# Adaptive Router Node Placement with Gateway Positions and QoS Constraints in Dynamic Wireless Mesh Networks

Chun-Cheng Lin*, Teng-Huei Chen, and Hui-Hsin Chin

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

**Abstract**

Conventionally, router node placement is concerned with placing only routers to serve clients; and gateway placement is concerned with placing only gateways to achieve some requirements for routers. More generally, this work considers the placement with routers and gateways simultaneously, while clients can move based on their own willingness. That is, this work investigates the adaptive placement problem of a dynamic wireless mesh network (dynWMN) consisting of mesh clients, mesh routers, and Internet gateways. Given fixed positions of Internet gateways, this problem is to adjust positions of mesh routers dynamically to make each mesh client connected with some gateway via multi-hop communication at different times, when each mesh client may switch on or off network access, so that both network connectivity and client coverage are maximized, subject to the Quality of Service (QoS) constraints of delay hops, relay load, and gateway capacity. To

☆A preliminary version of this work was presented at the 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), August 19-20, 2015, Taipei, Taiwan.

* Corresponding author. Tel.: +886-3-5731758; fax: +886-3-5729101.

*E-mail address:* cclin321@nctu.edu.tw (Chun-Cheng Lin)

avoid almost-overlapping routers and few-clients-covered routers in router node placement, this work further proposes a novel particle swarm optimization approach with three local search operators. In simulation of dynWMNs, dynamics of mesh clients can be characterized by a Markov chain, and hence, their stable states can be derived theoretically and are used as criteria of evaluating performance, by which the proposed approach shows promising performance and adaptability to topology changes at different times.

*Keywords:* Wireless mesh network, gateway, QoS, particle swarm optimization

---

**1. Introduction**

Wireless mesh network (WMN) [1], [2], [3] provides a low-cost and high-efficiency networking service. It consists of mesh clients, mesh routers, and Internet gateways, in which mesh clients communicate with each other via the backbone network consisting of mesh routers and Internet gateways, and they are connected to the Internet via multi-hop of mesh routers to some Internet gateways. Previous works on network placements are mainly classified into two categories: router node placement (RNP) and gateway placement (GP).

The RNP problem is further designed for static WMNs [4], [5], [6], [7], [8], and dynamic WMNs [9], [10], [11]. In dynamic WMNs, both mesh routers and mesh clients have mobility, and mesh clients can switch on or off their network access arbitrarily at different times, so that the RNP for dynamic WMNs become more difficult. In general, the RNP aims to find a placement of mesh routers to serve mesh clients, so that both network coverage and client connectivity are maximized at the same time. Most previous works focused on solving various RNP problems by various metaheuristic algorithms, e.g., genetic algorithm [4], [5], simulated annealing [6], [7], and particle swarm optimization (PSO) [9], [10]. On the other hand, the GP problem [12], [13] aims to deploy a number of gateways so that the total cost and the total number of hops are minimized. However, both the two objectives are restricted to quality of service (QoS), and hence, previous works were based on some known network topology models to design algorithms for finding candidate solutions of gateway placement.

Based on the above, most previous works on RNP were concerned with deployment of only routers and clients, so that the resultant local area networks without Internet gateways cannot access to the Internet. On the other hand, most previous works on GP were concerned with deployment of only Internet gateways and routers, so that the resultant placement without mesh clients cannot respond to diversified behaviors of mesh clients. Hence, this work integrates the above two placements to investigate the adaptive RNP problem for dynamic WMNs consisting of Internet gateways, mesh routers, and mesh clients. Since Internet gateways are introduced to this RNP problem, the network topology graph corresponding to the RNP should be created under QoS constraints, which was never considered in previous works on RNP problems.

This work proposes a PSO approach for the concerned RNP problem in dynamic WMNs under QoS constraints, because the decision variables in this problem are continuous and can be handled easily by PSO; and our previous works in [9] and [10] also showed high performance in solving RNP for dynamic WMNs. In visualizing the experimental results of using PSO to solving RNP for dynamic WMNs, we discovered two bad-quality placement issues in those results: first, positions of some mesh routers are too close so that their client coverages overlap too much; second, some mesh router covers too few mesh clients. Hence, this work proposes three local search operators in the proposed PSO approach to avoid the above two issues. To cope with QoS constraints, a network topology graph corresponding to a candidate solution is established under these QoS constraints, i.e., no constraint is violated, but the performance will be reflected in the objective function of the problem. By simulation, the proposed PSO approach is shown to adapt to dynamic changes of mesh clients at different times. Additionally, we find that in the long run, the objective value tends to a fixed level, which can be captured by a Markov chain process. Hence, the stable state of a Markov chain process for each simulation result is derived as the criterion of evaluating the experimental results, which was never concerned in previous related works.

Main contributions of this work are given as follows:

- This work is the first to investigate the RNP for dynamic WMNs consisting of Internet gateways, mesh routers, and mesh clients.
- To avoid almost-overlapping routers and few-clients-covered routers, a PSO with three local search operators is proposed.
- This work models the concerned problem with three QoS constraints.
- The dynamic behavior of mesh clients in simulation can be captured by a Markov chain. Hence, the static state of the Markov chain is derived theoretically and is used as the criterion of evaluating simulation results.

Note that the problem of finding an optimal placement of nodes on a plane in which positions of some nodes are known and fixed is different from the problem in which positions of all nodes are unknown, and deserves to be further investigated. For example, Eades [14] proposed a classical physical-analogy method to find a nice-looking placement of nodes of a graph structure in which positions of all nodes are unknown; and then Eades and Lin [15] further investigated the theoretical relationship of this method applied to place nodes of a graph structure in which some of the nodes have fixed positions. Another similar extension was also made in VLSI floorplanning, in which each circuit module is represented as a node, and the neighboring relationship of two nodes is represented as an edge. A notable work proposed in [16] is to extend the problem of determining a VLSI floorplan with the constraint of fixing some modules on some range of the floorplan, i.e., positions of their corresponding nodes are fixed within some range. Aside from fixed gateway positions, the problem concerned in this work considers QoS constraints and conducted a Markov chain analysis.

The rest of this work is organized as follows. Section 2 describes the concerned problem and some examples. Section 3 proposes a PSO approach with three local search operators for the problem. Section 4 gives experimental comparison of the PSO approaches with and without the proposed local search operators. Finally, Section 5 concludes this work with future work.

## 2. Preliminaries

This section first describes the concerned problem and then gives an example for illustrating the problem.

### *2.1. Problem description*

This work is based on our previous work in [9] to describe the concerned problem as follows. The notation used in the concerned problem is given in Table 1.

**[Insert Table 1 here]**

Consider a WMN consisting of Internet gateways, mesh routers, and mesh clients. Internet gateways make the WMN to be connected to the Internet, and hence, a mesh router or a mesh client can access the Internet only via multi-hop communication of mesh routers to some Internet gateway. That is, mesh routers serve as the function of bride between mesh clients and some Internet gateway, which further provides access to the Internet.

Each Internet gateway or mesh router is associated with a different-size coverage range. If a mesh client falls within some coverage of an Internet gateway (resp., mesh router), a link between the mesh client and the Internet gateway (resp., mesh router) is constructed. On the other hand, if coverages of two mesh routers (or a mesh router and an Internet gateway) are overlapped, it implies that positions of the two devices are close, and hence, this work supposes that they can communicate with each other, i.e., a link between the two devices is constructed. By doing so, a topology graph for the WMN can be constructed.

By extension of our previous model in [9], this work additionally considers Internet gateways in the model. Given a rectangular geographical area of width $A$ and height $B$, we consider to deploy a WMN with $\eta$ Internet gateways, $n$ mesh routers, and $m$ mesh clients, as denoted as follows:

- $W = \{w_1, w_2, \ldots, w_\eta\}$ where $w_i$ denotes the $i$-th Internet gateway, and is associated with a coverage with radius $\omega_i$ for $i \in \{1, 2, \ldots, \eta\}$;
- $R = \{r_1, r_2, \ldots, r_n\}$ where $r_i$ denotes the $i$-th mesh router, and is associated with a coverage with radius $\gamma_i$ for $i \in \{1, 2, \ldots, n\}$;
- $C = \{c_1, c_2, \ldots, c_m\}$ where $c_i$ denote the $i$-th mesh client for $i \in \{1, 2, \ldots, m\}$.

Let $U$ denote set of all nodes in this WMN, i.e., $U = W \cup R \cup C$.

This work considers a dynamic WMN, i.e., except positions of Internet gateways are fixed, mesh routers and mesh clients have mobility. Suppose that each mesh client can base on its will to switch on or off its network access at different times. When its network access is switched off, the WMN does not need to serve it. That is, positions of Internet gateways and mesh clients cannot be controlled, the decision variables of the concerned problem are positions of mesh routers at different times. To cope with such a dynamic problem, positions of mobile mesh routers are adjusted periodically to adapt to dynamic changes of mesh clients, including changes of their positions and network access. Let the time point when to adjust positions of mesh routers be called a *key time*, and suppose that the period length between two consecutive key times is fixed.

In what follows, consider deployment of a dynamic WMN. Let $D$ denote the deployed positions of three node types. Since positions of Internet gateways are always fixed, their positions are independent of time, and are denoted by $D(W) = \{D(w_1), D(w_2), \ldots, D(w_\eta)\}$, where $D(w_i) \in [0, A] \times [0, B]$ is position of the $i$-th Internet gateway for each $i$. And, the coverage area of Internet gateway $w_i$ is denoted by $\Omega_i$ (i.e., a circle centered at $D(w_i)$ with radius $\omega_i$). At the $t$-th key time, mesh client $c_i \in C$ is located at $D_t(c_i) \in [0, A] \times [0, B]$, and the decision variables of the concerned problem are positions of all mesh routers as denoted by $D_t(R) = \{D_t(r_1), D_t(r_2), \ldots, D_t(r_n)\}$, where $D(r_i) \in [0, A] \times [0, B]$ is position of the $i$-th mesh router for each $i$. And, the coverage area of mesh router $r_i$ at key time $t$ is denoted by $\Upsilon_i^t$ (i.e., a circle centered at $D_t(r_i)$ with radius $\gamma_i$). With the above position information, we establish a topology graph $G_t = (U_t, E_t)$ for the WMN at the $t$-th key time in which $U_t$ and $E_t$ are sets of nodes and edges, respectively, defined as follows:

- $U_t = W \cup R \cup (C \setminus S_t)$ where $S_t$ denotes set of the mesh clients that switch off their network access;
- For any Internet gateway $w_i \in W$ and any mesh router $r_j \in R$, if $\Omega_i \cap \Upsilon_j^t \neq \varnothing$, then $(w_i, r_j) \in E_t$;
- For any two mesh routers $r_i, r_j \in R$ where an edge linked with exact one of the two mesh router has existed in $E_t$, i.e., it can communicate with some Internet gateway via multi-hop communication of mesh routers, if $\Upsilon_i^t \cap \Upsilon_j^t \neq \varnothing$, then $(r_i, r_j) \in E_t$;
- For any Internet gateway $w_i \in W$ and any mesh client $c_j \in C \setminus S_t$, if $D_t(c_j) \in \Omega_i$ and degree of mesh client $c_i$ is 0, then $(w_i, c_j) \in E_t$;

- For any mesh client $r_i \in R$ and any mesh client $c_i \in C \setminus S_t$, if $D_t(c_i) \in \Upsilon^t_j$ and degree of mesh client $c_i$ is 0, then $(c_i, r_j) \in E_t$.

The objective of the concerned problem is to maximize two network performance measures: *network connectivity* and *client coverage*, which are used to evaluate performance of the topology graph $G_t$ established above. Since graph $G_t$ may not be connected, without loss of generality, we suppose that $G_t$ consists of $h$ subgraph components $G_t^1, G_t^2, ..., G_t^q, G_t^{q+1} ..., G_t^h$, i.e., $G_t = G_t^1 \cup G_t^2 \cup \cdots \cup G_t^h$, where $G_t^i \cap G_t^j = \varnothing$ for $i, j \in \{1, 2, \ldots, q, \ldots, h\}$ and $i \neq j$; and the first $q$ subgraph components (i.e., $G_t^1, G_t^2, ..., G_t^q$) are components with connection to some Internet gateway, i.e., mesh clients in those components can access the Internet, but the others cannot. Hence, different from the previous work in [9] that defined the network connectivity as the size of the greatest subgraph component, this work defines the *network connectivity* $\delta(G_t)$ corresponding to graph $G_t$ as follows:

$$\delta(G_t) = \sum_{i \in \{1,2...,q\}} \left|G_t^i\right| - \eta . \tag{1}$$

That is, this work aims to maximize the total sum of sizes of all the subgraph components (excluding Internet gateways) that are connected to the Internet.

Continuing the previous works, the *client coverage* $\phi(G_t)$ corresponding to graph $G_t$ is defined as follows:

$$\phi(G_t) = |\{i;\, d_t(c_i) = 1 \text{ for } i \in \{1, ..., m\}\}| \tag{2}$$

where $d_t(c_i)$ is one if mesh client $c_i$ is connected in graph $G_t$ at the $t$-th key time; otherwise, it is zero.

