Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 416–425
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Edge detection improvement by ant colony optimization

De-Sian Lu, Chien-Chang Chen *

Department of Computer Science, Hsuan Chuang University, Hsinchu 300, Taiwan

Received 11 September 2006; received in revised form 17 October 2007
Available online 13 November 2007

Communicated by G. Sanniti di Baja

Abstract

Edge detection is a technique for marking sharp intensity changes, and is important in further analyzing image content. However, traditional edge detection approaches always result in broken pieces, possibly the loss of some important edges. This study presents an ant colony optimization based mechanism to compensate broken edges. The proposed procedure adopts four moving policies to reduce the computation load. Remainders of pheromone as compensable edges are then acquired after finite iterations. Experimental results indicate that the proposed edge detection improvement approach is efficient on compensating broken edges and more efficient than the traditional ACO approach in computation reduction.

Keywords: Edge detection; Ant colony optimization (ACO); Pheromone trail

1. Introduction

Edges often carry important information about an object, when shown as large gradient magnitude. Edge detection strategies seek out obvious edges in an image. Traditional edge detection approaches, like Sobel or Canny operators (Canny, 1986; Gonzales and Woods, 2002; Sharifi et al., 2002), commonly extract edges by adopting specific templates, or in combination with smoothing functions. However, traditional edge filtering methods often result in some drawbacks like broken edges. Therefore, many methods have been proposed to link these broken edges in order to improve edge detection (Ghita and Whelan, 2002; Hajjar and Chen, 1999). Endpoints of edges preserve important information, and some methods try to draw reasonable direct lines between pairs of broken edges. Ghita and Whelan (2002) adopted a mask to acquire the direction of endpoints in order to estimate the cost of the linking line. The cost and direction of each directed line determines whether the line is selected. These methods are in general simple but their main inconvenience is the fact that they return incomplete edge structures.

Some edge linking approaches perform Hough transformation (Parker, 1997; Gonzales and Woods, 2002) on edge image, then extract the specific shape to connect broken edges. However, the edges do not always have fixed shapes. Some other methods (Ng et al., 1999; Sharifi et al., 2002) adopt hybrid techniques to connect broken edges. Insufficient information always leads to inappropriate results. Thus, broken edges are hard to connect accurately. Edge linking should consider at least two endpoints in order to generate linking edges. Therefore, possible compensable edges need to be sought from these endpoints. The local information of the original image can be further analyzed to help link broken pieces and search lost edges. This searching procedure is a globally tree-like topology.

Ant colony optimization (ACO) is a heuristic method that imitates the behavior of real ants to solve discrete optimization problems (Dorigo and Stützle, 2004). The created artificial ants behave like intelligent agents with memory and ability to see. These ants share their experiences in order to search optimal paths iteration by iteration. Since

* Corresponding author. Tel.: +886 35302255x5230; fax: +886 35391252.
E-mail address: cchen34@hcu.edu.tw (C.-C. Chen).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.10.021

the starting states of ants and the terminating rules are defined, ants adopt state transition rules to construct their optimal solutions.

ACO is efficient in solving tree-like problems. Furthermore, the constructive steps of each ant can be different in the same iteration. It means that no requirements are demanded to estimate the total number of probable optimal solutions. This condition differs from other evolutionary computation methods like genetic algorithms. Some researchers have adopted ACO to solve specific image processing or machine vision problems, like imitation and reproduction of human vision perception and optical illusions (Vallone and Mërigot, 2003), texture classification (Zhuang and Mastorakis, 2005; Zheng et al., 2003), and edge detection (Zhuang and Mastorakis, 2005).

In solving edge detection problem, each pixel of one image is assumed to be connected with its 8-neighborhood pixels. The distance between adjacent pixels is estimated from the original image, and the ants are placed on endpoints extracted from traditional edge detection approaches. These located ants attempt to repair breaks of edges, and extend their searching range to find compensable edges.

The rest of this study is organized as follows. Section 2 introduces the concept of ant colony optimization. Section 3 explains how to improve edge detection by ant colony optimization. Section 4 presents some experimental results and brief conclusions are given in Section 5.

2. Ant colony optimization

Ant colony optimization (ACO) is a multi-agent system (Dorigo and Stűtzle, 2004) that iteratively searches for optimal solutions. Elements of optimal solutions are extracted according to the shortest path of ant tours. Ants deposit their searching reward, pheromone, on their passed paths. These feedbacks may attract other ants to follow partially with a probability called state transition rules. This probability consists of two weighted factors, namely trail intensity and route length. State transition rules imply that shorter and more ant-experienced paths attract more ants to pass through. However, as with real ants, not all ants follow the most attractive paths, instead a few ants try to explore new paths. The process of taking the maximal probability path is called exploitation, and the process of selecting the next path by probability is called exploration.

ACO can be applied to other optimization problems by modifying evolutionary procedures. However, the following elements need to be clearly defined:

1. A problem graph that includes paths, cost of moving, and the number of ants.
2. Initial states and moving rules of ants.
3. Termination conditions of ants.
4. Definitions of pheromone reward and natural evaporation.

The ACO mechanism has the following desirable properties (Dorigo et al., 1996):

1. ACO is versatile in that it can be applied to similar versions of the same problem. An example of a similar version might be the traveling salesman problem, which can be directly extended to the asymmetric traveling salesman problem.
2. ACO is robust, and can be applied with only minimal changes to other combinational optimization problems, such as the quadratic assignment problem and the job-shop scheduling problem.
3. ACO is a population-based approach, and allows positive feedback exploitation to be adopted as a search mechanism.

