# Search Improvemnt Based on Ants Performance in Image Edge Detection using ACO

Syam Ajay Simha Gullipalli
Universität Paderborn
syam@mail.uni-paderborn.de

February 24, 2015

**Abstract.** Ant Colony Optimization (ACO) is a popular metaheuristic framework used to solve diverse combinatorial optimization problems. Graph problems can be seen as combinatorial optimization problems which could be solved using ACO algorithms. Image edge detection is already attained through ACO algorithms by representing image as graph. The existing mehods leave noise on the resultant image which is reduced through thresholding by attempting different threshold values. The proposed thecnique leads the ants towards good solutions based on the average performance of ants which does not require reduction of noise.

**Keywords:** Ant Colony Optimization, Edge Detection, Ant Colony System, Image Processing.

## 1 Introduction

Edge Detection is one of the early stages of machine vision applications, which is used to estimate structural features of a given image. The edges in an image are determined by finding abrupt changes in intensity levels of neighbor pixels. The conventional edge detecion methods use smoothing functions to reduce the noise, and complex mathematical functions like first and second order derivatices and thresholding techniques to find edges. These methods may fail to find all possible edges. Edge detection using Ant Colony Optimization (ACO) algorithm may dominate the conventional edge detection methods, because it takes the advantage of a number of ants to find partial solutions, and these partial solutions are collectively used to produce complete possible solution.

ACO is a population based metaheuristc which is inspispred by the foraging behavior of ants [1–3]. A number of ACO-based algorithms have been developed since Dorigo et al. first proposed in early 1990's. Ant System (AS) is the first algorithm which is the base for other varants. Rank-based Ant System (RAS), Max-Min Ant System (MMAS), Elitist Ant System (EAS), Ant Colony System (ACS) are the successors of AS in which the main difference lies in how the pheromone is updated [1, 2]. The proposed technique uses ACS which benefits from the two levels of pheromone update.

The image should be transformed into a graph representing a set of vertices and edges. Then ACO algorithm is applied on the resulting graph to construct pheromone matrix. Section 2 describes how the image is transformed and some state-of-the-art algorithms using ACO to find edges using the transformed graph. The proposed technique, how initialization of ants effect the results is expalined in section 3. The results of the proposed technique are analyzed in section 4.

## 2 State-of-the-art Algorithms for Edge Detection

Many experiments show that the ACO metaheuristic could be successfully used to find edges in an image. Edge detection using the transition and pheromone update rules of AS are presented in literature [4]. Many experiments prove that the ACS variant dominates AS in edge detection [5–7] because of it's two levels of pheromone update (local and global). A hybrid edge detection using Canny edge detector and ACS is presented in literature [8], and edge detection using quantam computing and ACS is presented in literature [7] which use complex mathematical functions like matrix multiplications and trignometric functions. The proposed technique does not use those complex functions. Instead, it extends the work presented in literature [5]. In order to apply ACS on an image, it should be transformed into a graph. Therefore, a matrix $I_{w \times h}$ representing the intensity at each pixel is constructed as shown in figure 1.

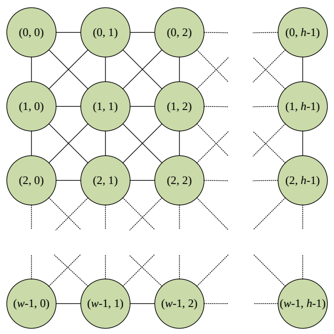


Fig. 1: Graph representation of $w \times h$ image. Each pixel in the image is considered as a node and each node is connected to its neighbors (horizontal, vertical, and diagonal) to form edges. [5]

The pseudocode for edge detection using ACS is as shown in algorithm 1. $K$ artificial ants are randomly distributed over the image and their movements are guided by the variation in pixel intensity levels.

