Information Sciences 232 (2013) 294–308

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis

Chun-Cheng Lin *

Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 12 April 2012
Received in revised form 5 November 2012
Accepted 30 December 2012
Available online 21 January 2013

Keywords:
Dynamic router node placement
Wireless mesh network
Particle swarm optimization
Metaheuristic
Convergence analysis
Stability analysis

ABSTRACT

Different from previous works, this paper considers the router node placement of wireless mesh networks (WMNs) in a dynamic network scenario in which both mesh clients and mesh routers have mobility, and mesh clients can switch on or off their network access at different times. We investigate how to determine the dynamic placement of mesh routers in a geographical area to adapt to the network topology changes at different times while maximizing two main network performance measures: network connectivity and client coverage, i.e., 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. As a result, this paper first models a mathematical form for our concerned problem, then proposes a particle swarm optimization (PSO) approach, and, from a theoretical aspect, provides the convergence and stability analysis of the PSO with constriction coefficient, which is much simpler than the previous analysis. Experimental results show the quality of the proposed approach through sensitivity analysis, as well as the adaptability to the topology changes at different times.

1. Introduction

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. Therefore, WMNs have broadly been identified as a promising cost-effective solution for ubiquitous wireless broadband connectivity in a metropolis, and have naturally emerged as a fast growing research topic in communications and networking. A detailed survey on WMNs can be found in [1]. Since the practical problems in all layers of WMNs are very complicated in general, previous studies often investigated simplified versions of those problems after some degree of abstraction and assumption. In order to meet practical requirements, a variety of research challenges for WMNs remain, e.g., the routing problems with different concerns [2], the assignment problems for multi-channel WMNs [10,24,25], the security problems in WMNs [20], among others.

The problem of router node placement (RNP) in WMNs has been studied, e.g., see [4,29,30]. Such a combinatorial optimization problem considers a 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 the heterogeneity of WMNs in practice, each mesh router is assumed to have a different size of radio coverage, and two mesh

* Tel.: +886 3 5731758; fax: +886 3 5729101.
E-mail address: cclin321@nctu.edu.tw

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.12.023

routers can communicate with each other via their radio coverage. 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 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 the 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 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 [3,12,18,28], 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, there are more and more studies that are taking into account the mobility in wireless networks, e.g., dynamic multicast problems in mobile ad hoc networks [7], distributed geographic service discovery in mobile sensor network [14], channel allocation in mobile wireless networks [26], among others. In this paper, we consider the mobility of WMN in a dynamic network scenario in which both mesh clients and mesh routers[1] have mobility, and mesh clients can switch on or off their network access at different times, while the geographical locations of mesh routers can be adjusted to adapt to the network topology changes. In such a dynamic scenario, we investigate the problem of dynamic router node placement in WMNs (WMN-dynRNP), which is to find a dynamic placement of mesh routers in a rectangular geographical area to adapt to the network topology changes at different times while maximizing both network connectivity and client coverage. In our formulation, network connectivity is measured by the size of the greatest component of the topology underlying the WMNs, while client coverage 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 most of the previous works focused on solving the static network scenario of our concerned problems by using metaheuristics, e.g., simulated annealing [29] and genetic algorithm [4,30].

Particle swarm optimization (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 it has been proven to be efficient and effective in a variety of applications, e.g., the design of water supply systems [19], manufacturing process [21], structural design optimization [23], among others. Hence, it is of interest and importance to design a PSO approach to the WMN-dynRNP problem. This paper provides an efficient and effective PSO approach for the WMN-dynRNP problem, discusses the effect of a variety of parameters on the influences of the WMNs, and evaluates the quality of the proposed approach through detailed simulations. In addition, from a theoretical aspect, we provide the convergence and stability analysis of the PSO with constriction coefficient, which is much simpler than the previous analysis [9]. Note that the constriction coefficient has been used to improve the performance of PSO [5,9].

The main contributions of this paper are summarized as follows:

- Previous works [4,30] only investigated the static network scenario of our concerned problem, and never formulated the static network problem in a mathematical form. In this paper, a mathematical formulation of the WMN-dynRNP problem is derived, so that interested readers are able to investigate the problem precisely in the future.
- This paper provides an efficient PSO approach for the WMN-dynRNP problem and evaluates the effect of a variety of parameters on the influences of WMNs experimentally.
- A version of PSO, called the PSO with constriction coefficient, is applied in this paper, but the previous algorithm analysis for this version is more complicated [9]. One of the most important contributions of this paper is to provide the convergence and stability analysis for this version of PSO, which is much simpler than the previous analysis.
- Previous works [30,4] focused on placing mesh routers in a grid area, which restricts the positions of mesh nodes. In contrast, our PSO approach allows mesh routers with high flexibility to be placed at floating-point positions in a continuous deployment area. In addition, previous approaches optimized network connectivity and client coverage in two stages, while ours optimizes them at the same time, because client coverage may also be crucial in some practical services. Some experimental results will be provided to show the advantage of simultaneous optimization.
- We also develop a visualization user interface for the WMN-dynRNP problem. The interface allows users to observe the evolution of solutions, and further get insight into how to adjust parameters of PSO to improve the solutions.

Although a large number of studies have been made on a wide variety of problems for wireless networks, to the best of our understanding, our concerned dynamic problem rooted from [30,4] is specific, because most of the other previous works were designed for network packets, or did not consider the mobility of nodes. For example, a recent work in

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

[22] surveyed routing and mobility approaches in IPv6 over LoWPAN mesh networks. Those approaches are different from our work because they were designed for the routing and mobility problems for network packets, but our work is concerned with the allocation of node resources, which is a kind of combinatorial optimization problem. In addition, another of our main contributions is to provide a PSO approach with a simpler convergence analysis, which makes our work specific. Consider another recent work in [31] which built a convex optimization formulation for the distributed scheduling for video streaming over multi-channel multi-radio multi-hop wireless networks that jointly considers channel assignment, rate allocation, and routing. This work did not consider the mobility of nodes, but our work does. Therefore, this work is also different from ours.

The rest of this paper is organized as follows. In Section 2, we describe our concerned problem in a formal form. We propose a PSO approach to the concerned problem in Section 3. Section 4 gives the convergence and stability analysis of the PSO with constriction coefficient. Section 5 shows the experimental results of our approach. A conclusion is given in Section 6.

2. 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 routers. To respond to the heterogeneity of WMNs in practice, we assume that mesh routers have different radio coverage ranges.

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

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

This paper considers a dynamic WMN scenario in which both mesh routers and mesh clients have mobility, and mesh clients can switch on or off their network access at different times so that the network topology changes in time, while we adjust the locations of mobile mesh routers to adapt to the topology changes. We assume that the locations of mesh routers are determined periodically, i.e., there is a fixed time period between any two key time points, and all the decisions are made only at key time points. At each different key time point, mesh clients with mobility may have different locations, and the placements of mesh routers are determined according to the modified deployment of mesh clients in the geographical area.

