International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:2, 2010

# Metaheuristics Methods (GA & ACO) for Minimizing the Length of Freeman Chain Code from Handwritten Isolated Characters

Dewi Nasien, Habibollah Haron, and Siti SophiayatiYuhaniz

***Abstract*—**This paper presents a comparison of metaheuristic algorithms, Genetic Algorithm (GA) and Ant Colony Optimization (ACO), in producing freeman chain code (FCC). The main problem in representing characters using FCC is the length of the FCC depends on the starting points. Isolated characters, especially the upper-case characters, usually have branches that make the traversing process difficult. The study in FCC construction using one continuous route has not been widely explored. This is our motivation to use the population-based metaheuristics. The experimental result shows that the route length using GA is better than ACO, however, ACO is better in computation time than GA.

***Keywords*—**Handwriting Recognition, Feature Extraction, Freeman Chain Code, Genetic Algorithm and Ant Colony Optimization.

## I. INTRODUCTION

HANDWRITING character recognition (HCR) is the ability of a computer to receive and interpret intelligible handwritten input then analyzed to many automated process system. Generally, HCR can be divided into three steps namely pre-processing, feature extraction and classification (recognition). Preprocessing stage is to produce a clean character image that can be used directly and efficiently by the feature extraction stage. Feature extraction stage is to remove redundancy from data. Classification stage is to recognize characters or words. This paper only concentrates in the feature extraction stage.

Feature extraction in HCR is a very important field of image processing and object recognition. Fundamental component of characters are called features. The basic task of feature extraction and selection is to find out a group of the most effective features for classification; that is, compressing from high-dimensional feature space to low-dimensional feature space, so as to design classifier effectively [1].

Dewi Nasien is with Soft Computing Research Group, Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia (phone: +60-163-095-888; fax: +60-755-650-44; e-mail: dwien82@gmail.com).

Habibollah Haron is with Soft Computing Research Group, Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia (e-mail: habib@utm.my).

Siti Sophiayati Yuhaniz is with Soft Computing Research Group, Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia (e-mail: sophia@utm.my).

Types of features depend on the system in which they are implemented. One of the features is shape feature. Shape feature is the most basic problem in image processing. There are two kinds of different shape based features. The first one is the features that are invariant to translation, rotation and scaling and the second one are features that do not have these qualities [4]. The first one is simpler to extract and have simpler procedures. There are two procedures for extracting these features. The first procedure is the boundary-procedure which is based on the outer boundary shape. The second one is region-procedure which works with the whole shape as an object.

Metaheuristic method is an approach to solve the optimization problems and to find the best of all possible of solutions. These two metaheuristics in this paper are Genetic Algorithm (GA) and Ant Colony Optimization (ACO). These methods are selected because they offer powerful methods to solve complex optimization problems. FCC is selected in the representation of a character image. In a handwritten character recognition, it is often to find several branches and this makes difficult to decide where it should go. Moreover, a revisit to the previous visited node is often needed to visit all of the nodes. These difficulties motivate us to use soft computing.

This paper is organized as follows. Section II presents the literature review of FCC, GA, and ACO. Section III describes the proposed algorithm and the two metaheuristic approaches. In section IV, data problem and parameter setting are presented. Section V shows experimental result and discussion. Finally, this paper is concluded in section VI.

## II. LITERATURE REVIEW

In this section describes the fundamental concept and methods used in this paper. This section covers the FCC, GA and ACO.

### *A. Freeman Chain Code (FCC)*

Chain coding is a common approach to represent different shapes. Chain code is one of the representation techniques that is useful for image processing, shape analysis and pattern recognition fields. Chain code is an object related data structure for representing the boundary of a binary object on a discrete grid [5]. Chain code representation gives boundary of character image where the codes represent the direction of where is the location of the next pixel and the connections to the starting point. The previous works in the differences of encoding and chain code representations are greatly discussed in [6,7,8].