In the initialization step all the values in pheromone matrix are set to a constant $\tau_0$ (usually small but not zero). An exploitation paramerter $q_0$, a constant between 0 and 1 is given. At each construction step every ant moves from it's current pixel to it's neighborhood pixel based on a random number $q$ uniformly

**Algorithm 1:** Pseudocode for edge detection using ACS. [5]

```
1 Initialization;
2 forall the iterations n in 1:N do
3     forall the construction steps l in 1:L do
4         forall the ants k in 1:K do
5             Choose and move to the next pixel;
6             Update local pheromone;
7     Update global pheromone values on visited pixels;
```

distributed between 0 and 1, using pseudorandom proportional rule as shown in equation (1).

$$j = \begin{cases} \arg\max_{j \in N_i^k} (\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}) & \text{if } q \leq q_0 \text{(Exploitation)} \\ J & \text{otherwise (Exploration)} \end{cases} \tag{1}$$

Where $N_i^k$ is the set of unvisited neighbors and $J$ is selected using trasition probability rule as in equation (2).

$$p_{ij}^{(k)}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} \tag{2}$$

$\tau_{ij}$ is the pheromone value and $\eta_{ij}$ is the heuristic information at pixel $(i,\ j)$ which are influenced by constants $\alpha$ and $\beta$ respectively. $\eta_{ij}$ is calculated as in equation (3).

$$\eta_{ij} = \frac{V_c(I_{ij})}{V_{max}} \tag{3}$$

$I_{ij}$ is the intensity of pixel $(i,\ j)$. $V_c(I_{ij})$ is the function (equation (4)) of intensity variation around the pixel $(i,\ j)$ as shown in figure 2. $V_{max}$ is the maximum intensity variation in given image.

$$\begin{aligned} V_c(I_{(i,j)}) = {} & |I_{(i-1,j-1)} - I_{(i+1,j+1)}| + |I_{(i,j-1)} - I_{(i,j+1)}| \\ & + |I_{(i+1,j-1)} - I_{(i-1,j+1)}| + |I_{(i+1,j)} - I_{(i-1,j)}| \end{aligned} \tag{4}$$

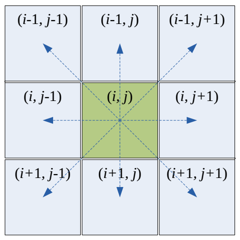


Fig. 2: Computing intensity variation at pixel $(i,\ j)$. [5]

At each move ants update the local pheromone value of the visited pixel as in equation (5). $\varphi$ is the pheromone decay coefficient. Repeated visit of same set of nodes by an ant is restricted by maintaining a memory at each ant.

$$\tau_{ij} \leftarrow (1-\varphi)\cdot\tau_{ij} + \varphi\cdot\tau_0 \tag{5}$$

After each construction step, global pheromone is updated on visited pixels using equation (6). $\rho$ is the pheromone evaporation coefficient in range (0, 1]. $\Delta\tau_{ij}^k$ is the amount of pheromone updated by ant k on pixel $(i,\ j)$. If ant k visited pixel $(i,\ j)$, then $\Delta\tau_{ij}^k$ is computed as the average of heuristic values associated to the pixel $(i,\ j)$. Otherwise, set it to zero.