Based on the above, such a dynamic framework is modeled as follows. At the tth 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 deployment of mesh clients at each tth 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 (i.e., the radio coverage of mesh router r_i at the tth key time point). For a determined placement of mesh routers at the tth key time point, we can establish a topology graph $G_t = (U_t, E_t)$ in which

- $U_t = R \cup (C \backslash S_t)$ where S_t is the set of mesh clients that switch off network access or do not access a network at the tth key time point;
- for any two mesh routers $r_i, r_j \in R$, edge $(r_i, r_j) \in E_t$ if $\Upsilon_i^t \cap \Upsilon_j^t \neq \emptyset$;
- for any mesh client $c_i \in C \backslash S_t$ and any mesh router $r_j \in R$, edge $(c_i, r_j) \in E_t$ if $D_t(c_i) \in \Upsilon_j^t$.

Note that, for similarity, we assume that two mesh routers can communicate only if their radio coverage is overlapped. The assumption is reasonable because two mesh routers with overlapping radio coverage imply that they are so close geographically that they can communicate with each other.

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; 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., $G_t = G_t^1 \cup G_t^2 \cup \cdots G_t^h$, 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:

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

The client coverage can be expressed as follows:

$$\psi(G_t) = |\{i; d_t(c_i) > 0 \quad \text{for } i \in \{1, \ldots, m\}\}|, \tag{2}$$

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

In order to explain the above notation, we demonstrate two examples for a WMN with five mesh routers $c_1, \ldots, c_5$ and 14 mesh clients $c_1, \ldots, c_{14}$ in Fig. 1, in which each mesh router has a different-size radio coverage range. If two mesh routers

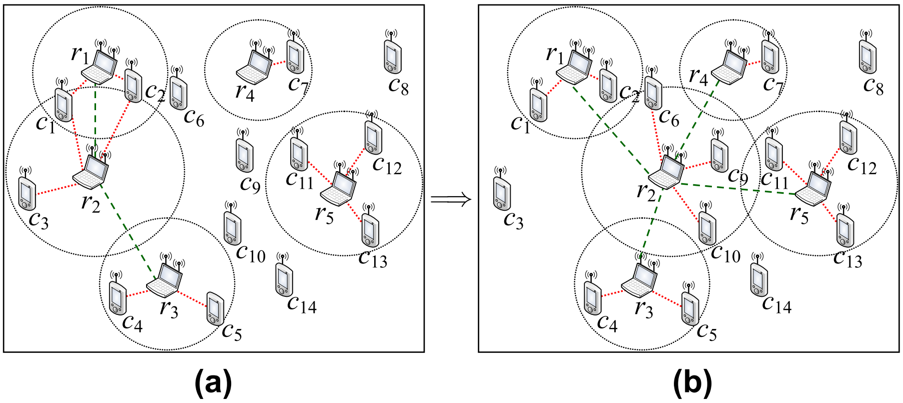


Fig. 1. Two placements for the same WNN with five mesh routers and 14 mesh clients.

have a radio coverage overlap, they are connected by a green dashed link (e.g., see the green dashed link between r_1 and r_2 in Fig. 1a). If a mesh client is located within the radio coverage range of a mesh router, they are connected by a red dotted link (e.g., c_1 is connected to r_1 and r_2 by two red dotted links in Fig. 1a because it is located within both of the radio coverage ranges of r_1 and r_2 simultaneously). Hence, the topology graph in Fig. 1a has three subgraph components among which the size of the greatest component is 8 (i.e., $\phi = 8$), and 10 mesh clients c_1–c_7, c_{11}–c_{13} are covered (i.e., $\psi = 10$). If we move mesh router r_2 to the center location of the deployment area as shown in Fig. 1b, then all the components are merged into a single one with size 16 (i.e., $\phi = 16$), and 11 mesh clients are covered (i.e., $\psi = 11$). That is, both of the two network performance measures are improved by moving only the mesh router r_2. As a result, it is of interest to investigate how to place the mesh routers so that both of the two network performance measures are optimal.

Furthermore, the problem becomes more complicated if mesh clients have mobility and can switch on or off their network access as time goes by. For example, consider a scenario in Fig. 2a transformed from Fig. 1b. In this scenario, mesh routers c_6 and c_{11} switch off their network access (i.e., they do not show in Fig. 2a), and some of the mesh clients move to new locations. Although all of the mesh routers are connected in Fig. 2a (in which $\phi = 9$), only 4 mesh clients are covered (i.e., $\psi = 4$). Hence, the new scenario performs worse than the original configuration in Fig. 1b. To increase the total client coverage, we move mesh router r_2 to the lower right location of the deployment area as shown in Fig. 2b, in which the size of the greatest component is 8 (i.e., $\phi = 8$), and seven mesh clients are covered (i.e., $\psi = 7$). Although the ψ value is increased by 3, the ϕ value is decreased by 1. However, this does not imply that this movement is better, because the two network measures have different weights in the objective function.

With the above notation, our concerned problem is to find a dynamic placement $D_t(R)$ of mesh routers in a WMN to adapt to the network topology change at any tth key time point so that both the size of greatest comments $\phi(G_t)$ and the client coverage $\psi(G_t)$ are as large as possible. Hence, our concerned problem can be stated as follows:

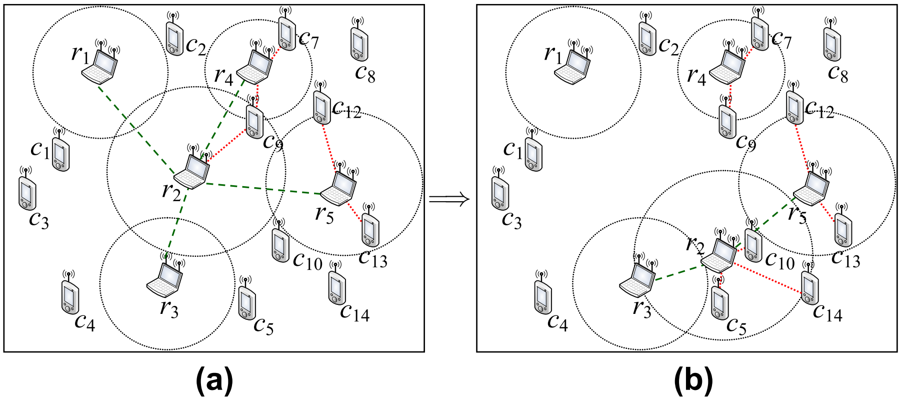


Fig. 2. (a) A scenario transformed from Fig. 1b, in which mesh routers c_6 and c_{11} switch off their network access, and some of the mesh clients move to new locations. (b) Mesh router r_2 moves to the lower right area to increase the client coverage.

Dynamic Router Node Placement in Wireless Mesh Network (WMN-dynRNP): Give a WMN with n mesh routers and m mesh clients, for $t = 1, 2, \ldots$, the topology graph $G_t = (U_t, E_t)$ underlies the WMN at the tth key time point (as described above) deployed in a two-dimensional $W \times H$ area in which mesh clients have mobility and can switch on or off their network access, while the locations of mesh routers need be determined to adapt to the topology changes. The objective of the problem is to find a dynamic placement $D_t(R) = \{D_t(r_1), \ldots, D_t(r_n)\}$ of the n mesh routers so that both the size of the greatest subgraph component $\phi(G_t)$ and the client coverage $\psi(G_t)$ are maximized.

