

## A Recursive Ant Colony System For Feature Extraction & Edge Detection

Miss. Hiteshri S.Khandre  
*M.Tech IV Sem, Computer Science & Engg.*  
*RTM Nagpur University ,*  
*Rajiv Gandhi College Of Engg., Research &*  
*Technology, Babupeth, Chandrapur,*  
*India.(+91)*

Prof. P.S.Kulkarni  
*Department Of Information Technology,*  
*Rajiv Gandhi College Of Engg., Research &*  
*Technology, Babupeth, Chandrapur,*  
*India(+91)*

### Abstract

*Edge detection is one of the important techniques in image processing which is used in many applications. It generally detects the contour of an image and thus provides important details about an image which can further be used in tasks like object recognition and image segmentation. The most important step in the edge detection, on which the success of generation of true edge map depends, lies on the determination of threshold. This paper gives nature inspired solution like ant colony optimization (ACO) in edge detection and feature extraction. Recursive ant colony optimization technique is proposed here to find out the solution. The success of the proposed work is tested visually with the help of test images and empirically tested on the basis of several statistical parameters of comparison. Wavelet thresholding method is used to check the performance of feature extraction method.*

### Keywords

*Recursive Ant Colony System, Edge detection, Partitioning, 2-opt, ACS, Heuristic Search*

## I. INTRODUCTION

Image processing and machine vision have for long been a vital element in various fields of technology. While automated and artificially intelligent systems in many cases require a significant amount of precise and robust image processing capabilities, robust techniques are yet to be proposed and demonstrated for many ongoing problems. Image edge detection refers to the extraction of the edges in a digital image. It is a process whose aim is to identify points in an image where discontinuities or sharp changes in intensity occur. This process is crucial to understand the content of an image and has its applications in image analysis and machine vision. As a meta-heuristic algorithm, ant colony optimization (ACO) has the features of robust, parallel, positive feedback, which prove it to be a useful means for searching optimal results from the problem. ACO has been used to solve the problem like Edge detection & Feature

extraction. Edge is one of the simplest and the most important features of image, and this feature is broadly used in image recognition, segmentation, enhancement and compression. The purpose of edge detection is not only to extract the edges of the interested objects from an image, but also to lay the foundation for image fusion, shape extraction, image segmentation, image matching, and image tracking. Ant colony optimization (ACO) is one of the most recent techniques for approximate optimization, which is based on the foraging behavior of real ants. Ants communicate by means of a kind of substance called pheromone which can enable them to find the shortest paths (the most preferred ways) between their nest and food sources. Edge detection is an important content of image processing and low-level computer vision. Edges represent important contour features in image. Image edge is usually defined as the sudden change areas of gray value. The basic idea of the traditional gray gradient-based edge detection algorithms is the comparison of pixel's gray value. Therefore, pixel's gray gradient value can be used to be the feature of edge

Automatic detection of edges in image is a classical problem in computer vision and image processing. Image edge detection refers to the extraction of the edges in a digital image. It is a process whose aim is to identify points in an image where discontinuities or sharp changes in intensity occur. This process is crucial to understand the content of an image and has its applications in image analysis and machine vision. As a meta-heuristic algorithm, ant colony optimization (ACO) has the features of robust, parallel, positive feedback, which prove it to be a useful means for searching optimal results from the problem. ACO has been used to solve many complex problems successfully such as the TSP, quadratic assignment problem, graph coloring, and data mining. Compared with other heuristic algorithms, ACO is a population-based approach which uses exploitation of positive feedback as well as greedy search. Because of its

parallel and discrete features, ACO is more suitable for image processing problems, such as segmentation, feature extraction, image matching and texture classification. Here our aim is to design an algorithm for image edge extraction which can be tuned using different parameters for satisfying performance in the presence of noise.

The rest of this paper is organized as follows. Section II presents basics of principle component analysis. Section III gives mechanism of PCA. Section IV review of existing algorithms. Section V describes the Ant Colony System in detail. Section VI presents Recursive Ant Colony System, a new approach based on Ant Colony System. Finally, in Section VII, we report the conclusion.