The first approach of chain code was introduced by Freeman in 1961 that is known as freeman chain code (FCC) [9]. Freeman (1974), states in general, a coding scheme for line structure must satisfy three objectives [10]:

- it must faithfully preserve the information of interest
- it must permit compact storage and convenient for display
- it must facilitate any required processing

There are two directions of chain code, namely 4-neighborhood and 8-neighborhood. This paper utilized 8-neighbourhood in extraction of characters as shown in Fig. 1 (b). As in many papers usually the researchers start from zero until seven in 8-neighbourhood. This paper will start from one until eight because is easy to distinguish the direction or non direction (value is zero) of chain code. The numbers between one to eight in the chain code represent position and different shapes of directions according to the 8-neighbourhood.

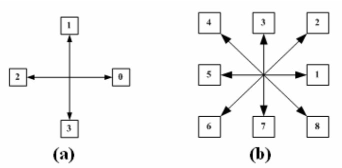



Fig. 1 Freeman chain codes : (a) 4-neighborhood and (b) 8-neighborhood

### B. Genetic Algorithm (GA)

Evolutionary computation (EC) is an umbrella term which covers genetic algorithms, evolution strategy and genetic programming and all of these techniques use process selection, mutation and reproduction to simulate evolution. Charles Darwin is the first person to introduce the theory of evolution in 1858. The purpose of EC is to improve the ability of an organism or system to survive in changing environment. This paper only describes the processes that are involved in GA.

In the early 1970s John Holland, one of the founder of evolutionary computation, introduced the concept of genetic algorithms [11]. GAs are a class of stochastic search based on biological evolution. A GA begins with a population (typically random) chromosomes and represented by binary of ones and zeros or decimal points. To measure the chromosomes performance (measure fitness), an evaluation function is used. Evolution fitness can be viewed as a measure of the organisms' ability to anticipate changes in their environment, ability to survive and reproduce in a specific environment. A set of chromosomes is called population. A population consists of three major fundamental operation which are, reproduction, cross over and mutation. First, reproduction is to replace population with good strings that have high fitness value. Second, cross over is to produce new chromosomes by combining the various pairs of chromosomes in the population. Finally, mutation is to flip a randomly selected gene in a chromosome. All these aspects of operations will establish one generation. The termination of GA process is occurred after a specified number of generation and the best chromosomes in the population are found. If no satisfactory solution is found then GA process is repeated.

### C. Ant Colony Optimization (ACO)

Swarm intelligence (SI) algorithms are based from the inspiration of the social behaviors of insects and other animals. There are two popular types of SI methods in computational intelligences: Ant colony optimization (ACO) and Particle swarm optimization (PSO). ACO utilizes the foraging behavior of ants to solve combinatorial optimization problems [12]. PSO is a population-based stochastic optimization technique that was developed by Kennedy and Eberhart in 1995 and is inspired by the social behavior of bird flocking or fish schooling [13]. ACO is used in our proposed algorithms.

ACO has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo and co-workers [13,14]. Metaheuristics is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems [15]. Tabu search, simulated annealing, ACO and EC are an examples of metaheuristics.

ACO is an algorithm inspired by foraging behavior of some ant species. These ants explore the area surrounding their nest in a random manner and find a food source. After that, they evaluate the quantity and quality of the food and back to the nest. These ants secrete a substance, called pheromone, and use its trails as medium of communicating information [16]. The other ants have the probability to enhance further deposition of pheromone that move on with path. Indirect communication between the ants via pheromone trails enables them to find shortest path between their nest and food sources [17]. This cooperative behavior of ants inspires the new computational paradigm for optimizing real life systems, which is suited for solving large scale problems [18].

The ACO consists of three major fundamental concepts which are constructing ant solutions, applying local search and updating pheromones. After initialization, a number of solutions are constructed by the ants, then, these solutions are then improved through a local search and finally the pheromone is updated.

## III. Proposed Algorithm

Thinned binary image (TBI) of pre-processing stage is used in the FE for extracting the character recognition. The TBI has the size of 50x50 pixels. The resulting TBI is shown in Fig. 2, for upper-case 'B' character. This paper proposes to use genetic algorithm (GA) and ant colony optimization (ACO) for minimizing the length of chain code.

The main problem in representation characters using FCC is the length of the FCC depends on the starting point, the branching node and the revisited walk. To solve these problems, population-based metaheuristics are used to generate the FCC which has the ability to produce FCC correctly in representing the characters and in the same way, for the classification stage must correctly represent and distinguish each characters.

