

## A Comparatives Study of Gateway Placement Optimization in Wireless Mesh Network using GA, PSO and ACO

Dac-Nhuong Le

Faculty of Information Technology, Haiphong University, Haiphong, Vietnam

---

### Article Info

#### Article history:

Received May 28<sup>th</sup>, 2013

Revised June 20<sup>th</sup>, 2013

Accepted July 10<sup>th</sup>, 2013

---

#### Keyword:

Wireless Mesh Networks  
Gateway Placement  
Multi-Traffic Flow Weight  
Genetic Algorithm  
Particle Swarm Optimization  
Ant Colony Optimization

---

### ABSTRACT

In this paper, we study the challenging problem of optimizing gateway placement for throughput in Wireless Mesh Networks. I compare my study of Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) to solve the optimal gateway placement in Wireless Mesh Networks problem. We generate the locations of gateway randomly and independently and choose the router with the highest weight as potential location for gateway placement use multi-traffic-flow weight. In the my algorithms based on GA and PSO, we calculate the fitness value of each scheme and update them step by step with the best method to quickly find the optimal. Furthermore, with ACO algorithm, we calculate the probability and pheromone values of ants will choose to go from current gateway  $i$  to next client  $j$ . After each iteration, the pheromone values are updated by all the number of ants that have reached to the destination successfully and found the optimal solution. Numerical results show that the proposed algorithm has achieved much better than previous studies.

*Copyright © 2013 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

### Corresponding Author:

Dac-Nhuong Le,

Faculty of Information Technology, Haiphong University, Vietnam

171 Phan Dang Luu, Kien An, Haiphong, Vietnam.

Email: nhuongld@hus.edu.vn

---

## 1. INTRODUCTION

A wireless mesh network (WMN) is a communication network made up of radio nodes organized in a mesh topology, which often consists of mesh clients, mesh routers and gateways [1]. The mesh clients are often laptops, cell phones and other wireless devices, which are connected to the Internet through the mesh routers. The mesh routers forward traffic to and from the gateways which connected to the Internet. The coverage area of the radio nodes working as a single network is sometimes called a mesh cloud. Access to this mesh cloud is dependent on the radio nodes working in harmony with each other to create a radio network. A mesh network is reliable and offers redundancy. When one node can no longer operate, the rest of the nodes can still communicate with each other, directly or through one or more intermediate nodes. Wireless mesh networks can be implemented with various wireless technology including 802.11, 802.16, cellular technologies or combinations of more than one type. A wireless mesh network has some features which are similar to wireless ad-hoc network. It is often assumed that all nodes in a wireless mesh network are immobile but this is not necessary. The mesh routers may be highly mobile and are not limited to power, memory, calculating ability and operate as intelligent switching devices. Figure 1 presents an example of a WMN.

In recent years, the optimizing WMN problem is interested in many researches. However it still remains open [1]. In there, gateway placement is the most interested problem in optimizing WMN. There are some analogous research results in wired or cellular networks. However, all the above investigation has been focused on network connectivity of WMNs by deploying the minimum number of backbone nodes [2]. Throughput is one of the most important parameters that affect the quality of service of WMN. So in this paper, we will improve a gateway placement algorithm to optimize throughput for WMNs. A similar problem

---

*Journal homepage:* <http://iaesjournal.com/online/index.php/IJINS>

was studied by Ping Zhou, Xudong Wang, B. S. Manoj and Ramesh Rao in [2], however, their scheme was not updated step by step, and the locations of gateways were determined sequentially, so the location of previously-placed gateways affects the location of those placed later. Constructing computation model to calculate the throughput of WMNs is very necessary, but it is not simple to build. There are many computation models built in [3-8], but all of them, except [8], are not suitable for calculating throughput of WMNs. In this paper, we use the computation model in [2], in which TDMA scheduling is assumed to coordinate packet transmissions in mesh clients, mesh routers, and gateways.

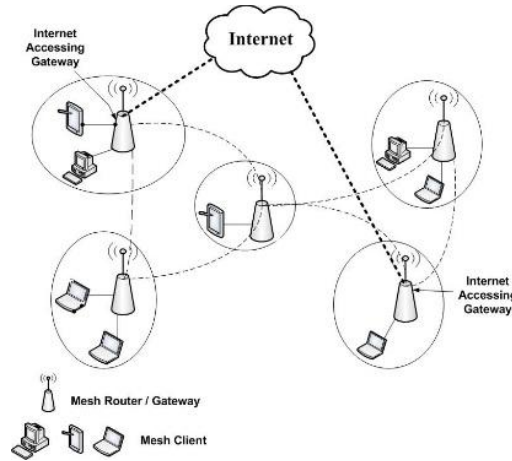


Fig.1. A typical WMN

The rest of this paper is organized as follows. Section 2 presents the computation model and briefly introduces the main idea of MTW-based gateway placement proposed in [2]. Section 3-5 presents my new algorithm for gateway placement in optimizing WMN based on GA, ACO and PSO. Section 6 presents my simulation and analysis results, and finally, section 7 concludes this paper

**2. MTW-BASED GATEWAY PLACEMENT**

In this section, we first present the computation model and briefly introduce the main idea of MTW-based gateway placement proposed in [2].

**2.1 Computation Model**

*2.1.1 Network Topology*

The computation model presented in [2] brings out a typical WMN topology for Internet accessing as follows and is illustrated in Figure 2.