## II. REVIEW OF EXISTING ALGORITHM

### A. An Ant Colony Optimization Algorithm for Image Edge Detection<sup>[4]</sup>

Gradient feature is simple but it is sensitive to noise and texture. Relative Difference of Statistical Means have strong ability to suppress noise but edge information may lose. Therefore authors have combined gray gradient value of pixel and relative difference of statistical means to image edge detection.

$$\Delta E_{\theta_n} = 0, E^1_{\theta_n} + E^2_{\theta_n} = 0$$

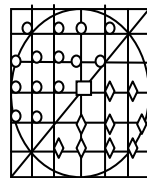
$$\frac{2|E^1_{\theta_n} - E^2_{\theta_n}|}{E^1_{\theta_n} + E^2_{\theta_n}}, E^1_{\theta_n} + E^2_{\theta_n} \neq 0$$

$\Delta E_{\theta_n}$  is the relative difference of Statistical Means.

$$E^k_{\theta_n} = \frac{\sum_{x,y \in D} f(x,y)}{N} \quad k=1,2; n=0-3$$

$$\theta_n = n\Delta\theta$$

$$\Delta\theta = \pi/4$$



### Probability Decision

$$P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta w_{ij}(\Delta)}{\sum_{j \in \Omega} (\tau_{ij})^\alpha (\eta_{ij})^\beta w_{ij}(\Delta)}$$

Where  $i, j \in \Omega$  indicates all the pixels that are in the 8-neighborhood of the pixel  $(i, j)$ .

$$\eta_{ij} = a\Delta E$$

$$\Delta E = \max\{\Delta E_{\theta_n}\}$$

$w(\Delta)$  is a weighting function, this function ensures that very sharp turns are much less likely than turns through smaller angles.

The feature of a pixel  $(i, j)$  is presented as below:

$$F_{ij} = a \cdot \frac{\Delta I_{ij}}{\Delta I_{\max}} + b \cdot \Delta E_{ij}$$

Where  $a$  and  $b$  are weighting factors.  $\Delta I_{\max}$  denotes the maximum value of the gradient in image.

### Decision Process

Finally, a binary decision is made at each pixel location to determine whether it is on the edge or not, by applying a threshold  $T$  on the final pheromone matrix.

### B. Ant Colony Search for Edge Detection<sup>[5]</sup>

In this paper, a heuristic ant colony search algorithm is proposed to overcome the shortcoming of traditional edge extracting methods. Algorithm uses Sobel operator to get the possible edge points.

**Heuristic measure** is the key process used in this paper and given by following equation. Heuristic information is related to the gradient and phase of the transition route.

$f(i,j)$  -> gradient of node  $i,j$   
 $\theta(i,j)$  -> phase of node  $i,j$   
 $\lambda_1, \lambda_2$  -> constants where  $\lambda_1 + \lambda_2 = 1$   
 $\Delta\Phi_{(i,j)}^{(r,s)}$  -> directional difference of ant move from node  $(r, s)$  to  $(i, j)$

$$\eta_{(i,j)}^{(r,s)} = w(\Delta\Phi_{(i,j)}^{(r,s)}) \left[ \lambda_1 f_{(i,j)} + \lambda_2 \frac{1}{|\theta_{(i,j)} - \theta_{(r,s)}|} \right]$$

### Stopping criterion

$S_{\max}$  is the stop threshold

Stopping point is the end of the edge transited by the ant. In each step of transition, the ant estimates whether the stopping criterion is satisfied. If satisfied, the search will stop on the current node; otherwise it will search the next transition node repeatedly until satisfying the criterion.

$$P(r,s) = e^{-M(r,s)S_{\max}}$$

$$u_{(i,j)}^{(r,s)} M_{(r,s)} > S_{\max}$$

$$\text{where } M_{(r,s)} = \sum_{(u,v) \in \Omega} [\tau_{(u,v)}^{(r,s)}]^\alpha [\eta_{(u,v)}^{(r,s)}]^\beta$$

We can conclude from this formula, the smaller of the pheromone and heuristic measure on the point,

the higher probability of the search stopped at that point.