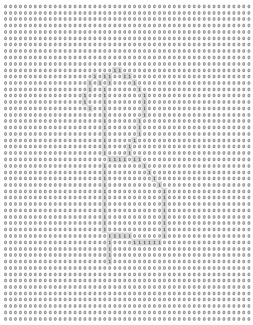


Fig. 2 TBI of uppercase ‘B’ character

### A. Solution Representation

Based on our problem, there are three solutions in representing a character: (1) character transformation into graph, (2) graph is a solution representation and (3) population-based metaheuristics approach to minimize the FCC length.

The solution of representation is presented by directed graph (digraph). A generation of FCC from a binary character can be modeled as a route of a graph problem. The complete graph based on the Fig. 2 can be seen in Fig. 3.

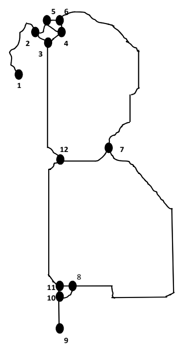



Fig. 3 Character transformation of fig. 2 into a graph

Table I provides all the edges in directed graph with theirs lengths. The lengths are obtained from the total number of nodes between two vertices.

The proposed metaheuristics use a sequence of edges to represent the FCC solutions. The edge is used as the solution representation. An edge is derived and ended from the same node. Two different edges can be derived from the same node or can also ended in another same node too. Thus, one edge can visited twice and as a result the solution representation can have a complete tour since a revisit to the previous visited nodes is often needed. Fig. 4 shows an example of the solution representation based on Fig. 2.

TABLE I
EDGES AND THEIR LENGTH GENERATED FROM FIG. 3

| Edge | Start Vertex | End Vertex | Length | Edge | Start Vertex | End Vertex | Length |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 5 | 17 | 6 | 7 | 0 |
| 2 | 2 | 1 | 5 | 18 | 7 | 6 | 15 |
| 3 | 2 | 3 | 0 | 19 | 7 | 8 | 15 |
| 4 | 3 | 2 | 0 | 20 | 8 | 7 | 23 |
| 5 | 2 | 5 | 0 | 21 | 8 | 10 | 23 |
| 6 | 5 | 2 | 0 | 22 | 10 | 8 | 0 |
| 7 | 3 | 4 | 0 | 23 | 8 | 11 | 0 |
| 8 | 4 | 3 | 0 | 24 | 11 | 8 | 0 |
| 9 | 3 | 12 | 10 | 25 | 10 | 9 | 2 |
| 10 | 12 | 3 | 10 | 26 | 9 | 10 | 2 |
| 11 | 4 | 5 | 0 | 27 | 11 | 10 | 0 |
| 12 | 5 | 4 | 0 | 28 | 10 | 11 | 0 |
| 13 | 4 | 6 | 0 | 29 | 11 | 12 | 11 |
| 14 | 6 | 4 | 0 | 30 | 12 | 11 | 11 |
| 15 | 5 | 6 | 9 | 31 | 12 | 7 | 3 |
| 16 | 6 | 5 | 0 | 32 | 7 | 12 | 3 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 26 | 22 | 20 | 18 | 14 | 11 | 6 | 2 | 1 | 3 |
| 9 | 30 | 24 | 20 | 32 | 10 | 4 | 2 | 1 | 5 |
| 12 | 13 | 17 | 32 | 30 | 27 | 25 | 26 | 22 | 20 |
| 18 | 14 | 11 | 6 | 2 | 1 | 3 | 9 | 30 | |

Fig. 4 An example of solution representation

The objective function is defined as the number of nodes which the FCC must visit from the starting node until all of the nodes are visited (revisit is counted too). For the solution representation in Fig. 4 the walk only uses the following sequence: (26 22 20 18 14 11 6 2 1 3 9 30 24) since all of the nodes are already visited with only that sequence. The objective function for this sequence is 185.

### B. Genetic Algorithm

Genetic Algorithm is one of the global optimization methods that belongs to the family of evolutionary algorithms. GA explores the solution space through an artificial evolutionary process, i.e. recombination, mutation, and selection. The solution representation used is described in section A. The implementation of GA to generate the FCC is depicted in Table II.

