Particle Swarm Optimization Applied to Mobile Node Placement in Wireless Mesh Networks

Chun-Cheng Lin* and Yi-Ling Lin
Department of Industrial Engineering and Management
National Chiao Tung University
Email: cclin321@nctu.edu.tw, celebratory105@hotmail.com

Abstract

This paper investigates the placement problem of mesh routers in wireless mesh networks (WMNs), in which the network access of mesh clients is accomplished through the gateway and bridging functions of mobile mesh routers with different radio coverage. Different from previous works that focused on static network scenarios, this paper considers a dynamic network scenario where both mesh clients and mesh routers have mobility, while the locations of mobile mesh routers can be adjusted to adapt to the topology changes. Under such a framework, this paper is concerned about the problem of how to determine the dynamic placements of mobile mesh routers in a geographical area at different time to maximize two main network performance measures: the network connectivity and client covering, which are measured by the size of the greatest component of the WMN topology and the number of the clients within radio coverage of mesh routers, respectively. In general, it is computationally intractable to solve the optimization problem for the above two performance measures even in static network scenarios. As a result, in this paper we define a formal mathematical program of our concerned problem, and propose to use a particle swarm optimization (PSO) approach. Experimental results show the quality of the proposed approach through sensitivity analysis, and also show that our proposed approach can adapt to the topology changes.

Keywords: Router node placement, wireless mesh network, particle swarm optimization, metaheuristic

* Corresponding author.

I. INTRODUCTION

As wireless mesh networks (WMNs) provide high flexibility of wireless connectivity everywhere within a wide geographical region due to their high-speed self-configuration capability and low set-up cost, there have existed a variety of research challenges, e.g., the routing problems with different concerns (Alotaibi and Mukherjee, 2012), the assignment problems for multi-channel WMNs (Ramamurthi et al., 2011), the security problems in WMNs (Muogilim et al., 2011), among others.

This paper is concerned with the problem of routing node placement in WMNs (Barolli et al., 2011; Xhafa et al., 2011; Xhafa et al., 2010). Consider the WMN composed of mesh clients and mesh routers in which mesh routers serve as the access point towards mesh clients and connect to other mesh routers through point-to-point wireless links. To respond to heterogeneity of WMNs in practice, each mesh router is assumed to have a different size of radio coverage, and two mesh routers can communicate only if their radio coverage overlaps. On the other hand, mesh clients only have the essential functions for network connectivity and routers, but do not have the function of gateways or bridges. Hence, mesh clients must go through mesh routers to communicate with other nodes. That is, the network access of WMNs is accomplished through the gateway and bridging functions of mesh routers.

The performance of WMNs mainly depends on the geographical placement of mesh routers and mesh clients, in which the placement of mesh routers plays a more crucial role as it determines the connectivity and coverage of the whole network. If the mesh routers are placed without taking into account specific restrictions of real geographic area and the topology underlying WMNs, it would lead to poor networking performance. In practice, however, it would be impossible to find an optimal placement of mesh nodes, since the distribution of real mesh clients cannot be predicted. Hence, we assume that the locations of mesh clients are given fixed in the deployment area by uniform distribution. But even by doing so, it is still computationally hard to achieve the optimality of the problem (Garey and Johnson 1979; Wang et al., 2007), and therefore, in practice, the mesh node deployment problems for WMNs are usually solved by metaheuristic approaches. The purpose of using metaheuristic approaches is to achieve near-optimal solutions in reasonable time, when the real optimal solution cannot be found in deterministic polynomial time. Although metaheuristic methods usually find local optimal solutions, they suffice for most practical situations.

With the advancement in communication technologies, more and more studies on wireless mobile networks have appeared, e.g., mobile ad hoc networks (Cheng and Yang, 2010), mobile sensor network (Hwang et al., 2011), among others. In this paper, we consider the WMN in which both mesh clients and mesh routers[1] have mobility, while the locations of mesh routers can be adjusted to adapt the topology changes. In the dynamic scenario, we investigate the problem of mobile node placement in wireless mesh networks (WMN-MNP) is to find a dynamic placement of mesh routers in the rectangular area at different time to maximize both the network connectivity and client covering. In the formulation, the network connectivity is measured by the size of the greatest component of the topology underlying the WMNs, while the client covering is the number of clients within the radio coverage of mesh routers, which can represent an index of the QoS for the WMNs. Note that the two measures have different values in different topologies. Some works for the static network scenario of our concerned problems have been solved by simulated annealing (Xhafa et al., 2011) and genetic algorithm (Barolli et al., 2011; Xhafa et al., 2010).

The PSO simulates a population of particles to solve optimization problems over continuous space by the guidance of both a particle's own experience and other particles' experiences, which allow simple mathematical formulas to find promising solutions and has been proven to be efficient and effective in various applications, e.g., multiuser detector for CDMA communications (Soo et al., 2007), minimum bit error rate multiuser transmission designs (Yao et al., 2009), optimal stochastic signaling for communications systems (Goken et al., 2010) among others. Hence, it is of interest and importance to investigate the WMN-MNP problem by the PSO approach. This paper provides an efficient and effective particle swarm optimization approach for the WMN-MNP problem, discusses the effect of different parameters on the influences of the WMNs, and evaluates the quality of the proposed approach through detailed simulations.