Note that if the WMN-dynRNP problem at a fixed key time point is considered independently, then it can be viewed as a static problem. It is obvious that the static problem contains the following NP-complete problem – *Minimum Geometric Disk Cover* [15,13], so the problem is an NP-hard problem, i.e., it cannot be solved by an efficient deterministic polynomial-time algorithm. Hence, we devise a PSO metaheuristic approach to the WMN-dynRNP problem, which provides an efficient promising solution, as compared to the NP-hard problem.

Minimum Geometric Disk Cover: Given a set $P \subseteq \mathbb{Z} \times \mathbb{Z}$ of points in the plane and a rational number $r > 0$, the problem is to find a set of centers $C \subseteq \mathbb{Q} \times \mathbb{Q}$ in the Euclidean plane, such that every point in P will be covered by a disk with radius r and center in one of the points in C.

3. A PSO approach to the WMN-dynRNP problem

The PSO is an algorithm used for solving optimization problems based on the social intelligence of a swarm of particles, proposed by [11,17]. The basic idea of PSO is to simulate a swarm of particles' social behavior of searching for 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-dynRNP 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 for the WMN-dynRNP problem.

3.1. Solution representation

The solution of our concerned WMN-dynRNP problem at each key time point is a placement of n mesh routers in a two-dimensional $W \times H$ area, whose lower-left corner is placed 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 in which $(x_{k(2i-1)}^t, x_{k(2i)}^t)$ denotes the (x, y)-coordinate of mesh router r_i for $i = 1, 2, \ldots, n$; 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 in which the superscript t denotes the case at the tth key time point.

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

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

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

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

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

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

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

where $V_{\max}$ is a given constant that is no more than $\max\{W, H\}$. The constraint for velocities is also reasonable to meet the practical condition of our problem in which mesh routers cannot move too far between two key time points, though they have mobility.

From the view point of the whole swarm, a vector and a fitness of the best solution that has 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.

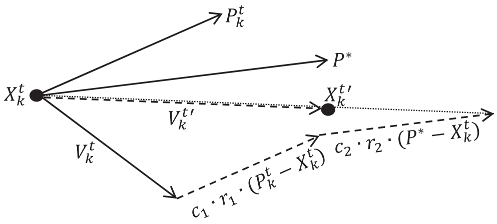

P_k^t
P^*
X_k^t
$X_k^{t\prime}$
$V_k^{t\prime}$
V_k^t
$c_2 \cdot r_2 \cdot (P^* - X_k^t)$
$c_1 \cdot r_1 \cdot (P_k^t - X_k^t)$

Fig. 3. Illustration of updating the location of particle k.

3.2. Fitness function

At tth key time point, if a placement X_k^t of mesh routers for particle k is determined, we can obtain a topology 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 $\psi(G_{t,k})$, which can be calculated by Eqs. (1) and (2), respectively.

The fitness function of $G_{t,k}$ is calculated as follows:

$$f(X_k^t) = \lambda \cdot \frac{\phi(G_{t,k})}{n+m} + (1-\lambda) \cdot \frac{\psi(G_{t,k})}{m}, \tag{6}$$

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 two objectives of the WMN-dynRNP problem. The first setting is bi-level optimization, which considers $\phi(G_{t,k})$ as the main objective, and $\psi(G_{t,k})$ 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 Eq. (6) applies the second setting. Note that the previous work [29] for the static network scenario of the WMN-dynRNP problem applies the first setting.

3.3. Updating position

Each iteration of the main loop of PSO updates the velocity vector V_k by the following formula (see also Fig. 3):

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

where $V_k^{t\prime}$ is the updated value of velocity vector V_k of particle k; $c = c_1 + c_2 > 4$; $\omega = 2/|2 - c - \sqrt{c^2 - 4c}|$; r_1 and r_2 are two random numbers between 0 and 1. The position of each particle k by the following formula:

$$X_k^{t\prime} = X_k^t + V_k^{t\prime}, \tag{8}$$

where $X_k^{t\prime}$ is the updated value of position vector X_k of particle k, respectively. Since the computation of the above two equations requires only basic arithmetic operations, they can be computed in $O(1)$ time. Note that a PSO approach with updating formula of Eq. (7) is called the *PSO with constriction coefficient* [9].

As illustrated in Fig. 3, the basic concept of the above formulas is that the velocity is updated from V_k^t to $V_k^{t\prime}$ according to the guidance of both the location P_k^t of the best solution found so far by particle k and the location P^* of the best solution found so far by all particles, and parameters c_1 and c_2 control their relative influence, respectively, while r_1 and r_2 are random numbers used to increase the probability of escaping local optimum. Since the value of the composite force vector may be exploded, the factor ω (which is called *constriction coefficient*) is used for reducing some undesirable explosive feedback effects [8]. In addition, it does not require any additional parameters, because the value is based on c_1 and c_2 (which are given initially).

3.4. Algorithm

The PSO algorithm works with a swarm of particles, each of which represents a candidate solution of the WMN-dynRNP problem at the tth 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 Eq. (7).

Algorithm 1. PSO (key time point t)

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

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 the initialization of all of the vector variables $X_k^t,\ P_k^t,\ P^*,\ V_k^t$ and all of 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)–(5), in which $\mathcal{U}(a,b)$ is a uniform distribution over [a, b]. Once X_k^t is determined, its fitness $f(X_k^t)$ is calculated by Eq. (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 $f(P^*)$ that have been found so far.

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 Eqs. (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 20. In Lines 21–26, 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.

4. Algorithm analysis

In this section, we derive the convergence and stability analysis for the version of the PSO with constriction coefficient, by analogy of the method of analyzing convergence and stability of the PSO in [23]. Our analysis has the advantage of providing a much easier method to realize the convergence and stability of the PSO with constriction coefficient, in comparison to the original analysis [9]. For notational simplicity, the superscript t of X_k^t, V_k^t, and P_k^t is omitted throughout the analysis in this section.

4.1. Convergence analysis

To conduct the convergence analysis of the PSO with constriction coefficient, we consider the time step value $\Delta\tau$ to describe the dynamics of the PSO, and rewrite the velocity and position update formulas in Eqs. (7) and (8), respectively, as follows:

$$V_k' = \omega\left(V_k + c_1 \cdot r_1 \cdot \frac{(P_k - X_k)}{\Delta\tau} + c_2 \cdot r_2 \cdot \frac{(P^* - X_k)}{\Delta\tau}\right), \tag{9}$$

$$X_k' = X_k + V_k'\Delta\tau. \tag{10}$$

By replacing Eq. (9) into Eq. (10), we obtain the following:

$$X_k' = X_k + \omega V_k \Delta\tau + \omega(c_1r_1 + c_2r_2)\left(\frac{c_1r_1P_k + c_2r_2P^*}{c_1r_1 + c_2r_2} - X_k\right)\Delta\tau, \tag{11}$$

which can be viewed as a general gradient line-search form as follows

$$X_k' = \widehat{X}_k + \alpha\overline{P}_k,$$

where $\widehat{X}_k = X_k + \omega V_k \Delta\tau$; $\alpha = \omega(c_1r_1 + c_2r_2)$ can be viewed as a stochastic step size, and $\overline{P}_k = (c_1r_1P_k + c_2r_2P^*)/(c_1r_1 + c_2r_2) - X_k$ can be viewed as a stochastic search direction. Since r_1, $r_2 \in [0, 1]$, we have $\alpha \in [0, \omega(c_1 + c_2)]$ and $\overline{P}_k \in [-X_k, (c_1P_k + c_2P^*)/(c_1 + c_2) - X_k]$.