TABLE II
THE PSEUDO-CODE OF GA

| **The Pseudocode of Genetic Algorithm** |
|---|
| Input data and settings parameter values |
| Generate random initial population |
| Repeat |
|     Generate offspring population |
|         Select two parent randomly |
|         Recombination |
|         Mutation |
|         Perform local search |
|     Selection |
| Until stopping criterion is achieved |

The input for the GA algorithm is a binary image as shown in Fig. 1. Then several parameter values must be determined. The algorithm starts with initial population generated randomly. Then, through recombination, mutation, and local search the new offspring population is constructed. After that, the algorithm selects the best solutions which will survive for the next iteration. Algorithm stops with the predetermine number of iterations. The following sections describe the algorithm in more detail.

*1. Initialization and Generating Initial Population*

The GA is influenced by its parameter values. They are the maximum number of iterations, number of population ($n$), Mutation Rate ($\rho$), and number of local search. In addition, the input binary image is processed to generate a more efficient data structure to be used in the main GA iterations.

*2. Recombination*

Recombination process tries to create two off springs solutions which have characteristics combined from previous population. In this GA implementation, two-point cross over is selected as the recombination operator. The illustration for two-point crossover is shown in Fig. 5.

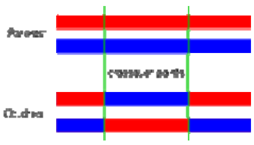


Fig. 5 Two-point crossover illustration

Initially, two-parent solutions are chosen randomly. Then, two points for each parent solutions are selected randomly. These two points are bounded only on the first half of the parent solutions. After that, everything between the two points is swapped between the two parent solutions, producing two offspring solutions. In addition, the swapping process is retained so it does not produce an offspring solution with an edge appear less or more than two times. This happens by discarding an addition of an edge which have appeared two times.

*3. Mutation*

Following the recombination process, a random number is provoked. If the random number value is less than the mutation rate (½), then the mutation process is performed to the offspring solution. Mutation process rearranges the sequence of an offspring solution randomly between two points. These two points are selected randomly and must be selected in the first half of the offspring solution.

*4. Local Search*

The offspring solution is improved in the local search process. The local search is chosen randomly from three choices:

1. randomly insert an edge to another location
2. randomly swap two edges
3. randomly rearrange a subset of solution sequence in opposite direction

Then it is performed to the offspring solution and whenever a better solution is obtained, the offspring solution will be updated. Moreover, the local search is iteratively executed for predetermined number of local search.

*5. Selection*

In the selection process, only $n$ best solutions survive to the next iteration. In total, there are 3n solutions ($n$ parent solutions and $2n$ offspring solutions). The algorithm sorts the $3n$ solutions by theirs objective function values and then selects $n$ best solution as the surviving population.

*C. ACO*

Ant Colony Optimization (ACO) imitates the behavior of an ant colony in finding the short path to the food source. It constructs ant solutions based on pheromone concentration in each path. The solution representation used is discrete representation is described in section A. The implementation of ACO to generate the FCC is depicted in Table III. In this implementation, the pheromone value $\tau_{ij}$ represents a tendency to continue the walk from edge $i$ to edge $j$. In addition, the pheromone value $\tau_{0i}$ represents a tendency to start a solution with edge $i$.

TABLE III
THE PSEUDO-CODE OF ACO

| **The Pseudocode of Ant Colony Optmization** |
|---|
| Input data and setting parameter values |
| Repeat |
|     Ant colony construction |
|     Apply local search |
|     Update pheromone value |
| Until stopping criterion is achieved |

Likewise, the input for the ACO algorithm is a binary image as shown in Fig. 1. Then several parameter values must be determined. The algorithm iterates with three main components: ant solution construction, local search procedures, and pheromone update. The algorithm stops with predetermine number of iterations. In addition, the proposed ACO algorithm keep record about a collection of the best solutions found. The following sections describe the algorithm in more detail.

*1. Initialization*

The proposed ACO uses several following parameters: maximum number of iterations, number of ants ($n$), Evaporation rate ($\rho$) and number of local search. In addition, the input binary image is processed to generate a more efficient data structure to be used in the main ACO iterations.

*2. Generate Initial Population*

Initial population is randomly generated. The population consists of $n$ solution which takes form of discrete representation. After that, the objective function for every solutions in the population is calculated.