After introducing Internet gateways to the problem, QoS plays an important role on performance of the network. Hence, by referring [12] and [17], this work takes into account the following three QoS constraints:

- *Constraint of delay hops*: When a mesh client attempts to access the Internet, it transmits messages to some Internet gateway via multi-hop relaying of mesh routers. However, too many relaying hops would lead to a too long delay of transmission. To achieve good QoS, number of hops from a mesh client to an Internet gateway cannot be greater than an upper bound $M_h$.
- *Constraint of relay load*: As each mesh router or Internet gateway serves as a relay node in transmission, the maximal load of each mesh router or Internet gateway must be restricted, i.e., the maximal degree of a mesh router or Internet gateway cannot be greater than an upper bound $M_r$.
- *Constraint of Internet gateway capacity*: The throughput of the WMN depends on bandwidth of each Internet gateway, which is determined by the number of mesh routers to be served by the Internet gateway. Hence, the maximal number of mesh routers (resp., mesh clients) to be served by an Internet gateway cannot be greater than an upper bound $M_{wr}$ (resp., $M_{wc}$).

With the above notations and setting, the problem concerned in this work is described as follows. Consider to deploy a WMN consisting of $\eta$ Internet gateways, $n$ mesh routers, and $m$ mesh clients on a rectangular area with width $A$ and height $B$, in which positions of all Internet gateways are given and fixed; each mesh client may change its position and may switch on or off its network access at different times. For $t$ = 0, 1, 2, …, the concerned problem is to determine positions of the $n$ mesh routers at the $t$-th key time so that network connectivity $\delta(G_t)$ and client coverage $\phi(G_t)$ of the network topology graph $G_t$ corresponding to the router node placement are maximized at the same time, under constraints of delay hops, relay load, and Internet gateway capacity.

In practice, a total solution of network access service is achieved by a hybrid of not only one technique. The system concerned in this work is assumed to be managed in a centralized way, and hence requires a centralized manager, which may be a base station (BS) that covers the whole deployment area. The BS collects the information on active states and positions of mesh clients and positions of mesh routers by GPS, and transmits messages of deployment decisions to all mesh routers. In reality, the centralized system may also be achieved by using multiple BSs (with small-range coverage) to collect information and making a centralized decision. Note that this system could also be implemented by other possible centralized solutions based on the operators.

*2.2. Examples*

This subsection gives some examples for explaining the concerned problem. Consider a WMN consisting of 2 Internet gateway ($w_1$, $w_2$), 4 mesh routers ($r_1$, $r_2$, $r_3$, $r_4$), and 15 mesh clients ($c_1$, $c_2$, …, $c_{15}$) as shown in Fig. 1(a), in which Internet gateways and mesh routers have different-size coverages. If coverages of Internet gateway $w_k$ and mesh router $r_j$ are overlapped (i.e., $\Omega_k \cap \Upsilon_j^t \neq \varnothing$), then the two devices are close so that they can communicate with each other, and hence, a gateway-to-router edge between them exists in the topology graph $G_t$. Similarly, if coverages of two mesh routers $r_j$ and $r_l$ are overlapped (i.e., $\Upsilon_j^t \cap \Upsilon_l^t \neq \varnothing$), then a router-to-router edge between them exists in graph $G_t$. And, if mesh client $c_i$ falls within coverage of mesh router $r_j$ (i.e., $D_t(c_i) \in \Upsilon_j^t$), then it can communicate via mesh router $r_j$, and hence, a router-to-client edge between them exists in graph $G_t$. By doing so, a topology graph $G_t$ is established in Fig. 1(a), in which all the edges are represented as dash line segments.

**[Insert Fig. 1 here]**

The topology graph $G_t$ in Fig. 1(a) has three subgraph components: the first component $G_t^1$ includes Internet gateway $w_1$, 3 mesh routers $r_1$, $r_3$, $r_4$, and 9 mesh clients $c_1$, $c_2$, $c_4$, $c_7$, $c_8$, $c_{10}$, $c_{11}$, $c_{13}$, $c_{14}$; the second component $G_t^2$ includes Internet gateway $w_2$ and 2 mesh clients $c_{12}$, $c_{15}$; the second component $G_t^3$ includes mesh router $r_2$ and 2 mesh clients $c_3$, $c_6$. Hence, network connectivity $\delta(G_t) = |G_t^1| + |G_t^2| - 2 = 13 + 3 - 2 = 14$ and client coverage $\phi(G_t) = 13$. Note that the WMN is dynamic, i.e., mesh routers are mobile. Hence, if mesh router $r_2$ moves to the left as shown in Fig. 1(b), then both performance measures are improved: $\delta(G_t) = 18$ and $\phi(G_t) = 14$.

Consider a more complex dynamic scenario. Since mesh clients make some changes, Fig. 1(b) becomes Fig. 2(a), in which positions of mesh routers are the same, but some mesh clients change their positions (i.e., $c_2$, $c_6$, $c_{11}$, $c_{12}$) and turn off their network access (i.e., $c_1$, $c_8$, $c_{15}$), so that both the two performance measures become worse: $\delta(G_t) = 14$ and $\phi(G_t) = 10$. But, if we move mesh router $r_1$ to the right as shown in Fig. 2(b), then both the two performance measures are improved: $\delta(G_t) = 16$ and $\phi(G_t) = 12$. Hence, it is of much challenge to find the optimal placement of mesh routers for a dynamic WMN at different times.

**[Insert Fig. 2 here]**

In what follows, an example of explaining QoS constraints is given in Fig. 3, which is a placement of the WMN consisting of an Internet gateway $w_1$ and 7 mesh routers $r_1$, …, $r_7$. In Fig. 3, if an edge between two devices exists in topology graph $G_t$, and all QoS constraints are satisfied, then the edge is represented as a dash line segment; otherwise, there is no edge between the two devices.

**[Insert Fig. 3 here]**

Three QoS constraints are explained as follows:

- *Constraint of delay hops*: The upper bound $M_h$ of number of hops from a mesh router to an Internet gateway is set to 2. In Fig. 3, the mesh routers with one hop to Internet gateway $w_1$ include $r_1$, $r_3$, $r_6$; and those with two hops include $r_4$, $r_5$. Note that other mesh routers that satisfy the constraint of delay hops but have no connections to Internet gateway $w_1$ because they violate other QoS constraints.
- *Constraint of relay load*: The upper bound $M_r$ of number of the mesh routers connected by an Internet gateway or a mesh router is set to 3. In Fig. 3, degree of Internet gateway $w_1$ or each mesh router is no greater than 3. Note that Internet gateway $w_1$ are connected to only three mesh routers $r_1$, $r_3$, $r_6$ under this QoS constraint, but cannot be connected to mesh router $r_2$.
- *Constraint of Internet gateway capacity*: The upper bound $M_{wr}$ of the maximal number of mesh routers to be served by an Internet gateway is set to 5. In Fig. 3, Internet gateway $w_1$ can serve at most 5 mesh routers $r_1$, $r_3$, $r_4$, $r_5$, $r_6$ but cannot serve mesh router $r_7$ under this QoS constraint. Additionally, aside from the capacity $M_{wr}$ for number of mesh routers, this work also considers the capacity $M_{wc}$ for number of mesh clients to be served by an Internet gateway.

### *2.3. Related works*

The problem concerned in this work is to incorporate the conventional problems of router node placement (RNP) and gateway placement (GP) in WMNs. Hence, this subsection reviews previous works on various problem settings and approaches for RNP and GP in WMNs.

The previous works on RNP problems in WMNs are classified into two categories: static WMNs [4], [5], [6], [7], [8]; and dynamic WMNs, [9], [10], [11]. Most of those works are solved by various metaheuristic algorithms. The work in [4] proposed a genetic algorithm (GA) for RNP in static WMNs; then the same team investigated the effect of mutation in GAs in [5]. Later, they proposed a simulated annealing (SA) approach [6] and a tabu search approach [8] for RNP in static WMNs. Our work in [7] further proposed an approach of SA with momentum terms for the RNP with service priority in static WMNs. For RNP in dynamic WMNs, our work in [9] proposed a PSO approach with constriction coefficient. Then, we considered the RNP with social awareness in dynamic WMNs in [10], and further proposed a PSO approach with social-supporting vectors. Then, we considered the weighted version of RNP in dynamic WMNs, and proposed a novel bat-inspired algorithm (BA) for the problem.

On the other hand, most of the previous works on the GP problem were based on some known network topology models of mesh routers to design algorithms for finding candidate GP solutions [18]. A variety of metaheuristic approaches have been proposed for the GP problems in WMNs, e.g., GA [19] and fuzzy differential evolution (fuzzy DE) [20]. A survey on those approaches for GP problems can be found in [21].

In addition to different approaches for GP, a variety of problem settings for GP were considered, e.g., GP with cross-layer throughput optimization [22], GP with optimal mobile multicast design [23], and GP with QoS requirements. Among them, most previous works focused on finding the GP with QoS requirements. For example, the work in [24] proposed a polynomial time near-optimal recursive algorithm for GP, while QoS requirements are preserved. The work in [25] considered the load balancing GP problem with QoS requirements, and proposed a greedy algorithm incorporated with GA. The work in [13] proposed a heuristic algorithm for the GP problem with multiple QoS factors, including transmission range, number of gateways, number of nodes served by each gateway, gateway location, relay load, and access fairness. The work in [26] proposed a nature-inspired metaheuristic algorithm for the GP with the QoS requirement of end-to-end bounded delay communications. The work in [12] proposed a zero-degree algorithm for the problem of placing a minimum number of gateways with QoS requirements.

Some works investigated the GP problems combined with other problems (e.g., the joint GP and spatial reuse problem [27]) or those applied in industrial networking environments [28].

## 3. The Proposed Approach

Particle swarm optimization (PSO) [29] imitates the behavior of a number of particles to search the optimal solution. First, the search space of the concerned problem is mapped into a landscape, in which each location is corresponding to a candidate solution of the problem and the performance of the location is evaluated by a *fitness* function. Consider a number of particles in which each particle is associated with a *position* in the landscape. The PSO considers a main loop. In each iteration of the PSO main loop, each particle moves to a new position with better fitness at a *velocity*, which depends on its original velocity, the best position found by it so far, and the best position found by all particles so far. The PSO main loop is executed until all particles are located at the same position or the maximal number of iterations is achieved. The best position among all particles' is outputted as the final solution of the PSO.

This section first introduces main components of the proposed PSO with three local search operators, and then gives the PSO algorithm.

*3.1. Solution Encoding*

The decision variables of the concerned problem are (x, y)-coordinates of $n$ mesh routers on the rectangular deployment area with width $A$ and height $B$ where the bottom left corner of the area is its origin (0, 0). Hence, they constitute the position of a particle in the PSO. Suppose that there are $K$ particles in the PSO. Then, each particle $\kappa \in \{1, 2, \ldots, K\}$ at the $\tau$-th iteration of the PSO main loop is associated with the following three vectors:

- $X$ vector (position): $X_{\kappa}^{\tau} = (x_{\kappa 1}^{\tau}, x_{\kappa 2}^{\tau}, \ldots, x_{\kappa(2n)}^{\tau})$ records the current position of particle $\kappa$ at the $\tau$-th key time, where position of mesh router $r_i$ is represented by $(x_{\kappa(2i-1)}^{\tau}, x_{\kappa(2i)}^{\tau})$ for $i \in \{1, 2, \ldots, n\}$.
- $P$ vector (previous best position): $P_{\kappa}^{\tau} = (p_{\kappa 1}^{\tau}, p_{\kappa 2}^{\tau}, \ldots, p_{\kappa(2n)}^{\tau})$ records the best position found by particle $\kappa$ so far, i.e., for the 0-th, 1st, …, $\tau$-th iterations.
- $V$ vector (velocity): $V_{\kappa}^{\tau} = (v_{\kappa 1}^{\tau}, v_{\kappa 2}^{\tau}, \ldots, v_{\kappa(2n)}^{\tau})$ records the velocity of particle $\kappa$ at the $\tau$-th iteration.

Since all mesh routers fall within a deployment area with width $A$ and height $B$, each element in $X$ vector and $V$ vector must satisfy the following constraints:

$$0 \le x_{\kappa(2i-1)}^{\tau} \le A,\ 0 \le x_{\kappa(2i)}^{\tau} \le B, \forall i \in \{1, \ldots, n\}; \quad (3)$$

$$-A \le v_{\kappa(2i-1)}^{\tau} \le A,\ -B \le v_{\kappa(2i)}^{\tau} \le B, \forall i \in \{1, \ldots, n\}. \quad (4)$$

To avoid too violent movement of each particle, each element of V vector must satisfy the following constraint:

$$-V_{\max} \leq v_{\kappa(2i-1)}^{\tau} \leq V_{\max}, \ -V_{\max} \leq v_{\kappa(2i)}^{\tau} \leq V_{\max}, \forall i \in \{1, \ldots, n\} \qquad (5)$$

where $V_{max}$ is a fixed value no greater than $\max\{A, B\}$. This constraint is reasonable because each mesh router can move within a restricted range between two key times. Aside from the above three vectors, each particle $\kappa$ at the $\tau$-th iteration is associated with the following two fitness values, used for evaluating performance of positions:

- $f(X_{\kappa}^{\tau})$ records the fitness value of vector $X_{\kappa}^{\tau}$, i.e., the fitness value corresponding to the current position of particle $\kappa$.
- $f(P_{\kappa}^{\tau})$ records the fitness value of vector $P_{\kappa}^{\tau}$, i.e., the best fitness value found by particle $\kappa$ so far.