3. Edge detection improvement by ant colony optimization

3.1. Redundant search reduction

The proposed approach relies on ants searching paths among break edges and exploring compensable edges. Ants

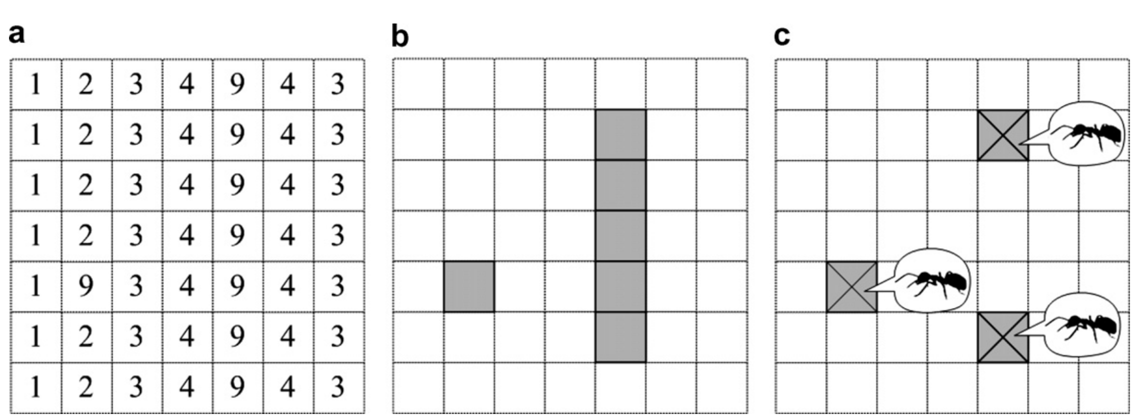

Fig. 1. (a) Original image. (b) Traditional edge detection of (a). (c) Placing ants on all endpoints.

are placed on endpoints of edge image filtered by traditional edge detection approaches. Each endpoint is laid down one ant as shown in Fig. 1c.

However, an image always contains hundreds of endpoints. Therefore, these ants may start a time-consuming evolutionary search. Some ants may search the same region redundantly. To reduce the number of redundant searches, the ants are split into several groups with different labels. One labeled ant finds other labeled ants by following four redundancy-reducing policies:

1. *Meeting face to face*. Two ants belonging to different groups both stop when the next moves bring them together, shown in Fig. 2a.
2. *Touching tracks*. An ant stops moving when it touches the track passed by another ant, shown in Fig. 2b.
3. *Adaptive tabu-list*. Each ant records his passed path, and uses a tabu-list to explore where he has never passed. However, this method causes some problems like blind alleys, as shown in Fig. 2c. The proposed approach adopts adaptive tabu-list to allow an ant to move to the passed routes again while time of ants moving into the blind alley is smaller than a predefined threshold.
4. *Track density measurement*. An ant stops when the passed tracks nearby the current point is larger than a predefined threshold, as shown in Fig. 2d.

These four policies improve the evolution efficiency by enhancing the path clarity and reducing the lower computation load.

3.2. Compensable edges search

The movement of an ant is determined by the 8-neighborhood difference. However, as shown in Fig. 3b, ants may unsteadily seek the whole edge-like region and easily cross to the error region. The global pheromone updating strategy may result in burr edges. Therefore, we adopt two following factors to select suitable paths:

- *Maximal neighboring difference*. Ants can identify possible edges from high neighboring differences.
- *Maximal connective similarity*. Ants may search possible edge irresolutely, as shown in Fig. 3c, when they search on pixels near the real edges. Edges with similar intensity are considered as real edges. Therefore, the path length estimation equation should estimate the affection of this consideration. Furthermore, the connective similarity is more important than the maximal neighboring difference.

Some important parameters used in the ACO approach are first defined. When an ant located at pixel i, the path visibility between pixels i and j is defined as follows:

$$\eta_{ij} = \frac{V(p_j)}{\max\{1, |p_j - p_i|\}}, \tag{1}$$

where p_i denotes the pixel intensity of pixel i, and p_j denotes the intensity of j as an adjacent pixel to i. The term $V(p_j)$ represents the neighboring difference of p_j, and is defined as follows:

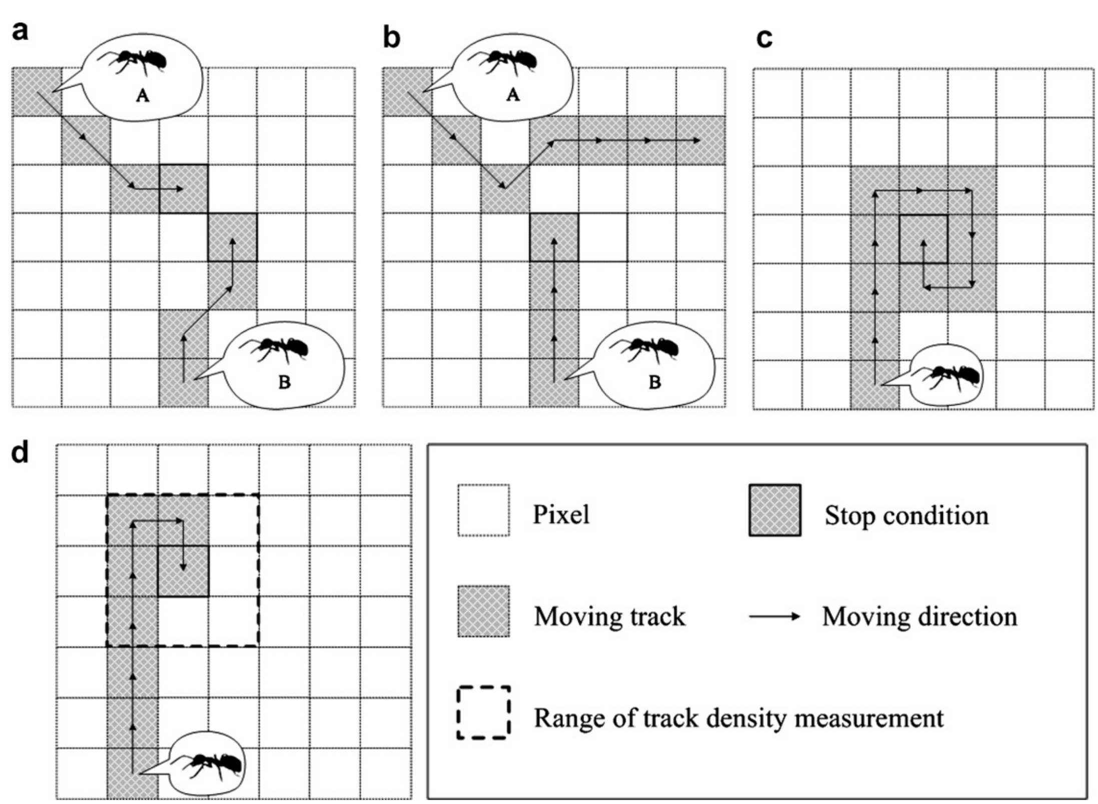

Fig. 2. (a) Two ants meet face to face. (b) The ant touches the track passed by ants of a different group. (c) Blind alley. (d) Measurement of track density.