### C. An ant-inspired algorithm for detection of image edge features<sup>[6]</sup>

The proposed model is based on the fact that an image is composed of a number of pixels, creating a map of cells. A neighborhood is defined for each pixel which identifies where the ants are permitted to move next. Pheromone is a decisive component in ant colony algorithms. Authors have defined two types of pheromone for the problem.

Each ant lays a trail of pheromone type-I as it forages through the 2D map. Each ant is assigned a short term memory, which it uses to remember its last place that it visited also to follow a constraint:

**Constraint:** An ant is not permitted to directly return to its previous cell.

Applying constraint significantly increases the mobility of the agents. If each ant is influenced by any type of pheromone in the environment, it can still be easily entangled in a loop of pheromone laid by it or other agents. To avoid this situation, we define pheromone type-II. This type of pheromone is the component responsible for the decision making process of the ants.

#### Post-processing

The threshold value determines the minimum amount of normalized pheromone required for a pixel to be accepted as an extracted feature. To be able to evaluate the features with regard to other edge detection methods, a threshold value, T, is introduced into the procedure.

### D. Edge Detection Using Adaptive Thresholding and Ant Colony Optimization<sup>[18]</sup>

#### Edge Detection using Adaptive Thresholding

In the proposed approach, initially edges are extracted using adaptive thresholding. The connectivity of the edges so obtained is then increased using modified ACO.

Adaptive thresholding typically takes a gray scale or color image as input and, outputs a binary image representing the edge information. For each pixel in the image, a threshold is calculated. If the pixel value is below the threshold it is set to the background value, otherwise it assumes the foreground value. The edges obtained using adaptive thresholding contains some thick edges also therefore a thinning algorithm is implemented for the pre-processing for an efficient end point analysis. The processed image is then analysed to obtain the end point information of the

broken edges. The edges extracted from the above steps provide larger end point information as compared with that provided by Sobel operator.

#### Edge Improvement

Several discontinuities appear in the image after the application of adaptive thresholding.

The ACO algorithm is used to increase the connectivity of the edges in the image obtained after applying local adaptive thresholding. The steps are as follows:

- 1) Initialize the ant's position by placing them only at end points.
- 2) Initialize the pheromone matrix and calculate the heuristic information using Eq.,

$$\eta_{ij} = \frac{\max_j \begin{bmatrix} |I(x-1, y-1) - I(x+1, y+1)| \\ |I(x-1, y+1) - I(x+1, y-1)| \\ |I(x, y-1) - I(x, y+1)| \\ |I(x-1, y) - I(x+1, y)| \end{bmatrix}}{\eta_{\max}}$$

- 3) Construction Process:

For the ant index 1 : k

Move the kth ant for L steps according to the probabilistic transition matrix.

$$P_{ij}^{(n)} = \frac{(\tau_{ij}^{(n)})^\alpha (\eta_{ij})^\beta}{\sum_{j \in \Omega_i} (\tau_{ij}^{(n)})^\alpha (\eta_{ij})^\beta}$$

where  $\tau_{ij}^{(n)}$  is the pheromone information value of the arc linking the node i to the node j,  $\eta_{ij}$  represents the heuristic information for pixel (x,y) for going from node i to node j.

- 4) Calculate maximum probability of transition as per the *transition rule* and move the ant accordingly.
- 5) Perform local pheromone update process.
- 6) Check whether all ants have moved one step, if yes, perform the global pheromone update.
- 7) Check whether the ant can move to the next position by applying the stopping rules, if not, stop the ant.

Stopping rules:

- The movement of the ant is stopped when it touches the track already traversed by another ant.

- When all the neighboring pixels (8 pixels in 3\*3 grid) are already traversed by the ant, then the movement of ant stops.

#### 8) Decision Process:

The pheromone matrix so produced is used to extract the complete edge trace by applying thresholding.

9) The edge pixels obtained are combined with the edge pixels obtained by adaptive thresholding to get the complete edge information.

Finally, the results are analyzed using Shannon's Entropy function.

### III. ANT COLONY SYSTEM<sup>[7]</sup>

AS was the first algorithm inspired by real ants behavior. AS was initially applied to the solution of the traveling salesman problem but was not able to compete against the state-of-the art algorithms in the field.