By rearranging the terms in Eq. (11), we obtain

$$X_k' = X_k(1 - \omega c_1r_1 - \omega c_2r_2) + \omega V_k\Delta\tau + \omega c_1r_1P_k + \omega c_2r_2P^*. \tag{12}$$

Also, by rearranging the terms in Eq. (9), we obtain

$$V_k' = -X_k\frac{\omega(c_1r_1 + c_2r_2)}{\Delta\tau} + \omega V_k + \omega c_1r_1\frac{P_k}{\Delta\tau} + \omega c_2r_2\frac{P^*}{\Delta\tau}. \tag{13}$$

We combine the above two equations to have the following matrix form:

$$\begin{bmatrix} X_k' \\ V_k' \end{bmatrix} = \begin{bmatrix} 1 - \omega c_1r_1 - \omega c_2r_2 & \omega\Delta\tau \\ -\frac{\omega(c_1r_1+c_2r_2)}{\Delta\tau} & \omega \end{bmatrix}\begin{bmatrix} X_k \\ V_k \end{bmatrix} + \begin{bmatrix} \omega c_1r_1 & \omega c_2r_2 \\ \frac{\omega c_1r_1}{\Delta\tau} & \frac{\omega c_2r_2}{\Delta\tau} \end{bmatrix}\begin{bmatrix} P_k \\ P^* \end{bmatrix}, \tag{14}$$

which can be thought of as a discrete dynamic system representation for the PSO in which $[X, V]^T$ is the state subject to an external input $[P, P^*]^T$, and the two terms on the right side of the equation correspond to the dynamic and input matrices, respectively. Different from Eqs. (8) and (7), the above equation gives another explanation for the new location X_k' and velocity V_k' of particle k. It means that the new location X_k' (resp., velocity V_k') is a linear combination of four terms: X_k, V_k, P_k, P^*, in which a part of the new values of X_k' and V_k' are derived from the original X_k and V_k in the system, while P_k and P^* are additional inputs of the system.

Supposing that no external excitation exists in the dynamic system, $[P,P^*]^T$ is constant, i.e., other particles cannot find better positions. Then, a convergent behavior could be maintained. If convergent, as $k \to \infty$, $[X_K', V_k']^T = [X_k, V_k]^T$. That is, the dynamic system becomes:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\omega c_1r_1 - \omega c_2r_2 & \omega\Delta\tau \\ -\frac{\omega(c_1r_1+c_2r_2)}{\Delta\tau} & \omega - 1 \end{bmatrix}\begin{bmatrix} X_k \\ V_k \end{bmatrix} + \begin{bmatrix} \omega c_1r_1 & \omega c_2r_2 \\ \frac{\omega c_1r_1}{\Delta\tau} & \frac{\omega c_2r_2}{\Delta\tau} \end{bmatrix}\begin{bmatrix} P_k \\ P^* \end{bmatrix},$$

which holds only when $V_k = 0$ and $X_k = P_k = P^*$, where the convergent point is an equilibrium point if there is no external excitation, but better points are found by optimization process with external excitation.

4.2. Stability analysis

We further get insight into the dynamic system in Eq. (14). First, we calculate the characteristic equation of the dynamic system as follows:

$$\lambda^2 - (\omega - \omega c_1r_1 - \omega c_2r_2 + 1)\lambda + \omega = 0.$$

The eigenvalues are obtained as follows:

$$\lambda_{1,2} = \frac{1 + \omega - \omega c_1r_1 - \omega c_2r_2 \pm \sqrt{(1 + \omega - \omega c_1r_1 - \omega c_2r_2)^2 - 4\omega}}{2},$$

where $\lambda_1 \geqslant \lambda_2$. Note that the necessary and sufficient condition for stability of a discrete-dynamic system is that all the derived eigenvalues lies inside a unit circle around the origin on the complex plane, i.e., $|\lambda_{1,2}| < 1$ in this case.

When $\lambda_1 < 1$, we derive

$$\begin{aligned}&\omega(c_1r_1 + c_2r_2) > 0,\\ &1 - \omega + \omega(c_1r_1 + c_2r_2) > 0,\end{aligned}$$

where the first condition implies $c_1r_1 + c_2r_2 > 0$ since $\omega > 0$, while the second condition implies $\omega < 1$ since $r_1, r_2 \in [0, 1]$.

When $\lambda_2 > -1$, we derive

$$\begin{aligned}&\omega\Big(\frac{c_1r_1 + c_2r_2}{2} - 1\Big) < 1,\\ &\omega(c_1r_1 + c_2r_2) < 3 + \omega.\end{aligned}$$

in which the first condition implies

$$\max\left\{0, 1 - \frac{1}{\omega}\right\} < c_1r_1 + c_2r_2 < \min\left\{2 + \frac{2}{\omega}, 1 + \frac{3}{\omega}\right\}.$$

Since $r_1, r_2 \in [0, 1]$, we have the following conditions for stability:

$$\max\left\{0, 1 - \frac{1}{\omega}\right\} < c_1 + c_2 < \min\left\{2 + \frac{2}{\omega}, 1 + \frac{3}{\omega}\right\}, \quad 0 < \omega < 1.$$

Hence, the system is stable if the above conditions hold.

5. 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 of a variety of cases in static and dynamic scenarios.

5.1. Data and environment

Similar to [29], 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 range with radius $\sim \mathcal{U}(3, 6)$.

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 range with radius $\sim \mathcal{U}(4\sqrt{2} - 2, 8\sqrt{2} - 2)$.

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 range with radius $\sim \mathcal{U}(7, 14)$.

Note that the deployment area is a grid in [29], meaning that each coordinate is an integer, while the coordinates in our setting are floating-point numbers, which allow for more general applications. For each case, we generate 10 instances, in which the mesh clients are distributed in the deployment area according to a uniform distribution.

The simulation of our proposed PSO algorithm is performed using the C++ programming language, and, unless otherwise described in the rest of this paper, our PSO algorithm applies the parameter setting given in Table 1. The simulation considers the WSNs deployed in the above three cases, i.e., on 32×32, 64×64, and 128×128 areas.

We explain the parameter setting as follows. 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 Fig. 4a), 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 Fig. 4c). That is, the λ value determines the tradeoff between our concerned two objectives (ϕ and ψ) to some degree. Hence, we take a λ value that has a high probability to satisfy the two concerned objectives simultaneously. After considerable trial and error, we found that the results using $\eta = 0.3$ perform better (e.g., see Fig. 4b). In addition, $V_{\max}$ is set to 0.1, because mesh routers cannot move fast between any two time steps.