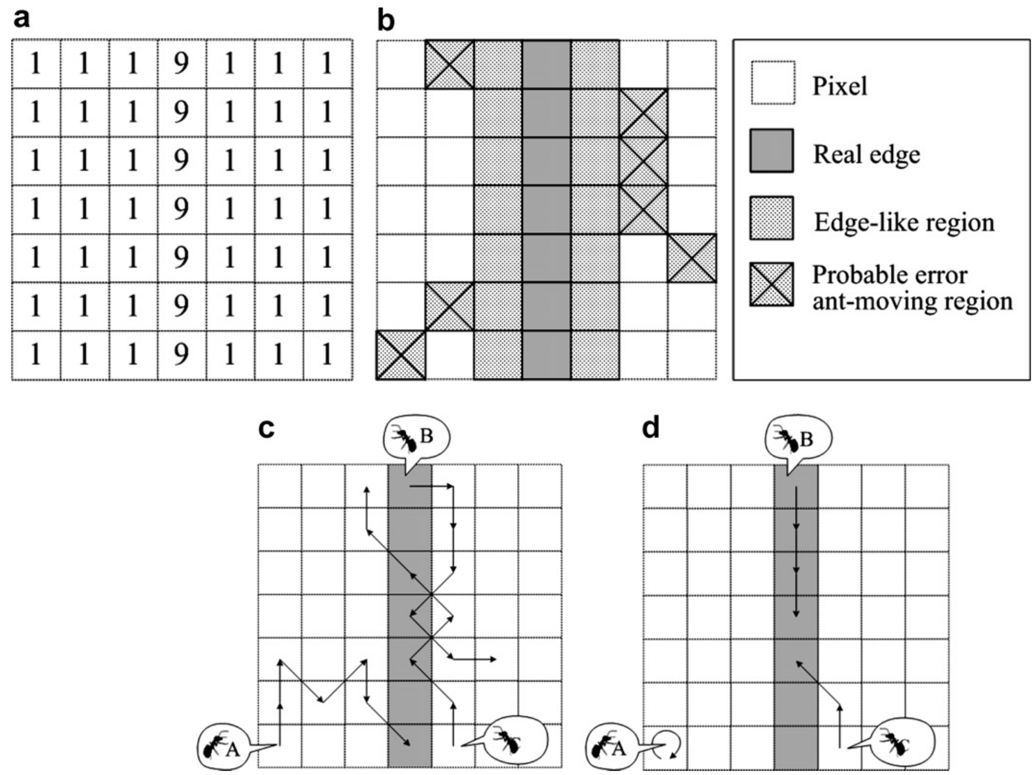

Fig. 3. (a) Original image. (b) Probable ant-moving region. (c) Irresolute searching. (d) Improved searching on redundancy-reducing policies.

$$V(p_j) = \frac{\sum_{l \in NE_j} |p_j - p_l|}{8}, \tag{2}$$

where NE_j denotes the collection of all 8-neighborhood pixels. η_{ij} in Eq. (1) represents the edge-like magnitude on the undirected path between i and j, thus $\eta_{ij} = \eta_{ji}$. A large value for η_{ij} means that the ideal edge may exist. The denominator of Eq. (1) denotes the factor of maximal connective similarity. If $V(p_j) = 0$, as defined in Eq. (2), then the ant stops here, as indicated by ant A in Fig. 3d. The ant located at an error edge cannot move to the adjacent place because of fully plane neighborhoods.

An ant goes to its next stop by the following path selection rule:

$$\text{STR} = \begin{cases} \arg\max\limits_{j \in NE_i} \{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}\}, & \text{if } q \leqslant q_0, \\ J, & \text{otherwise}, \end{cases} \tag{3}$$

$$prob_{ij} = \left\{ \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{j \in NE_i} [\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}, \quad \text{if } j \in NE_i, \right. \tag{4}$$

where η_{ij} is defined in Eq. (1); τ_{ij} denotes the pheromone trail between pixel i and j; α and β are two parameters determining the relative influence of the pheromone trail and the path visibility; q denotes a random variable, and J denotes a random variable selected according to the probability distribution given by Eq. (4). Eq. (3) indicates that a parameter q_0, satisfying $0 \leqslant q_0 \leqslant 1$, determines the probability of an ant selecting exploitation or exploration as its next stop. In exploitation, as a probability of q_0, the ant chooses the next stop by the largest relative influence. Conversely, in exploration, the ant chooses the next stop according to the probability distribution of the relative influence with the probability of $1 - q_0$.

After an ant moves from the current stand i to next stop j, the pheromone trail of the path is evaporated as follows:

$$\tau_{ij} = (1 - \xi) \cdot \tau_{ij} + \xi \cdot \tau_0, \tag{5}$$

where ξ denotes the local pheromone evaporation ratio satisfying $0 < \xi < 1$, and τ_0 is defined as the same as the initial value for the pheromone trails. The pheromone trail is globally updated after each iteration. The pheromone trail of each route, whether or not it is passed through, needs to be renewed by the following global updating rule:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}, \tag{6}$$

$$\Delta\tau_{ij} = \sum_{m=1}^{k} \frac{avg(D_m)}{\tau_{\max}}, \tag{7}$$

where ρ denotes the global pheromone evaporation ratio satisfying $0 \leqslant \rho \leqslant 1$, and $avg(D_m)$ denotes the average step length of ant m. The score of a route is the total score of resulting from the passing of all ants in an iteration. The denominator of Eq. (7) $\tau_{\max}$ is set in conjunction with the size of image as $\max\{M, N\}$, where $M \times N$ denotes the image size.