#### A. General ACS algorithm

1. Initialize pheromone trails and place  $M$  ants on the nodes of AS graph

2. Repeat until system convergence

2.1 For  $i = 1$  to  $n$

2.1.1 For  $j = 1$  to  $M$

2.1.1.1 Choose the node  $s$  to move to, according to the transition probability specified in (1).

2.1.1.2 Move the ant- $k$  to the node  $s$

2.2 Update the pheromone using the pheromone update formula (3)

Ants change pheromone level of edges by applying local updating rule as described in equation (3).

$$\tau(i,j) \leftarrow (1-\rho) \cdot \tau(i,j) + \rho \cdot \tau_0 \text{ -----(3)}$$

where,  $0 < \rho < 1$  is the coefficient representing pheromone evaporation, and  $n$  is the number of cities and  $\tau_0 = (n \cdot L_{nn})^{-1}$ , where  $L_{nn}$  is the tour length produced by nearest neighbour heuristic [13]. After all ants complete their cyclic tour, only the globally best ant (i.e. ant belonging to shortest tour) changes trail following global updating rule as given in equation (4).

$$\tau(i,j) \leftarrow (1-\alpha) \cdot \tau(i,j) + \alpha \cdot \Delta \tau(i,j) \text{ -----(4)}$$

where,  $0 < \alpha < 1$  is pheromone decay parameter,  $L_{gb}$  is the length of globally best tour, and

$$\Delta \tau(i,j) = (L_{gb})^{-1}, \text{ if } (i,j) \text{ global best tour} \\ 0, \text{ otherwise -----(5)}$$

Global updating rule is similar to a reinforcement learning process as in this case better solutions get higher reinforcement, thus providing high amount of trail to shorter tours.

#### B. RECURSIVE ANT COLONY SYSTEM

The Recursive Ant Colony System (RACS) algorithm applies a partitioning scheme to the problem in a manner analogous to the recursive merge sort based on the divide and conquer technique. The algorithm is based on the fact that the efficiency of Ant Colony applications is better for problems of smaller size having less number of cities. This occurs due to the random nature of the algorithm, in which a large number of good random decisions made on weighed choices are required to come together to construct an efficient solution and as the size of the problem increases, so do the number of decisions to be made to generate a single tour. The RACS algorithm partitions the set of all nodes for a problem; say  $S$ , into two disjoint sets, say  $S1$  and  $S2$ , and then proceeds to find solutions independently for the two sub-problems now created by focusing on reducing the lengths of the segments formed by these sets in the original tour, keeping the end points of any new path same as that in the original path. As the search space for these sub-problems gets reduced, resulting from the division of the nodes for the original problem, the exploration efficiency and hence the accuracy of the ACS algorithm is much greater for these sub-problems. The accuracy of the overall solution obtained by the conjunction of the solutions obtained from these sub-problems is upper-bounded by the accuracy of the division of the nodes for each subset, which in turn depends upon the accuracy of the initial candidate tour generated. Thus, the RACS algorithm employs a strategy of generating a candidate tour initially using an iterative ACS procedure, followed by partitioning of the tour and recursive implementation of the ACS and Greedy 2-opt(for symmetric TSPs) algorithms on the sub-problems created at each recursive level, to further improve the candidate solution initially generated. Recursive ACS is based on Ant Colony System. This technique is used to solve Traveling Salesman Problem. Now the same technique we have applied on Edge Detection & Feature Extraction problem. The proposed ACO-based image edge detection approach aims to utilize a number of ants to move on a 2-D image for constructing a pheromone matrix, each entry of which represents the edge information at each pixel location of the image.

In edge detection problem before applying RACS we have to apply ACS on the problem, in which first the image has been read. Parameters like:

$p = 0.0001 \cdot \text{ones}(\text{size}(\text{img}))$

$\alpha = 10$

$\beta = 0.1$

$\rho = 0.1$

$\phi = 0.05$

$T=0.01$  has been initialized.