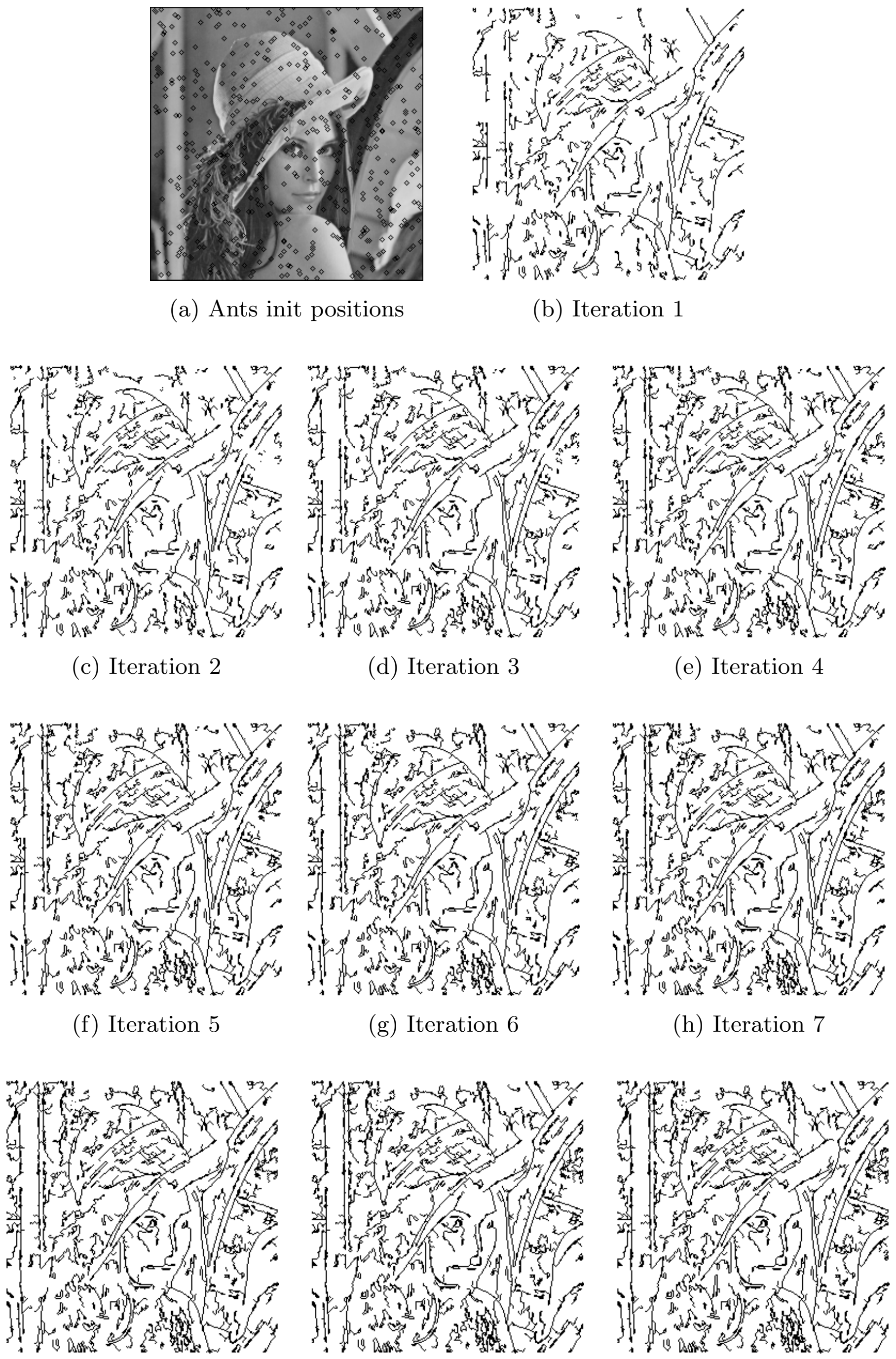
$$\tau_{ij}^n \leftarrow (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij}^k \tag{6}$$

The global pheromone update here differs from the original ACS. The ACS is initially designed for Travelling Salesman Problem (TSP) which benefits from updating global pheromone only on best so far solutions. Unlike TSP, in edge detection the global pheromone is updated on each solution, because the combination of all the solutions is considered as complete possible solution.

The technique proposed in literature [5] is implemented and results are presented in figure 3. The edge images (b-k) are binary images of pheromone matrix after each iteration without thresholding. In ACS algorithm, as initialization step, ants are initialized randomly. The original technique uses thresholding technique to remove the noise since it leaves noise on the resultant image. Often, the thresholding removes the details of edges apart from the noise. The next sections present how to overcome loosing the details on edges on resultant image.

## 3 Improving Search

A good set of parameters for edge detection using ACS is presented in literature [5] by varying $q_0$ valie. The same set of parameter values are used for all the experiments proposed in this paper by fixing the value of $q_0$ to 0.7. Standard test image, Lena of size $256\times256$ is used to test the experiments. The value of each parameter is as follows:


(a) Ants init positions (b) Iteration 1

(c) Iteration 2 (d) Iteration 3 (e) Iteration 4

(f) Iteration 5 (g) Iteration 6 (h) Iteration 7

(i) Iteration 8 (j) Iteration 9 (k) Iteration 10

Fig. 3: Ants are (a) initialized on random positions and (b)-(k) results are shown after each iteration is finished.