Selection of the parameters is important. Parameters α and β determines the relative influence of the pheromone trail and the path visibility. Information in image content is always more important than in pheromone trail, thus, $\beta > \alpha$ is a general selection and our experiments adopt

$\alpha = 1$, $\beta = 2$. Parameter q_0 determines the probability of the ant selecting the most important edge. The value of q_0 is always close to 1 and our experiment adopt $q_0 = 0.9$. Three parameters τ_0, ξ and ρ determine the pheromone trail change, where τ_0 is the initial value for the pheromone trails and the value is very close to 0. ξ and ρ are always small to let new pheromone trail mainly from previous pheromone trail. Our experiment adopt $\tau_0 = 0.000001$, $\xi = 0.2$, and $\rho = 0.3$. At last, finite number of iterations finishes the improving cycle. The number selection is a trade-off between compensation edges integrity and computation load, in which large number of iterations completes the compensation edges but accompanying with heavy computation load. However, too many iterations may generate too much false edges. Experimental results show that the number of iterations between 100 and 400 is adequate for acquiring compensation edges.

Fig. 4 shows the flowchart of the proposed approach, which includes the origin switching step and the improving cycle. The proposed approach improves traditional edge result in many improving cycles as iterations. The remain pheromone generated in each iteration forms the compensatory edge, which is then combined with the previous edge result to acquire a new edge result for next iteration as the origin switching step, as shown in Fig. 4. Moreover, the first edge result is the edge detected by traditional edge detection approach. Edges are gradually compensated according to the increase in number of improving cycles. The image edge is improved from the traditional edge detection approach after a finite number of iterations of improving cycle.

4. Experimental results

This section presents the experimental results of our improvements in traditional edge images filtered by the Sobel and Canny edge detection approaches. Comparisons with public available Multiresolution Sequential Edge Linking method (M-SEL) are provided (Cook and Delp, 1995). The results of ACO with and without the proposed redundancy-reducing policies were also compared. In these experiments, traditional edge detectors are executed by MATLAB toolbox. Both the proposed ACO based approach and traditional ACO were programmed by Visual Studio.NET. Our experiments adopt parameters as illustrated in Section 3.2.

Fig. 5 depicts the proposed compensation result of an edge image detected by the Sobel operator. Fig. 5a shows the original image. Fig. 5b presents the Sobel edge detection image with a threshold of 0.05 in MATLAB, in which the left and upper-right edges are lost. These lost edges are compensated by the proposed approach in 200 iterations,

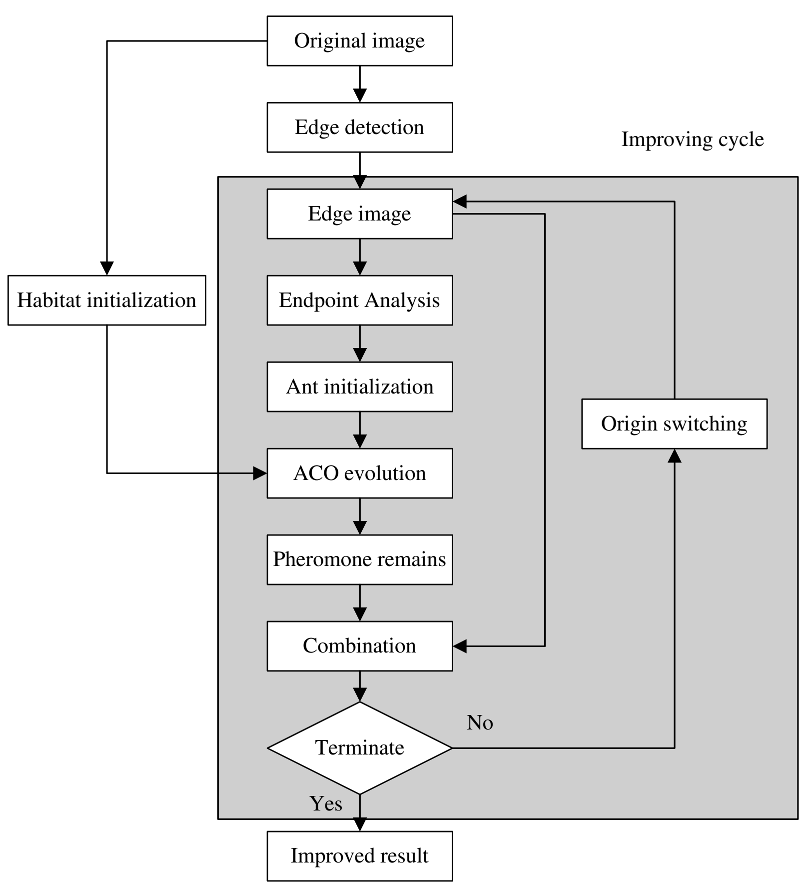

Fig. 4. Flowchart of the proposed approach.