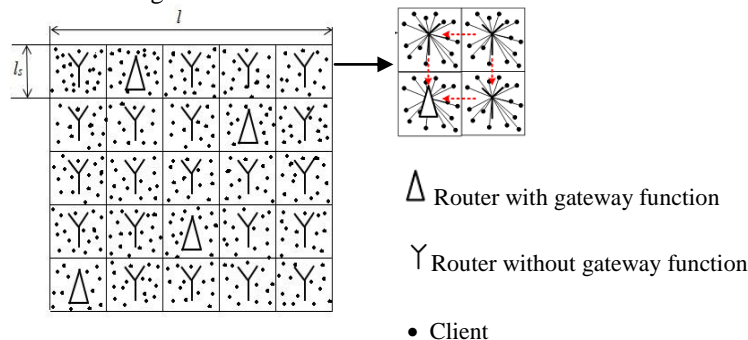


Figure 2. Network topology of an WMN infrastructure with gateways

This topology has  $N_c$  mesh clients which are assumed to be distributed on a square  $R$ ,  $N_r$  routers, and  $N_g$  gateways with the constraint of  $1 \leq N_g \leq N_r \leq N_c$ . According to [9]  $R$  is partitioned evenly into  $N_r$  cells  $R^j$ , and a mesh router is placed in the centre of each cell. In each cell, mesh clients are connected to the mesh router like a star topology and are not communicated with each other directly. Data transmission is carried

out among mesh clients, which are equivalent such that they always have the same amount of packets to send or receive during a certain time, while the mesh routers find the best route and forward data to its destination. All traffic is assumed to go through gateways. Each mesh router determines its nearest gateway to relay packets to or from that. If there is more than one nearest gateways, the router will load its traffic to all its nearest gateways by a round robin. A mesh client is said to be associated with a gateway if its connected router is associated with the gateway. Thus, traffic load of a mesh client will also be shared by all its potentially associated gateways.

There are some definitions of communications which will be frequently used:

- *Local communications*: it is referred as the communications between a mesh router and a mesh client;
- *Backbone communications*: it is referred as the communications between two mesh routers, which includes the communications between a gateway and a mesh router;
- *Downlink communications*: it is referred as the communications from a gateway to a mesh client, in which a data packet is first relayed among mesh routers in backbone communications and is then sent by a mesh router to one of its connected mesh clients;
- *Uplink communications*: it is referred as the communications from a mesh client to a gateway, in which a data packet is sent in the exact reverse direction as described in the downlink communications.

### 2.1.2 Transmission Model

Each mesh router is often equipped with two virtual radio interfaces over one physical radio interface, in which one transmitting at  $W_1$  bits/s for backbone communications and the other transmitting at  $W_2$  bits/s for local communications. Each mesh client transmits  $W_2$  bits/s in local communications. It is assumed that  $W_1$  and  $W_2$  are orthogonal so that local communications and backbone communications do not influence each other. Moreover, mesh routers or mesh clients can receive packets from only one sender at a time. Transmission and reception can occur in either time-division duplex (TDD) or frequency division duplex (FDD), depending on how the physical and MAC layers are implemented.

### 2.1.3 Throughput

The computation model proposed in [2] introduces two criterions to evaluate the performance of gateway placement algorithms: the total of throughput and the minimal throughput of each client. In this paper, we also use these criterions to evaluate the performance of our algorithm.

**Problem 1: Optimal gateway placement for maximizing aggregate throughput of WMNs**, i.e., in the above WMN model, given  $N_c$ ,  $N_r$ ,  $N_g$ ,  $W_1$ ,  $W_2$  and specific clients' distribution, routers' distribution, transmission, scheduling and routing protocols,  $N_g$  gateways are chosen among  $N_r$  mesh routers such that

$$\sum_{i=1}^{N_c} TH(i, N_g) \rightarrow \max \quad (1)$$

where  $TH(i, N_g)$  denotes the per client throughput of the  $i^{\text{th}}$  mesh client when  $N_g$  gateways are deployed.

**Problem 2: Optimal gateway placement for maximizing the worst case of per client throughput in the WMN**, i.e., in the above WMN model, given  $N_c$ ,  $N_r$ ,  $N_g$ ,  $W_1$ ,  $W_2$  and specific clients' distribution, routers' distribution, transmission, scheduling and routing protocols,  $N_g$  gateways are chosen among  $N_r$  mesh routers such that

$$\min_{i=1}^{N_c} TH(i, N_g) \rightarrow \max \quad (2)$$

### 2.1.4 Sharing Efficiency of Gateways

*IntD* is defined as *Interfering Distance of Gateways*. If distance of two gateways less than *IntD*, they interfere with each other. Interfering gateways have to share the same wireless channel in the backbone communications. The algorithm to calculate the sharing efficiency of gateways is presented as follows.

#### CALCULATE THE SHARING EFFICIENCY OF GATEWAYS ALGORITHM

1. Constructing the table of non-overlapping interfering groups arranged in descending order of the number of elements in the group.
2. Assigning percentage value for each gateway from the top to the last row in the above table.