*3. Construct Ant Solutions*

The algorithm constructs ant solution start from an empty solution. Then, it iteratively adds an edge based on the pheromone values and problem constraints. The problem constraints include an edge must not appear than twice and an edge must have a connection to the previous edge.

*4. Local Search*

The next solution is improved in the local search process as described in section B.4.

*5. Pheromone Update*

The pheromone values are updated using all of the ant solution. Equation (1) is represents the update pheromone method.

$$\tau_{ij} = (1 - \rho) * \tau_{ij} + 1/ofv_k \quad (1)$$

with $ofv_k$is an objective function value for ant solution $k$

## IV. Problem Data and Parameter Setting

We will evaluate the performance of both methods from metaheuristics approaches. The scope area is isolated handwritten on upper-case English characters. The database used is CEDAR (Center of Excellence for Document Analysis and Recognition) which consists of 126 characters that have been pre-processed to a TBI using the method proposed by [19].

The specification of hardware and software are shown below:

a) Notebook Compaq Presario CQ40
   - Memory (RAM) 3GB
   - Processor is AMD Turion™X2 Dual Core Mobiles RM-72 2.1GHz

b) Matlab R2008a (version 7.6)

Parameter values used in the proposed GA are as follow: (1) maximum iteration = 100, (2) population = 100, (3) mutation probability = 0.3 and (4) maximum local search = 10. Conversely, for ACO uses the following parameter values such as: (1) maximum iteration = 100, (2) number of ants = 100, (3) evaporation rate = 0.3 and (4) maximum local search = 10. Both of methods uses same values.

## V. Experimental and Discussion

Both of the proposed algorithms consist of 10 replications where each replication is 100 FCC solutions for every TBI Table IV shows the comparison for both of the proposed algorithms based on route length and computation time.

TABLE IV
Comparison both of the Proposed Algorithms Based on Route Length and Computation Time

| Proposed Algorithm | Route Length | | | | Computation Time | |
|---|---|---|---|---|---|---|
| | Best | Average | Worst | Std. Deviation | Average | Total |
| GA | 2,316.62 | 2,334.37 | 2,349.23 | 11.06 | 1,152.69 | 54,521.5 |
| ACO | 2,343.32 | 2,354.15 | 2,380.95 | 8.01 | 1,133.62 | 53,822.2 |

The result shows that the proposed GA performed better than ACO in route length such as best, average, worst but no good result in standard deviation. The better performance of the proposed GA is an impact of solution space through an artificial evolutionary process such as recombination, mutation and selection.

Moreover, the proposed GA is not better than the proposed ACO in term of computation time. For solving the whole thinned binary images, the proposed ACO needs 53,822.2 seconds. Meanwhile the proposed GA needs 54,521.5 seconds.

For future works, other soft computing methods can be explored for instance, particle swarm optimization (PSO), differential evolution (DE) and fuzzy logic.

## VI. Conclusion

To sum up, metaheuristics algorithm has been presented namely genetic algorithm and ant colony optimization to generate FCC from handwritten character recognition to a digraph.

Both of the proposed algorithms are for minimizing the length of the FCC. Based on the experiments, both of the proposed algorithms have advantages and disadvantages. The route length of FCC using GA is shorter than ACO. On the contrary, ACO is better in computation time than PSO.

The resulting FCC will become the input to the classification stage. Every feature in chain code is fed to the classifier for recognition. The efficiency of FCC in the representation can be seen by the number of image characters that can be recognized.

## Acknowledgment

The authors honorably appreciate Ministry of Science, Technology and Innovation (MOSTI) for the eScienceFund grant with VOT number 79369 and Research Management Center (RMC), University of Technology Malaysia (UTM) for the support in making this projects success.