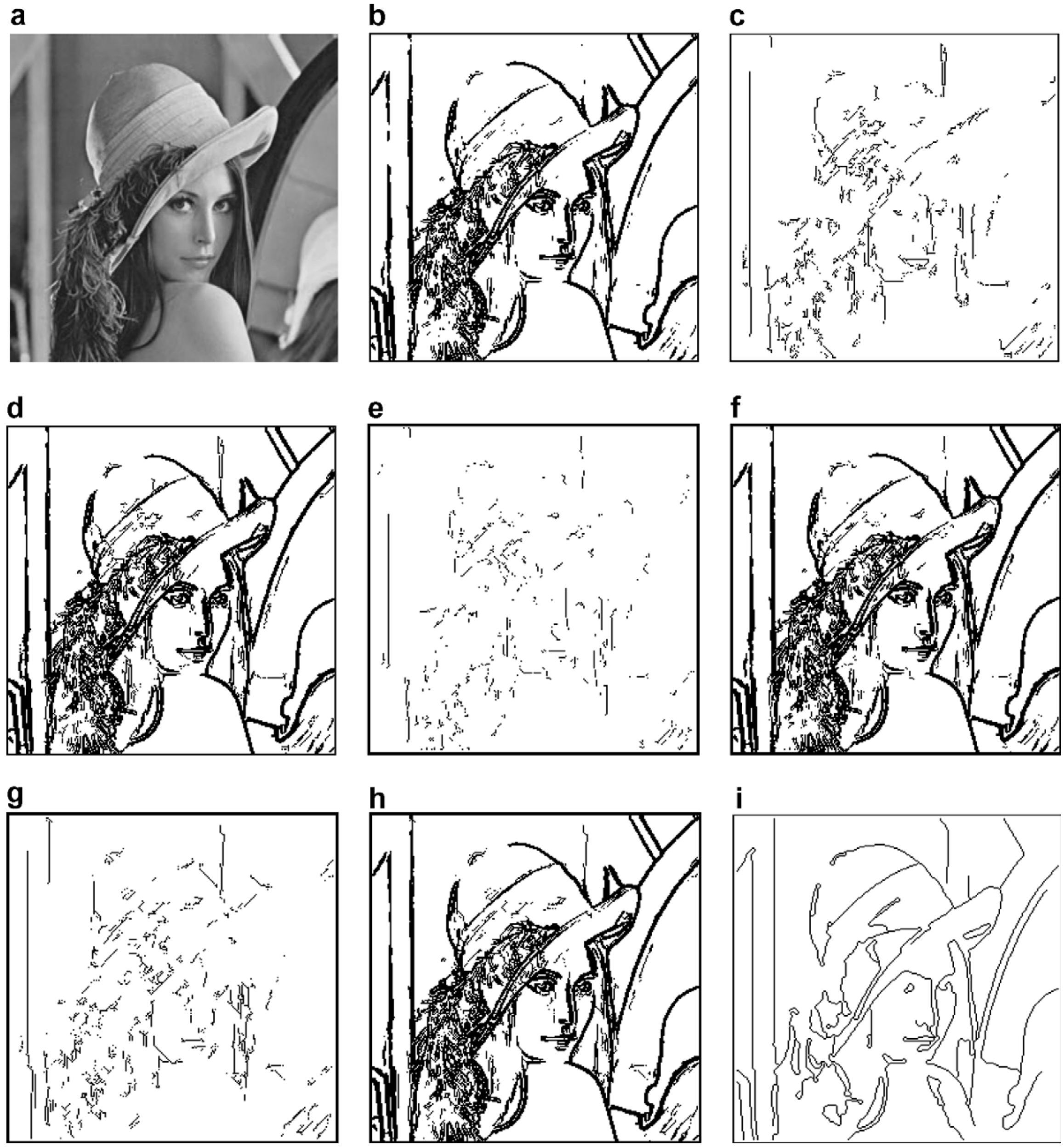

Fig. 5. (a) Original image. (b) Performance of the Sobel edge detector (threshold = 0.05). (c) The compensation with 200 iterations. (d) The improved result from the proposed approach. (e) The compensation with 200 iterations of adjusting parameters $\xi = 0.9$, and $\rho = 0.8$. (f) The improved result of merging (b) and (e). (g) The compensation with 200 iterations of adjusting parameters $\alpha = 10$, $\beta = 0.1$. (h) The improved result of merging (b) and (g). (i) The edge image detected by the M-SEL method with 30 minimum root nodes.

as shown in Fig. 5c. Fig. 5d depicts the combination of Fig. 5b and c, which achieves the improved result. Two examples of adopting different parameters are illustrated in Fig. 5e and g. Fig. 5e shows the compensation with 200 iterations of adjusting parameters $\xi = 0.9$, and $\rho = 0.8$, the other parameters are listed in Section 3.2. Large values of ξ and ρ lead to high probability of searching new paths. Thus, many short compensation edges existed in Fig. 5e. Fig. 5f depicts the improved result of merging Fig. 5b and e. Fig. 5g shows the compensation with 200 iterations of adjusting parameters $\alpha = 10$, $\beta = 0.1$. Selection of $\alpha > \beta$ lets ants select next step mostly by pheromone trail, rather than the path visibility. Many short compensation edges that not placed on real edges are then acquired. Fig. 5h depicts the improved result of merging Fig. 5b and (g). Consequently, parameters selection determines ants to select next step mainly by pheromone trail, path visibility, or new path. Fig. 5i compares with the edge image detected by the M-SEL method with 30 minimum root nodes.

Fig. 6 shows another result of Sobel edge detection. Fig. 6a shows the original image “house”. Fig. 6b shows the edge image detected by the Sobel operator with a threshold of 0.25 in MATLAB, revealing that this lower