Additionally, from the viewpoint of the whole particle swarm, the best position found by all particles so far is recorded as $P^* = (p_1^*, p_2^*, \ldots, p_{(2n)}^*)$, and its fitness value is recorded as $f(P^*)$. After all iterations, the $P^*$ vector records the final solution of the PSO algorithm.

*3.2. Solution Decoding*

If position $X_{\kappa}^{\tau}$ of particle $\kappa$ at the $\tau$-th iteration is obtained, a corresponding network topology graph $G_{\kappa;\tau} = (U_{\kappa;\tau}, E_{\kappa;\tau})$ for the WMN can be constructed. However, both QoS constraints and the ordering of all nodes to be considered affect the graph to be constructed. Hence, this work proposes a heuristic for constructing such a topology graph in Algorithm 1.

**[Insert Algorithm 1 here]**

To avoid spectrum interference and transmission packet loss, it is general to minimize the total number of delay hops when constructing the WMN topology graph, e.g., see [30], [31]. Hence, Algorithm 1 is also based on the objective of minimizing the total number of delay hops. Lines 4 – 13 consider to link mesh routers and clients with each Internet gateway, if QoS constraints are not violated. Then, spreading from one-hop mesh routers to multi-hop mesh routers, Lines 15 – 27 consider to link mesh routers with mesh clients, until any QoS constraint is not satisfied or no node can be added to the topology graph.

**Lemma 1.** *Given position* $X_{\kappa}^{\tau}$ *of particle* $\kappa$ *at the* $\tau$*-th iteration, the corresponding topology graph can be established by Algorithm 1 in* $O(n \cdot m \cdot (\eta + n))$ *time.*

*Proof.* It is easy to check that the graph can be established by Algorithm 1, and hence, it suffices to show the time complexity of Algorithm 1.

Consider the initialization steps in Algorithm 1. Line 1 bases on $X_{\kappa}^{\tau}$ to determine ($x$, $y$)-coordinates of $n$ mesh router, and further to determine the coverage of each mesh router centered at its ($x$, $y$)-coordinate. Hence, Line 1 is done in $O(n)$ time. Line 2 initializes $\eta$ subgraphs each of which has only one node (Internet gateway), and hence is done in $O(\eta)$ time. Line 3 initializes states of $n$ mesh routers and $m$ mesh clients, and hence is done in $O(n + m)$ time.

In the first main loop of Algorithm 1 (Lines 4 – 13), Line 4 considers each of $\eta$ Internet gateways; and Line 5 checks connectivity of this Internet gateway with at most $n$ available mesh routers. Note that in Line 5, checking three QoS constraints can be done in $O(1)$, because the three QoS measures can be recorded in the data structures of their corresponding nodes. Hence, Lines 4 and 5 generate $O(\eta \cdot n)$ iterations of Lines 6 – 11. In each of those iterations, except Line 8 considers at most $m$ mesh clients, the other lines (i.e., Lines 6, 7, 9 – 11) can be done in $O(1)$ time. Hence, each iteration is done in $O(m)$ time. Hence, the first main loop can be done in $O(\eta \cdot n \cdot m)$ time.

Line 14 is done in $O(1)$ time. In the second main loop of Algorithm 1 (Lines 15 – 27), Lines 15 – 16 considers each mesh router at most once in Lines 15 – 16, and hence have $O(n)$ iterations of Lines 17 – 24. In each of those iterations, except Lines 17 and 20 consider at most $n$ mesh routers and $m$ mesh clients, respectively, the other lines can be done in $O(1)$ time. Hence, each iteration runs in $O(n \cdot m)$ time. Hence, the second main loop runs in $O(n^2 \cdot m)$ time.

In summary, initialization and two main loops of Algorithm 1 runs in time $O(n + \eta + n + m) + O(\eta \cdot n \cdot m) + O(n^2 \cdot m) = O(n \cdot m \cdot (\eta + n))$ time, as required.

$\square$

### *3.3. Fitness Evaluation*

Based on the position vector $X_\kappa^\tau$ of particle $\kappa$ at the $\tau$-th iteration of the PSO main loop, a corresponding network topology graph $G_{\kappa;\tau}$ can be constructed in the last subsection. And, in the PSO, a *fitness* value corresponding to the position vector is used for evaluating its performance. The concerned problem has two objectives: maximizing the network connectivity $\delta(G_{\kappa;\tau})$ and client coverage $\phi(G_{\kappa;\tau})$. Hence, this work integrates them into a single fitness function (e.g., see [32]) as follows:

$$f(X_\kappa^\tau) = \lambda \frac{\delta(G_{\kappa,\tau})}{m+n} + (1-\lambda)\frac{\phi(G_{\kappa,\tau})}{m} \tag{6}$$

where $\lambda$ is a parameter that controls the weighting between the two performance measures, and it falls within [0, 1]. Since $0 \leq \delta(G_{\kappa,\tau}) \leq m + n$ and $0 \leq \phi(G_{\kappa,\tau}) \leq m$, they are normalized by their respective upper bound, for convenience of comparing experimental performance.

**Lemma 2.** *Given position $X_\kappa^\tau$ of particle $\kappa$ at the $\tau$-th iteration, if the topology graph $G_{\kappa,\tau}$ has been established by Algorithm 1, then the fitness of $X_\kappa^\tau$ can be computed in $O(\eta + n + m)$ time.*

*Proof.* Since the maximal size of topology graph $G_{\kappa,\tau}$ is $\eta + n + m$, network connectivity $\delta(G_{\kappa,\tau})$ and client coverage $\phi(G_{\kappa,\tau})$ can be computed by traveling all nodes of graph $G_{\kappa,\tau}$. Hence, fitness of $X_\kappa^\tau$ can be computed in $O(\eta + n + m)$ time, as required.

□

Note that this work is theoretical, and tries the best to make the proposed system framework meet more real-world conditions, though it is impossible. In reality, mesh clients move continuously, but it is unavoidable that the decision of deploying mesh routers is made periodically. Hence, reducing the time period between two consecutive decisions could make the decisions better reflect the dynamic behavior of mesh clients. A possible solution is to implement the proposed algorithm by hardware, but the length of the time period still depends on technical advances in hardware or software. Hence, to simplify the problem, the proposed fitness function only captures the performance of some static time point of the dynamics, because the proposed algorithm does not make any prediction on the moving behavior of each mesh client.

*3.4. Updating position*

In the PSO with constriction coefficient [33], each particle $\kappa$ at the $\tau$-th iteration updates its velocity as follows:

$$V_\kappa^\tau = \varpi\left[V_\kappa^{\tau-1} + c_1 \cdot r_1 \cdot (P_\kappa^{\tau-1} - X_\kappa^{\tau-1}) + c_2 \cdot r_2 \cdot (P^* - X_\kappa^{\tau-1})\right]. \qquad (7)$$

That is, the new velocity $V_\kappa^\tau$ is updated according to its own original velocity $V_\kappa^{\tau-1}$, its previous best position $P_\kappa^{\tau-1}$, its own original position $X_\kappa^{\tau-1}$, and the best position $P^*$ found by all particle so far. Parameters $c_1$ and $c_2$ control the degrees to which the previous best position and the global best position affect the velocity, respectively. $r_1$ and $r_2$ are random numbers from [0, 1]. To avoid that $V_k^\tau$ is too large so that the best position is missed due to fast movement, constriction parameter $\varpi$ is used to reduce the total velocity. It is generally set to be linearly decreased from 0.9 to 0.4 from previous works. A proper parameter setting should make particles with global search ability at the early stage and with local search ability at the later stage, e.g., $c = c_1 + c_2 > 4$ and $\varpi = 2/|2 - c - \sqrt{c^2 - 4c}|$ in [34]. After velocity is determined, position $X_\kappa^\tau$ of particle $\kappa$ at the $\tau$-th iteration is updated as follows:

$$X_\kappa^\tau \leftarrow X_\kappa^{\tau-1} + V_\kappa^\tau. \qquad (8)$$

*3.5. Local search*

By observing visualization of our previous experimental results in [9] that applied the legacy PSO to find placement of router nodes in dynamic WMNs, we find that although most of those results achieve good quality, the dynamic cases when some mesh clients suddenly switch on or off their network access at different times lead to the following two main bad-quality placement problems:

- *Almost-overlapping router*: When positions of two mesh routers are too close, their coverages are almost overlapped so that they cover almost the same mesh clients. For example, in Fig. 4(a), a large ratio of coverage of mesh router $r_j$ falls within coverage of mesh router $r_i$. In this case, the client coverage becomes worse.
- *Few-clients-covered router*: Some mesh routers may cover too few or almost no mesh clients, so that those mesh routers are idle (e.g., mesh router $r_j$ in Fig. 4(b)).

**[Insert Figure 4 here]**

To avoid the above two problems, this work proposes the following three local search operators:

- *Dispersing local search operator*: To avoid almost-overlapping routers, this operator aims to disperse mesh routers to promote the total coverage of all mesh routers to serve more mesh clients. Hence, this operator considers each pair of mesh routers $r_i$ and $r_j$ in which their respective coverage radiuses are $\gamma_i$ and $\gamma_j$; without loss of generality, $\gamma_i \geq \gamma_j$; and mesh router $r_j$ is located within coverage of mesh router $r_i$ (Fig. 5). Given a threshold parameter $L_o$, this operator moves mesh router $r_j$ to a random position away from mesh router $r_i$ with a distance no more than $l_1$ if the following condition holds:

$$\frac{dis_\tau(r_i, r_j) + \gamma_j}{\gamma_i} \le L_o, \ \forall i, j \in \{1, \ldots, n\}, i \ne j \tag{9}$$

where $dis_\tau(r_i, r_j)$ is the distance between mesh routers $r_i$ and $r_j$ at the $\tau$-th iteration. If $L_o$ is set to 1, then it implies that we would like the two coverages to be overlapped totally (see Fig. 5(b)). If $L_o$ is set to approach to 2, then it implies that we would like the two mesh routers depart farther from each other (see Fig. 5(a)). Hence, $L_o$ should be set within range [1, 2]. The ratio on the left side of the above condition (9) reflects both distance $dis_\tau(r_i, r_j)$ radius $\gamma_i$, and radius $\gamma_j$, i.e., aside from the distance between two mesh routers, if the smaller radius $\gamma_j$ is large as compared to radius $\gamma_i$, the ratio of coverage of mesh router $r_i$ that is not overlapped with coverage of mesh router $r_j$ is larger, and hence, the operator is not needed.

**[Insert Figure 5 here]**

- *Covermore local search operator*: To avoid few-clients-covered routers, this operator finds all of those routers and move their positions locally to increase their client coverage. Specifically, consider each mesh router $r_j$ for $j \in \{1, \ldots, n\}$. Given a threshold parameter $L_c$, this operator moves mesh router $r_j$ locally within a circle centered at $r_j$ with a radius no more than $l_2$ if the following condition holds:

$$d_\tau(r_j) \le L_c \tag{10}$$

where $d_\tau(r_j)$ is number of mesh clients connected with mesh router $r_j$ at the $\tau$-th iteration. For example, $d_t(r_i) = 3$ and $d_t(r_j) = 1$ in Fig. 6. If $L_c = 2$, then this operator moves mesh router $r_j$.

**[Insert Figure 6 here]**

- *Random local search operator*: Given a parameter $L_r$, this operator randomly selects $L_r$ mesh routers, and then move each of them to a random position in its neighborhood, i.e., within a circle centered at this mesh router with a radius no more than $l_3$. That is, this operator tries to find a potential good-quality solution without any problem information.

The reason why the proposed local search operators are crucial is given as follows. Genetic algorithm (GA) is a notable metaheuristic algorithm, and has been applied to solve various problems. The GA with local search improvement is called the memetic algorithm (MA). Recently, lots of works have focused to solve various problems by variants of MA, e.g., max-bisection problem [35], tardiness minimization problem of a single machine [36], and dynamic pickup-and-delivery problem [37]. Hence, the PSO with novel local search designs has also received a lot of attention recently, e.g., [38]. When designing local search operators in PSO, a key point is to design those operators to be simple. Since each iteration of the main loop of the PSO conducts the operator once, a simple local search operator would not increase too much overhead in computation.

### *3.6. The PSO algorithm with three local search operators*

Based on the designs on PSO components in the last subsections, the PSO algorithm with three local search operators is given in Algorithm 2, which aims to find the best placement of mesh routers at the $t$-th key time. That is, the first placement of mesh routers is generated by Algorithm 2 when setting $t = 0$. Then, the dynamic WMN using this mesh router placement is operated until the next key time. Then, the $(t + 1)$-th placements of mesh routers are generated by Algorithm 2 when setting $t = 1, 2, \ldots$, respectively.

Note that the algorithm is almost the same with our previous algorithm in [9] except Lines 20 – 25 in Algorithm 2 conducts one of the three proposed local search operators.

