image of Canny,s edge detection method is shown in Fig. 4. The result images after implementing the variations to the original image with greedy selection method and roulette wheel selection method with various memory sizes of ants are given in Fig. 5. Table 1 shows the variation used for each image in Fig. 5 with the correlation value with respect to the canny's result image and the cpu utilization time to run that algorithm. The betterment can be viewed by observing the result images in Fig. 5 that are of good clarity and we can conclude that the variants that applies roulette wheel selection method with new initialization phase and ant's memory size 5-15 gives better results than all others.

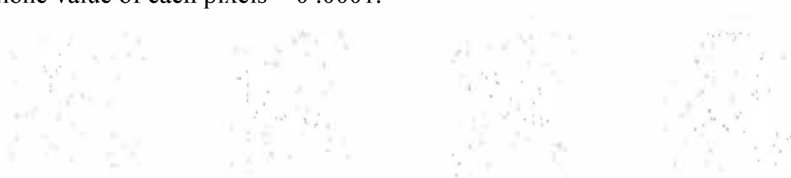
Table 1
Experimental results for different variations.

Fig.	Variant used	Memory size of Ant's	Correlation value with respect to Canny's result	Cpu utilization time in (sec)
5.a	GZM	0	0.1249	59.019
5.b	GTM	5	0.0995	97.595
5.c	GTM	10	0.1187	98.566
5.d	GTM	15	0.1062	126.841
5.e	GIZM	0	0.1696	61.562
5.f	GITM	5	0.1928	85.955
5.g	GITM	10	0.1696	61.562
5.h	GITM	15	0.1988	95.954
5.i	RZM	0	0.4043	76.119
5.j	RTM	5	0.3409	112.474
5.k	RTM	10	0.3764	118.248
5.l	RTM	15	0.3672	136.324
5.m	RIZM	0	0.4973	76.268
5.n	RITM	5	0.4636	119.897
5.o	RITM	10	0.4468	123.69
5.p	RITM	15	0.4842	117.938



(a) (b)

Fig. 4 (a) Shows an input image. (b) Image after edge detection by using Canny's detector operator.



(a) (b) (c) (d)