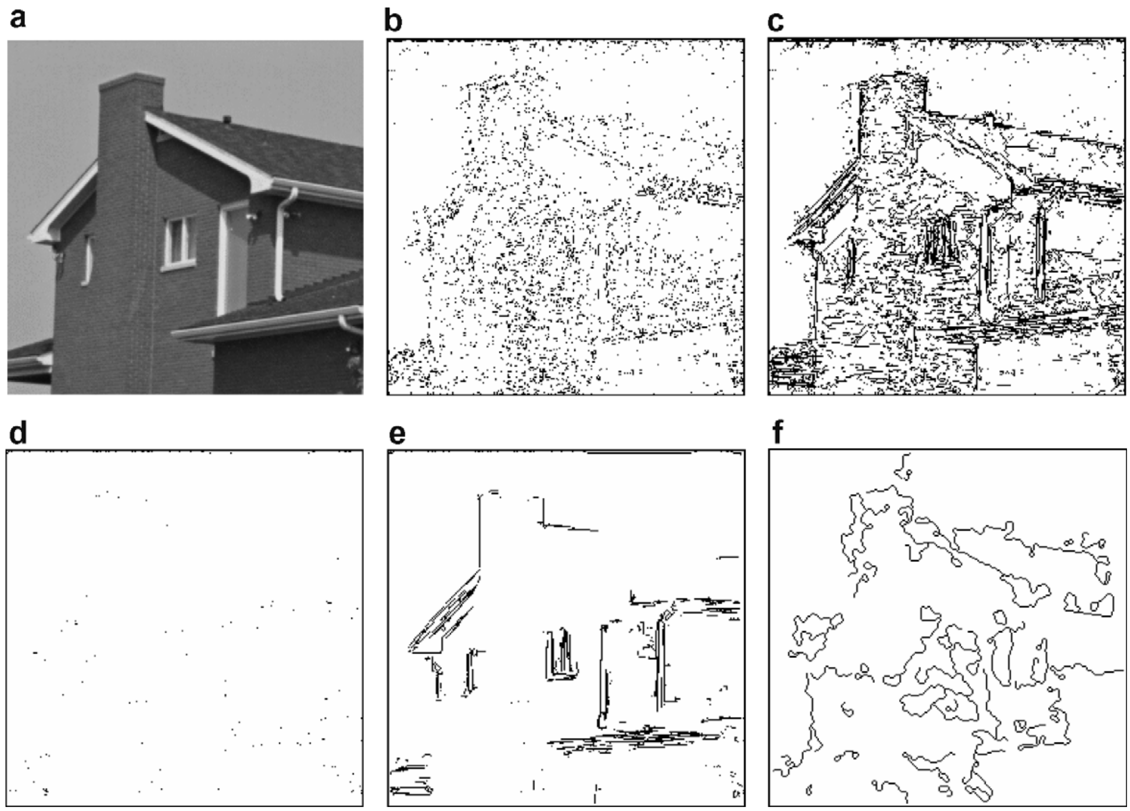

Fig. 6. (a) Original image. (b) Performance of the Sobel edge detector (threshold = 0.25). (c) The improved result by the proposed approach with 180 iterations. (d) Performance of the Sobel edge detector (threshold = 0.38). (e) The improved result by the proposed approach with 420 iterations. (f) The result by the M-SEL method with 150 minimum root nodes.

threshold detects many noises and broken edges. Fig. 6c indicates that the proposed approach improve the edge detection with 180 iterations. Fig. 6d and e demonstrates other experimental results based on a threshold of 0.38 and 420 iterations. Fig. 6d, comparing with Fig. 6b, shows that fewer edges are filtered with this higher threshold. Fig. 6e shows the improved result by the proposed approach with 420 iterations, in which some important edges are detected. Higher threshold detects only some edges as shown in Fig. 6d, and therefore, this experiment selects larger iterations as 420 to compensate a lot of edges. A comparison with the M-SEL method with 150 minimum root nodes is illustrated in Fig. 6f. Experimental results reveal that the proposed approach can compensate most important edges only from a few broken edges.

Fig. 7 shows the results of applying the Canny edge detection approach. Fig. 7b displays the Canny edge detection result with parameters ($\sigma = 0.2$, low threshold = 0.55, high threshold = 1.0) in MATLAB. Fig. 7c depicts the result of applying the proposed approach with 160 iterations on edge image Fig. 7b. Fig. 7c shows that important edges in eaves are almost compensated to acquire a better edge image.

Results in Figs. 5–7 indicate that the proposed approach compensates important edges efficiently, even if they are not detected by traditional edge detection approach, as in Figs. 6d and 7b. The proposed method also creates some noise edges, since traditional edge detected images include some noise-like edges, which tends to generate noise-like compensation edges when ants are placed on them.

Fig. 8 shows the proposed compensation result of edge images from two noise images detected by the Sobel operator. Fig. 8a shows the noise image as adding Gaussian noise with mean and variance of 0 and 0.01 in MATLAB. Fig. 8b presents the Sobel edge detection image with a threshold of 0.19 in MATLAB. Fig. 8c demonstrates the compensation result with 180 iterations. Fig. 8d–f depicts another example of different Gaussian noise added image. Fig. 8d shows the noise image as adding Gaussian noise with mean and variance of 0 and 0.02 in MATLAB. Fig. 8e presents the Sobel edge detection image with a threshold of 0.26 in MATLAB. Fig. 8f demonstrates the compensation result with 180 iterations. These two examples show that some broken edges are compensated. However, broken or false edges are still existed mainly from the high probability of ants reaching to the noise pixels at the image edges when an image is added noises. More Gaussian noises added leads to more false edges generated. Fig. 8f generates much more false edges than Fig. 8c does due to more noises added. Thus, the proposed approach compensates image edges well when low noise added. Moreover, few false edges still exist in the improved results shown in Figs. 5–8 because of ants preserving certain probability to search new paths and these paths mostly forms false edges.

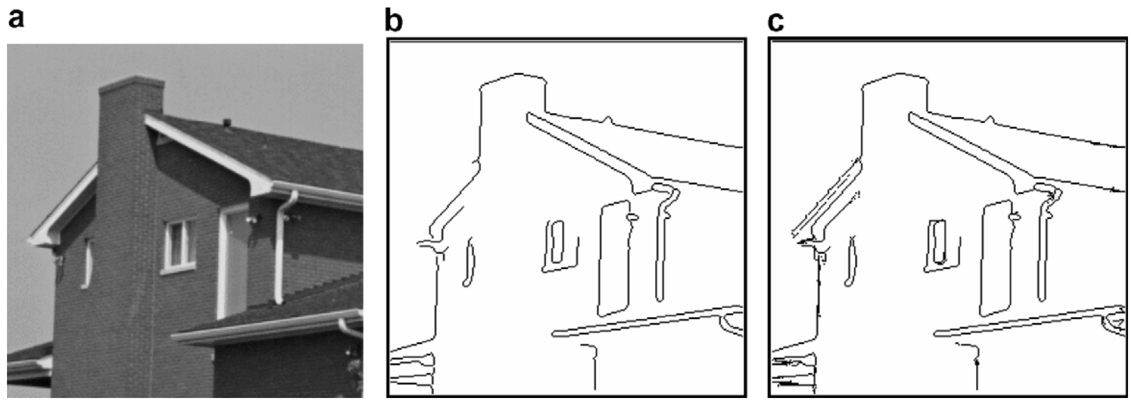

Fig. 7. (a) Original image. (b) Performance of the Canny edge detector ($\sigma = 0.2$, low threshold = 0.55, high threshold = 1.0). (c) The improved result from (b) with 160 iterations.