**[Insert Algorithm 2 here]**

The parameterized time complexity of Algorithm 2 is given as follows:

**Theorem 1.** *Algorithm 2 runs in* $O(T \cdot (K \cdot n \cdot m \cdot (\eta + n)))$ *time.*
*Proof.* In Algorithm 2, the loop of Lines 1 – 13 has $K$ iterations (Line 1), and each iteration in this loop can be done by assigning positions of $n$ mesh routers in $O(n)$ time except Line 3 establishes a topology graph in $O(n \cdot m \cdot (\eta + n))$ time by Lemma 1 and computes fitness in $O(\eta + n + m)$ time by Lemma 2. Hence, the loop of Lines 1 – 13 runs in $O(K \cdot n \cdot m \cdot (\eta + n))$ time.

The loop of Lines 14 – 33 considers $T$ iterations (Line 33), in which the loop of Lines 15 – 32 considers $K$ iterations (Line 15). And, each iteration in this loop can be done by assigning positions of $n$ mesh routers in $O(n)$ time except Line 21 checks coverages of each pair of mesh routers in $O(n^2)$ time when applying the dispersing local search operator; Lines 20 and 22 establishes a topology graph and computes fitness in $O(n \cdot m \cdot (\eta + n))$ time. Hence, the loop of Lines 14 – 33 runs in $O(T \cdot (K \cdot n \cdot m \cdot (\eta + n)))$ time, as required

□

Note that number of iterations $T$ and number of particles $K$ are given parameters, and $\eta \leq n$ in general; otherwise, the problem becomes trivial. Hence, the above time complexity becomes $O(n^2 \cdot m)$. That is, the bottleneck of the time complexity is in establishing the topology graph, in which number of links between $n$ mesh routers is connected in $O(n^2)$ time during which connectivity between each mesh router and $m$ mesh clients is checked in $O(m)$

time. It would be of interest to propose a special data structure to improve the time complexity.

### *3.7. Convergence and Stability Analysis of the Proposed PSO Approach*

In the proposed PSO algorithm, all particles are classified into two possible types of movement: either with or without local search operators (i.e., depending on whether one of three local search operators in Line 21 of Algorithm 2 is conducted or not).

Let $\Delta\tau$ is a time step. For each particle $\kappa$ without local search operators at iteration $\tau$, its velocity $V_\kappa^\tau$ and position $X_\kappa^\tau$ are the same with those in the classical PSO approach, and hence follow almost the same analysis with that in [39], which is summarized as follows, for clarification for the later analysis. Velocity $V_\kappa^\tau$ and position $X_\kappa^\tau$ are updated, respectively, according to the following generalized equations:

$$V_\kappa^\tau = \varpi\left[V_\kappa^{\tau-1} + c_1 \cdot r_1 \cdot \frac{(P_\kappa^{\tau-1} - X_\kappa^{\tau-1})}{\Delta\tau} + c_2 \cdot r_2 \cdot \frac{(P^* - X_\kappa^{\tau-1})}{\Delta\tau}\right]; \tag{11}$$

$$X_\kappa^\tau \leftarrow X_\kappa^{\tau-1} + V_\kappa^\tau \cdot \Delta\tau . \tag{12}$$

The above equation becomes the following by replacing (11) into (12):

$$X_\kappa^\tau = X_\kappa^{\tau-1} + \varpi \cdot V_\kappa^\tau \cdot \Delta\tau$$

$$+ \varpi \cdot (c_1 \cdot r_1 + c_2 \cdot r_2) \cdot \left(\frac{c_1 \cdot r_1 P_\kappa^{\tau-1} + c_2 \cdot P^*}{c_1 \cdot r_1 + c_2 \cdot r_2} - X_\kappa^{\tau-1}\right) \tag{13}$$

Let $\hat{X}_\kappa^\tau = X_\kappa^{\tau-1} + \varpi \cdot V_\kappa^{\tau-1} \cdot \Delta\tau$ , $\alpha = \varpi \cdot (c_1 \cdot r_1 + c_2 \cdot r_2)$ , and $\overline{P}_\kappa^\tau = (c_1 \cdot r_1 P_\kappa^{\tau-1} + c_2 \cdot P^*) / (c_1 \cdot r_1 + c_2 \cdot r_2) - X_\kappa^{\tau-1}$. Then, the above equation can be regarded as a general gradient line-search form as follows:

$$X_\kappa^\tau = \hat{X}_\kappa^\tau + \alpha \overline{P}_\kappa^\tau \tag{14}$$

By rearranging (11) and (13), we have the following matrix form:

$$\begin{bmatrix} X_\kappa^\tau \\ V_\kappa^\tau \end{bmatrix} = \begin{bmatrix} 1 - \varpi c_1 r_1 - \varpi c_2 r_2 & \varpi \Delta\tau \\ -\dfrac{\varpi(c_1 r_1 - c_2 r_2)}{\Delta\tau} & \varpi \end{bmatrix} \begin{bmatrix} X_\kappa^{\tau-1} \\ V_\kappa^{\tau-1} \end{bmatrix} + \begin{bmatrix} \varpi c_1 r_1 & \varpi c_2 r_2 \\ \dfrac{\varpi c_1 r_1}{\Delta\tau} & \dfrac{\varpi c_2 r_2}{\Delta\tau} \end{bmatrix} \begin{bmatrix} P_\kappa^{\tau-1} \\ P^* \end{bmatrix}. \tag{15}$$

The above form can be regarded as a discrete-dynamic system for positions and velocities of PSO, in which $P_\kappa^{\tau-1}$ and $P^*$ are external inputs. And, this system has been shown to achieve convergence only when $V_\kappa^\tau$ = 0 and $X_\kappa^{\tau-1} = P_\kappa^{\tau-1} = P^*$ as iteration number $\tau$ approaches infinity in [39].

On the other hand, for each particle $\kappa$ with local search operators at iteration $\tau$, an additional movement due to one of three local search operators is added to its position $X_\kappa^\tau$. Remind that the *dispersing*, *covermore*, and *random* local search operators lead to maximal movement distances $l_1$, $l_2$, and $l_3$, respectively. And, the probabilities of conducting each of the three local search operators are equal, i.e., 1/3. Hence, the position updating equation transforms from (13) to the following:

$$X_\kappa^\tau = X_\kappa^{\tau-1} + \varpi \cdot V_\kappa^{\tau-1} \cdot \Delta\tau$$
$$+ \varpi \cdot (c_1 \cdot r_1 + c_2 \cdot r_2) \cdot (\frac{c_1 \cdot r_1 P_\kappa^{\tau-1} + c_2 \cdot P^*}{c_1 \cdot r_1 + c_2 \cdot r_2} - X_\kappa^{\tau-1})$$

$$+\frac{1}{3}\cdot(r_3\cdot l_1+r_4\cdot l_2+r_5\cdot l_3) \tag{16}$$

where $r_3$, $r_4$, and $r_5$ are random numbers from [0, 1]. Hence, the following matrix form in (15) for the case with local search operators is rewritten as follows:

$$\begin{aligned}\begin{bmatrix} X_\kappa^\tau \\ V_\kappa^\tau \end{bmatrix} &= \begin{bmatrix} 1-\varpi c_1 r_1-\varpi c_2 r_2 & \varpi\Delta\tau \\ -\dfrac{\varpi(c_1 r_1-c_2 r_2)}{\Delta\tau} & \varpi \end{bmatrix}\begin{bmatrix} X_\kappa^{\tau-1} \\ V_\kappa^{\tau-1} \end{bmatrix}+\begin{bmatrix} \varpi c_1 r_1 & \varpi c_2 r_2 \\ \dfrac{\varpi c_1 r_1}{\Delta\tau} & \dfrac{\varpi c_2 r_2}{\Delta\tau} \end{bmatrix}\begin{bmatrix} P_\kappa^{\tau-1} \\ P^* \end{bmatrix} \\ &+\begin{bmatrix} \dfrac{r_3\cdot l_1+r_4\cdot l_2+r_5\cdot l_3}{3} \\ 0 \end{bmatrix}\end{aligned} \tag{17}$$

If no external excitation exists in the above dynamic system, then $P_\kappa^{\tau-1}$ and $P^*$ are constant, i.e., no better positions can be found by other particles. In this case, this system could lead to convergence. That is, when iteration number $\tau$ goes to ∞, we have $X_\kappa^\tau = X_\kappa^{\tau-1}$ and $V_\kappa^\tau = V_\kappa^{\tau-1}$. Hence, the above system becomes the following:

$$\begin{aligned}\begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} -\varpi c_1 r_1-\varpi c_2 r_2 & \varpi\Delta\tau \\ -\dfrac{\varpi(c_1 r_1-c_2 r_2)}{\Delta\tau} & \varpi-1 \end{bmatrix}\begin{bmatrix} X_\kappa^{\tau-1} \\ V_\kappa^{\tau-1} \end{bmatrix}+\begin{bmatrix} \varpi c_1 r_1 & \varpi c_2 r_2 \\ \dfrac{\varpi c_1 r_1}{\Delta\tau} & \dfrac{\varpi c_2 r_2}{\Delta\tau} \end{bmatrix}\begin{bmatrix} P_\kappa^{\tau-1} \\ P^* \end{bmatrix} \\ &+\begin{bmatrix} \dfrac{r_3\cdot l_1+r_4\cdot l_2+r_5\cdot l_3}{3} \\ 0 \end{bmatrix}\end{aligned} \tag{18}$$

And, the above form holds only when $V_\kappa^\tau = 0$, $X_\kappa^{\tau-1} = P_\kappa^{\tau-1} = P^*$, and $l_1 = l_2 = l_3 = 0$. As compared with the one without local search operators, this condition additionally requires $l_1 = l_2 = l_3 = 0$. That is, if the distance bounds $l_1$, $l_2$, $l_3$ used in the proposed three local search operators are set as decreasing functions and finally approach to zero, the dynamic system could achieve convergences as iteration number goes to infinity.

As for stability analysis, if $l_1 = l_2 = l_3 = 0$ finally, the stability analysis of the PSO is the same with the original analysis in [39]. Hence, the analysis is skipped in this paper.

## 4. Simulation Design and Results

This section gives simulation results of the proposed algorithm with three local search schemes. First, the simulation environment is introduced. Then, the simulation results are compared with our previous approach in [9] that did not consider local search operators and was not designed for coping with constraints of gateway postions and QoS. Finally, the simulation results for dynamic WMNs are analyzed in detail.

### *4.1. Simulation Environment*

This work refers the network scenario in [4] and the gateway placement in [40] to modify the experimental dataset in [9] as follows:

- Case 1: Consider to deploy 4 Internet gateways, 16 mesh routers, and 48 mesh clients on a 32 × 32 geographical area, i.e., $\eta = 4$, $m = 16$, $n = 48$, and $A = B = 32$. Coverage radius of each gateway and mesh router is a real number from a uniform distribution $U(3, 6)$.
- Case 2: Consider to deploy 4 Internet gateways, 32 mesh routers, and 96 mesh clients on a 64 × 64 geographical area, i.e., $\eta = 4$, $m = 32$, $n = 96$, and $A = B = 64$. Coverage radius of each gateway and mesh router is a real number from a uniform distribution $U(4\sqrt{2} - 2, 8\sqrt{2} - 4)$.
- Case 3: Consider to deploy 4 Internet gateways, 64 mesh routers, and 192 mesh clients on a 128 × 128 geographical area, i.e., $\eta = 4$, $m = 64$, $n$

= 192, and $A$ = $B$ = 128. Coverage radius of each gateway and mesh router is a real number from a uniform distribution $U(7, 14)$.

Note that each device is located at a grid point in [6], but this work allows each device to be located at a floating-point position within the deployment area.

In the above experimental datasets, Cases 1, 2, and 3 are from a small to a large scale, and hence, they are also called small-, middle-, and large-scale cases, respectively. We generate 10 different instances for each of the three cases, in which positions of Internet gateways are distributed in the center positions of four uniform sub-regions of the deployment area [40] as shown in Fig. 7.

**[Insert Fig. 7 here]**

The proposed PSO algorithm is implemented in C++ programming language. Parameter setting used in the concerned problem and the proposed algorithm is given in Table 2. In the experiments, four approaches with and without local search operators are conducted. For notational convenience, *PSO*, *PSO with dispersing*, *PSO with covermore*, and *PSO with random* are given to denote the PSO approach without any local search operator and the PSO approaches with dispersing, covermore, and random local search operators, respectively.

**[Insert Table 2 here]**

*4.2. Experimental Results for Static WMNs*

This subsection analyzes the experimental results of four approaches for static WMNs. First, experimental results with different numbers of nodes are analyzed. Then, convergence using different numbers of iterations is analyzed. Last, three different local search operators are analyzed.

*4.2.1. Analyzing number of nodes*

Experimental results of four approaches for the 10 instances of case 1 is given in Fig. 8. From Fig. 8, the fitness value for the PSO without any local search operator (or say, the legacy PSO) is about 0.86; and the fitness value for the PSO with each local search operator is about 0.90. Furthermore, from the ratio of two performance measures in each bar in Fig. 8, the PSO with each local search operator perform better than the legacy PSO in terms of both measures on average. Hence, local search operators are indeed beneficial.

**[Insert Fig. 8 here]**

*4.2.2. Convergence analysis*

Fitness values of 150 iterations of four approaches for an instance of case 1 are plotted in Fig. 9, from which fitness of each approach converges to a fixed value before 40 iterations. From Fig. 9, each of the three PSO approaches with local search operators converges to a larger fitness value than the legacy PSO, though the PSO with *random* local search operator converges slower than the legacy PSO.