## References

[1] Shi-Fei Ding; Wei-KuanJia; Chun-Yang Su; Zhong-Zhi Shi; Research of pattern feature extraction and selection. Machine Learning and Cybernetics, 2008 International Conference on Volume 1, 12-15 July 2008 Page(s):466 – 471.
[2] ZhaoqiBian, Xuegong Zhang. Pattern Recognition. 2nd Edition, Tsinghua University Press, Beijing, 2000.
[3] Jixiang Sun, Modern pattern recognition, Defense University of Science and Technology Publishing House, Changsha, 2002.
[4] Kunaver, M.; Tasic, J.F.; Image feature extraction - an overview. Computer as a Tool, 2005. EUROCON 2005.The International Conference on Volume 1, 21-24 Nov. 2005 Page(s):183 – 186.
[5] Jähne, Bernd; Digital Image Processing: Concepts, Algorithms, and Scientific Applications. Edition: 6, Published by Springer, 2005.
[6] Liu, Y.K., Zalik, B.: An efficient chain code with Huffman coding, Pattern Recognition, 38(4), 2005, 553-557.
[7] Sánchez-Cruz, Hermilo., Bribiesca, Ernesto., Rodríguez-Dagnino, R.M. Efficiency of Chain Codes to Represent Binary Objects. Volume 40, Issue 6, June 2007, Pages 1660-1674.
[8] Wulandhari. L.A., HaronHabibolah. The Evolution and Trend of Chain Code Scheme. ICGST-GVIP, ISSN 1687-398X, Volume (8), Issue (III), October 2008.
[9] Freeman. H, Techniques for the Digital Computer Analysis of Chain-Encoded Arbitrary Plane Curves, Proc. Natn. Electron. Conf. 18 (1961) 312-324.
[10] Freeman H, Computer Processing of Line-Drawing Images, ACM Computing Surveys 6, 1974, 57-97.
[11] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, New York.
[12] Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, London (1999) 11–32.
[13] M. Dorigo and G. Di Caro, "The Ant Colony Optimization meta-heuristic," in New Ideas in Optimization, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11–32, 1999.
[14] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.
[15] Dorigo, M., Birattari, M., Stutzle, T., Ant Colony Optimization. IEEE Computational Intelligence Magazine. November 2006.
[16] Dorigo M (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26:1–13.

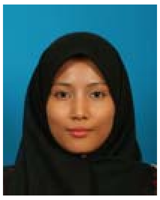
[17] J.-L. Deneubourg, S. Aron, S. Goss, J.-M. Pasteels, The self-organizing exploratory pattern of the argentine ant, J. Insect Behav. 3 (1990) 159–168.
[18] Socha K, Dorigo M (2008) Ant colony optimization for continuous domain. Eur J Oper Res 185:1155–1173.
[19] Engkamat, A.A. Enhancement of Parallel Thinning Algorithm for Handwritten Characters Using Neural Network. MSc Thesis. Universiti Teknologi Malaysia, 2005.

**Dewi Nasien** was born in Pekanbaru, Indonesia, June 30, 1982. She received her degree in the field of Information and Technology at Universitas Islam Negeri SUSQA Pekanbaru, Indonesia in 2005. She obtained an M.Sc in Computer Science at Universiti Teknologi Malaysia, Johor Bahru, Malaysia in 2008.

Currently, she is concluding a Ph.D at the Department modeling and industrial computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia. Her research interests are in the field of pattern recognition and soft computing.

**Habibollah Haron** was born in Malacca, Malaysia in 1966. He received his degree in Computer Science from Universiti Teknologi Malaysia in 1989, then Master degree in 1995 from University of Sussex, UK. He obtained his PhD in the area of Computer Aided Geometric Design (CAGD) in the year of 2004 at Universiti Teknologi Malaysia.

He has been with Universiti Teknologi Malaysia since 1991 as a faculty member, and was appointed as Associate Professor in 2008. He was also the Head of Department for Modeling and Industrial Computing Department, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia before currently attached with Imam Mohammad Ibn Saud University, Riyadh for his sabbatical work. Some of his research areas are computer aided design and soft computing.



**Siti Sophiayati Yuhaniz** was born in Selangor, Malaysia, March 23rd 1980. She received her Bachelor degree in the field of computer science at Universiti Teknologi Malaysia, then obtained her PhD in the field of on-board satellite image processing at University of Surrey, UK in 2008.

She is one of the faculty members of Faculty of Computer Science and Information System, UTM, under the Modeling and Industrial Computing Department. She began lecturing in July 2008, and a member of Soft Computing Research Group (SCRG) which is under the K-Economy Research Alliance, Universiti Teknologi Malaysia. Her research interests are in the field of soft computing, pattern recognition, and remote sensing.