– Initial pheromone value ($\tau_0$) = 0.1
– Number of ants ($K$) = 512
– Number of iterations ($N$) = 10
– Number of constructions ($L$) = 40
– Parameter influencing pheromone trail ($\alpha$) = 1 (fixed to 1 for ACS)
– Parameter influencing heuristic information ($\beta$) = 1
– Pheromone decay coefficient ($\varphi$) = 0.05 (used for local pheromone update)
– Pheromone evaporation coefficient ($\rho$) = 0.1 (used for global pheromone update)
– Degree of exploration ($q_0$) = 0.7

## 3.1 Initialize Ants on Better Heuristics

Most of the literature including [7, 8] concentrate on improving the heuristic. The proposed technique try to improve the search instead, to best fit for the image edge detection. The biggest disadventage of initializing ants on random positions is, it leaves noise on the image, and even the theresholding technique to remove noise may remove some important details on the edges. Removing noise means, the work done by most of the ants is simply wasted. Since, the heuristic matrix for the entire image could be constructed before starting the algorithm, it is easier to guide all the ants to start at good locations where most of the solutions produced by ants could be used without thresholding.

## 3.2 Reinitialize Ants Based on Performance

Initializing ants on best heuristic positions has shown a drastic improvement over the method that initializes the ants on random positions. However, after certain iterations the result stagnates. Consequently, the ants may not find the remaining possible edges. Therefore, this experiment extends the previous one by adding an initialization step after each iteration.

On each iteration, the amount of pheromone deposited by each ant at local and global pheromone deposit steps are maintained. After global pheromone deposit (step 7 in algorithm 1) the average amount of pheromone ($\tau_{avg}$) deposited by all the ants is calculated. Before starting the next iteration, the ants which deposit the pheromone less than the avarage amount ($\tau_{avg}$) calculated are reinitialized to the next heighest heuristic position and the ant's memory of positions visited in the past is reset. Thus, the ants which performed above the average are left on their latest location to find the connected edges. The ants which performed below the average assume that there are no connected edges to find and move to new positions to find further possible edges. The result after each iteration is as shown in figure 5. The results show that, this method tends to find edges at other positions without loosing most of the edges found so far. Comparision of three experiments made in this paper is presented in next section.

The best heuristic depends on the nature of task. For edge detection using ACS, the best heuristics are the positions where the heuristic values are high.