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 is about 0.2647, 0.6675, and 2.0660 s, respectively. It implies that our PSO algorithm has the ability to efficiently cope with the WMN-dynRNP problem. As shown in Fig. 4, we developed a visualization user interface. With the guide of the visualization, users can adjust the parameters of PSO

Table 1
Parameters used in simulation.

Parameter	Value
The number of iterations between two key time points	10
The number of mesh clients that change their locations	0, 16, 32, 48
The probability that each mesh client can switch its network access (%)	1
The maximum distance to which mesh clients can move	10
The parameter λ that controls the balance between two objectives	0.3
The cognitive learning rate, c_1	3
The social learning rate, c_2	2
The limit $V_{\max}$ of velocities	0.1
The total number of particles, η	100
The maximum iteration, τ	10

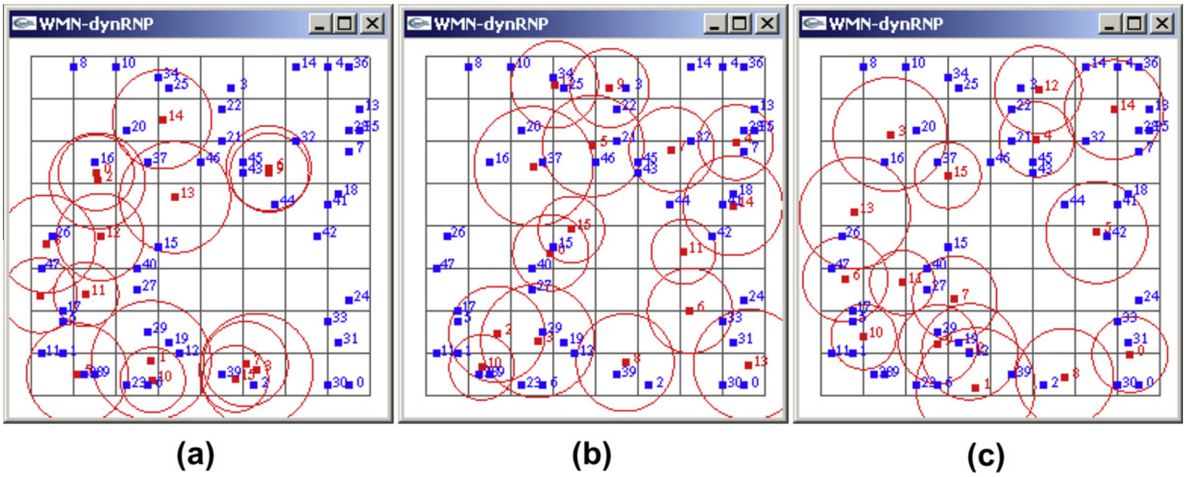


Fig. 4. Visualization of the WMNs under three different λ values, in which each blue point is a mesh client, while each red point is a mesh router that acts as the center of a radio coverage circle. (a) λ = 1.0: ϕ = 42, ψ = 26. (b) λ = 0.3: ϕ = 54, ψ = 38. (c) λ = 0.0: ϕ = 32, ψ = 36. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

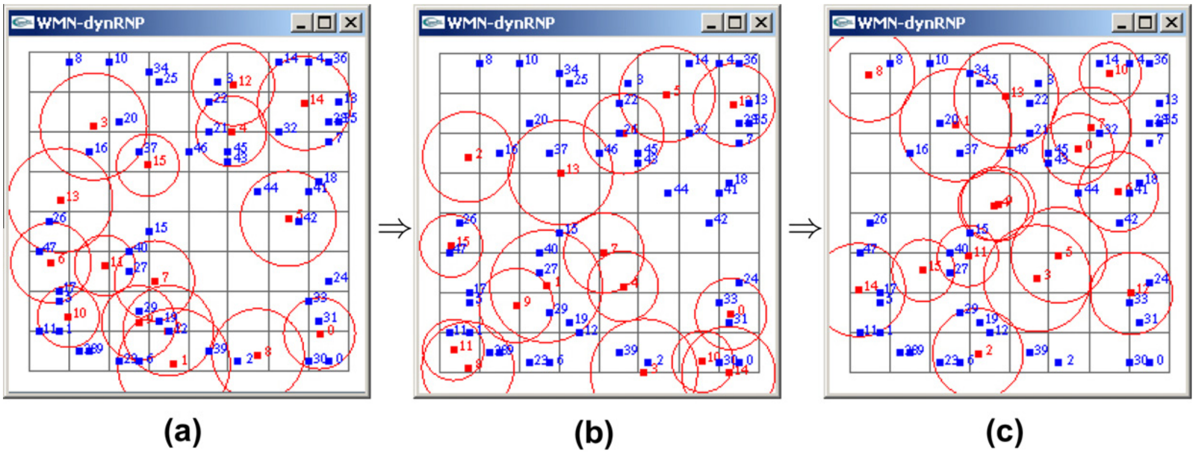


Fig. 5. Visualization of the dynamic process of a WMN instance in (a) iterations 0–2; (b) iterations 3–5; (c) iterations 6–10.

accordingly to improve the solutions. For example, we show the visualization of the dynamic topology changes of a WMN instance in Fig. 5, from which we observe that all of the mesh routers are merged to form a single subgraph component and cover more mesh clients from the left subfigure to the right subfigure.

5.2. Experimental results

In order to observe the convergence of the best fitness values in our PSO method, we plot the best fitness values vs. the number of iterations of the main loop of the PSO algorithm for a case-1 instance in Fig. 6a, and the plots of their corresponding ϕ and ψ values vs. the iteration numbers in Fig. 6a and b, respectively. From Fig. 6a, we can observe that the best fitness value converges at iteration 11. From Figs. 6b and c, both of the two objective values can influence the best fitness value before convergence.

Fig. 7 shows the bar chart of the best fitness values of 100 runs of our PSO algorithm for the same case-1 problem instance. We can see that the 100 runs obtain the results with different best fitness values, but all of them fall in the range [0.6, 0.8]. In addition, each bar in Fig. 7 uses blue and yellow colors to represent the ratios of the ϕ and ψ objectives, respectively. Those different ratios imply that different runs are able to generate diversified results, which can be viewed as an advantage of the PSO algorithm.

The statistics of all of the problem instances are given in Table 2, in which three major columns indicate the three problem cases, in each of which four columns store the best fitness, average fitness, worst fitness, and the standard deviation of fitness values in the 100 runs of our PSO algorithm for each instance of the case; rows indicate 10 instances of each problem case. The averages of each column are stored at the bottom row of Table 2, from which we observe that our PSO approach performs best as the problem size increases. But we cannot conclude that our PSO approach is more suitable for larger problems, because many parameters exist to determine the resultant statistics. However, at least we found that small deviation shows the robustness of implementation, so that the algorithm is always able to deliver good solutions.