The rest of this paper is organized as follows. In Section II, we describe our concerned problem in a formal form, and propose a PSO approach to the problem in Section III. Section IV shows the experimental results. A conclusion is given in Section V.

II. PROBLEM DESCRIPTION

A wireless mesh network (WMN) consists of two types of nodes: mesh routers and mesh clients. Each mesh client can only communicate with the node within the same radio coverage or any node that can be accessed via multi-hop router communications. That is, any mesh client cannot communicate with other nodes in the network if it is not located in the radio coverage of some mesh router. To respond to heterogeneity of WMNs in practice, we assume that mesh routers have different radio coverage ranges. Note that this paper considers a dynamic scenario, i.e., both mesh routers and mesh clients have mobility, and the network topology changes in time.

Consider a WMN with n mesh routers and m mesh clients deployed in a two-dimensional area. The mesh nodes in the WMN are denoted by $U = R \cup C$ in which

[1] A product of mobile mesh router can be found in http://www.tropos.com/pdf/datasheets/tropos_datasheet_4210.pdf

- $R = \{r_1, r_2, \ldots, r_n\}$ where each r_i is a mesh router and has a radio coverage γ_i;
- $C = \{c_1, c_2, \ldots, c_m\}$ where each c_i is a mesh client.

In dynamic scenario, each mesh client changes its location in time such that the network topology evolves, while we adjust the locations of mobile mesh routers to adapt to the topology changes. We assume that the location of each mesh router is determined periodically, i.e., at each different key time point, mesh clients may have different locations, and the placements of mesh routers are determined according to the deployment of mesh clients in the geographical area. Such a framework is modelled as follows. At the t-th key time point, each mesh client $c_i \in C$ is located at $D_t(c_i) \in \mathbb{R}^2$ in the deployment area. According to the mesh client deployment at each t-th key time point, we determine the placements of mesh routers, denoted by $D_t(R) = \{D_t(r_1), D_t(r_2), \ldots, D_t(r_n)\}$. Let the circle centered at the location $D_t(r_i)$ of node r_i with radius γ_i be denoted by Υ_i^t. For a determined placement of mesh routers at the t-th key time point, we can establish a topology graph $G_t = (U, E_t)$ in which

- if $\Upsilon_i^t \cap \Upsilon_j^t \neq \emptyset$ for any two mesh routers $r_i, r_j \in$ R, then $(r_i, r_j) \in E_t$;
- if $D_t(c_i) \in \Upsilon_j^t$ for any mesh client $c_i \in C \setminus S_t$ and any mesh router $r_j \in R$, then $(c_i, r_j) \in E_t$.

It should be noticed that graph G_t may not be connected, i.e., G_t may consist of several subgraph components. In order to increase the network connectivity of WMN, we would like to make the size of the greatest subgraph component as large as possible. However, a large size of the greatest component does not imply a wide radio coverage of mesh clients, and hence, we consider client coverage as the other concerned design on placement of mesh routers.

Assume that there are h subgraph components $G_t^1, \ldots, G_t^h$ in G_t, i.e., $G_t = G_t^1 \cap G_t^2 \cap \ldots G_t^h$, and $G_t^i \cap G_t^j = \emptyset$ for $i, j \in \{1, \ldots, h\}$. The size of the greatest subgraph component in G can be expressed as follows:

$$\phi(G_t) = \max_{i \in \{1,\ldots,h\}} \{|G_t^i|\}. \tag{1}$$

The client coverage can be expressed as follows:

$$\varphi(G_t) = |\{i; d_t(c_i) > 0 \text{ for } i \in \{1,\ldots,h\}\}|. \tag{2}$$

where $d_t(c_i)$ is the degree of node c_i in topology graph G_t.

With the above notation, our concerned problem is to find a dynamic placement $D_t(R)$ of mesh routers in a WMN to adapt the topology change at any t-th key time point so that both the size of greatest comments $\phi(G_t)$ and the client coverage $\varphi(G_t)$ are as large as possible. Hence, our concerned problem can be stated as follows: *For t = 1, 2, …, we are given a topology graph $G_t = (U, E_t)$ underlying a WMN at the t-th key time point (as described above) distributed in a two-dimensional $W \times H$ area where mesh clients have mobility, while the locations of mesh routers need be determined to adapt the topology changes. The objective of the problem is to find a dynamic placement $D_t(R) = \{D(r_1), \ldots, D(r_n)\}$ of the n mesh routers so* *that both the size of the greatest subgraph component $\phi(G_t)$ and the client coverage $\varphi(G_t)$ are maximized.*

Note that the WMN-MNP problem at a fixed key time point is a static scenario for the problem. It is obvious that the static scenario problem contains the following NP-complete problem – *Minimum Geometric Disk Cover* (Johnson, 1982; Hochbaum and Maass, 1985). Hence, the problem is an NP-hard problem, i.e., it cannot be solved by an efficient deterministic polynomial-time algorithm. Hence, this paper devises a PSO metaheuristic approach to the WMN-MNP problem, which provides an efficient promising solution, as compared to the NP-hardness.