Heuristic function has been initialized & calculated. In feature extraction module, color feature has been considered as an extracted feature, in which RACS has been implemented in similar manner as in edge detection using RACS. After calculation of pheromone & heuristic information, histogram technique is used for color feature extraction. After comparing the Feature extraction using ACS & RACS, we concluded that the execution time taken by feature extraction using RACS is far less than that of using ACS.

### IV RESULT & DISCUSSIONS

The figure of merit is the useful measure for assessing the performance of edge detectors. This measure uses the distance between all pairs of points corresponding to quantify, with precision, the difference between the contours. The figure of merit, which assesses the similarity between two contours is defined as:

$$FM = \frac{1}{\max(N_i, N_b)} \sum_{i=1}^{N_b} \frac{1}{1 + \alpha * d_i^2}$$

where  $N_i$  &  $N_b$  are the points of edges in the image and ground truth image, respectively,  $d_i$  is the distance between a edge pixel and the nearest edge pixel of the ground truth and  $\alpha$  is a empirical calibration constant and was used  $\alpha = 1/9$ . The figure of merit is an indicator of the quality of edge, and reflects the overall behavior of the distances between the edges, being a relative measure, which varies in the range [0,1], where 1 represents the optimal value, i.e., the edges detected coincide with the ground truth.

We have test the algorithm on following image for edge detection

- 1) Cameraman
- 2) Clown
- 3) Lena

Results for algorithm is shown as follows:



1) Fig: Input Image



Fig: Flat Function

Fig: Square Function



Fig: Sine Function

Fig: Wave Function



2) Fig: Clown



Fig: Flat Function

Fig: Square Function

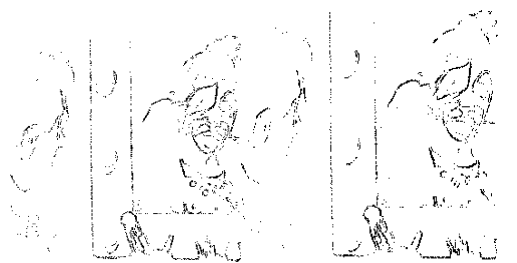


Fig: Sine Function

Fig: Wave Function





3) Fig : Lena



Fig: Flat Function



Fig: Square Function

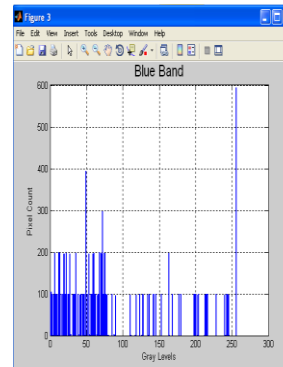
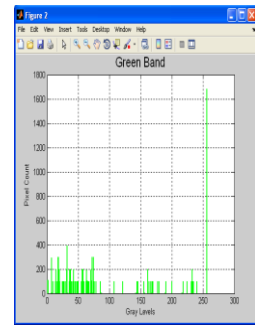
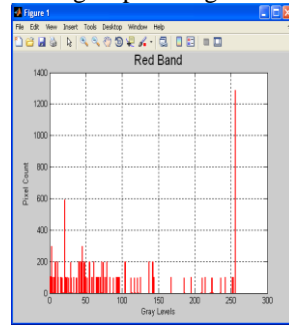


Fig: Sine Function



Fig: Wave Function

Fig: Input Image



Thus, RACS also work better for feature extraction. Color feature has been taken for extraction. Above diagram shows no. of different color pixels (R,G,B) has been extracted.

### V. CONCLUSION

Ant Colony Optimization algorithm is widely used in edge detection problems. Here we proposed Recursive ant colony optimization algorithm to solve edge detection and feature extraction. Algorithm for edge detection is implemented on three images like clown, cameraman and Lena. Performance is evaluated on parameters like time and figure of measure. RACS performs better as compared to original ACS. RACS also performs better for Feature extraction. The comparison is done on execution speed. Future work is to used other evolutionary techniques in edge detection and compare the result with proposed work.

### REFERENCES

[1] M. Dorigo, "Ant colony optimization".  
 [2] A. Colomi, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, Proceedings of ECAL'91, *European Conference on Artificial Life*, Elsevier Publishing, Amsterdam, 1991.  
 [3] M. Dorigo, V. Maniezzo, and A. Colomi, The ant system: an autocatalytic optimizing process, *Technical Report TR91-016*, Politecnico di Milano (1991).