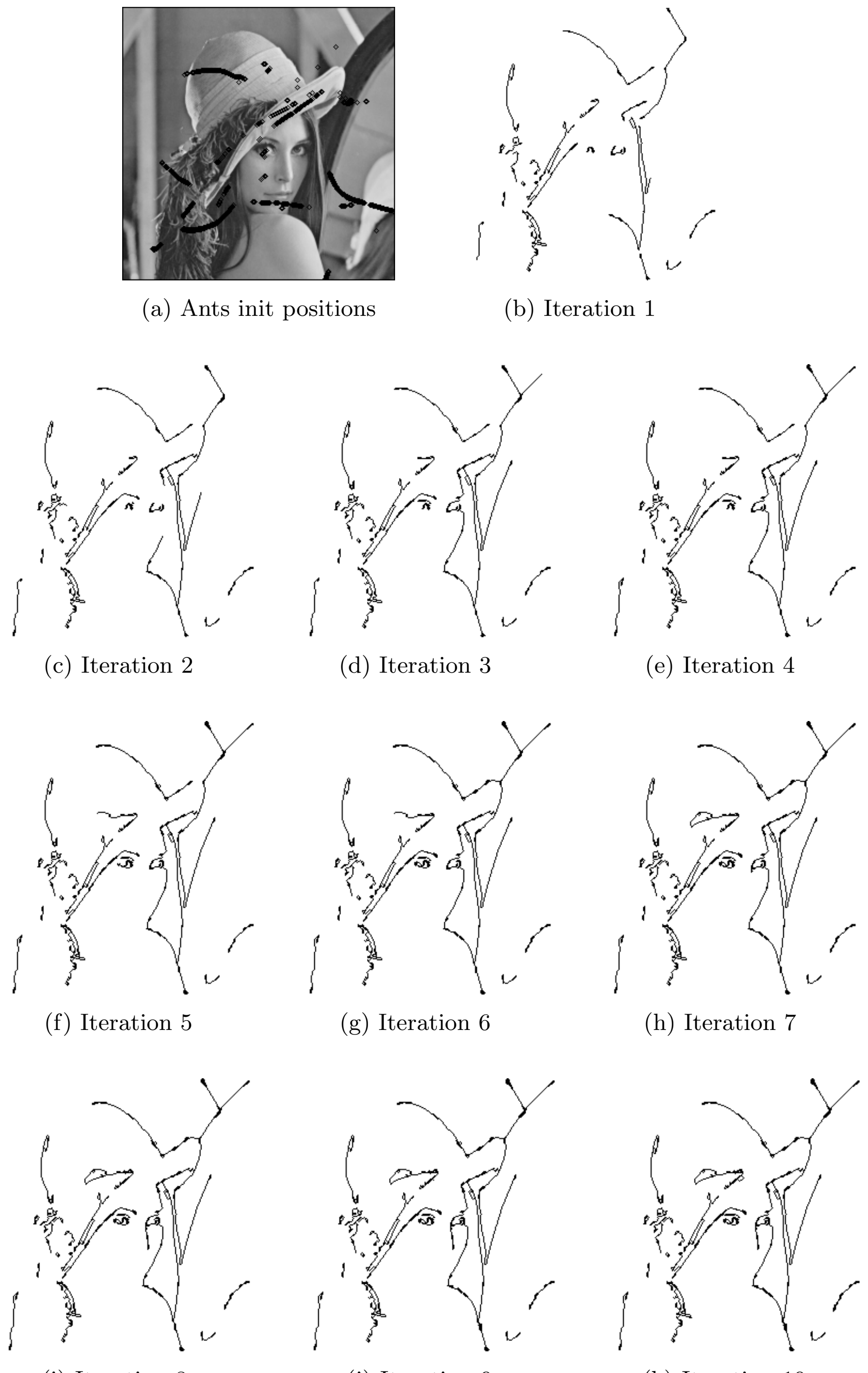
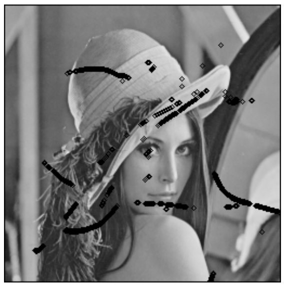
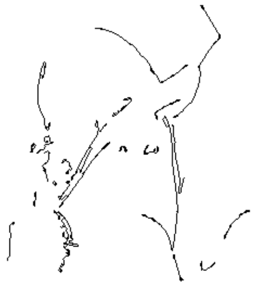
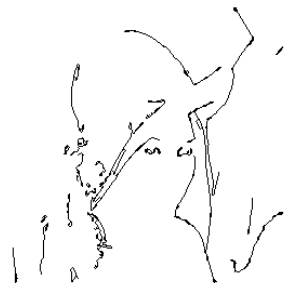
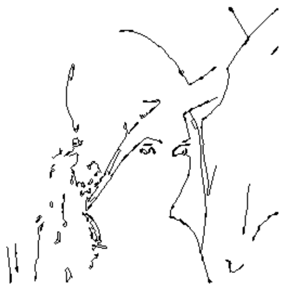


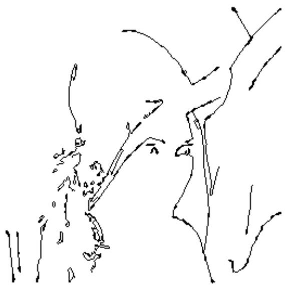
Fig. 4: Ants are (a) initialized on positions where heuristic values are high and (b)-(k) results are shown after each iteration is finished.