In the first step, any two elements of each group that interfere with each other, and a group appearing later must have at least one gateway which does not belong to the previous groups. The procedure that calculates percentage value for the gateways is described as follows:

CALCULATE THE PERCENTAGE VALUE OF GATEWAYS ALGORITHM	
1.	Assign value of 100% for all the gateways.
2.	<b>FOR</b> the top row <b>TO</b> the last row of the table in the first step $k=1/\text{the number of gateways}$ in current group.
3.	<b>FOR</b> the first gateway <b>TO</b> the last gateway in current group
4.	<b>IF</b> percentage value $> k$ <b>THEN</b> push into subgroup1
5.	<b>ELSE</b> push into subgroup2.
6.	<b>END FOR.</b>
7.	$P = \frac{1 - \text{sum of all the percentage value in subgroup 2}}{\text{the number of gateway in subgroup 1}}$
8.	Assign value of P for all gateways in subgroup1.
9.	<b>END FOR.</b>

The final computing value is stored in  $G_{eff}(k)$ ,  $k=1..N_g$ .

### 2.1.5 Throughput Computation

Throughput of the  $i^{\text{th}}$  mesh client when  $N_g$  gateways are deployed, denoted as  $TH(i, N_g)$ , is calculated as follow:

$$TH(i, N_g) = \min_{i=1..N_c} \left\{ TH_{W_1}(i, N_g), TH_{W_2}(i) \right\}, \quad (3)$$

Here,  $TH_{W_1}(i, N_g)$  is defined as the throughput of the  $i^{\text{th}}$  mesh client in backbone communications and  $TH_{W_2}(i)$  is defined as the throughput of  $i^{\text{th}}$  mesh client in local communications. Because  $W_1$  and  $W_2$  are orthogonal, so we can compute  $TH_{W_1}(i, N_g)$  and  $TH_{W_2}(i)$  separately. Note that  $TH_{W_2}(i)$  is independent of  $N_g$  in WMN model and if a mesh client is connected directly to a gateway, its throughput is decided only by the per-client throughput in local communications.

$TH_{W_1}(i, N_g)$  is computed as follows:

$$TH_{W_1}(i, N_g) = \frac{\sum_{\substack{k=\text{all the associated gateways} \\ \text{with the mesh router } R^j}} TH'_g(k)}{N_g(j)} \quad (4)$$

Here,  $N_g(j)$  is the number of gateways associated with the mesh router  $R^j$ . A gateway is associated with a router if the distance between them is less than or equal the radius of that gateway. The computation of the radius of gateway is proposed in sub-section 2.1.2.  $TH'_g(k)$  is the throughput per client that the  $k^{\text{th}}$  gateway can guarantee for all its associated mesh clients in backbone communications.

$$TH'_g(k) = \frac{G_{eff}(k) \times c_1 W_1}{\sum_{\substack{l=\text{all associated routers} \\ \text{with the } k^{\text{th}} \text{ gateway}}} \frac{N_c(l) \times N'_{hop}(l)}{N_g(l)}} \quad (5)$$

Here  $c_1 W_1$  is the throughput that the  $k^{\text{th}}$  gateway can guarantee in backbone communications,  $N_c(l)$  is the number of clients associated with the mesh  $l^{\text{th}}$  router.  $N'_{hop}(l)$  is the actual time slot that the  $R^l$ -connected mesh client uses to transmit data to the gateway.

$$\begin{aligned} N'_{hop}(j) &= N_{hop}(j), \text{ if } N_{hop}(j) < SRD; \\ N'_{hop}(j) &= SRD, \text{ if } N_{hop}(j) \geq SRD; \end{aligned} \quad (6)$$

Where,  $N_{hop}(j)$  is the number of hops from the mesh client to the gateway.  $SRD$  is defined as *Slot Reuse Distance*. Next,  $TH_{W_2}(i)$  is computed simply as follows:

$$TH_{W_2}(i) = \frac{c_2 W_2}{CRF \times N_c(j)}, i = 1..N_c \quad (7)$$

Here,  $c_2 W_2$  is the throughput that  $R^j$  can guarantee for all associated mesh clients.  $CRF$  is defined as *Cell Reuse Factor*.

## 2.2 The original MTW-based Gateway Placement

In this algorithm, a traffic-flow weight, denoted as  $MTW(j)$ , is calculated iteratively on the mesh router  $R^j$ ,  $j=1..N_r$ . Each time, the router with the highest weight will be chosen to place a gateway. The weight computation is adaptive to the following factors:

### CALCULATE MIW(J) ALGORITHM

1. The number of mesh routers and the number of gateways.
2. Traffic demands from mesh clients.
3. The location of existing gateways in the network.
4. The interference from existing gateways.

First of all, this algorithm proposes a formula to compute the gateway radius:

$$R_g = \text{round}\left(\frac{\sqrt{N_r}}{2\sqrt{N_g}}\right) \quad (8)$$

Assuming all mesh clients are similar in WMN model, then local traffic demand on each mesh router, denoted as  $D(j)$ ,  $j=1..N_r$ , represented by the number of mesh clients connected to  $R^j$ .  $MTW(j)$  is calculated with  $D(j)$  and  $R_g$  as follows:

$$\begin{aligned} MTW(j) = & (R_g + 1) \times D(j) \\ & + R_g \times (\text{traffic demand on all 1-hop neighbors of } R^j) \\ & + (R_g - 1) \times (\text{traffic demand on all 2-hop neighbors of } R^j) \\ & + (R_g - 2) \times (\text{traffic demand on all 3-hop neighbors of } R^j) \\ & + \dots \end{aligned} \quad (9)$$

Place the first gateway on the router with highest  $MTW(j)$ . If more than one gateways are requested, re-adjust  $D(j)$ ,  $j=1..N_r$  with  $R_g$  as follows: set the value 0 for all routers within  $(R_g-1)$  hops away from  $R^j$  (including  $R^j$ ) and reduce to half for gateways which are  $R_g$  hops away from  $R^j$ . Re-calculate  $MTW(j)$  with the new  $D(j)$ , and perform the following procedure.