III. A PSO APPROACH TO THE CONCERNED PROBLEM

The PSO is an algorithm used for solving optimization problems based on the social intelligence of a swarm of particles, proposed by (Eberhart and Kennedy, 1995; Kennedy, 1997). The basic idea of PSO is to simulate a swarm of particles' social behavior of searching a food source in a multi-dimensional search space. It is assumed that each particle has a position and a velocity, which are updated by iteration. Each particle moves toward the food source by the guidance of its local best known position (which has been found so far by itself) and also the global best known position (which has been found so far by any particle of the swarm). The food source is associated with the global optimal solution of the concerned optimization problem, while the location of each particle in the search space is associated with a candidate solution. Hence, the global optimal solution can be found if almost all the candidate solutions move to the same position in the search space.

This section gives in detail our PSO approach to the WMN-MNP problem. We first give the solution representation of each particle, then the fitness function used in the PSO approach, then the scheme of updating positions of each particle, and finally our PSO algorithm.

3.1 Solution Representation

The solution of our concerned WMN-MNP problem is a placement of n mesh routers in a two-dimensional $W \times H$ area, whose lower-left corner is place at the origin of an $x \times y$ plane. That is, the (x, y)-coordinates of the n mesh routers should be determined for each candidate solution.

In PSO, each particle k represents a candidate solution, which is determined by the following three vectors and two fitness values:

- the x-vector $X_k^{\,t} = (x_{k1}^{\,t}, x_{k2}^{\,t}, \ldots, x_{k(2n)}^{\,t})$ records the current position of the particle in the search space where $(x_{k(2i-1)}^{\,t}, x_{k(2i)}^{\,t})$ denotes the (x, y)-coordinate of mesh router r_i for i = 1, 2, …, n;
- the p-vector $P_k^{\,t} = (p_{k1}^{\,t}, p_{k2}^{\,t}, \ldots, p_{k(2n)}^{\,t})$ records the location of the best solution found so far by particle k;

- the v-vector $V_k{}^t = (v_{k1}{}^t, v_{k2}{}^t, \ldots, v_{k(2n)}{}^t)$ records the velocity along which particle k will move;
- $f(X_k{}^t)$ records the fitness of $X_k{}^t$;
- $f(P_k{}^t)$ records the fitness of $P_k{}^t$.

Since all the mesh routers are placed within a $W \times H$ area, we have the following constraints: $\forall\, i \in \{1, \ldots, n\}$,

$$0 \le x_{k(2i-1)}{}^t \le W, \quad (3)$$

$$0 \le x_{k(2i)}{}^t \le H, \quad (4)$$

$$-W \le v_{k(2i-1)}{}^t \le W,$$

$$-H \le v_{k(2i)}{}^t \le H.$$

In order to avoid drastic change of velocities, we have the following constraints for velocities: $\forall\, i \in \{1, \ldots, n\}$,

$$-V_{max} \le v_{k(2i-1)}{}^t \le V_{max}, \quad (5)$$

where V_{max} is a given constant that is no more than $\max\{W, H\}$.

From the view point of the whole swarm, a vector and a fitness of the best solution that have been found so far are stored in each iteration:

- $P^* = (p_1^*, p_2^*, \ldots, p_{2n}^*)$ records the location of the best solution found so far by all particles;
- $f(P^*)$ records the fitness of P^*.

That is, after finishing the PSO algorithm, P^* and $f(P^*)$ store the location and the fitness value (objective value) of the final solution, respectively.

3.2 Fitness Function

Given a placement $X_k{}^t$ of mesh routers for particle k, we can obtain a graph $G_{t,k}$ underlying the WMN. Recall that the objective of our concerned problem is to maximize the size of the greatest subgraph component $\phi(G_{t,k})$ and the client coverage $\varphi(G_{t,k})$, which can be calculated by Equation (1) and Equation (2), respectively.

The fitness function of X_k is calculated as follows:

$$f(X_k^t) = \lambda \cdot \phi(G_{t,k})/(n+m) + (1-\lambda) \cdot \varphi(G_{t,k})/m \quad (6)$$

where λ is a parameter in the range [0, 1] that controls the balance between the two terms of the equation. Note that the denominator of each term of the equation is used for normalization.

There are two possible settings for the WMN-RNP problem with two objectives. The first setting is bi-level optimization, which considers $\phi(G_{t,k})$ as the main objective, while $\varphi(G_{t,k})$ as the second one. That is, the setting has high priority to optimize the first objective, and then optimize the second objective without worsening the best value of the first objective. The second setting is simultaneous optimization, which optimizes the two objectives simultaneously. Our fitness function in Equation (6) applies the second setting, while the previous work (Xhafa et al., 2011) applies the first setting. This is a difference of our work from the previous work.