In order to realize the influence of the number of particles (η) and the maximal number of iterations (τ) on the resultant fitness values, we give a sensitivity analysis for the two parameters in Table 3, in which the best fitness and average fitness values of each case are calculated by the average of running 10 instances 100 times of our PSO algorithm. From Table 3, we observe the following inferences, though some variation exists due to the randomness in the PSO scheme:

- If the number of particles (η) is fixed, more iterations (τ) result in a larger average fitness value, i.e., τ has a positive influence on the average fitness. This inference holds for all situations in Table 3, e.g., see the average fitness values (.7918, .7991, .8035) of different τ values in case 1 when $\eta = 50$. Note that this inference does not hold for the best fitness.
- In contrast to the above, if the maximal number of iterations (τ) is fixed, more particles (η) result in a larger average fitness value, i.e., η has a positive influence on the average fitness. This inference holds for all situations in Table 3, e.g., see the average fitness values (.7918, .8108, .8204) of different η values in case 1 when $\tau = 5$.
- In order to investigate the interwork of η and τ, we observe from Table 3 that in case 2, the average fitness of $\eta = 50$, $\tau = 15$ (.8098) is less than that of $\eta = 100$, $\tau = 5$ (.8166), and the average fitness of $\eta = 100$, $\tau = 15$ (.8241) is less than that of $\eta = 150$, $\tau = 5$ (.8242). Such observation combined with the first inference above leads us to wonder whether the number of iterations τ can be transformed into the number of particles η to obtain similar or better results. If this is true, then the average fitness values in a column can respect its increasing ordering, e.g., see case 2 in Table 3: $.8026 < .8076 < .8098 < .8166 < .8223 < .8241 < .8242 < .8283 < .8314$. However, the relationship does not hold for the other two cases.

Subsequently, in order to demonstrate the ability of our approach to adapt to the topology changes (in which we suppose that the topology graph changes in each 10 iterations), we run 100 iterations of the PSO algorithm on a case-1 instance in a static and three different dynamic scenarios, and their plots of best fitness vs. the iteration number are given in Fig. 8a–d, respectively. The three dynamic scenarios assume three different degrees of dynamics: 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 in Fig. 8b–d, respectively. Note that for comparison, we use the same seed number for the pseudorandom function in our programming implementation in each case in Fig. 8, from which we observe that the run chart of the best fitness values of the first 10 iterations in Figs. 8a–d has the same trajectory.

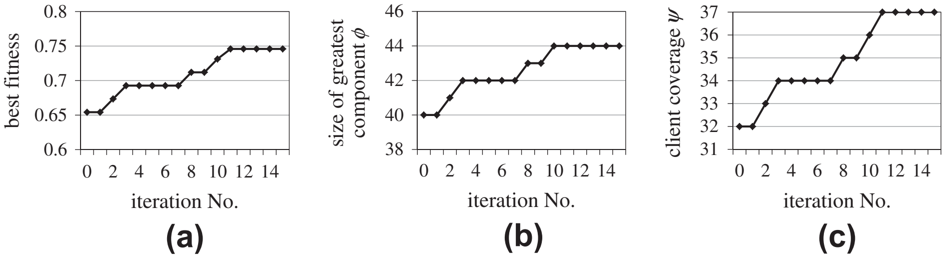


Fig. 6. (a) Plot of best fitness values vs. the iteration number. (b) Plot of the ϕ values vs. the generation number. (c) Plot of the ψ values vs. the generation number.

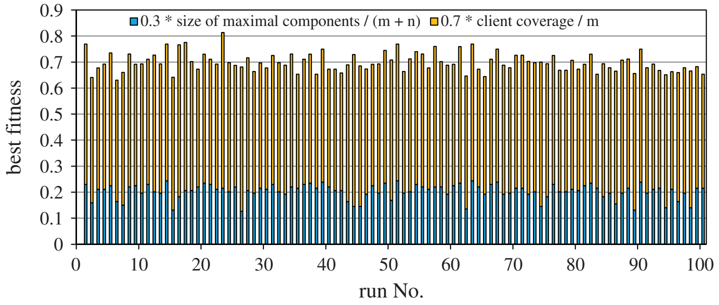


Fig. 7. Bar chart of the best fitness values of 100 runs of our PSO algorithm.

Table 2
Statistics of all of the problem instances.

Instance	Case 1				Case 2				Case 3			
	Best	Average	Worst	SD	Best	Average	Worst	SD	Best	Average	Worst	SD
Inst 1	.7880	.6910	.6214	.0337	.9229	.8523	.8169	.0178	.8844	.8299	.7930	.0184
Inst 2	.9615	.8866	.8458	.0229	.8747	.7938	.7453	.0271	.9036	.8504	.8173	.0163
Inst 3	.9188	.8544	.7880	.0269	.9232	.8695	.8292	.0180	.8844	.8361	.7978	.0172
Inst 4	.9036	.7890	.7302	.0331	.8799	.8217	.7664	.0206	.8796	.8354	.7961	.0151
Inst 5	.8901	.8090	.7510	.0285	.8844	.8056	.7474	.0237	.8678	.8319	.8066	.0139
Inst 6	.8651	.7825	.7214	.0293	.8464	.7822	.7495	.0224	.8940	.8413	.8099	.0178
Inst 7	.9328	.8686	.8130	.0285	.8628	.8186	.7742	.0192	.8785	.8285	.7887	.0170
Inst 8	.9422	.8676	.8073	.0291	.8940	.7999	.7461	.0267	.9181	.8585	.8217	.0161
Inst 9	.9036	.8233	.7688	.0271	.8940	.8313	.7784	.0250	.8892	.8469	.8174	.0151
Inst 10	.8844	.7992	.7411	.0273	.8940	.8320	.7839	.0242	.8701	.8268	.7977	.0152
Average	.8990	.8171	.7588	.0286	.8876	.8207	.7737	.0225	.8870	.8386	.8046	.0162

SD denotes standard deviation.

Table 3
Sensitivity analysis.

Particle number (η)	Maximal iteration (τ)	Case 1		Case 2		Case 3	
		Best fitness	Average fitness	Best fitness	Average fitness	Best fitness	Average fitness
50	5	.8879	.7918	.8791	.8026	.8783	.8258
	10	.8885	.7991	.8752	.8076	.8871	.8289
	15	.8881	.8035	.8803	.8098	.8818	.8299
100	5	.8962	.8108	.8847	.8166	.8871	.8289
	10	.9029	.8170	.8886	.8223	.8845	.8382
	15	.9042	.8235	.8847	.8241	.8839	.8402
150	5	.8980	.8204	.8872	.8242	.8886	.8415
	10	.9024	.8279	.8883	.8283	.8938	.8446
	15	.9133	.8318	.8883	.8314	.8905	.8465

In the static scenario (Fig. 8a), the best fitness values converge and do not have any change after the 13th iteration. When the number of the mesh clients that change their locations is small (Fig. 8b), the best fitness values almost do not change during each time period (i.e., within each 10 iterations), meaning that the best fitness value is still the optimal after a slight topology change. However, as the number of the mesh clients that change their locations increases (Fig. 8c and d), the best fitness value has to be improved to achieve the optimality during each time period, e.g., see the best fitness values from the 40th to 45th iterations in Fig. 8d. It can also be seen that the best fitness values are modified drastically when there are more dynamics. In addition, we observe that the range accommodating all of the best fitness values in slight dynamics (e.g., range [0.6, 0.75] in Fig. 8b) is smaller than that in large dynamics (e.g., range [0.45, 0.75] in Fig. 8d).