### RE-CALCULATE MIW(J) ALGORITHM

1. Choose the router with the highest weight as potential location for gateway placement, namely  $R^j$ .
2. Re-construct the table of non-overlapping interfere groups with  $R^j$  and previous gateways.
3. Compute the sharing efficiency for  $R^j$ .
4.  $MIW'(j) = MIW(j) \times Geff(j)MIW'(j)$ .
5. If  $MIW'(j)$  is still larger than the second highest weight, then place the gateway in the location. Otherwise, repeat the above steps from 1 to 5 until obtaining the location.

## 3. OPTIMIZING GATEWAY PLACEMENT IN WMNS BASED ON GENETIC ALGORITHM

### 3.1 Expressing the individual

There are three common types of expressing the individual: encoding as a real number, an integer and a binary. In this paper, we use integer encoding to express the individual. The individual is a  $K$ -dimensional vector ( $K$  is the number of gateways), where each of its component is an integer corresponding to the position to be located in the WMN.

Specifically, gateways are denoted by  $\{g_1, \dots, g_k\}$ , in which if the  $j^{\text{th}}$  individual is  $\{a_1^j, \dots, a_k^j\}$  then  $a_i^j$  would correspond to the gateway  $g_i$ , and its value will be a random integer generated correlatively. Assume that the WMN model, presented in Section 2, is divided into  $N$  cells and numbered from left to right and from top to bottom.  $a_i^j$  will then receive the value in the range of  $[0..(N-1)]$ . The pseudo-code of the procedure for each individual:

**CALCULATE THE THROUGHPUT OF GATEWAYS ALGORITHM**

1. Determine the location of gateways.
2. Compute the throughput achieved.

### 3.2 Population Initialization

The initial individuals are generated with  $P$  elements ( $P$  is a designated parameter). Each individual is a  $K$ -dimensional vector ( $K$  is the number of gateways) that each component is an integer, randomly generated, corresponding to the interval of  $[0, N-1]$ .

### 3.3 Crossover Operator

This operator mimics the mating process in the nature. To do crossover, two individuals are picked first and an integer number (crossover point) is generated randomly between  $[1, k]$  (where  $k$  is the number of gateways). Then the offspring is generated by interchanging the second halves of its parent, as illustrated in figure 3.

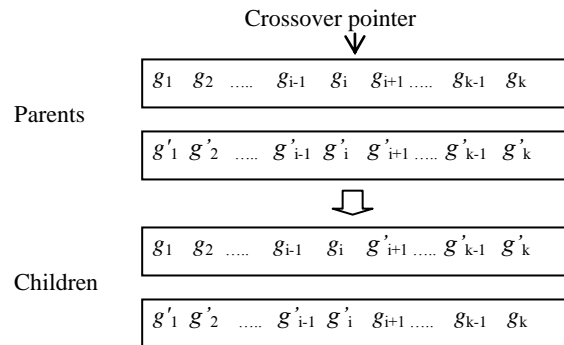


Figure 3. An example of crossover operator

In the crossover stage, the algorithm examines all pairs of individuals. It begins with the pairs that include the individual with a higher fitness value until the population size becomes twice of the original size.

### 3.4 Mutation Operator

The mutation operation is one kind of random change in the individual. In our algorithm, pointwise mutation is adopted, in which one gene in the individual is changed with a certain probability, referred to as the mutation probability. This operator allows the algorithm to search for new and more feasible individuals in new corners of the solution spaces. To do mutation, an element is randomly selected from the gateway and the selected gateway is called the mutation point, as illustrated in figure 4.

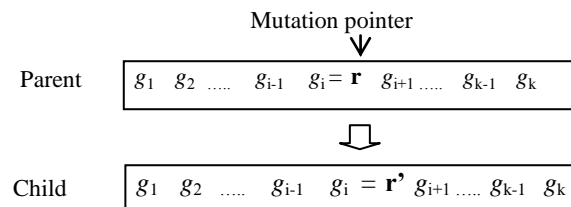


Figure 4. An example of mutation operator

The mutation stage is implemented until either population size becomes twice of the original size or all individuals in the current generation are examined.

### 3.5 Evaluation function

Evaluation value of  $j^{\text{th}}$  individual is calculated by the following formula:

$$F_j = 1 - \frac{1}{\sum_{i=1}^{N_c} TH(i, K)} \quad (10)$$

In which,  $N_c$  is the number of clients,  $K$  is the number of gateways,  $TH(i, K)$  is computed by the formula (3).

### 3.6 My GA Algorithm

After applying the genetic algorithm operators above, the reproduction stage selects the  $P$  fittest individuals that have higher fitness values from both parents and childrens for the next generation. This process is repeated until the stopping conditions are satisfied.

I must define the conditions for stopping the algorithm. The algorithm will stop after  $G$  generations ( $G$  is a design parameter) or when the values of evaluation function is unmodified. The pseudo-code of the genetic algorithm for the optimal gateway placement in WMN, as follows:

GENETIC ALGORITHM (GA)	
1.	<b>BEGIN</b>
2.	INITIALISE <i>population</i> with random candidate solutions;
3.	EVALUATE FUNCTION each candidate;
4.	<b>REPEAT</b>
5.	SELECT parents;
6.	RECOMBINE pairs of parents;
7.	CROSSOVER the resulting offspring;
8.	MUTATION the resulting offspring;
9.	EVALUATE FUNCTION new candidates;
10.	SELECT individuals for the next generation;
11.	<b>UNTIL</b> ( <i>TERMINATION CONDITION</i> is satisfied);
12.	<b>END</b>

## 4. OPTIMIZING GATEWAY PLACEMENT IN WMN BASED ON PSO

Particle Swarm Optimization (PSO) was invented by Kennedy and Eberhart [15-16] while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a socio cognitive study investigating the notion of “*collective intelligence*” in biological populations.