3.3 Updating Position

Each iteration of the main loop of PSO updates the velocity vector V_k^t by the following formula:

$$V_k^{t\prime} = \omega [V_k^t + c_1 \cdot r_1 \cdot (P_k - X_k^t) + c_2 \cdot r_2 \cdot (P_k^* - X_k^t)] \quad (7)$$

where $c = c_1 + c_2 > 4$; $\omega = 2/|2 - c - (c^2 - 4c)|$; r_1 and r_2 are two random numbers 0 and 1. The position of each particle k by the following formula:

$$X_k^{t\prime} = X_k^t + V_k^{t\prime} \quad (8)$$

where $X_k^{t\prime}$ and $V_k^{t\prime}$ are new values of position vector X_k^t and velocity vector V_k^t of particle k, respectively. Note that a PSO approach with updating formula of Equation (7) is called the PSO with *constriction coefficient* (Clerc and Kennedy, 2002).

3.4 Algorithm

The PSO algorithm works with a swarm of particles, each of which represents a candidate solution of the WMN-MNP problem at the t-th key time point. For dynamic scenario, we consider two cases to initialize each particle: in the case when $t = 0$, the particle is located at a random position of the search space; otherwise (i.e., $t > 0$), the particle is located at its position of the search space at the (t-1)-th key time point. By doing so, particles can inherit the intelligence at the previous key time points. Then each particle searches the food source (optimal solution) in the search space by iteration, i.e., it moves toward the food source at a velocity, which is updated by iteration. The key design of the PSO is to update the velocity by the guidance of its own experience (local best known solution) and also the experience of other particles (the global best known solution), as described in Equation (7).

Let η denote the total number of particles. The algorithm of PSO is given in Algorithm 1, in which the key steps are explained as follows. Lines 1 – 13 are initialization of all the vector variables X_k^t, P_k^t, P^*, V_k^t and all the fitness values for each particle k. Line 1 considers each particle k. Line 2 judges whether t is the starting key time point. If the condition is true, then Lines 3 and 4 initialize X_k^t and V_k^t by their values X_k^{t-1} and V_k^{t-1} at the previous key time point, respectively; otherwise, Lines 6 and 7 initialize X_k^t and V_k^t randomly under Constraints (3), (4), and (5), where $U(a, b)$ is a uniform distribution over $[a, b]$. Once X_k^t is determined, its fitness $f(X_k^t)$ is calculated by Equation (6). Line 9 assigns the current position X_k^t and the current fitness value $f(X_k^t)$ to the local best position P_k^t and the corresponding fitness value $f(P_k^t)$, respectively. Lines 10 – 12 updates the global best position P^* and its fitness value $f(F^*)$ that have been found so far.

Algorithm 1 PSO(key time point t)

1: **for each** $k \in \{1, 2, \ldots, \eta\}$ **do**
2: **if** $t > 0$ **then**
3: $X_k^t \leftarrow X_k^{t\text{-}1}$, and calculate its fitness $f(X_k^{t\text{-}1})$
4: $V_k^t \leftarrow V_k^{t\text{-}1}$
5: **else**
6: initialize particle k's position $X_k^t = (x_{k1}^{\ t}, \ldots, x_{k(2n)}^{\ t})$ randomly where $x_{k(2i\text{-}1)}^{\ t} \sim U(0, W)$ and $x_{k(2i)}^{\ t} \sim U(0, H)$ for each $i \in \{1, 2, \ldots, n\}$, and calculate its fitness $f(X_k^t)$.
7: initialize particle k's velocity $V_k^t = (v_{k1}^{\ t}, \ldots, v_{k(2n)}^{\ t})$ randomly where $v_{k(2i\text{-}1)}^{\ t} \sim U(0,W)$ and $v_{k(2i)}^{\ t} \sim U(0, H)$ for each $i \in \{1, 2, \ldots, n\}$.
8: **end if**
9: $P_k^{\ t} \leftarrow X_k^{\ t}$ and $f(P_k^{\ t}) \leftarrow f(X_k^{\ t})$.
10: **if** $f(P_k^{\ t}) > f(P^*)$ **then**
11: $P^* \leftarrow P_k^{\ t}$ and $f(P^*) \leftarrow f(P_k^{\ t})$
12: **end if**
13: **end if**
14: **repeat**
14: **for each** $k \in \{1, 2, \ldots, \eta\}$ **do**
17: update particle k's velocity $V_k^{\ t}$ by Equation (7).
18: $\forall\ i \in \{1, \ldots, 2n\}$, $v_i^{\ t}$ is truncated if violating Constraint (5).
19: update particle k's position $X_k^{\ t}$ by Equation (8).
20: $\forall\ i \in \{1, \ldots, 2n\}$, $x_i^{\ t}$ is truncated if violating Constraints (3) and (4).
21: calculate $f(X_k^{\ t})$.
22: **if** $f(X_k^{\ t}) > f(P_k^{\ t})$ **then**
23: $P_k^{\ t} \leftarrow X_k^{\ t}$ and $f(P_k^{\ t}) \leftarrow f(X_k^{\ t})$.
24: **if** $f(P_k^{\ t}) > f(P^*)$ **then**
25: $P^* \leftarrow P_k^{\ t}$ and $f(P^*) \leftarrow f(P_k^{\ t})$.
26: **end if**
27: **end if**
28: **end for**
29: **until** {the maximum iteration τ is reached or $f(P^*)$ exceeds a threshold}
30: output $f(P^*)$ as the solution at the t-th key time point