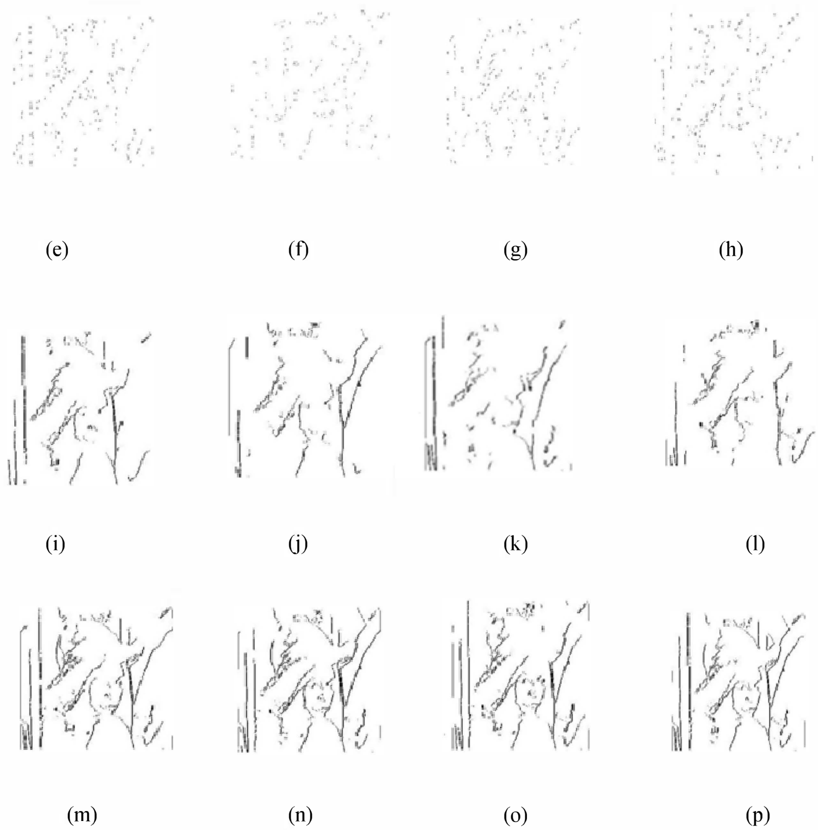


Fig. 5 (a) Result image of the variants GZM, (b, c, d) The result of GTM with memory size 5, 10, 15 respectively. (e) Shows the result image of variants GIZM. (f, g, h) Output image of GITM with memory size 5, 10, 15. (i) Denotes the outcome of RZM variants. (j, k, l) The edge detection by RTM variant is given with ants memory of size 5, 10, 15. (m) Shows the output image of variants RIZM and (n, o, p) are the result images of best variants RITM with memory size of 5, 10, 15.

V. CONCLUSION AND FUTURE SCOPES

Ant colony algorithm is a metaheuristic algorithms inspired by behavior of natural insect; ants. In this paper we have proposed and successfully implemented eight variants by modifying the initialization and construction phase of existing ACO algorithm, such as: GZM, GTM, GIZM, GITM, RZM, RTM, RIZM and RITM. We have applied the variants for detection of edges in binary images. From the resulting images we conclude that the variant RITM i.e. with Roulette Wheel selection method with tabu list memory of ants and with constraints given in the initialization phase is proved to be better than others.

Ant colony algorithm is a metaheuristic algorithms inspired by behavior of natural insect; ants. In this paper we have proposed and successfully implemented eight variants by modifying the initialization and construction phase of existing ACO algorithm, such as: GZM, GTM, GIZM, GITM, RZM, RTM, RIZM and RITM. We have applied the variants for detection of edges in binary images. From the resulting images we conclude that the variant RITM i.e. with Roulette Wheel selection method with tabu list memory of ants and with constraints given in the initialization phase is proved to be better than others.

EFERENCES

[1] Marco Dorigo, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications and advances."

[2] J. Tian, W.Yu and S. Xie, "An Ant Colony Optimization Algorithm For Image Edge Detection", *IEEE Congress on Evolutionary computation*, pp.751-756, 2008.

[3] A. V. Baterina and C. Oppus, " Ant Colony Optimization for Image Edge Detection", *Recent advances in signal processing, robotics and automation*, pp. 220-225, 2010.

[4] O. P. Verma, M. Hanamandlu and A.K. Sultania, Dhruv, " A Novel Fuzzy Ant System For Edge Detection", *IEEE/ACIS International Conference on Computer and Information Science 9*, pp. 228-233, 2010.

[5] J. Zhang, K. He, X. Zheng and J. Zhou (2010), "An Ant Colony Optimization Algorithm for Image Edge Detection", *IEEE International Conference on Artificial Intelligence and Computational Intelligence*, pp-215-219, 2010.

[6] Gonzalez. R. C,Woods R.E., "Digital Image Processing" ,3rd edi.

[7] https://www.library.cornell.edu/preservation/tutorial/intro/intro-01.html.

[8] L. Caponetti, M. Abbattista and G. Carapella , " A Genetic approach to edge detection, Image Processing", *ICIP-94, IEEE International Conference Proceeding*, vol 1, pp.318-322, 1994.

[9] H. L. Tan, S. B. Gelfand and E. J. Delp, " A cost minimization approach to edge detection using Simulated annealing", *IEEE Comput. Vision Patt. Recogn. Conf.(San Diego)*, pp. 86-91, June 1989.

[10] M. Setayesh, M. Zhang and M. Johnston , "Improving Edge Detection using Particle Swarm Optimization", *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of IEEE,* pp.1-8, 2010.

[11] http://www.csse.uwa.edu.au/~pk/Research/matlabfns/LineSegments/example/index.html.

[12] http://www.funpecrp.com.br/gmr/year2005/vol3-4/wob09_full_text.html