### 4.1. Expressing an element

There are three common types of expressing an element: encoding as a real number, an integer and a binary. In this paper, we use integer encoding to express an element. An element is a  $K$ -dimensional vector ( $K$  is the number of gateways), where each of its component is an integer corresponding to the position to be located in the WMN. Specifically, gateways are denoted by  $\{g_1, \dots, g_k\}$ , in which if the  $j^{\text{th}}$  element is  $\{a_1^j, \dots, a_k^j\}$  then  $a_i^j$  would correspond to the gateway  $g_i$ , and its value will be a random integer generated correlatively. Assume that the WMN model, presented in Section 2, is divided into  $N$  cells and numbered from left to right and from top to bottom.  $a_i^j$  will then receive the value in the range of  $[0 .. (N-1)]$ .

### 4.2. Population Initialization

The initial population is generated with  $P$  elements ( $P$  is a designated parameter). Each element is a  $K$ -dimensional vector ( $K$  is the number of gateways) that each component is an integer, randomly generated, corresponding to the interval of  $[0, N-1]$ .

### 4.3. Fitness function

Fitness value of  $j^{\text{th}}$  element is calculated by the following formula:

$$F_j = 1 - \frac{1}{\sum_{i=1}^{N_c} TH(i, K)} \quad (11)$$

In which,  $N_c$  is the number of clients,  $K$  is the number of gateways,  $TH(i,K)$  is computed by the formula (3).

#### 4.4. Evaluation

Elements in each generation are updated according to formula (12) and (13) described below. In which  $present[j]$  and  $v[j]$  are respectively the  $j^{th}$  element in the current generation and its speed. In the context of the current problem,  $present[j]$  and  $v[j]$  are  $K$ -dimensional vectors.

$$v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) \quad (12)$$

$$present [] = present [] + v[] \quad (13)$$

Since PSO is a stochastic process, we must define the conditions for stopping the algorithm. The algorithm will stop after  $G$  generations ( $G$  is a design parameter) or when the values of  $gBest$  and  $pBest$  are unmodified.

#### 4.5. My PSO algorithm

The pseudo-code of the PSO algorithm for the optimal gateway placement in WMN, as follows:

PARTIAL SWARM OPTIMIZATION ALGORITHM (PSO)	
1.	<b>BEGIN</b>
2.	<b>FOR EACH</b> particle
3.	Initialize particle;
4.	<b>ENDFOR</b>
5.	<b>DO</b>
6.	<b>FOR EACH</b> particle
7.	Calculate fitness value according to (11);
8.	<b>IF</b> the fitness value is better than the best fitness value (pBest) in history
9.	Set current value as the new pBest;
10.	<b>ENDIF</b>
11.	<b>ENDFOR</b>
12.	Choose the particle with the best fitness value of all the particles as the gBest (or Choose the particle with the best fitness value of all the neighbors particles as the lBest)
13.	<b>FOR EACH</b> particle
14.	Calculate particle velocity according to (12);
15.	Update particle position according to (13);
16.	<b>ENDFOR</b>
17.	<b>WHILE</b> (stop condition is true);
18.	<b>END.</b>

### 5. OPTIMIZING GATEWAY PLACEMENT IN WMN BASED ON ACO

The ACO algorithm is originated from ant behavior in the food searching. When an ant travels through paths, from nest food location, it drops pheromone. According to the pheromone concentration the other ants choose appropriate path. The paths with the greatest pheromone concentration are the shortest ways to the food. The optimization algorithm can be developed from such ant behavior. The first ACO algorithm was the Ant System [12], and after then, other implementations of the algorithm have been developed [13-14].

#### 5.1 Expressing an element

In our case the pheromone matrix is generated with matrix elements that represent a location for ant movement, and in the same time it is possible receiver location. Assume that the WMN model, presented in Section II-1, is divided into  $N$  cells and numbered from left to right and from top to bottom. In this paper, we use integer encoding to express an element of matrix  $A_{m*n}$  ( $m$  is the number of gateways,  $n$  is number of client).  $a_{ij}$  will then receive a random integer generated correlatively .



### 5.2 Population Initialization

The ant population is randomly generated (50 ants) and each ant is associated to one matrix location (node). Location of ants is an integer, randomly generated, corresponding to the interval of  $[0, N-1]$ .

### 5.3 Cost function

The next node (location) is selected according to the probability with which ant  $k$  will choose to go from current gateway  $i$  to next client  $j$  is calculated by the following formula:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (14)$$

In which,  $\tau_{ij}$  is the pheromone content of the path from gateway  $i$  to client  $j$ ,  $N_i^k$  is the neighborhood includes only locations that have not been visited by ant  $k$  when it is at gateway  $i$ ,  $\eta_{ij}$  is the desirability of client  $j$ , and it depends of optimization goal so it can be our cost function. The influence of the pheromone concentration to the probability value is presented by the constant  $\alpha$ , while constant  $\beta$  do the same for the desirability. These constants are determined empirically and our values are  $\alpha=1$ ,  $\beta=10$ . The ants deposit pheromone on the locations they visited according to the relation.

$$\tau_j^{new} = \tau_j^{current} + \Delta\tau_j^k \quad (15)$$

where  $\Delta\tau_j^k$  is the amount of pheromone that ant  $k$  exudes to the client  $j$  when it is going from gateway  $i$  to client  $j$ . This additional amount of pheromone is defined by

$$\Delta\tau_j^k = \frac{1}{f_j} \quad (16)$$

Where  $f_j$  is the cost function of the client  $j$  is computed by the formula (3). The pheromone evaporates during time and diminishes if there are no new additions.

