ACO and PSO Algorithms Applied to Gateway Placement Optimization in Wireless Mesh Networks

Dac-Nhuong Le 1, Nhu Gia Nguyen 2, Nguyen Dang Le 1, Nghia Huu Dinh 3 and Vinh Trong Le 4
1 Faculty of Information Technology, Haiphong University, Vietnam
2 Duytan University, Danang, Vietnam
3 School of Graduate Studies, Vietnam National University, Vietnam
4 Hanoi University of Science, Vietnam National University

Abstract. In this paper, we study the challenging problem of optimizing gateway placement for throughput in Wireless Mesh Networks and propose a novel algorithm based on Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) for it. A gateway placement algorithm was proposed based on ACO, we generate the locations of gateway randomly and independently then 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. Our algorithm was proposed based on PSO, we calculate the fitness value of each scheme and update them step by step with the best method to quickly find the optimal. Numerical results show that the proposed algorithm has achieved much better than previous studies.

Keywords: Wireless Mesh Networks, Gateway Placement, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO).

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.

Fig.1. A typical WMN
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. Fig. 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 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.

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 presents our new algorithm for gateway placement in optimizing WMN based on ACO and PSO. Section 4 presents our simulation and analysis results, and finally, section 5 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 Fig. 2.

![Network topology of an WMN infrastructure with gateways](image)

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 $R'$, 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,$\sum_{i=1}^{N_c} TH(i, N_g)$ (1) is maximized, where $TH(i, N_g)$ denotes the per client throughput of the $i^{th}$ 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, $\min_{i=1}^{N_c} TH(i, N_g)$ (2) is maximized.

### 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.

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:

*Assign value of 100% for all the gateways. For the top row to the last row of the table in the first step $k=1/the number of gateways in current group.*
For the first gateway to the last gateway in current group
If percentage value > k then push into subgroup1
Else push into subgroup2.
End for.

\[ P = \frac{\text{1-sum of all the percentage value in subgroup2}}{\text{the number of the gateways in subgroup1}} \]

Assign value of P for all gateways in subgroup1.
End for.

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

2.1.5 Throughput Computation

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

\[ TH(i, N_g) = \min_{i=1..N_c} \left\{ TH_{W_1}(i, N_g), TH_{W_2}(i) \right\}, \]  

(3)

Here, \( TH_{W_1}(i, N_g) \) is defined as the throughput of the \( i \)th mesh client in backbone communications and \( TH_{W_2}(i) \) is defined as the throughput of \( i \)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:

\[ TH_{W_1}(i, N_g) = \frac{\sum_{k=\text{all the associated gateways with the mesh router } R_i} TH_g'(k)}{N_g(j)} \]

(4)

Here, \( N_g(j) \) is the number of gateways associated with the mesh router \( R_i \). 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 \( k \)th gateway can guarantee for all its associated mesh clients in backbone communications.

\[ TH_g'(k) = \frac{G_{\text{eff}}(k) \times c_1 W_1}{\sum_{l=\text{all the associated routers with the } k \text{th gateway}} (N_c(l) \times N_{\text{hop}}'(l) + N_c(l))} \]

(5)

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

\[ N_{\text{hop}}'(j) = N_{\text{hop}}(j), \text{if } N_{\text{hop}}(j) < \text{SRD}; \]

\[ N_{\text{hop}}'(j) = \text{SRD}, \text{if } N_{\text{hop}}(j) \geq \text{SRD}; \]

(6)

Here, \( N_{\text{hop}}(j) \) is the number of hops from the mesh client to the gateway. \text{SRD} is defined as Slot Reuse Distance. Next, \( TH_{W_2}(i) \) is computed simply as follows: \( TH_{W_2}(i) = \frac{c_2 W_2}{CRF \times N_c(j)}, i=1..N_c \)  

(7)

Here, \( c_2 W_2 \) is the throughput that \( R_i \) 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 \( R_i \), \( j=1..N_c \). Each time, the router with the highest weight will be chosen to place a gateway. The weight computation is adaptive to the following factors:

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: 
\[
R_g = \text{round}\left(\frac{\sqrt{N}}{2\sqrt{N_r}}\right) \quad (8)
\]

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 \(R^i\). \(MTW(j)\) is calculated with \(D(j)\) and \(R_g\) as follows:
\[
MTW(j) = (R_g+1) \times D(j) + R_g \times \text{(traffic demand on all 1-hop neighbors of } R^i) + (R_g-1) \times \text{(traffic demand on all 2-hop neighbors of } R^i) + \ldots
\]

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 \(R^i\). Re-calculate \(MTW(j)\) with the new \(D(j)\), and perform the following procedure.