We further consider the dynamic scenario in which mesh clients not only can change their locations but also can switch on or off their network access. Let each mesh client c_i have a network access state $s_t(c_i)$ at the tth key time point defined as follows:

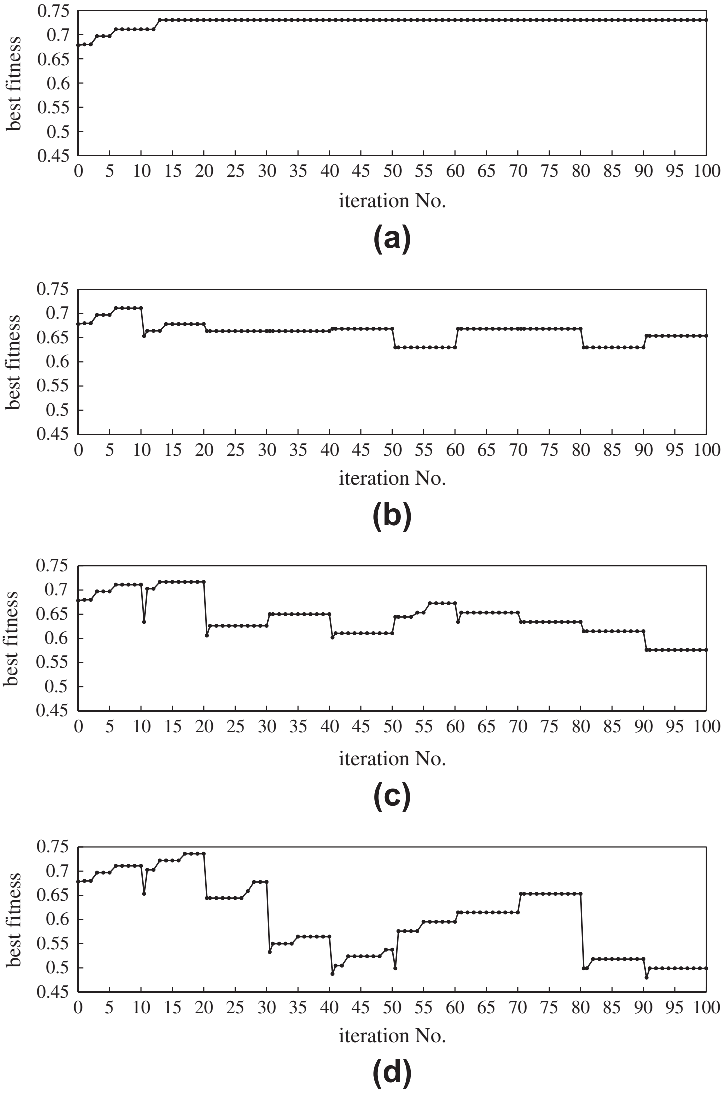


Fig. 8. Plot of best fitness vs. the iteration number in the dynamic scenario in which mesh clients cannot switch its network access state. The number of the mesh clients that change their locations in each 10 iterations is (a) 0, (b) 16, (c) 32, and (d) 48.

$$s_t(c_i) = \begin{cases} 1, & \text{if } c_i \text{ is accessing the network;} \\ 0, & \text{otherwise (i.e., } c_i \text{ switches off the network access).} \end{cases} \tag{15}$$

We establish a dynamic scenario in which each mesh client has a probability threshold to switch its network access state, i.e., at each key time point, if a random number in [0, 1] exceeds the probability threshold, its network access state alternates between 0 and 1 as follows:

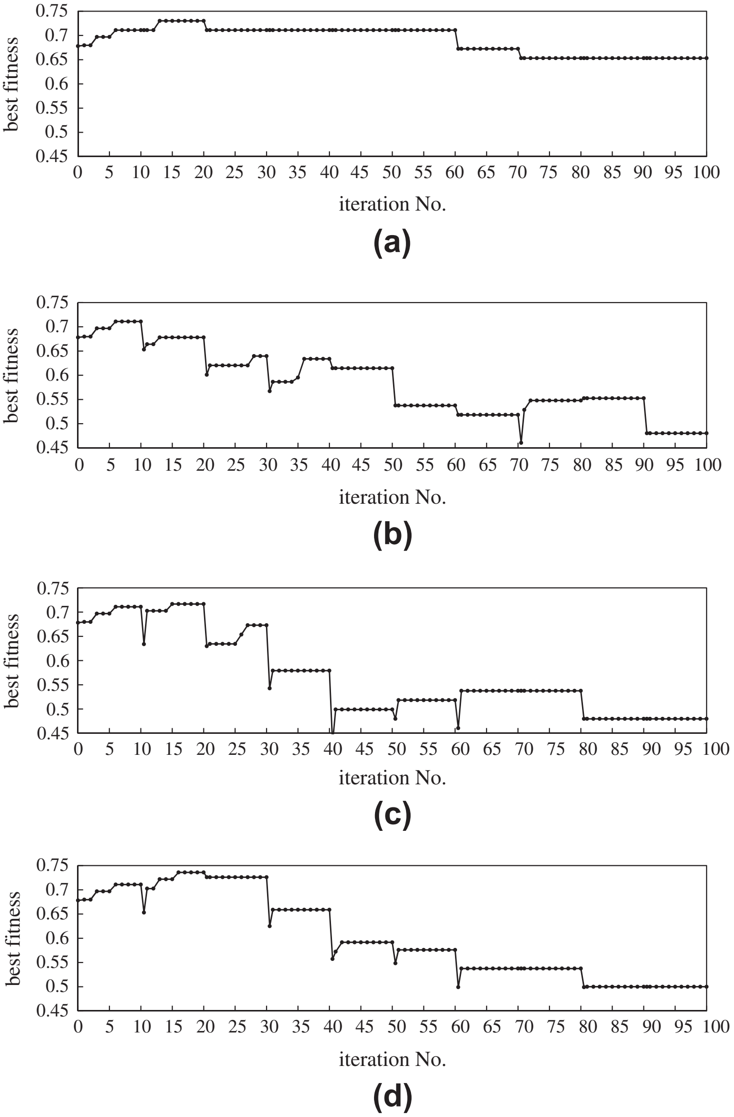
$$s_t(c_i) \leftarrow (s_t(c_i) + 1) \bmod m, \tag{16}$$

Fig. 9 gives the plot of the best fitness values vs. the iteration number in the dynamic scenario where each mesh client can switch its network access state under 1% probability, and 0, 16, 32, and 48 mesh clients change their locations in each 10 iterations, respectively, in Figs. 9a–d. As compared to Fig. 8 (which does not consider the network access states), the best fitness values in Fig. 9 decrease and almost cannot come back to the initial best fitness level as iterations increase. But it is reasonable because more and more mesh clients may switch off their network access as iterations increase, so that a smaller-size topology graph decreases both the network connectivity and the client coverage.