The pheromone evaporation is applied to all locations as follows

$$\tau_j^{new} = (1-\rho)\tau_j \quad (0 < \rho \leq 1) \quad (17)$$

The value of  $\rho$  is selected empirically, what is in our case  $\rho=0.1$ .

### 5.5. My ACO algorithm

The pseudo-code of the ACO algorithm for the optimal gateway placement in WMN, as follows:

ACO ALGORITHM	
1.	<b>Initialization:</b>
2.	Algorithm parameters: $\alpha=1, \beta=10$ ;
3.	Ant population size: $K$ ;
4.	Maximum number of iteration: $N_{Max}$ ;
5.	<b>Generation:</b>
6.	Generating the pheromone matrix for the ant $k$ ;
7.	Update the pheromone values and set $x^*=k$ ;
8.	$i=1$ ;
9.	<b>Repeat</b>
10.	<b>FOR</b> $k = 1$ <b>TO</b> $K$ <b>DO</b>
11.	Computing the cost function for the ant $k$ by the formula (3);
12.	Computing probability move of ant individual by the formula (14);
13.	<b>IF</b> $f(k) < f(x^*)$ <b>THEN</b>
14.	Update the pheromone values by the formula (17);
15.	Set $x^*=k$ ;
16.	<b>ENDIF</b>
17.	<b>ENDFOR</b>
18.	<b>UNTIL</b> $i > N_{Max}$ ;

## 6. NUMERICAL RESULTS AND DISCUSSION

### 6.1. Experiments Senario

According to numeric results in [2], the MTW-based Gateway Placement Algorithm is better than three gateway placement algorithms: *Random Placement (RDP)*, *Busiest Router Placement (BRP)*, and *Regular Placement (RGP)*. Therefore in this paper we only compare our algorithm with MTW-based gateway placement algorithm.

I study three experiments:

- In the first experiment I assume  $N_c=200$ ,  $N_r=36$ ,  $l=1000m$ , i.e. there are 1000 mesh clients distributed in a square region of  $1000m \times 1000m$ ; the square is split evenly into 36 small square cells and a mesh router is placed in the center of each cell. Concurrently, I assume  $CRF = 4$ ,  $SRD=3$ ,  $IntD=2$ , the backbone bandwidth is 20Mbps and the local bandwidth is 10Mbps.
- The second experiment is similar to the first one, but in which  $N_c=400$ ,  $N_r=64$ . The local traffic demand of each mesh router in all experiments is generated randomly.
- In the third experiment I assume  $N_c=1000$ ,  $N_r=36$ ,  $N_g=17$ ,  $l=1000m$ , i.e. there are 1000 mesh clients distributed in a square region of  $1000m \times 1000m$  is showed in Figure 3; the square is split evenly into 36 small square cells and a mesh router is placed in the center of each cell. Concurrently, I assume  $CRF = 4$ ,  $SRD=3$ ,  $IntD=2$ , the backbone bandwidth is 20Mbps and the local bandwidth is 10Mbps.

22	5	56	8	48	40
41	33	42	29	34	39
40	18	48	25	16	23
5	12	2	16	44	9
37	17	47	24	38	4
14	50	12	17	47	38

Figure 5. 1000 mesh clients distributed in a square region of  $1000m \times 1000m$

### 6.2. GA, PSO, and ACO parameters

In my experiment, I have implemented GA, PSO and ACO algorithms in C language with parameters can be defined bellows:

GA Parameters		PSO Parameters		ACO Parameters	
Population size:	500	Population size:	100	Population size:	100
Initialization:	Random	Initialization:	Random	Initialization:	Random
Mutation rate:	0.1	Cognitive parameter:	c1=1	Number of Ants:	50
Number of offspring:	500	Social parameter:	c2=1	$\rho = 0.1$	
Recombination:	<i>One point crossover</i>	Number of neighbor:	K=3	$\alpha = 1$	
Recombination probability:	70%			$\beta = 10$	
Parent selection:	<i>Best out of random two</i>				
Survival selection:	<i>Generational</i>				
Iterations:	100	Iterations:	100	Iterations:	100

### 6.3. Results and Comparisions

In each experiment, I optimize the gateway placement problem by maximum one of two parameters: the total throughput of all mesh clients, denoted as  $GA\_Sum$ ,  $PSO\_Sum$ ,  $ACO\_Sum$ , and the minimal throughput of each mesh client, denoted as  $GA\_Min$ ,  $PSO\_Min$ ,  $ACO\_Min$ . Then I compare my results with the results achieved by MTW-based gateway placement algorithm.

Firstly, I compare the aggregate throughput and the worst case throughput achieved by each algorithm in the first experiment as shown in Figure 6 and Figure 7. And the second experiment results compare as shown in Figure 8 and Figure 9.