1. Choose the router with the highest weight as potential location for gateway placement, namely \(R^i\).
2. Re-construct the table of non-overlapping interfere groups with \(R^i\) and previous gateways.
3. Compute the sharing efficiency for \(R_j\).
4. \(MTW'(j) = MTW(j) \times Geff(j)\)MTW'(j). If \(MTW'(j)\) 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 WMN using ACO and PSO

3.1. Apply ACO algorithm to Gateway Placement Problem

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 [10], and after then, other implementations of the algorithm have been developed [11-12].

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 \(A_{m \times n}\) (where \(m\) is the number of gateways, \(n\) is number of client). \(a_{ij}\) will then receive a random integer generated correlatively.

The pseudo code of the procedure for each element

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

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 \([0, N-1]\). 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:
\[
p_{ij}^k = \frac{\left(\tau_{ij}\right)^\alpha \left(\eta_{ij}\right)^\beta}{\sum_{i \in N_i^k} \left(\tau_{ij}\right)^\alpha \left(\eta_{ij}\right)^\beta}
\]

(9)

In which, \(\tau_{ij}\) is the pheromone content of the path from gateway \(i\) to client \(j\). \(N_i^k\) is the neighbourhood 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.
\[
\tau_{ij}^{\text{new}} = \tau_{ij}^{\text{current}} + \Delta \tau_{ij}^k
\]

(10)
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

$$\Delta \tau_j^k = \frac{1}{f_j}$$  \hspace{1cm} (11)

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:

$$\tau_j^\omega = (1 - \rho) \tau_j \hspace{0.5cm} (0 < \rho \leq 1)$$  \hspace{1cm} (12)

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

### 3.2. Apply PSO algorithm to Gateway Placement Problem

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.

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 $\{g_1, \ldots, g_k\}$, in which if the $j^{th}$ element is $\{a'_1, \ldots, a'_k\}$ then $a'_j$ would correspond to the gateway $g_j$, and its value will be a random integer generated correlatively. Assume that the WMN model, presented in Section 2.1, is divided into $N$ cells and numbered from left to right and from top to bottom. $a'_j$ will then receive the value in the range of $[0,(N-1)]$.

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 $[0,N-1]$. Fitness value of $j^{th}$ element is calculated by the following formula:

$$F_j = 1 - \frac{1}{\sum_{i=1}^{N_c} TH(i,K)}$$  \hspace{1cm} (13)

In which, $N_c$ is the number of clients, $K$ is the number of gateways, $TH(i,K)$ is computed by the formula (3). Elements in each generation are updated according to formula (14) and (15) described below. In which $\text{present}[j]$ and $\text{v}[j]$ are respectively the $j^{th}$ element in the current generation and its speed. In the context of the current problem, $\text{present}[j]$ and $\text{v}[j]$ are $K$-dimensional vectors.

$$\text{v}[j] = \text{v}[j] + c_1 \times \text{rand()} \times (\text{pbest}[j] - \text{present}[j]) + c_2 \times \text{rand()} \times (\text{gbest}[j] - \text{present}[j])$$  \hspace{1cm} (14)

$$\text{present}[j] = \text{present}[j] + \text{v}[j]$$  \hspace{1cm} (15)

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. Our results have been published in [13-14].

### 4. Numerical Results and Discussion

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.

We study two experiments. In the first experiment we assume $N_c=200$, $N_r=36$, $l=1000m$, i.e. there are 200 mesh clients distributed in a square region of 1000m x 1000m; the square is split evenly into 36 small square cells and a mesh router is placed in the centre of each cell. Concurrently, we 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 each experiment, we optimize the gateway placement problem by maximizing one of two parameters: the total throughput of all mesh clients, denoted as \(ACO\_Sum, PSO\_Sum\), and the minimal throughput of each mesh client, denoted as \(ACO\_Min, PSO\_Min\). Then we compare our results with the results achieved by MTW-based gateway placement algorithm.

Firstly, we compare the aggregate throughput and the worst case throughput achieved by each algorithm, as shown in Fig. 3 and Fig. 4.

![Fig. 3. The comparison of the aggregate throughput (a) and the worst case of per client throughput (b) in the first experiment](image)

![Fig. 4. The comparison of the aggregate throughput (a) and the worst case of per client throughput (b) in the second experiment](image)

We find that the results achieved by our algorithm are better than the results achieved by MTWP 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 choose the number of gateways suitably to maximize the throughput of WMN and reduces the cost.

Final, we compare throughput per gateway of two gateway placement algorithm, as shown in Fig. 5.

![Fig. 5. The comparison of the aggregate throughput per gateway](image)

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

5. Conclusion

The problem of gateway placement in WMNs for enhancing throughput was investigated continuously in this paper. A gateway placement algorithm was proposed based on ACO, PSO. 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 algorithm has achieved much better performance than other schemes. It is also proved to be a cost-effective solution. The best performance we got with the PSO algorithm 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.

6. References