**[Insert Fig. 9 here]**

*4.2.3. Analyzing experimental results using three local search operators*

Statistics of experimental results of running 20 times of four approaches for 10 instances of case 1 are given in Table 3, in which the best, mean, worst, and standard deviation (stdDev) of 20 fitness values are recorded. Box plots of those results of four approaches are shown in Figs. 11(a), 11(b), 11(c), and 11(d), respectively, in which the maximum, median, and minimum among 20 fitness values can be observed by bars; the first and third quartiles can be observed by the up and down sides of the box, respectively.

**[Insert Table 3 here]**

From Table 3 and Fig. 10, the PSO with each local search operator (i.e., Figs. 10(a), 10(b), and 10(c)) always has smaller variation than the legacy PSO (i.e., Fig. 10(d)), and hence, it is concluded that the PSO with each local search operator performs more stably. For all instances, the PSO with each local search operator always has better fitness values than the legacy PSO. Among all PSO approaches, *the PSO with covermore local search operator* performs the best because the best fitness values of 8 instances achieve 1 (see Table 3); and its results are the most stable because its difference between best and worst fitness values is the smallest (see Fig. 10(b)). Since the PSO with covermore local search operator has the best performance, we then comparison the results of PSO with dispersing and random local search operators. From Fig. 10(a) and Fig. 10(c), the PSO with random local search operator performs better and is more stable than that with dispersing local search operator.

**[Insert Fig. 10 here]**

The box plot of experimental results of running 20 times of four approaches for 10 instances of 3 cases are given in Fig. 11. From Fig. 11, the results using three PSO approaches with local search operators have significantly differences from the legacy PSO in terms of fitness value and stableness, and perform better. Among them, the PSO with covermore local search operator performs the best, followed by the PSO with random local search operator, the

PSO with dispersing local search operator, and finally the legacy PSO. Among the three cases, the variation of case 1 is the largest.

**[Insert Fig. 11 here]**

*4.3. Experimental Results for Dynamic WMNs*

Dynamic WMNs have two dynamic scenario: First, mesh clients may change their positions at different times, but never switch off their network access, i.e., all mesh clients request the network access. Second, mesh clients may change their positions and switch on or off their network access at different times.

Although those two dynamic scenarios were considered in our previous works in [9], this work further discovers that the dynamic process in the second scenario is a stochastic process [41], in which the process that a mesh client switches on or off its network access is to consider its state at the pervious key time. That is, a mesh client's current state (whose network access is on or off) influences the probability of its next state, which is a conditional probability, called Markov chain [42]. The corresponding Markov chain is illustrated in Fig. 12(a), in which state 0 represents that the network access of the concerned mesh client is off; while state 1 represents it to be on.

**[Insert Fig. 12 here]**

As shown in Fig. 12(b), the transition matrix of this Markov chain presents all probabilities that each state at the $t$-th key time is transited to another state at the $(t + 1)$-th key time. For example, $p_{00}$ is the probability that state 0 at the t-th key time keeps state 0 at the $(t + 1)$-th key time, i.e., the network access keeps off; and the probability when the network access becomes on is $p_{01}$.

Hence, $p_{00} + p_{01} = 1$. This Markov chain is a stochastic process whose associated probability distributions are stable, i.e., as time goes by, a stable state exists. Let $\pi_0$ (resp., $\pi_1$) denote the probability that a mesh client switch off (resp. on) its network access. Based on the stable state of the Markov chain, the expected number of mesh clients that switch off (resp., on) network access is $m \cdot \pi_0$ (resp., $m \cdot \pi_1$). That is, although $m$ mesh clients are considered to switch on or off their network access initially, the two expected number of mesh clients should be used to be the benchmark of the experimental results in the long run when each mesh client can choose to switch on or off its network access arbitrarily.

*4.3.1. The first dynamic scenario (where only positions are dynamic)*

On dynamic settings in simulation, it would be impossible to consider all possible behaviors of mesh clients. Hence, most works conducted simulations in some random scenarios, e.g., the simulation in the previous work in [43] considered that each mesh client is initially assigned to a random position, and moves according to a random direction.

In the first dynamic scenario (where mesh clients may change their positions at different times but never switch off their network access) of our simulation, the experimental result of four approaches for an instance of cases 1, 2, and 3 are given in Figs. 14(a), 14(b), 14(c), respectively. Note that positions of mesh clients are changed at each key time (i.e., 15 iterations in our setting), and hence, the fitness value at each 15 iterations has a drop in each case in Fig. 13.

**[Insert Fig. 13 here]**

From all cases in Fig. 13, the legacy PSO performs worst, as it is trapped to a lower fitness level at each key time and almost cannot recover to a higher fitness level. Conversely, the PSO approach with each local search operator recovers the fitness value to a higher level after a drop at each key time, and has remarkably better performance than the legacy PSO. Among them, the PSO with covermore local search operator has the best performance in terms of fitness and stability, and its fitness value goes above 0.95 in almost all cases.

#### *4.3.2. The second dynamic scenario (where both positions and network access are dynamic)*

This subsubsection considers the second dynamic scenario, in which mesh clients may change their positions and switch on or off their network access at different times. To the best of our understanding, pervious works on RNP problems in dynamic WMNs never analyzed the reason why fitness becomes lower as time goes by in the second dynamic scenario. Hence, this work finds that the dynamic process can be captured by a Markov chain, and hence, this work applies its transition matrix to derive probabilities of stable states, the expected number of survival mesh clients, and further the expected fitness value in the long run.

To find the stable state in the long run, we consider case 1 (i.e., $\eta = 4$, $n = 16$, $m = 48$) for 6000 iterations (i.e., 6000/15 = 400 key times) under the following three conditions of probabilities of switching network access (Fig. 14):

- Condition 1: This condition is the same as the setting in our previous work in [9], i.e., the probability that a mesh client switches its network access state is 0.01. Hence, its transition matrix is shown in Fig. 14(a). Based on the matrix, the stable probability $\pi_0$ (resp., $\pi_1$) that a mesh client switch off (resp. on) its network access in the long run is 0.5 (resp., 0.5). Hence, the expected number of mesh clients that switch on network access is $\pi_1 \times m = 0.5 \times 48 = 24$. Hence, the theoretical optimal fitness value is $\lambda \cdot (\pi_1 \cdot m + n) / (m + n) + (1 - \lambda) \cdot \pi_1 \cdot m / m = 0.3 \cdot (24 + 16) / (48 + 16) + 0.7 \cdot 24 / 48 = 0.5375$.

- Condition 2: Since $p_{10} = p_{01}$ in condition 1 is too small and not practical, condition 2 sets them to be 0.5 (Fig. 14(b)) so that it also derives the same stable probabilities (i.e., $\pi_0 = \pi_1 = 0.5$) with condition 1. Hence, the theoretical optimal fitness value is also 0.5375.
- Condition 3: This condition considers a more practical requirement where mesh clients tend to keep network access. Hence, this condition sets that $p_{10} = 0.1$ (i.e., the probability that an online mesh client turns off its network access) is less than $p_{01} = 0.5$ (i.e., the probability that an offline mesh clients turns on its network access) (Fig. 14(c)). We obtain the stable probabilities $\pi_0 = 0.1667$ and $\pi_1 = 0.8333$, and the expected number of mesh clients is 42. Hence, the theoretical optimal fitness value is 0.8458.

**[Insert Fig. 14 here]**

Experimental results of running four approaches on an instance of case 1 under conditions 1, 2, and 3 for 6000 iterations (i.e., 400 key times) in the second dynamic scenario are given in Figs. 15(a), 15(b), and 15(c), respectively. In each subfigure in Fig. 15, the theoretical optimal fitness value in the long run under each condition is represented as a horizontal line.

From Fig. 15(a) and 15(b), although conditions 1 and 2 have the same stability probabilities (i.e., $\pi_0$ and $\pi_1$), their results are much different. In Fig. 15(a), the fitness value of each of the proposed PSO approaches tends slower and stably to the theoretical optimal fitness value, and achieves it at after about the 2000-th iteration. But, in Fig. 15(b), the fitness value of each of the proposed PSO approaches keeps vibrating more near the theoretical optimal fitness value. At the two conditions, the three proposed PSO approaches always recover to the theoretical optimal fitness level and obviously perform better than the legacy PSO. Among them, the PSO with covermore local search operator perform best.

**[Insert Fig. 15 here]**

Since the expected number of mesh clients in condition 3 is greater than those in the other two conditions, the theoretical optimal fitness value is also greater, and hence, the ratio when the theoretical optimal fitness value is greater than others is higher (Fig. 15(c)). From Fig. 15(c), it is more obvious to observe that the PSO with covermore local search operator performs best, followed by the PSO with random and dispersing local search operators, and the legacy PSO.

Furthermore, the statistics of the stable results (i.e., those generated after 2000 iterations) using four PSO approaches under three conditions are given in Tables 4, 5, and 6, respectively.

**[Insert Table 4 here]**

**[Insert Table 5 here]**

**[Insert Table 6 here]**

From Tables 4, 5, and 6, the results of the proposed three PSO approaches are close to the theoretical optimal fitness value, in which those of the PSO with covermore local search operator are the most close. Due to their larger stand deviations, their confidential levels under large samples are compared. Under 95% confidence interval, the PSO with covermore local search operator always performs well in both fitness values of lower and upper confidence founds at three different conditions, i.e., the good performance is not affected by large variance. By similar analysis, the performances of the PSO with random local search operator is the second best, followed by the PSO with dispersing local search operator, and finally the legacy PSO.

## 5. Conclusion

This work has proposed a PSO approach with three local search operators for placement router nodes to adapt to changes of mesh clients in dynamic WMNs under constraints of positions of Internet gateways and QoS. Since the previous PSO approach for a similar problem is easily trapped at a local optimal solution at each change, the proposed three local search operators are helpful in escaping the local optimal solution, and improve convergence and stability. Experimental results show that the proposed PSO approaches with three local search operators perform better than the legacy PSO in all cases and dynamic conditions. Furthermore, we find that the dynamic conditions can be described as Markov chains, and the theoretical optimal fitness values can be derived and is used as the criteria for experimental comparison.

The major novelty of this work is that this work is the first to incorporate two placement problems: gateway placement as well as router node placement. Such a problem has never been considered from the literature. And, comprehensive theoretical and experimental analyses for the proposed model and method have been provided. In the proposed model, although positions of gateways are fixed and only positions of mesh routers are decision variables, this problem is still novel, e.g., the notable works in [15] and [16] applied similar concepts in graph drawing and VLSI floorplanning, respectively. Like most previous works in communications networks, the proposed model can only be verified via simulation, because real-world verification would be impossible and costly. But, our simulation results can provide a reference for real next-generation networking applications and management.

In the future, weights or priority [44] and community behavior [45] of mesh clients can be considered. Additionally, power consumption [46], [47] can also be considered in the concerned problem.

**Acknowledgements**

This work was supported in part by MOST 104-2221-E-009-134-MY2, Taiwan.

**References**

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

[2] D. Benyamina, A. Hafid, and M. Gendreau, Wireless mesh networks design—A survey, IEEE Communications Surveys & Tutorials 14(2) (2012) 299-310.

[3] A. Malik, J. Qadir, B. Ahmad, K.-L. A. Yau, U. Ullah, QoS in IEEE 802.11-based wireless networks: A contemporary review, Journal of Network and Computer Applications 55 (2015) 24-46.

[4] F. Xhafa, C. Sánchez, and 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), 2010, pp. 465-472.

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

[6] F. Xhafa, A. Barolli, C. Sánchez, and 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.

[7] C.-C. Lin, L. Shu, and D.-J. Deng, Router node placement with service priority in wireless mesh networks using simulated annealing with momentum terms, IEEE Systems Journal, in press.
[8] F. Xhafa, C. Sánchez, A. Barolli, M. Takizawa, Solving mesh router nodes placement problem in wireless mesh networks by tabu search algorithm, Journal of Computer and System Sciences 81(8) (2015) 1417-1428.
[9] C.-C. Lin, Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis, Information Sciences 232 (2013) 294-308.
[10] C.-C. Lin, P.-T. Tseng, T.-Y. Wu, and D.-J. Deng, Social-aware dynamic router node placement in wireless mesh networks, Wireless Networks, in press.
[11] C.-C. Lin, Y.-S. Li, D.-J. Deng, A bat-inspired algorithm for router node placement with weighted clients in wireless mesh networks, in: Proceedings of 9th International Conference on Communications and Networking in China (CHINACOM 2014), 2014, pp. 139-143.
[12] M. Seyedzadegan, M. Othman, B. M. Ali, and S. Subramaniam, Zero-degree algorithm for Internet gateway placement in backbone wireless mesh networks, Journal of Network and Computer Applications 36(6) (2013) 1705-1723.
[13] Y. Drabu and H. Peyravi, Gateway Placement with QoS Constraints in Wireless Mesh Networks, in: Proceedings of 7th International Conference on Networking (ICN 2008), IEEE Press, 2008, pp. 46-51.
[14] P. Eades, A heuristic for graph drawing, Congressus Numerantium 42 (1984) 149-160.
[15] P. Eades and X. Lin, Spring algorithms and symmetry, Theoretical Computer Science 240 (2000) 379-405.
[16] F.Y. Young, D.F. Wong and Hannah H. Yang, Slicing floorplan with range constraints, IEEE Transactions of Computer-Aided Design of Integrated Circuits and Systems 19(2) (2000) 272-278.