Lines 14 – 28 are the main loop of the PSO algorithm. For each iteration, we repeat Lines 15 – 27 until the total number of iterations is greater than the maximum iteration number or the best fitness value $f(P^*)$ found so far exceeds a threshold value. Each iteration of the main loop considers each particle k. The velocity V_k^t and the position X_k^t are updated according to Equations (7) and (8, respectively, in Lines 16 and 18. Since V_k^t and X_k^t need to satisfy Constraints (5), (3) and (4), they are truncated if those constraints are violated. Once the position X_k^t is updated, we calculate its fitness $f(X_k^t)$ in Line 21. In Lines 22 – 25, the local best position P_k^t and the global best position P^* are updated if a better position is found; their fitness values are updated at the same time.

Since $f(P^*)$ stores the global best solution of each iteration, Line 29 outputs the final best solution at the end of the PSO algorithm.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Based on the proposed PSO approach described in the previous sections, we implemented the proposed method. Some experiments were conducted for evaluating the performance of our proposed PSO approach. In this section, we first explain how the data used in the experiments were generated and the environment where our experiments were tested, and then give the experimental results in a variety of cases.

4.1 Data and Environment

Similar to (Xhafa et al., 2011), we consider the following three cases:

case 1: There are 32 mesh routers (m = 16) and 48 mesh clients (n = 48) on a 32 × 32 area (W = H = 32). Each mesh router is associated with a circular radio coverage with radius ~ U(3,6).

case 2: There are 64 mesh routers (m = 32) and 96 mesh clients (n = 96) on a 64 × 64 area (W = H = 64). Each mesh router is associated with a circular radio coverage with radius ~ $U(4\sqrt{2} - 2, 8\sqrt{2} - 2)$.

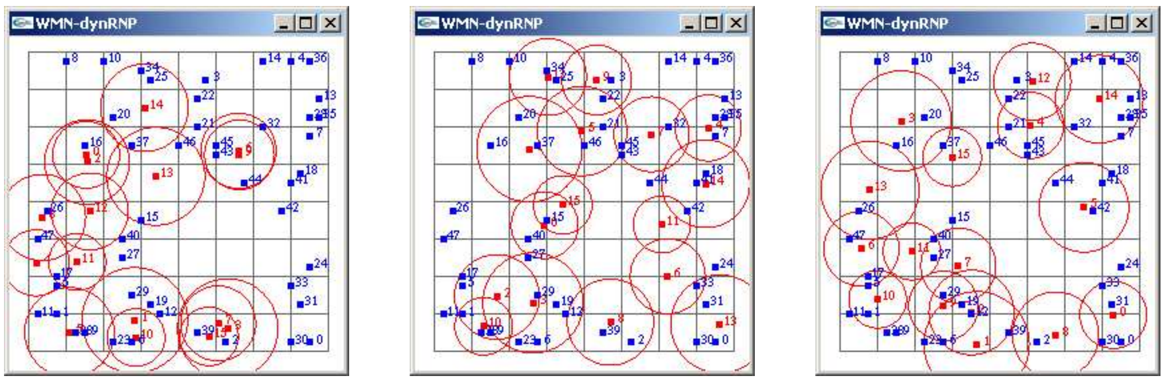
case 3: There are 128 mesh routers (m = 64) and 192 mesh clients (n = 192) on a 128 × 128 area (W = H = 128). Each mesh router is associated with a circular radio coverage with radius ~ $U(7, 14)$.

Note that the area is a grid in (Xhafa et al., 2011), meaning that each coordinate is an integer, while the coordinate in our setting is a floating-point number, which allows more general applications. For each case, we generate 10 instances, in which the mesh clients are distributed in the area according to a uniform distribution.

In our experiments, our PSO algorithm applies the following parameter settings: $\lambda = 0.3$, $c_1 = 3$, $c_2 = 2$, $V_{max} = 0.1$, $\eta = 100$, $\tau = 10$. The reason why we use $\lambda = 0.3$ is explained as follows. After executing a lot of experiments under a variety of λ values, we found that in the case of using a larger λ value, the radio coverage circles tend to gather in a dense area so that many mesh clients are not covered (e.g., see Figure 1(a)), while in the case of using a smaller λ value, the radio coverage circles dispersed over the area can cover most mesh clients but cannot constitute a larger component (e.g., see Figure 1(c)). That is, the λ value determines the tradeoff between our concerned two objectives (ϕ and φ) to some degree. Hence, we take a λ value that have a high probability to satisfy the two concerned objectives simultaneously.

After a lot of trial and error, we found that the results using $\lambda = 0.3$ perform better (e.g., see Figure 1(b)).