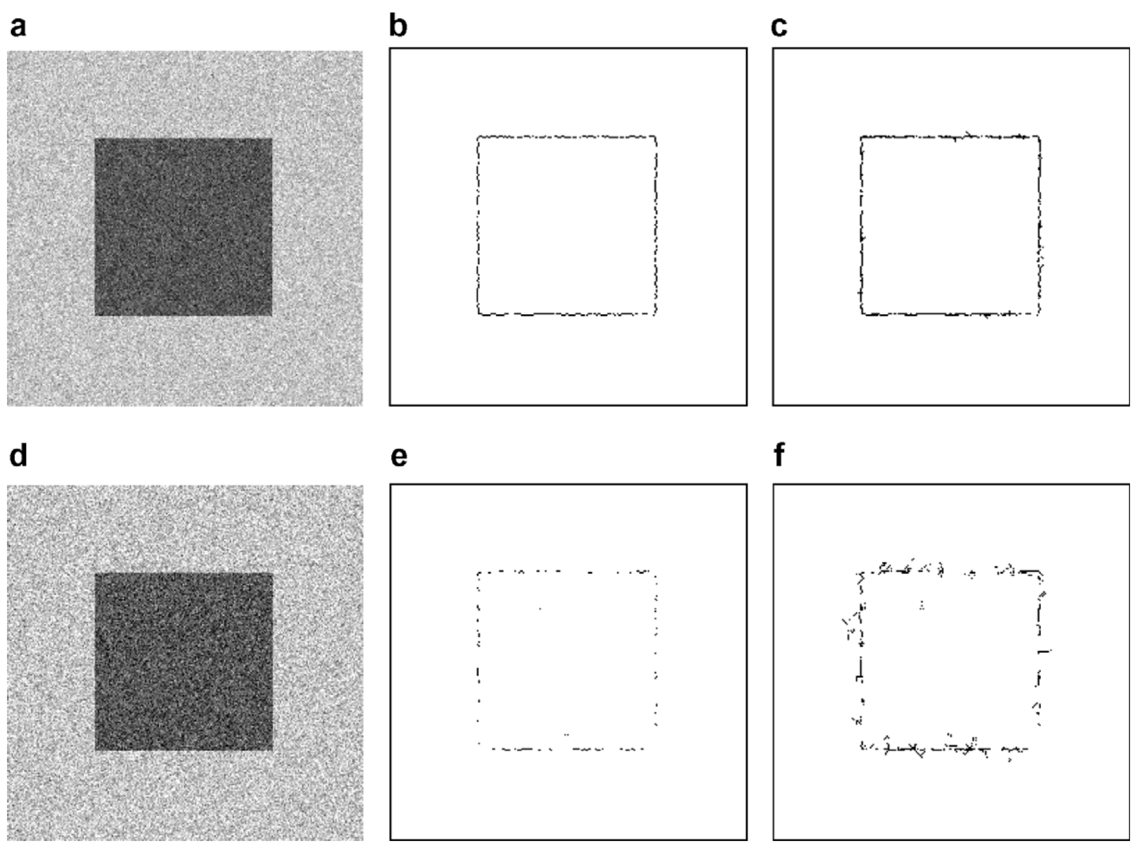

Fig. 8. (a) Original image with random noise. (b) Performance of the Sobel edge detector (threshold = 0.19). (c) The improved result by the proposed approach with 180 iterations. (d) Original image with random noise. (e) Performance of the Sobel edge detector (threshold = 0.26). (f) The improved result by the proposed approach with 180 iterations.

The traditional and proposed improved ACO were also compared. Fig. 9 compares the compensated results of the proposed improved ACO and traditional ACO. Fig. 9b shows the edge image by the Canny edge detection approach with parameters ($\sigma = 0.1$, low threshold = 0.2, high threshold = 1). Fig. 9c shows the result of edge improvement by traditional ACO. The proposed improved ACO approach adopts four redundancy-reducing policies to improve the edges as shown in Fig. 9d. Experimental results illustrate that the proposed approach reduces fake edges and noise more effectively than traditional ACO.

Table 1 shows the computation times of the experiments shown in Figs. 6–8 with the same 200 iterations. The computation times of previous M-SEL method are also included. The computation time in the proposed approach