6. Conclusions and future work

A particle swarm optimization approach for optimizing dynamic router node placement in wireless mesh networks (WMN-dynRNP) has been proposed and implemented. The main difference of our model from previous models is the introduction of the dynamic scenario in which at different times, mesh routers and mesh clients can change their locations, and mesh clients can switch on or off their network access. In addition, our PSO approach 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. One of our most important contributions in this paper is to propose a mathematical formulation of the WMN-dynRNP problem, so that interested readers are able to investigate the problem precisely in the future, and, from a theoretical aspect, to derive the convergence analysis and stability analysis of the PSO with constriction coefficient, which is much simpler than the previous work. The performance of our PSO approach is evaluated by discussing the effect of a variety of parameters on the influences of the WMNs. In addition, we also developed a visualization user interface to allow users to observe the evolution of solutions to get more insight into the PSO algorithm.

In the future, we intend to solve the weighted version of the WMN-dynRNP problem and further the dynamic version of the problem in multi-channel WMNs. It is also of importance and interest to consider the optimization of other objectives at the same time. Another line of the future research is to propose some variants of PSO or the PSO mixed with other techniques



(a)

(b)

(c)

(d)

Fig. 9. Plot of best fitness vs. the iteration number in the dynamic scenario in which each mesh client can switch its network access state under 1% probability. The number of the mesh clients that change their locations in each 10 iterations is (a) 0, (b) 16, (c) 32, and (d) 48.

to improve the computational performance, e.g., the scheme of parameters in PSO [27], fuzzy PSO [16], among others. It is also of interest to derive the convergence analysis of the multi-objective version of the proposed PSO, by analogy from [6].

Acknowledgements

The author thanks the anonymous referees for comments that improved the content as well as the presentation of this paper. This work has been supported in part by NSC 101-2219-E-009-025, Taiwan.

References

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.

[2] E. Alotaibi, B. Mukherjee, A survey on routing algorithms for wireless ad-hoc and mesh networks, Computer Networks 56 (2012) 940–965.

[3] E. Amaldi, A. Capone, M. Cesana, I. Filippini, F. Malucelli, Optimization models and methods for planning wireless mesh networks, Computer Networks 52 (2008) 2159–2171.

[4] A. Barolli, F. Xhafa, C. Sánchez, M. Takizawa, A study on the effect of mutation in genetic algorithms for mesh router placement problem in wireless mesh networks, in: Proceedings of 5th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2011), IEEE Press, 2011, pp. 32-39.

[5] L.T. Bui, O. Soliman, H.A. Abbass, A modified strategy for the constriction factor in particle swarm optimization, in: Proceedings of ACAL 2007, Lecture Notes in Artificial Intelligence, vol. 4828, 2007, pp. 333-344.

[6] P. Chakraborty, S. Das, G.G. Roy, A. Abraham, On convergence of the multi-objective particle swarm optimizers, Information Sciences 181 (8) (2011) 1411–1425.

[7] H. Cheng, S. Yang, Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks, Engineering Applications of Artificial Intelligence 23 (2010) 806–819.

[8] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proceedings of Congress on Evolutionary Computation (CEC99), IEEE Press, 1999, pp. 1951-1957.

[9] M. Clerc, J. Kennedy, The particle swarm – explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.

[10] J. Crichigno, M.-Y. Wu, W. Shu, Protocols and architectures for channel assignment in wireless mesh networks, Ad Hoc Networks 6 (2008) 1051–1077.

[11] R. Eberhart, J. Kennedy, A new optimizer using particles swarm theory, in: Proceedings of 6th International Symposium on Micro Machine and Human Science, 1995, pp. 39-43.

[12] M. Garey, D. Johnson, Computers and Intractability – A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[13] D. Hochbaum, W. Maass, Approximating schemes for covering and packing problems in image processing and VLSI, Journal of the ACM 32 (1) (1985) 130–136.

[14] C. Hwang, E. Talipov, H. Cha, Distributed geographic service discovery for mobile sensor networks, Computer Networks 55 (1) (2011) 1069–1082.

[15] D. Johnson, The NP-completeness column: an ongoing guide, Journal of Algorithms 3 (2) (1982) 182–195.

[16] Y.-T. Juang, S.-L. Tung, H.-C. Chiu, Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions, Information Sciences 181 (20) (2011) 4539–4549.

[17] J. Kennedy, The particle swarm: social adaptation of knowledge, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE Press, 1997, pp. 303-308.

[18] A. Lim, B. Rodrigues, F. Wang, Z. Xua, k-center problems with minimum coverage, Theoretical Computer Science 332 (2005) 1–17.

[19] I. Montalvo, J. Izquierdo, R. Pérez, M. Tung, Particle swarm optimization applied to the design of water supply systems, Computers & Mathematics with Applications 56 (2008) 769–776.

[20] O.E. Muogilim, K.-K. Loo, R. Comley, Wireless mesh network security: a traffic engineering management approach, Journal of Network and Computer Applications 34 (2011) 478–491.

[21] T. Navalertporn, N.V. Afzulpurkar, Optimization of tile manufacturing process using particle swarm optimization, Swarm and Evolutionary Computation 1 (2) (2011) 97–109.

[22] L.M.L. Oliveira, A.F. de Sousa, J.J.P.C. Rodrigues, Routing and mobility approaches in IPv6 over LoWPAN mesh network, International Journal of Communication Systems 24 (2011) 1445–1466.

[23] R. Perez, K. Behdinan, Particle swarm approach for structural design optimization, Computers and Structures 85 (13–14) (2007) 1579–1588.

[24] A. Rad, V. Wong, Congestion-aware channel assignment for multi-channel wireless mesh networks, Ad Hoc Networks 6 (2008) 1051–1077.

[25] V. Ramamurthi, A. Reaz, D. Ghosal, S. Dixit, B. Mukherjee, Channel, capacity, and flow assignment in wireless mesh networks, Computer Networks 55 (2011) 2241–2258.

[26] T.-L. Sheu, Y.-J. Wu, A preemptive channel allocation scheme for multimedia traffic in mobile wireless networks, Information Sciences 176 (3) (2006) 217–236.

[27] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Information Sciences 177 (22) (2007) 5033–5049.

[28] J. Wang, B. Xie, K. Cai, D. Agrawal, Efficient mesh router placement in wireless mesh networks, in: Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2007), IEEE Press, 2007, pp. 1-9.

[29] F. Xhafa, A. Barolli, C. Sánchez, L. Barolli, A simulated annealing algorithm for router nodes placement problem in wireless mesh networks, Simulation Modelling Practice and Theory 19 (10) (2011) 2276–2284.

[30] F. Xhafa, C. Sánchez, L. Barolli, Genetic algorithms for efficient placement of router nodes in wireless mesh networks, in: Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), IEEE Press, 2010. pp. 465-472.

[31] L. Zhou, X. Wang, W. Tu, G.-M. Muntean, B. Geller, Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks, IEEE Journal on Selected Areas in Communications 28 (3) (2010) 409–419.