**Table: Performance Evaluation for ACS & RACS**

Image	ACS			RACS		
	Time	Pixels	FOM	Time	Pixels	FOM
Clown	282	2020	0.023	0.37	1944	0.024
Cameram an	221	2282	0.74	0.09	2074	0.7
Lena	685	2289	0.07	0.59	2161	0.07

Above images were tested upon ACS & RACS. It executes extremely fast for RACS. It also performs better in terms of figure of merit for two images.

For feature extraction it executes extremely fast with RACS as compared to ACS.



- [4] Jian Zhang, Kun He, Xiuqing Zheng, Jiliu Zhou, "An Ant Colony Optimization Algorithm for Image Edge Detection", International Conference on Artificial Intelligence and Computational Intelligence, IEEE, 2010, pp 215-219.
- [5] Yanfang Che, Yong Yu, "Ant Colony Search for Edge Detection", 4th International Congress on Image and Signal Processing, IEEE 2011, pp 874-878.
- [6] S. Ali Etemad, Tony White, "An ant-inspired algorithm for detection of image edge features", Applied Soft Computing 11 (2011) 4883–4893.
- [7] Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luigi, "Ant Colony Optimization".
- [8] M. Dorigo and L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transaction on Evolutionary Computation* 1 (1997), 53–66.
- [9] W.J. Gutjahr: ACO algorithms with guaranteed convergence to the optimal solution. *Information Processing Letters* 82(3): 145-153 (2002).
- [10] Nitish Sabharwal, Harshit Sharma, "A Recursive Ant Colony System Algorithm for the TSP" 2011 International Conference on Advancements in Information Technology With workshop of ICBMG 2011 IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore.
- [11] G. Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications, volume 840 of LNCS. Springer Verlag, 1994.
- [12] Dorigo M., Maniezzo V. & Colorni A., "The Ant System: Optimization by a colony of cooperating agents," IEEE Trans. Systems, Man, and Cybernetics-Vol.26,N 1, pp.1-13, 1996.
- [13] Dogan Aydin, Aybars Ugur, "Extraction of flower regions in color images using ant colony optimization", 1877-0509 2010 Published by Singhal, Sakshi Garg and Deepti Singh Chauhan "Edge Detection Using Adaptive Thresholding and Ant Colony Optimization".
- [14] You-yi Zheng, Ji-lai Rao, Lei Wu, "Edge Detection Methods in Digital Image Processing", The 5th international Conference on Computer Science & Education Hefei, China. August 24–27, 2010.
- [15] Om Prakash Verma, Prerna Singhal, Sakshi Garg and Deepti Singh Chauhan "Edge Detection Using Adaptive Thresholding and Ant Colony Optimization"
- [16] G. Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications, volume 840 of LNCS. Springer Verlag, 1994.
- [17] Christian Nilsson, "Heuristics for the Traveling Salesman Problem".
- [18] Om Prakash Verma, Prerna Singhal, Sakshi Garg and Deepti Singh Chauhan, "Edge Detection Using Adaptive Thresholding and Ant Colony Optimization", World Congress on Information and Communication Elsevier Ltd.
- [19] Xuechuan Wang, Kuldip K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition", 2002.
- [20] G.W. Cottrell, Principal components analysis of images via back propagation, SPIE Proceedings in Visual Communication and Image Processing, Vol. 1001, 1988, pp. 1070–1077.
- [21] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
- [22] Keyun Tong, "Wavelet Transform And Principal Component Analysis Based Feature Extraction", June 3, 2010.
- [23] Om Prakash Verma, Prerna Singhal, Sakshi Garg and Deepti Singh Chauhan, "Edge Detection Using Adaptive Thresholding and Ant Colony Optimization", World Congress on Information and Communication Technologies, 2011.
- [24] Christian Nilsson, "Heuristics for the Traveling Salesman Problem".
- [25] Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant System for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)*, 34, 39–53.
- [26] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26 (1), 29–41.