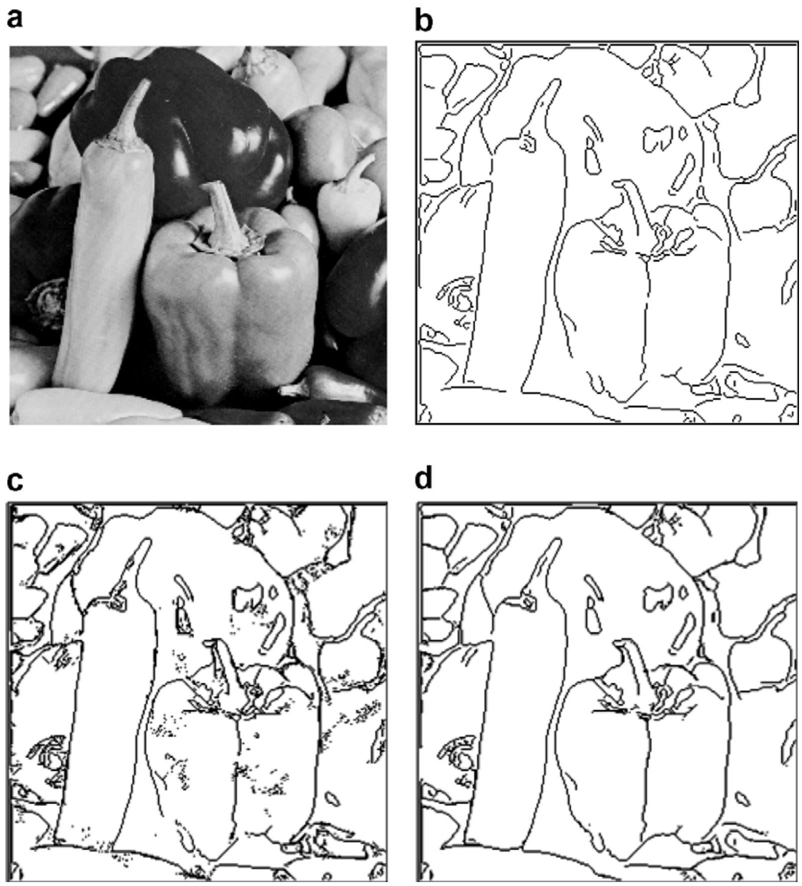

Fig. 9. (a) Original image. (b) Performance of the Canny edge detector ($\sigma = 0.1$, low threshold = 0.2, high threshold = 1). (c) The improved result by traditional ACO. (d) The improved result by the proposed approach.

was almost half that of the traditional ACO, although they are larger than that of M-SEL method. Fig. 10 estimates the computation time of image Lena every 20 iterations. Experimental results show that computation time of the proposed approach was very stable. In contrast, traditional computation time of ACO varied significantly. These results show that the proposed ACO approach significantly reduces the number of uncertain and redundant searches. Consequently, the proposed approach with four redundancy-reducing policies produces fewer fake edges and has a lower computation complexity than the traditional ACO approach.

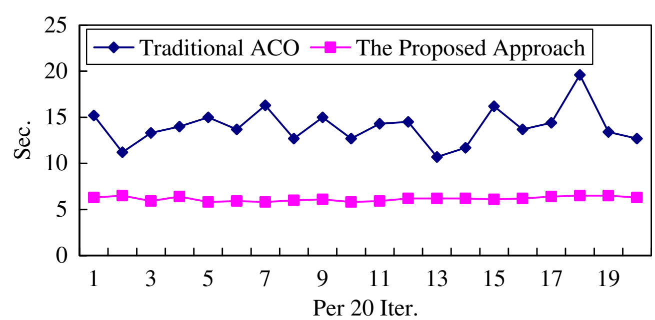

Fig. 10. Computation time comparison per 20 iterations.

Table 1
Computation comparison between traditional ACO, the proposed approach and M-SEL approach

Image	Ants	Iter.	Computation time					
			Traditional ACO		The proposed approach		ACO ratio	M-SEL time (s)
			Time (s)	Avg. (Iter.)	Time (s)	Avg. (Iter.)		
Peppers (Fig. 8(b))	199	200	104.2	0.521	54.5	0.2725	0.523	3.7
Lena (Fig. 6(d))	265	200	139.7	0.6985	64.7	0.3235	0.4631	4.3
House (Fig. 7(c))	138	200	97.5	0.4875	51.9	0.2595	0.5323	4.8
Total	602	600	341.4		171.1			
Avg. (Ants)				0.000945183		0.000473699	0.5012	

5. Conclusions

This study presents a novel method that adopts ACO to improve traditional edge detection. Ants in the proposed method break the limited range of local analysis, and proceed along the edge-like path to extract whole compensation edges. Additionally, the global pheromone updating rules and path length measurement are adjusted. This method is effective because the proposed redundancy-reducing policies can shorten the ant searching path and reduce the computation load. Experimental results indicate that the proposed method efficiently improves edge images detected by traditional approaches like Sobel or Canny. Reducing the computation complexity of ACO merits future study.

Acknowledgements

The authors would like to thank the reviewers for valuable comments.

References

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal Machine Intell. 8, 679–698.

Cook, G.W., Delp, E.J., 1995. Multiresolution sequential edge linking. In: Proc. IEEE Internat. Conf. on Image Processing, October, pp. 41–44.

Dorigo, M., Stűtzle, T., 2004. Ant Colony Optimization. MIT Press.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The ant system: Optimization by a colony of cooperating agents. IEEE Trans. Systems Man Cybernet. – Part B 26, 1–13.

Ghita, O., Whelan, P.F., 2002. Computational approach for edge linking. J. Electron. Imag. 11, 479–485.

Gonzales, R.C., Woods, R.E., 2002. Digital Image Processing. Prentice Hall.

Hajjar, A., Chen, T., 1999. A VLSI architecture for real-time edge linking. IEEE Trans. Pattern Anal. Machine Intell. 21, 89–94.

Ng, C.M., Leung, W.L., Lau, F., 1999. Edge detection using evolutionary algorithms. In: IEEE Internat. Conf. on System, Man and Cybernetics, October, pp. 865–868.

Parker, J.R., 1997. Algorithms for Image Processing and Computer Vision. Wiley Computer Publication.

Sharifi, M., Fathy, M., Mahmoudi, M.T., 2002. A classified and comparative study of edge detection algorithms. In: IEEE Proc. Internat. Conf. on Information Technology: Coding and Computing, April, pp. 117–120.

Vallone, U., Mërigot, A., 2003. Imitating human visual attention and reproduction optical allusion by ant scan. Internat. J. Comput. Intell. Appl. 3, 157–166.

Zheng, H., Wong, A., Nahavandi, S., 2003. Hybrid ant colony algorithm for texture classification. In: IEEE Congress on Evolutionary Computation Proc., pp. 2648–2652.

Zhuang, X., Mastorakis, N.E., 2005. Image processing with the artificial swarm intelligence. WSEAS Trans. Comput. 4, 333–341.