[17] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, Gateway placement optimization in wireless mesh networks with QoS constraints, IEEE Journal on Selected Areas in Communications 24(11) (2006) 2127-2136.

[18] B. He, B. Xie, and D. P. Agrawal, Optimizing deployment of internet gateway in wireless mesh networks, Computer Communications 31(7) (2008) 1259-1275.

[19] A. M. Ahmed, A. H. A. Hashim, and W. H. Hassan, Investigation of gateway placement optimization approaches in wireless mesh networks using genetic algorithms, in: Proceedings of International Conference on Computer and Communication Engineering (ICCCE 2014), pp. 68 - 71, 2014.

[20] G. M. Sheeba, A. Nachiappan, Fuzzy differential evolution based gateway placements in WMN for cost optimization, in: Intelligent Systems Technologies and Applications, Volume 385 of the series Advances in Intelligent Systems and Computing, 2015, pp. 137-145.

[21] A. M. Ahmed and A. H. A. Hashim, Metaheuristic approaches for gateway placement optimization in wireless mesh networks: A survey, International Journal of Computer Science and Network Security 14(12) (2014) 1-5.

[22] F. Li, Y. Wang, X.-Y. Li, Gateway placement for throughput optimization in wireless mesh networks, in: Proceedings of IEEE International Conference on Communications (ICC '07), pp 4955-4960, 2007.

[23] M. L. Sanni, A.-H. A. Hashim, F. Anwar, A. W. Naji, G. S. M. Ahmed, Gateway placement optimisation problem for mobile multicast design in wireless mesh networks, in: Proceedings of 2012 International Conference on Computer and Communication Engineering (ICCCE 2012), IEEE Press, 2012, pp. 446-451.

[24] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, Gateway placement optimization in wireless mesh networks with QoS constraints, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 2127-2136.

[25] F. Zeng and Z. Chen, Load balancing placement of gateways in wireless mesh networks with QoS constraints, in: Proceedings of 9th International Conference for Young Computer Scientists (ICYCS 2008), pp. 445-450, 2008.

[26] D. Benyamina, A. Hafid, M. Gendreau, Optimal placement of gateways in multi-hop wireless mesh networks: a clustering-based approach, in: Proceedings of IEEE 34th Conference on Local Computer Networks (LCN 2009), pp. 625 - 632, 2009.

[27] V. Targon, B. Sansò, and A. Capone, The joint gateway placement and spatial reuse problem in wireless mesh networks, Computer Networks 54(2) (2010) 231-240.

[28] H. Wang, J. Li, and H. Gao, Minimized cost gateway deployment in cyber-physical systems, International Journal of Distributed Sensor Networks 2015 (2015), Article ID 813656.

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

[30] L. Pan, H. XiaoXia, F. Yuguang, and L. Phone, Optimal placement of gateways in vehicular networks, IEEE Transactions on Vehicular Technology 56(6) (2007) 3421-3430.

[31] S. P. T. Srinivasan and C. Chellappan, Semi-randomised propagation for secure routing in Wireless Sensor Networks, in: Proceedings of International Conference on Recent Trends in Information Technology (ICRTIT 2011), IEEE Press, 2011, pp. 428-432.

[32] R. L. Keeney and H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Trade-offs, Cambridge University Press, 1993.

[33] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6(1) (2002) 58-73.
[34] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, 2010, pp. 760-766.
[35] G. Lin and W. Zhu, An efficient memetic algorithm for the max-bisection problem, IEEE Transactions on Computers 63(6) (2014) 1365-1376.
[36] M. A. González and C. R. Vela, An efficient memetic algorithm for total weighted tardiness minimization in a single machine with setups, Applied Soft Computing 37 (2015) 506-518.
[37] Z. Zhu, J. Xiao, S. He, Z. Ji, and Y. Sun, A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem, Information Sciences 329(1) (2016) 73-89.
[38] H. Samma, C. P. Lim, and J. M. Saleh, A new reinforcement learning-based memetic particle swarm optimizer, Applied Soft Computing 43 (2016) 276-297.
[39] R. Perez and K. Behdinan, Particle swarm approach for structural design optimization, Computers and Structures 85 (13-14) (2007) 1579-1588.
[40] F. Li, Y. Wang, X.-Y. Li, A. Nusairat, and Y. Wu, Gateway placement for throughput optimization in wireless mesh networks, Mobile Networks and Applications 13(1-2) (2008) 198-211.
[41] J. L. Doob, Stochastic Processes, Wiley, New York, 1953.
[42] H. A. Taha, Operations Research: An Introduction, Pearson/Prentice Hall Press, 2007.
[43] W.-L. Shen, C.-S. Chen, K. C. Lin, and K. A. Hua, Autonomous mobile mesh networks, IEEE Transactions on Mobile Computing 13(2) (2014) 364-376.

[44] K.-W. Chin, S. Soh, and C. Meng, A novel scheduler for concurrent Tx/Rx wireless mesh networks with weighted links, IEEE Communications Letters 16(2) (2012) 246-248.
[45] T. Oda, A. Barolli, E. Spaho, L. Barolli, F. Xhafa, and M. Younas, Effects of population size for location-aware node placement in WMNs: Evaluation by a genetic algorithm-based approach, Personal and Ubiquitous Computing 18(2) (2014) 261-269.
[46] B. Wang, X. Huan, L. T. Yang, and Y. Mo, Hybrid placement of Internet gateways and rechargeable routers with guaranteed QoS for green wireless mesh networks, Mobile Networks and Applications 20(5) (2015) 1-13.
[47] S. Mamechaoui, S. Senouci, F. Didi, and G. Pujolle, Energy efficient management for wireless mesh networks with green routers, Mobile Networks and Applications 20(5) (2015) 1-16.

Table 1. Notation used in the concerned problem.

| Notation | Definition |
|---|---|
| $A$ | Width of deployment area. |
| $B$ | Height of deployment area. |
| $\eta$ | Number of Internet gateways. |
| $w_i$ | The $i$-th Internet gateway, for $i \in \{1, 2, \ldots, \eta\}$. |
| $\omega_i$ | The radius $\omega_i$ of the coverage of the $i$-th Internet gateway, for $i \in \{1, 2, \ldots, \eta\}$ |
| $W$ | Set of $\eta$ Internet gateways, i.e., $W = \{w_1, w_2, \ldots, w_\eta\}$. |
| $n$ | Number of mesh routers. |
| $r_i$ | The $i$-th mesh router, for $i \in \{1, 2, \ldots, n\}$. |
| $\gamma_i$ | Radius of the coverage of the $i$-th mesh router, for $i \in \{1, 2, \ldots, n\}$. |
| $R$ | Set of $n$ mesh routers, i.e., $R = \{r_1, r_2, \ldots, r_n\}$. |
| $m$ | Number of mesh clients. |
| $c_i$ | The $i$-th mesh client for $i \in \{1, 2, \ldots, m\}$. |
| $C$ | Set of $m$ mesh clients, i.e., $C = \{c_1, c_2, \ldots, c_m\}$. |
| $U$ | Set of all nodes in the WMN, $U = W \cup R \cup C$. |
| $D(w_i)$ | Position of the $i$-th Internet gateway $w_i$ for $i \in \{1, 2, \ldots, \eta\}$, in which $D(w_i) \in [0, A] \times [0, B]$. |
| $D(W)$ | *Set of positions of all Internet gateways, i.e.,* $D(W) = \{D(w_1), D(w_2), \ldots, D(w_\eta)\}$ |
| $\Omega_i$ | The coverage area of Internet gateway $w_i$ (i.e., a circle centered at $D(w_i)$ with radius $\omega_i$). |
| $D_t(c_i)$ | Position of mesh client $c_i \in C$ at the $t$-th key time, in which $D_t(c_i) \in [0, A] \times [0, B]$. |
| $D(r_i)$ | Position of the $i$-th mesh router for each $i$,in which $D(r_i) \in [0, A] \times [0, B]$. |
| $D_t(R)$ | Positions of all mesh routers, i.e., $D_t(R) = \{D_t(r_1), D_t(r_2), \ldots, D_t(r_n)\}$. |
| $\Upsilon_i^t$ | The coverage area of mesh router $r_i$ at key time $t$ (i.e., a circle centered at $D_t(r_i)$ with radius $\gamma_i$). |
| $S_t$ | Set of the mesh clients that switch off their network access at the $t$-th key time. |
| $U_t$ | Set of nodes in the topology graph for the WMN at the $t$-th key time, i.e., $U_t = W \cup R \cup (C \setminus S_t)$. |
| $E_t$ | Set of edges in the topology graph for the WMN at the $t$-th key time. |
| $G_t$ | Topology graph $G_t = (U_t, E_t)$ for the WMN at the $t$-th key time. |
| $\delta(G_t)$ | *Network connectivity* corresponding to graph $G_t$. |
| $\phi(G_t)$ | *Client coverage* corresponding to graph $G_t$. |
| $M_h$ | Parameter used in the delay-hop constraint. |
| $M_r$ | Parameter used in the relay-load constraint. |
| $M_{wr}$ | Maximum number of mesh routers used in the gateway capacity constraint. |
| $M_{wc}$ | Maximum number of mesh clients used in the gateway capacity constraint. |

Table 2. Parameter setting used in the concerned problem and the proposed algorithm.

| Parameter | Value |
|---|---|
| Number of iterations between two key times in dynamic scenario | 15 |
| Number of dynamic clients between two key times | 0 (static), 1 |
| Probability that a mesh client switch on its network access when it is switched off | 0.01,0.50,0.90 |
| Probability that a mesh client switch off its network access when it is switched on | 0.01,0.50,0.50 |
| The maximal distance by which a mesh client can move | 10 |
| The weighting parameter $\lambda$ in fitness function | 0.3 |
| Maximal velocity $V_{\max}$ | 0.1 |
| Number of particles $K$ | 50 |
| Number of iterations $T$ | 150, 6000 |
| Parameter $c_1$ in updating velocity | 3 |
| Parameter $c_2$ in updating velocity | 2 |
| Parameter $L_o$ used in the *dispersing* local search operator | 1.5 |
| Parameter $L_c$ used in the *covermore* local search operator | 2 |
| Parameter $L_r$ used in the *random* local search operator | 4, 8, 16 |
| Parameter $M_h$ used in the delay-hop constraint | 2, 3, 5 |
| Parameter $M_r$ used in the relay-load constraint | 3, 4, 5 |
| Maximum number of mesh routers $M_{wr}$ used in the gateway capacity constraint | 5, 10, 20 |
| Maximum number of mesh clients $M_{wc}$ used in the gateway capacity constraint | 15, 32, 55 |

Table 3. Statistics of experimental results of running 20 times of four approaches for 10 instances of case 1.

| inst. | PSO with dispersing | | | | PSO with covermore | | | | PSO with random | | | | PSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | best | mean | worst | stdDev | best | mean | worst | stdDev | best | mean | worst | stdDev | best | mean | worst | stdDev |
| 1 | 1.000 | 0.980 | 0.962 | 0.013 | 1.000 | 0.999 | 0.981 | 0.004 | 1.000 | 0.991 | 0.981 | 0.010 | 0.923 | 0.886 | 0.832 | 0.023 |
| 2 | 0.971 | 0.943 | 0.923 | 0.012 | 1.000 | 0.980 | 0.962 | 0.009 | 0.981 | 0.961 | 0.942 | 0.010 | 0.923 | 0.848 | 0.807 | 0.032 |
| 3 | 0.923 | 0.890 | 0.865 | 0.015 | 0.962 | 0.935 | 0.923 | 0.010 | 0.942 | 0.920 | 0.904 | 0.010 | 0.813 | 0.778 | 0.745 | 0.018 |
| 4 | 0.981 | 0.956 | 0.942 | 0.012 | 1.000 | 0.985 | 0.981 | 0.008 | 0.995 | 0.978 | 0.962 | 0.007 | 0.813 | 0.876 | 0.846 | 0.022 |
| 5 | 0.981 | 0.948 | 0.923 | 0.016 | 1.000 | 0.985 | 0.981 | 0.007 | 0.981 | 0.966 | 0.947 | 0.009 | 0.904 | 0.853 | 0.798 | 0.021 |
| 6 | 0.947 | 0.925 | 0.904 | 0.014 | 1.000 | 0.977 | 0.962 | 0.012 | 1.000 | 0.961 | 0.942 | 0.016 | 0.870 | 0.830 | 0.803 | 0.020 |
| 7 | 0.947 | 0.911 | 0.885 | 0.017 | 0.981 | 0.967 | 0.952 | 0.009 | 0.962 | 0.950 | 0.942 | 0.009 | 0.880 | 0.824 | 0.779 | 0.028 |
| 8 | 0.981 | 0.962 | 0.942 | 0.011 | 1.000 | 0.991 | 0.981 | 0.009 | 0.986 | 0.979 | 0.962 | 0.007 | 0.923 | 0.888 | 0.861 | 0.018 |
| 9 | 0.981 | 0.959 | 0.942 | 0.013 | 1.000 | 0.985 | 0.967 | 0.010 | 0.986 | 0.978 | 0.962 | 0.007 | 0.918 | 0.880 | 0.856 | 0.016 |
| 10 | 0.981 | 0.954 | 0.938 | 0.011 | 1.000 | 0.995 | 0.981 | 0.008 | 1.000 | 0.983 | 0.967 | 0.008 | 0.933 | 0.889 | 0.857 | 0.016 |
| Mean | 0.969 | 0.943 | 0.923 | 0.013 | 0.994 | 0.980 | 0.967 | 0.008 | 0.983 | 0.967 | 0.951 | 0.009 | 0.901 | 0.855 | 0.818 | 0.021 |

Table 4. Statistics of the results after 2000 iterations using four PSO approaches under condition 1 (whose theoretical optimal fitness value = 0.5375) under 95% confidence interval.