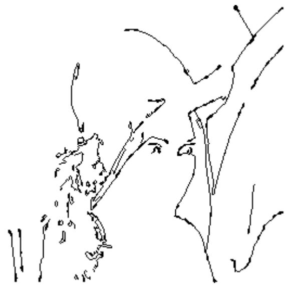

(a) Ants init positions

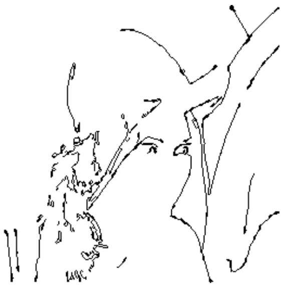

(b) Iteration 1

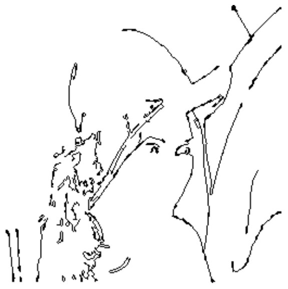

(c) Iteration 2

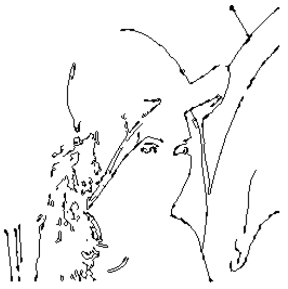

(d) Iteration 3

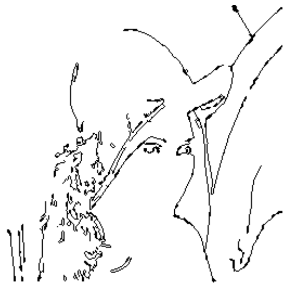

(e) Iteration 4


(f) Iteration 5


(g) Iteration 6


(h) Iteration 7


(i) Iteration 8


(j) Iteration 9

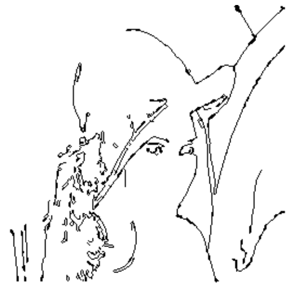

(k) Iteration 10

Fig. 5: Ants are (a) initialized on positions where heuristic values are high and (b)-(k) after each iteration ants pheromone deposit below the average are reinitialized and results are shown.

Therefore, $K$ ants are initialized on $K$ positions in decreasing order of heuristic values as part of ACS, and the results after each iteration are as shown in figure 4. The binary image after ecah iteration is presented and no thresholding is applied. Thre results show that, all the edges found by ants are potential edges and there is a great performance improvement compared to previous method (random initialization), because no ant's effort is wasted. After few iterations, the results will not be improved (stagnation) as shown in figure 4i - 4k, there is no much difference bwteen images means that all the ants move on previously visited positions (by any ant) without finding further possible edges. Section 3.2 proposes a technique to solve the stagnation, and improves the search to find further possible edges.

## 4 Analysis of Results

The amount of pheromone deposited by each ant on each iteration is maintained, and the average of pheromone deposit on each iteration is plotted as shown in figure 6. For all the three experiments made, box plot for the ants pheromone deposit in each method on each iteration is plotted as in figure 7. Initializing ants on random positions shows poor performance over the other two methods as the avrage pheromone deposit is too low on each iteration.

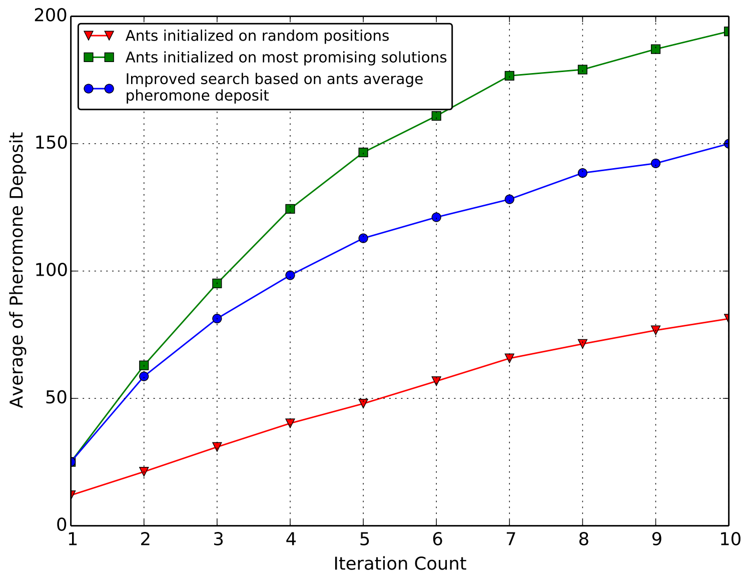
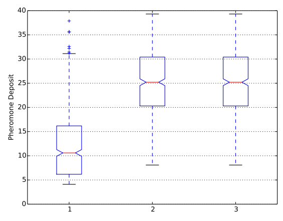
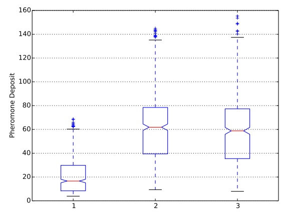


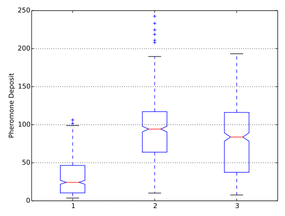
Fig. 6: Plot showing the average amount of pheromone deposited by all the ants on each iteration.