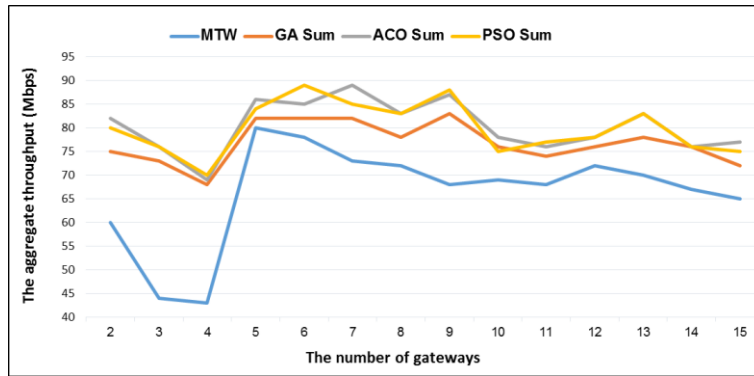


Figure 6. The comparison of the aggregate throughput of per client throughput in the first experiment

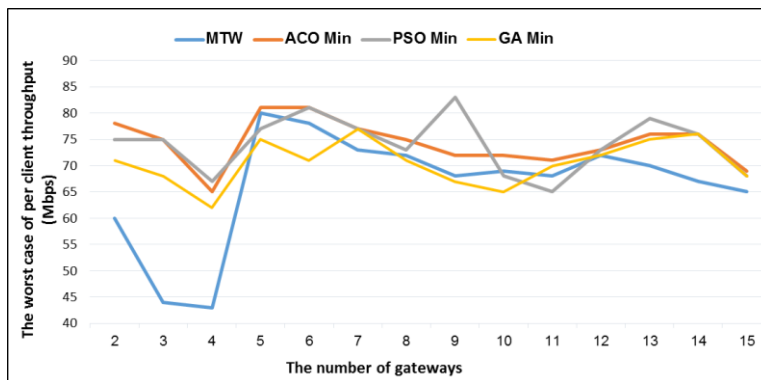


Figure 7. The comparison of the worst case of per client throughput in the first experiment

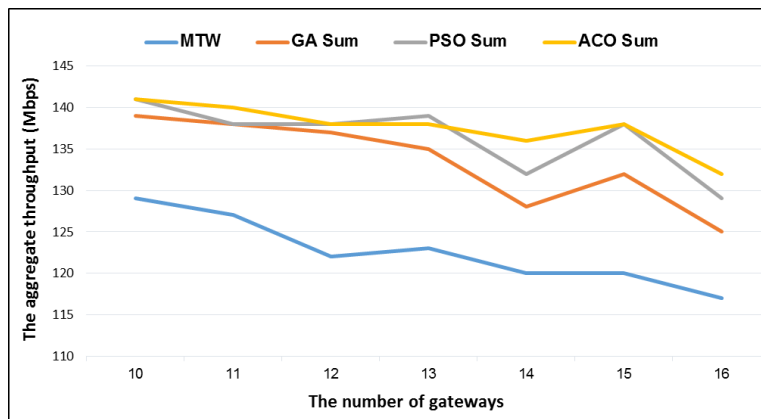


Figure 8. The comparison of the aggregate throughput of per client throughput in the second experiment

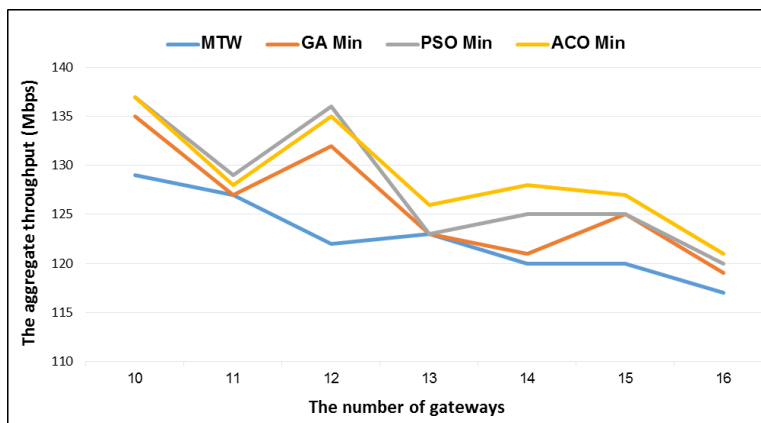


Figure 9. The comparison of the worst case of per client throughput in the second experiment

We find that the results achieved by our algorithm are better than the results achieved by MTW algorithm in all experiments. Next, we easily realize the fact that when the number of gateways increase, the throughput might not be better. So when designing the WMN, it is necessary to choice the number of gateways suitably to maximum the throughput of WMN and reduces the cost.

Final, the optimal solution of my algorithm in the third experiment is:

- Gateway placements optimal:	8 28 10 25 6 5 15 34 13 32 2 17 30 22 27 35 18
- The total throughput of all mesh clients:	76.65
- The minimal throughput of each mesh client:	0.0119

The results show us once again the superiority of the algorithms proposed in this paper. The results show that both PSO and ACO has better properties compared to GA algorithm.

## 7. CONCLUSIONS

The problem of gateway placement in WMNs for enhancing throughput was investigated continuously in this paper. A gateway placement algorithm was proposed based on GA, PSO, and ACO. A non-asymptotic analytical model was also derived to determine the achieved throughput by a gateway placement algorithm. Based on such a model, the performance of the proposed gateway placement algorithm was evaluated. Numerical results show that the proposed algorithms has achieved much better performance than other schemes. It is also proved to be a cost-effective solution. The best performance I got with the PSO and ACO algorithms with the lowest cost function and consuming time. While the ACO algorithm doesn't converges as fast and its results is litter worse than PSO. Optimizing gateway placement together with throughput maximization is our next research goal.