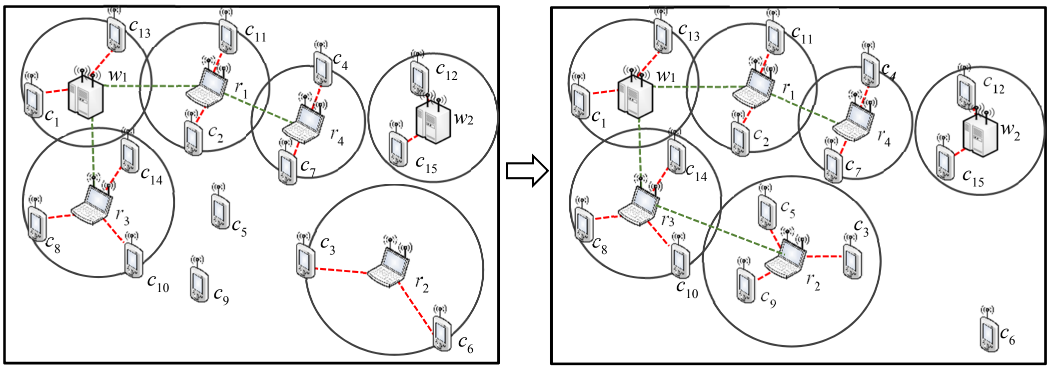
| | Mean | stdDev | Lower confidence bound | Upper confidence bound |
|---|---|---|---|---|
| PSO with dispersing | 0.5130 | 0.0486 | 0.4178 | 0.6083 |
| PSO with covermore | 0.5382 | 0.0594 | 0.4219 | 0.6546 |
| PSO with random | 0.5259 | 0.0554 | 0.4173 | 0.6345 |
| PSO | 0.3475 | 0.0540 | 0.2416 | 0.4534 |

Table 5. Statistics of the results after 2000 iterations using four PSO approaches under condition 2 (whose theoretical optimal fitness value = 0.5375) under 95% confidence interval.

| | Mean | stdDev | Lower confidence bound | Upper confidence bound |
|---|---|---|---|---|
| PSO with dispersing | 0.4691 | 0.0570 | 0.3575 | 0.5808 |
| PSO with covermore | 0.5236 | 0.0613 | 0.4035 | 0.6437 |
| PSO with random | 0.5068 | 0.0630 | 0.3833 | 0.6302 |
| PSO | 0.3433 | 0.0619 | 0.2219 | 0.4646 |

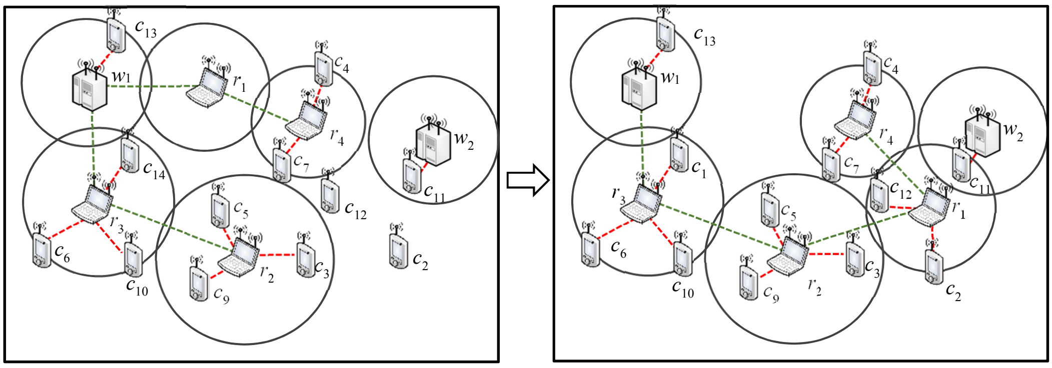
Table 6. Statistics of the results after 2000 iterations using four PSO approaches under condition 3 (whose theoretical optimal fitness value = 0.8458) under 95% confidence interval.

| | Mean | stdDev | Lower confidence bound | Upper confidence bound |
|---|---|---|---|---|
| PSO with dispersing | 0.7211 | 0.0512 | 0.6208 | 0.8215 |
| PSO with covermore | 0.8028 | 0.0476 | 0.7096 | 0.8960 |
| PSO with random | 0.7890 | 0.0495 | 0.6919 | 0.8860 |
| PSO | 0.4610 | 0.0675 | 0.3286 | 0.5934 |



(a) (b)

Fig. 1. Example of showing improvement when moving mesh router $r_2$ in a WMN with two Internet gateways.



(a) (b)

Fig. 2. Example of showing improvement of moving mesh router $r_1$ after some mesh clients have changes.

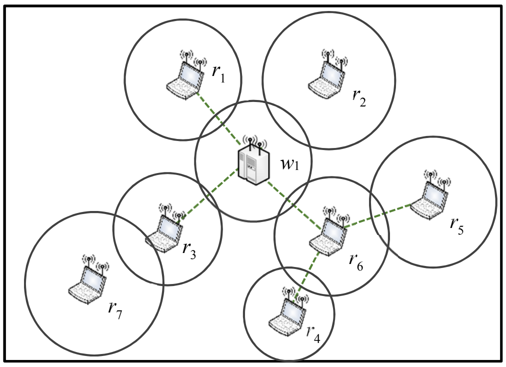
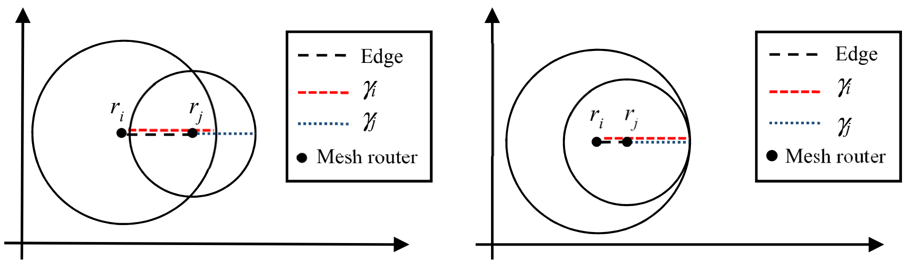



Fig. 3. Example of explaining three QoS constraints.


(a) Almost-overlapping router (b) Few-clients-covered router

Fig. 4. Two bad-quality placement problems.




(a) Partial-overlapped coverage (b) Total-overlapped coverage

Fig. 5. Two cases of overlapped coverage.

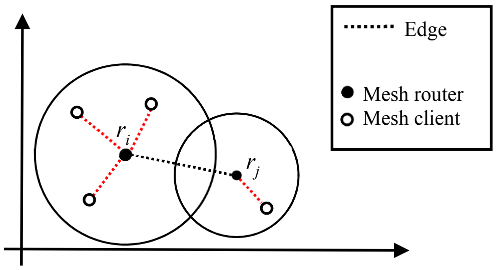



Fig. 6. An example used for explaining covermore local search operator.

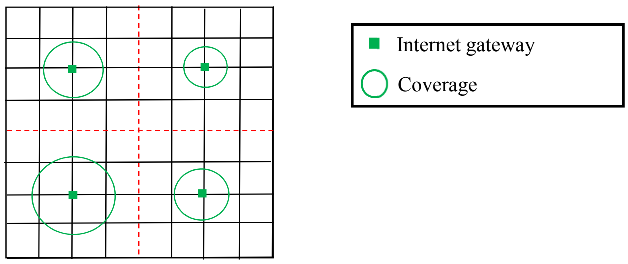



Fig. 7. Deployment of 4 Internet gateways in the experimental dataset.

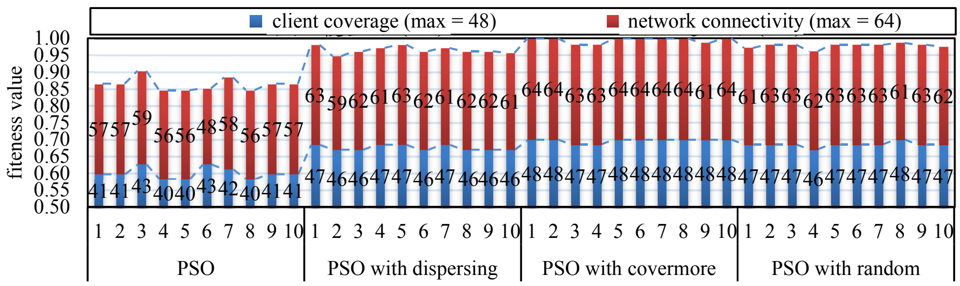



Fig. 8. Experimental comparison of four approaches for the 10 instances of case 1 in terms of two performance measures.

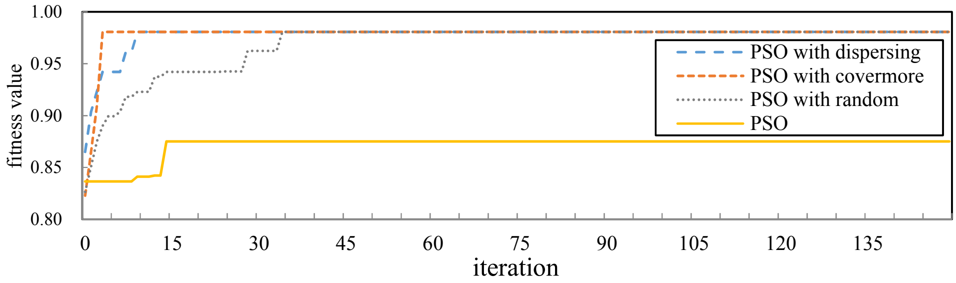



Fig. 9. Convergence analysis.

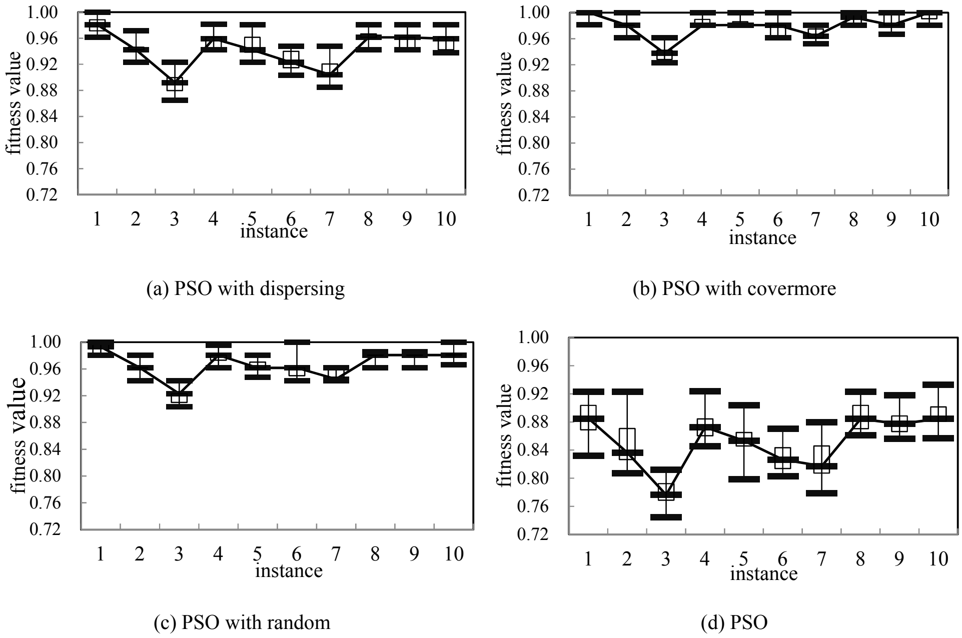



Fig. 10. Box plot of experimental results using four approaches for 10 instances of case 1.