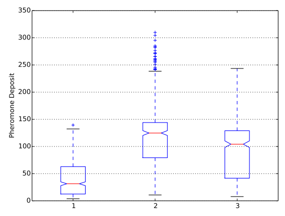
(a) Iteration 1



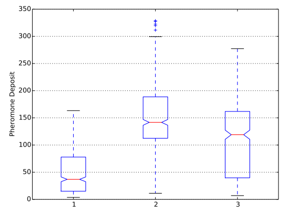
(b) Iteration 2



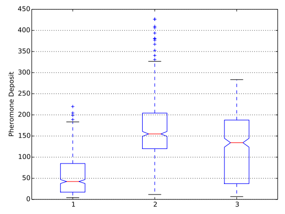
(c) Iteration 3



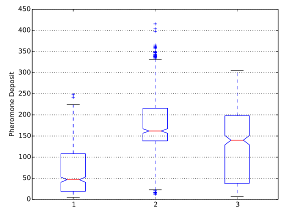
(d) Iteration 4



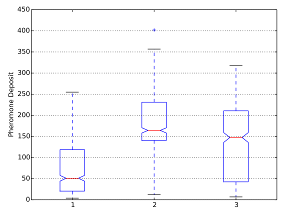
(e) Iteration 5



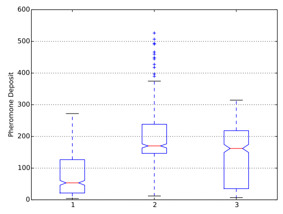
(f) Iteration 6



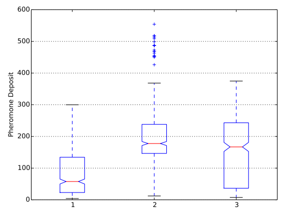
(g) Iteration 7



(h) Iteration 8



(i) Iteration 9



(j) Iteration 10

Fig. 7: Box plot for pheromone deposit by each ant on each iteration for three methods, (1) Ants initialized on random positions, (2) ants initialized on positions where heuristic values are high, and (3) reinitialize ants after each iteration based on their average performance.

Although, the second approach, the method that initializes ants on better heuristics has more average pheromone deposit on each iteration, the outliers above the upper whisker in boxplot (position 2 on X-axis in figure 7) prove that the ants deposited large amount of pheromone have influenced the average value, and ants try to move towards the best solutions already found rather than finding other possibilities. Third approach, reinitialize ants below the average performance, (position 3 on X-axis in figure 7) the wide spread of inter quartile range (IQR) shows that the ants pheromone deposit is more deviated which means, several ants try to find new edges on each iteration.

## 5 Conclusion

Reassigning ants below the average performance has excelled, because the result never gets stagnated. Especially, when the image has more disconnected edges (for example, the hair part on the test image), this method eventually finds all the edges. There is a possibility that the edges found during early iterations may disappear in the later because of pheromone evaporation. But, blending all the results after each iteration would give the complete possible edges. A simple heuristic is used for the experiments made in this paper. Improving heuristic and hybridization on top of the proposed method may further improve the solutions.

## References

1. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Company, Scituate, MA, USA (2004)
2. Dorigo, M.: Ant colony optimization. Scholarpedia **2**(3) (2007) 1461 revision 90969.
3. Christian, B.: Ant colony optimization: Introduction and recent trends. Physics of Life Reviews **2**(4) (2005) 353–373
4. Rezaee, A.: Extracting edge of images with ant colony. Journal of Electrical Engineering **59**(1) (2008) 57–59
5. Baterina, A.V., Oppus, C.: Ant colony optimization for image edge detection. In: Proceedings of the 9th WSEAS International Conference on Signal Processing, Robotics and Automation. ISPRA'10, Stevens Point, Wisconsin, USA, World Scientific and Engineering Academy and Society (WSEAS) (2010)
6. Shweta, A.: A review paper of edge detection using ant colony optimization techniques. Internation Journal of Latest Research in Science and Technology **1**(2) (2012) 120–123
7. Jian, Z., Jiliu, Z., Kun, H., Huanzhou, L.: Image edge detection using quantum ant colony optimization. International Journal of Digital Content Technology and its Applications(JDCTA) **6**(11) (2012) 402–405
8. Manish, T., Murugan, D., Kumar, G.: Hybrid edge detection using canny and ant colony optimization. Communications in Information Science and Management Engineering **3**(8) (2013) 402–405