Our PSO algorithm was tested on an Intel Core i3-2100 CPU @ 3.1 GHz with 10 GB memory. The average running time for determining a placement of an instance of case~1, case~2, and case~3 are about 0.0185, 0.0733, and 0.2919 seconds, respectively. It implies that our PSO algorithm has the ability to cope with the WMN-dynRNP problem efficiently. As shown in Figure 1, we develop a visualization user interface, with which users can interact to adjust parameters of PSO accordingly to improve solutions.



(a) $\lambda = 1.0$: $\phi = 42$, $\varphi = 26$ (b) $\lambda = 0.3$: $\phi = 54$, $\varphi = 38$ (c) $\lambda = 0.0$: $\phi = 32$, $\varphi = 36$

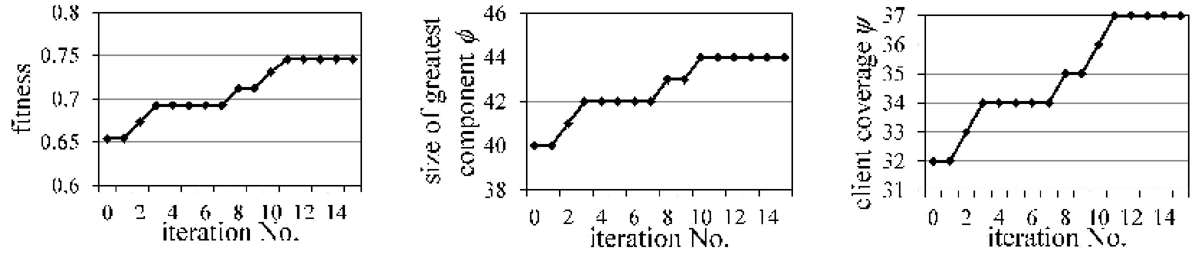
Figure 1. Visualization of the WMNs under three different λ values, in which each blue point is a mesh client, while each red point is a mesh router and acts as the center of a radio coverage circle.

4.2 Experimental Results

In order to observe the convergence of the best fitness values in our PSO method, we plot the best fitness values versus the number of iterations of the main loop of the PSO algorithm for a case-1 instance in Figure 2(a), and the plots of their corresponding ϕ and φ values versus the iteration numbers in Figures 2(a) and 2(b), respectively. From Figure 2(a), we can observe that the best fitness value converges at iteration 6. From Figures 2(b) and 2(c), both the two objective values can influence the best fitness value before convergence.

Subsequently, in order to demonstrate the ability of our method to adapt the topology changes, we run 100 iterations of the PSO with 100 particles on a case-1 instance in a static and three different dynamic scenarios, and their plots of fitness versus the iteration number are given in Figure 3 and Figure 3(b)-(d), respectively.

The three dynamic scenarios assume three different degrees of dynamics: in Figure 3(b)-(d), 16, 32, and 48 mesh clients (i.e., one-third, two thirds, and all of the mesh clients) change their locations in each 10 iterations, respectively. Note that for comparison, the same seed number for pseudorandom function in programming is used in each case in Figure 3, from which we observe that the run chart of the fitness values in the first 10 iterations has the same track.



(a) (b) (c)

Figure 2. (a) Plot of fitness values versus the iteration number. (b) Plot of the φ values versus the generation number. (c) Plot of the φ values versus the generation number.

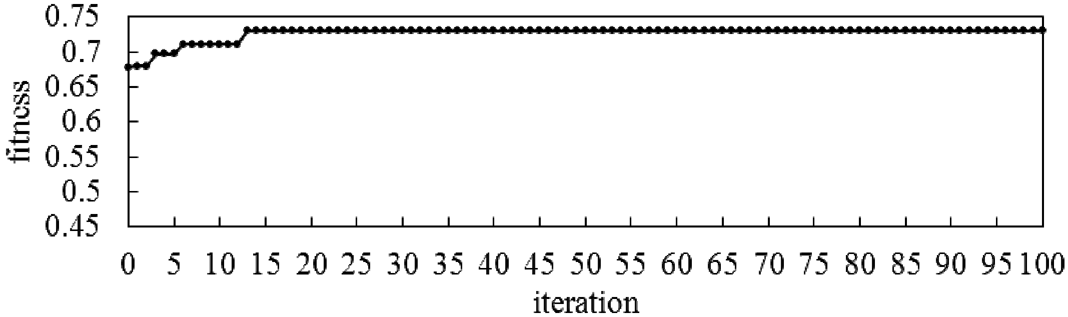
In the static scenario (Figure 3(a)), fitness values converge and do not have any change after 13-th iteration. When the number of mesh clients that change their locations is small (Figure 3(b)), fitness values almost do not change during each time period (i.e., within each 10 iterations), because the fitness value is still the optimal after a slight topology change. However, as the number of changing mesh clients is more and more (Figure 3(c) and 3(d)), the fitness has to improve to achieve the optimality during each time period, e.g., see the fitness values from 40th to 45th iterations in Figure 3(d). It can also be seen that a drastic modification of fitness values when there are more dynamics. In addition, we observe that the range accommodating all the fitness values in slight dynamics (e.g., range [0.6, 0.75] in Figure 3(b)) is smaller than that in large dynamics (e.g., range [0.45, 0.75] in Figure 3(d)).