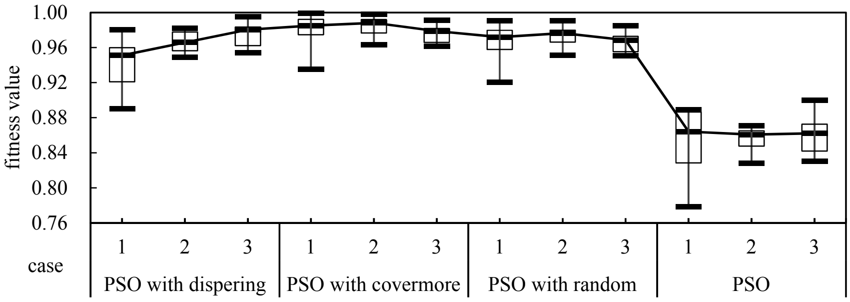
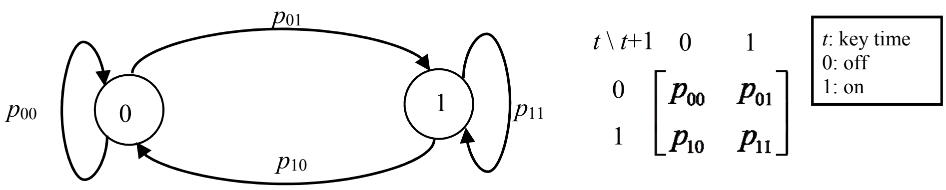
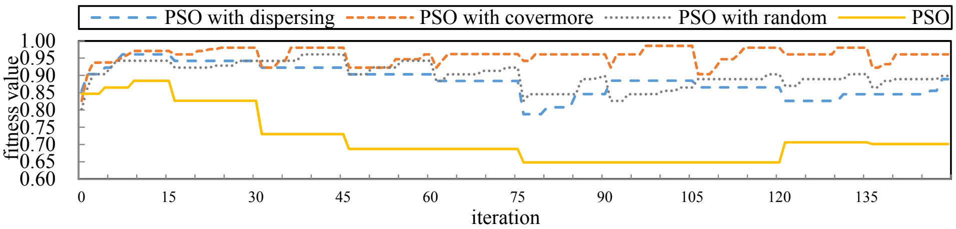



Fig. 11. Box plot of experimental results using four approaches for 3 cases.

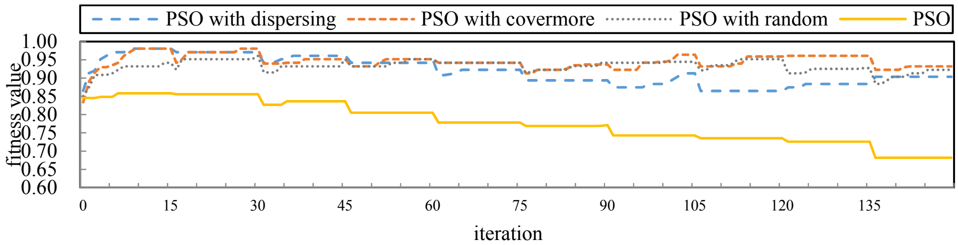



(a) (b)

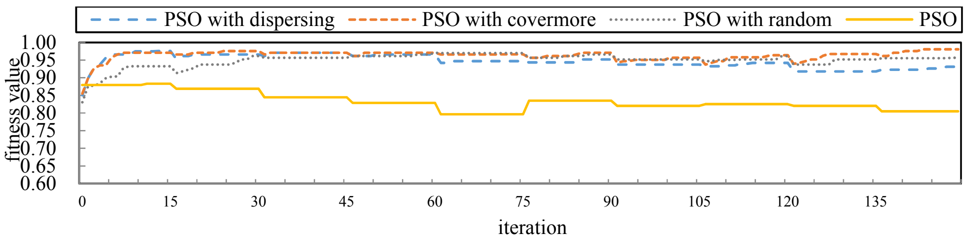
Fig. 12. Markov chain and transition matrix.




(a) Case 1




(b) Case 2




(c) Case 3

Fig. 13. Convergence analysis of four approaches for an instance of each case in the first dynamic scenario.

*t*: key time
0: off
1: on

$$\begin{array}{c|cc} t \setminus t+1 & 0 & 1 \\ \hline 0 & 0.99 & 0.01 \\ 1 & 0.01 & 0.99 \end{array}$$

(a)

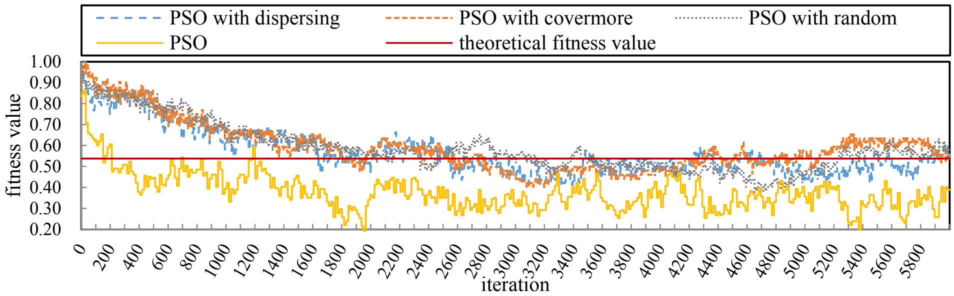
$$\begin{array}{c|cc} t \setminus t+1 & 0 & 1 \\ \hline 0 & 0.50 & 0.50 \\ 1 & 0.50 & 0.50 \end{array}$$

(b)

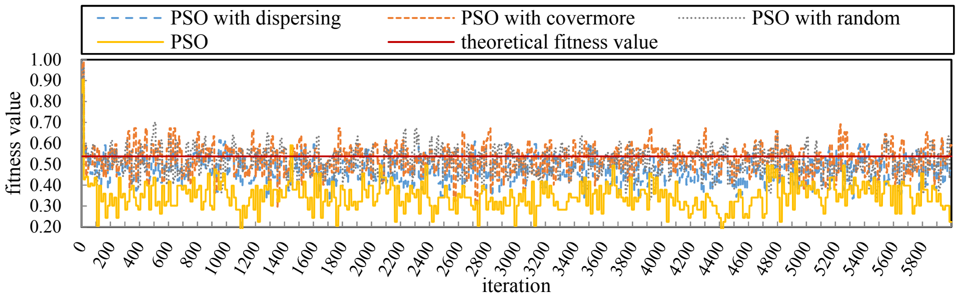
$$\begin{array}{c|cc} t \setminus t+1 & 0 & 1 \\ \hline 0 & 0.50 & 0.50 \\ 1 & 0.10 & 0.90 \end{array}$$

(c)

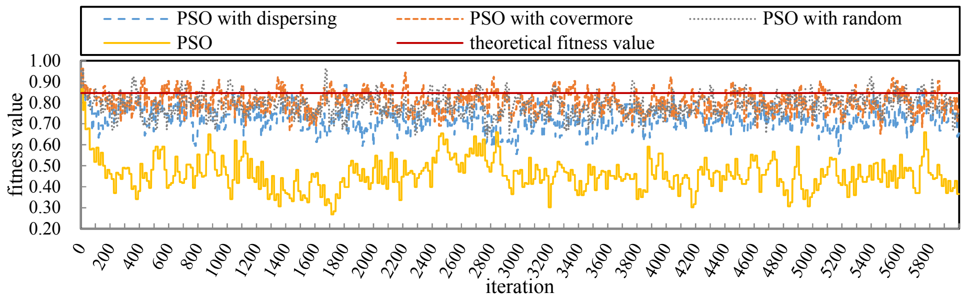
Fig. 14. Transition matrix of (a) condition 1, (b) condition 2, and (c) condition 3.




(a) Condition 1




(b) Condition 2




(c) Condition 3

Fig. 15. Experimental results of running four approaches on an instance of case 1 under three conditions for 6000 iterations (i.e., 400 key times) in the second dynamic scenario.

**Algorithm 1.** CONSTRUCT_GRAPH

**Input:** Position $X_{\kappa}^{\tau}$ of particle $\kappa$ at the $\tau$-th iteration, given coverages $\Omega_k$ of each Internet gateway $w_k$

**Output:** The corresponding network topology graph $G_{\kappa,\tau} = (U_{\kappa,\tau}, E_{\kappa,\tau})$

Based on $X_{\kappa}^{\tau}$, coverage $\Upsilon_{\kappa j}^{\tau}$ of each mesh router $r_j$ is calculated
Let $g_k$ be a subgraph with only a node (Internet gateway) $w_k$ for each $k \in \{1, \ldots, \eta\}$
Record that each mesh router is available to join a graph, and each mesh client is not served
**for** each Internet gateway $w_k$ **do**
  **for** each available mesh router $r_j$ with $\Omega_k \cap \Upsilon_{\kappa j}^{\tau} \neq \varnothing$ and adding $r_j$ to graph $g_k$ does not violate any QoS constraint **do**
    Add node $r_j$ and an edge between $r_j$ and $w_k$ to graph $g_k$
    Record that $r_j$ is a one-hop mesh router and is not available to join other graphs
    **for** each unserved mesh client $c_i$ with $D(c_i) \in \Upsilon_{\kappa j}^{\tau}$ and adding $c_i$ to $g_k$ does not violate any QoS constraint **do**
      Add node $c_i$ and an edge between $c_i$ and $r_j$ to graph $g_k$
      Record that mesh client $c_i$ is served
    **end for**
  **end for**
**end for**
$\mu \leftarrow 1$
**while** there is at least one $\mu$-hop mesh router **do**
  **for** each $\mu$-hop mesh router $r_i$ **do**
    **for** each available mesh router $r_j$ with $\Upsilon_i^{\tau} \cap \Upsilon_j^{\tau} \neq \varnothing$ and adding $r_j$ to the subgraph with $r_i$ does not violate any QoS constraint **do**
      Add node $r_i$ and an edge between $r_i$ and $r_j$ to the subgraph with mesh router $r_i$
      Record that $r_j$ is a $(\mu + 1)$-hop mesh router
      **for** each unserved mesh client $c_k$ with $D(c_k) \in \Upsilon_{\kappa j}^{\tau}$ and adding $c_k$ to the subgraph with $r_j$ does not violate any QoS constraint **do**
        Add node $c_k$ and an edge between $c_k$ and $r_j$ to subgraph with $r_j$
        Record that mesh client $c_k$ is served
      **end for**
    **end for**
  **end for**
  $\mu \leftarrow \mu + 1$
**end while**

**Algorithm 2.** PSO with local search ($t$-th key time)

**for** each $\kappa \in \{1, 2, \ldots, K\}$ **do**
  **if** $t > 0$ **then**
    $X_{\kappa}^{0} \leftarrow X_{\kappa}^{T}$, and evaluate its fitness $f(X_{\kappa}^{0})$
    $V_{\kappa}^{0} \leftarrow V_{\kappa}^{T}$
  **else**
    Initialize particle $\kappa$'s position $X_{\kappa}^{0} = (x_{\kappa 1}^{0}, \ldots, x_{\kappa(2n)}^{0})$ randomly where $x_{\kappa(2i-1)}^{0} \sim U(0, W)$ and $x_{\kappa(2i)}^{0} \sim U(0, H)$ for each $i \in \{1, 2, \ldots, n\}$, and evaluate its fitness $f(X_{\kappa}^{0})$
    Initialize particle $\tau$'s velocity $V_{\kappa}^{0} = (v_{\kappa 1}^{0}, \ldots, v_{\kappa(2n)}^{0})$ randomly where $v_{\kappa(2i-1)}^{0} \sim U(0, W)$ and $v_{\kappa(2i)}^{0} \sim U(0, W)$ for each $i \in \{1, 2, \ldots, n\}$
  **end if**
  $P_{\kappa}^{0} \leftarrow X_{\kappa}^{0}$ and $f(P_{\kappa}^{0}) \leftarrow f(X_{\kappa}^{0})$
  **if** $f(P_{\kappa}^{0}) > f(P^{*})$ **then**
    $P^{*} \leftarrow P_{\kappa}^{0}$ and $f(P^{*}) \leftarrow f(P_{\kappa}^{0})$
  **end if**
**end for**
**repeat**
  **for** each $\kappa \in \{1, 2, \ldots, K\}$ **do**
    Update particle $\kappa$'s velocity $V_{\kappa}^{\tau}$ by Eq. (7)
    For $i \in \{1, \ldots, n\}$, $v_{i}^{\tau}$ is truncated if violating Constraint (5)
    Update particle $\kappa$'s velocity $X_{\kappa}^{\tau}$ by Eq. (8)
    For $i \in \{1, \ldots, n\}$, $x_{i}^{\tau}$ is truncated if violating Constraints (3) and (4)
    Evaluate $f(X_{\kappa}^{\tau})$
    Generate a particle $\kappa'$ which is a clone from particle $\kappa$, and is conducted by one of *dispersing*, *covermore*, and *random* local search operators
    $X_{\kappa'}^{\tau} \leftarrow X_{\kappa}^{\tau}$ and evaluate $f(X_{\kappa'}^{t})$
    **if** $f(X_{\kappa'}^{t}) > f(X_{\kappa}^{t})$ **then**
      $X_{\kappa}^{\tau} \leftarrow X_{\kappa'}^{\tau}$ and $f(X_{\kappa}^{\tau}) \leftarrow f(X_{\kappa'}^{\tau})$
    **end if**
    **if** $f(X_{\kappa}^{\tau}) > f(P_{\kappa}^{\tau})$ **then**
      $P_{\kappa}^{\tau} \leftarrow X_{\kappa}^{\tau}$ and $f(P_{\kappa}^{\tau}) \leftarrow f(X_{\kappa}^{\tau})$
      **if** $f(P_{\kappa}^{\tau}) > f(P^{*})$ **then**
        $P^{*} \leftarrow P_{\kappa}^{\tau}$ and $f(P^{*}) \leftarrow f(P_{\kappa}^{\tau})$
      **end if**
    **end if**
  **end for**
**until** {the maximum iterations $T$ is reached or $f(P^{*})$ exceeds a threshold}
Output $P^{*}$ as the placement solution at the $t$-th key time