## ACKNOWLEDGEMENT

This research is partly supported by the NC107-2012 project of Haiphong University, Haiphong, Vietnam.

## REFERENCES

- [1] F. Akyildiz, Xudong Wang, Weilin Wang: "Wireless mesh networks: a survey", "Computer Networks and ISDN Systems", vol.47 no.4, pp. 445-487, 15 March 2005.
- [2] Ping Zhou, Xudong Wang, B. S. Manoj and Ramesh Rao (2010), "On Optimizing Gateway Placement for Throughput in Wireless Mesh Networks", Journal on Wireless Communications and Networking Volume 2010 (2010), Article ID 368423, 12 pages doi:10.1155/2010/368423
- [3] P. Gupta and P. R. Kumar, "Internets in the sky: The capacity of three dimensional wireless networks," Commun. Inform and Syst., vol.1, no. 1, pp. 33-49, Jan 2001.
- [4] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc IEEE INFOCOM'01*, 2001.
- [5] B. Liu, Z. Liu and D. Towsley, "On the capacity of hybrid wireless networks," in *Proc IEEE INFOCOM '03*, 2003.
- [6] U. C. Kozat and L. Tassiulas, "Throughput capacity of random ad hoc networks with infrastructure support," in *Proc ACM MOBICOM'03*, 2003.
- [7] A. Zemlianov and G. de Veciana, "Capacity of ad hoc wireless networks with infrastructure support," IEEE J. Sel. Areas Commun., vol.23, no.3, pp. 657-667, Mar 2005.
- [8] P. Zhou, X.Wang, and .R. Rao, "Asymptotic Capacity of Infrastructure Wireless Mesh Networks," IEEE Transaction on Mobile Computing, vol. 7, no.8, pp. 1011-1024, Aug. 2008.
- [9] J. Robinson and E. Knightly, "A Performance Study on Deployment Factors in Wireless Mesh Networks," in *Proc. IEEE Infocom 2007*, pp. 2054-2062, May 2007.
- [10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents", IEEE Trans. on System, MAN, and Cybernetics-Part B, vol. 26, pp. 29-41, February 1996
- [11] E. Rajo-Iglesias, O. Quevedo-Teruel, "Linear Array Synthesis using an Ant Colony Optimization based Algorithm", IEEE Trans. on Antennas and Propagation, 2005.
- [12] M. Dorigo, M. Birattari, and T. Stitzle, "Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique", IEEE computational intelligence magazine, November, 2006.
- [13] James Kennedy and Russell Eberhart, "Particle swarm optimization" in *Proceedings of IEEE International Conference on Neural Networks*, pages 1942-1948, Piscataway, NJ, USA, 1995.

- [14] Vinh Trong Le, Nhu Gia Nguyen, Dinh Huu Nghia, "A Novel PSO-Based Algorithm for Gateway Placement in Wireless Mesh Networks", in *Proc, 3<sup>rd</sup> IEEE International Conference on Communication Software and Networks (ICCSN)*, China, pp. 41-46, 2011.
- [15] Dac-Nhuong Le, Nhu Gia Nguyen, Nguyen Dang Le, and Vinh Trong Le (2012), A New Evolutionary Approach for the Optimal Location of Controllers in Wireless Networks, *The 2<sup>nd</sup> International Conference on Information Communication and Management (ICICM 2012)*, Hongkong.
- [16] Dac-Nhuong Le, Nhu Gia Nguyen, Nghia Huu Dinh, Nguyen Dang Le, and Vinh Trong Le (2012), Optimizing Gateway Placement in Wireless Mesh Networks based on ACO Algorithm, *The International Conference on Communication and Network Security (ICCNS 2012)*, Wuhan, China.
- [17] Dac-Nhuong Le and Nhu Gia Nguyen, *A New Evolutionary Approach for Gateway Placement in Wireless Mesh Networks*, International Journal of Computer Networks and Wireless Communications (IJCNWC), Vol.2, No.5, pp.550-555, 2012, India
- [18] Dac-Nhuong Le, Nhu Gia Nguyen, and Vinh Trong Le, *A Novel PSO-Based Algorithm for the Optimal Location of Controllers in Wireless Networks*, International Journal of Computer Science and Network Security (IICSNS), Vol.12 No.08, pp.23-27, 2012, Korea.
- [19] Dac-Nhuong Le, *PSO and ACO Algorithms Applied to Optimizing Location of Controllers in Wireless Networks*, International Journal of Computer Science and Telecommunications (IJCST), Vol.3, No.10, pp.1-7, 2012, England.
- [20] Dac-Nhuong Le, *PSO and ACO Algorithms Applied to optimal Resource Allocation to Support QoS Requirements in Next Generation Networks*, International Journal of Information & Network Security (IJINS), ISSN: 2089-3299, Vol.2, No.3, pp.216-228, 2013.

#### BIOGRAPHY OF AUTHOR



##### **Dac-Nhuong Le**

He received the BSc degree in computer science and the MSc degree in information technology from College of Technology, Vietnam National University, Vietnam, in 2005 and 2009, respectively. He is a lecturer at the Faculty of information technology in Haiphong University, Vietnam. He is currently a Ph.D student at Hanoi University of Science, Vietnam National University.

His research interests include algorithm theory, computer network and networks security.