V. CONCLUSION

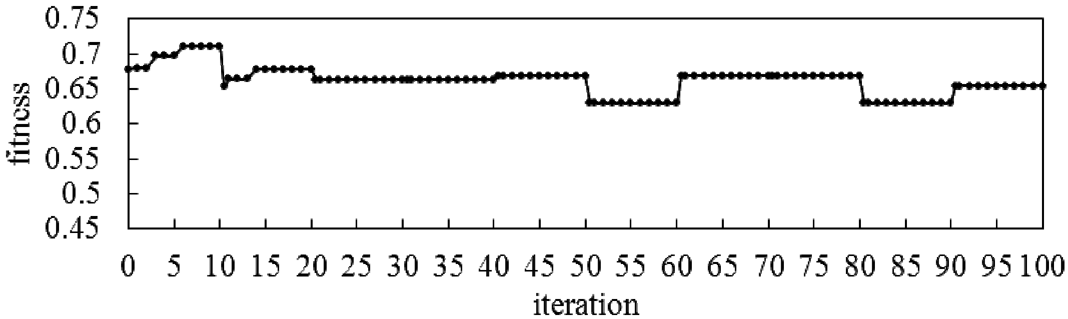
A PSO approach for optimizing the placement of mesh router nodes in WMNs has been proposed and implemented. One of our main contributions is to propose a formal mathematical formulation of the WMN-MNP problem, so that interested readers are able to investigate the problem precisely in the future. Our PSO approach is evaluated by discussing the effect of different parameters on influences of WMNs. A main difference from previous works is that our PSO allows mesh routers with high flexibility to be placed at floating-point positions in a continuous deployment area, and our problem optimizes two objectives at the same time. We also develop a visualization user interface to allow to observe the evolution of solutions to get more insight of the PSO algorithm.

REFERENCES

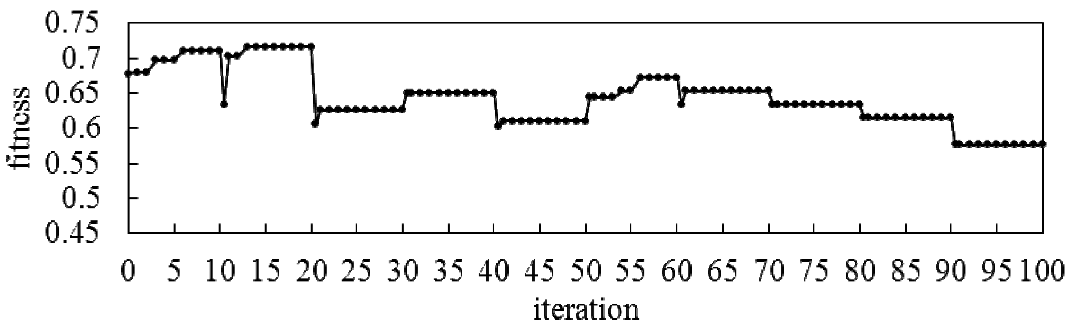
1. Alotaibi, E., and Mukherjee, B. “A survey on routing algorithms for wireless ad-hoc and mesh networks,” *Computer Networks* (56) 2012, pp: 940-965.
2. Barolli, A., Xhafa, F., Sanchez, C., and Takizawa, M. “A study on the effect of mutation in genetic algorithms for mesh router placement problem in wireless mesh networks,” *Proceedings of 5th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2011)*, IEEE Press, 2011, pp. 32-39.



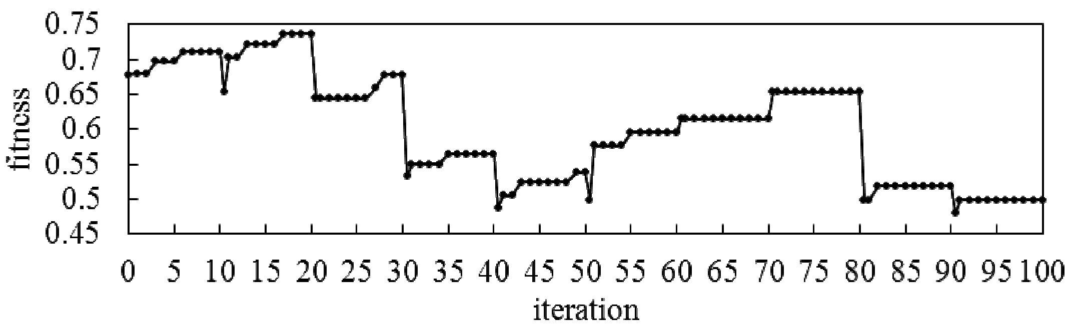
(a)



(b)



(c)



(d)

Figure 2. Plot of fitness versus the iteration number in the dynamic scenario. The number of the mesh clients that change their locations in each 10 iterations is (a) 0, (b) 16, (c) 32, and (d) 48.

3. Cheng, H., and Yang, S. “Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks,” *Engineering Applications of Artificial Intelligence* (23) 2010, pp: 806-819.
4. Clerc, M., and Kennedy, J. “The particle swarm explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation* (6:1) 2002, pp: 58-73.
5. Eberhart, R., and Kennedy, J. “A new optimizer using particles swarm theory,” *Proceedings of 6th International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.
6. Garey, M., and Johnson, D. *Computers and Intractability – A Guide to the Theory of NP-Completeness,* Freeman, San Francisco, 1979.
7. Goken, C., Gezici, S., and Arikan, O. “Optimal stochastic signaling for power-constrained binary communications systems,” *IEEE Transactions on Wireless Communications* (9) 2010, pp: 3650-3661.
8. Hochbaum, D., and Maass, W. “Approximating schemes for covering and packing problems in image processing and VLSI,” *Journal of the ACM* (32:1)1985, pp: 130-136.
9. Hwang, C., Talipov, E., and Cha, H. “Distributed geographic service discovery for mobile sensor networks original research article,” *Computer Networks* (55:1) 2011, pp: 1069-1082.
10. Johnson, D. “The NP-completeness column: An ongoing guide,” *Journal of Algorithms* (3:2) 1982, pp: 182-195.
11. Kennedy, J. “The particle swarm: Social adaptation of knowledge,” *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, 1997, pp. 303-308.
12. Muogilim, O. E., Loo, K.-K., and Comley, R. “Wireless mesh network security: A traffic engineering management approach,” *Journal of Network and Computer Applications* (34) 2011, pp: 478-491.
13. Ramamurthi, V., Reaz, A., Ghosal, D., Dixit, S., and Mukherjee, B., “Channel, capacity, and flow assignment in wireless mesh networks,” *Computer Networks* (55) 2011, pp: 2241-2258.
14. Soo, K., Siu, Y., Chan, W., Yang, L., and Chen, R. “Particle-swarm-optimization-based multiuser detector for CDMA communications,” *IEEE Transactions on Vehicular Technology* (56) 2007, pp: 3006-3013.
15. Wang, J., Xie, B., Cai, K., and Agrawal, D. “Efficient mesh router placement in wireless mesh networks,” *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*, IEEE Press, 2007, pp. 1-9.
16. Xhafa, F., Barolli, A., Sanchez, and C., Barolli, L. “A simulated annealing algorithm for router nodes placement problem in wireless mesh networks,” *Simulation Modelling Practice and Theory* (19:10) 2011, pp: 2276-2284.
17. Xhafa, F., Sanchez, C., and Barolli, L. “Genetic algorithms for efficient placement of

router nodes in wireless mesh networks," *Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*, IEEE Press, 2010, pp. 465-472.

18. Yao, W., Chen, S., Tan, S., and Hanzo, L. "Minimum bit error rate multiuser transmission designs using particle swarm optimisation," *IEEE Transactions on Wireless Communications* (8) 2009, pp: 5012-5017.

用粒子群優化法求解無線網狀網路之行動節點配置問題

林春成[*]　林奕伶
國立交通大學工業工程與管理學系
E-mail: cclin321@nctu.edu.tw, celebratory105@hotmail.com

摘要

本研究係探討無線網狀網路之行動節點最適配置問題。一般而言，無線網狀網路是由網狀路由節點與網狀客戶節點兩類節點所組成的，當中各節點之間之網路連線是藉由擁有不同無線網路覆蓋範圍之網狀路由節點的閘道與橋接器之功能所達成的。不同於過去研究著重於靜態網路情境，本研究考慮一個動態的配置情境，當中網狀客戶節點與網狀路由節點均有行動的能力，可在一個二維的地理佈局區域中作移動，因此網路之拓樸圖形也會隨著不同的時間而改變，我們根據這樣的變化來動態地調整網狀路由節點之地理位置之佈局。在這樣的架構下，本研究關心的問題是如何在此區域中動態決定網狀路由節點之地理配置使得網路連接性與伺服節點覆蓋率可同時為最大(此二目標均可被視為是無線網狀網路效能之主要量測值)。在我們的設定中，網路連接性指的是無線網狀網路所隱含之拓樸圖形中最大圖形分量之大小，而伺服節點覆蓋率則是指被網狀路由節點所覆蓋的伺服節點的總個數，當中後者可代表為無線網狀網路的服務品質的一項指標。一般而言，即使針對有較少數目的網狀路由節點與小型區域的靜態情境問題，考慮了上述兩個網路品質目標之最適化問題一般而言是計算上難解的。因此，本研究首先針對此問題建立一個數學模式，接著提供一粒子群優化演算法來解決此問題。粒子群優化演算法是模擬一群粒子藉由粒子自己的經驗與其他粒子的經驗的引導來搜群問題的解答, 這樣的方法允許簡單的數學公式來找出不錯的解答。最後，我們使用詳細的模擬結果來評估我們所提出來的方法的品質。

關鍵詞：路由節點配置、無線網狀網路、粒子群優化

[*] 聯繫作者。