Social-aware Dynamic Router Node Placement in Wireless Mesh Networks

Chun-Cheng Lin · Pei-Tsung Tseng · Ting-Yu Wu · Der-Jiunn Deng*

Abstract The problem of dynamic router node placement (dynRNP) in wireless mesh networks (WMNs) is concerned with determining a dynamic geographical placement of mesh routers to serve mobile mesh clients at different times, so that both network connectivity (i.e., the greatest topology subgraph component size) and client coverage (i.e., the number of the served mesh clients) are maximized. Mesh clients are wireless devises associated with users, and in real world, the users with same interests or some social relationship have higher chance to gather and move together geographically, i.e., they form a community, and the WMN with multiple communities can be regarded as a social network. Therefore, this paper investigates the so-called social-aware WMN-dynRNP problem assuming that mesh routers should be aware of the social community structure of mesh clients to dynamically adjust their placement to improve network performance. To cope with this problem, this paper proposes a social-based particle swarm optimization approach, which additionally includes a social-supporting vector to direct low-loading mesh routers to support the heavy-loading mesh routers in the same topology subgraph component (community), so as to dynamically adopt to the social community behavior of mesh clients. As compared with the previous approach, our experimental results

C.-C. Lin · P.-T. Tseng · T.-Y. Wu
Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan
E-mails: cclin321@nctu.edu.tw, jacky.tpt@gmail.com, tingyuwu777@gmail.com

* D.-J. Deng (Corresponding author)
Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 500, Taiwan
E-mail: djdeng@cc.ncue.edu.tw

show that the proposed approach is capable of effectively reducing number of the unserved mesh clients and increasing network connectivity in dynamic social scenarios.

Keywords Social network · wireless mesh network · router node placement · community movement · particle swarm optimization

1 Introduction

Wireless mesh network (WMN) has been a popular wireless communications technology. Different from other communications technologies, e.g., GSM, satellite, 3G, and WLAN, WMNs combine characteristics of WLANs and ad hoc networks, and enjoy merits of high transmission bandwidth, low cost, and high mobility. They have various applications and provide last-few-miles solutions [1], [2], e.g., home broadband networks, community networks, enterprise network connectivity, building automation, wireless multimedia sensor network [3], [4], etc. WMNs are robust as nodes in the network have self-organizing and self-configuring ability, i.e., the WMN system can find other alternative routing paths for those disconnected mesh nodes [5].

This paper considers the WMNs composed of mesh routers and mesh clients. Mesh clients are wireless devices associated with users; mesh routers are access points that constitute the backbone of WMNs, and have the gateway function to connect to the outside Internet. A topology graph can be established as follows. Each mesh router provides a circular radio coverage region, and the mesh clients within the radio coverage region can connect to the Internet via multi-hop communication. Two mesh routers can communicate with each other if their radio coverage re-

gions are overlapped. As illustrated in Figure 1, the connection from a mesh router to the Internet is represented by a black solid line segment; the connection between a mesh router and a mesh client is represented by a red dotted line segment; the connection between two mesh routers is represented by a green dashed line segment. Note that if a mesh client exists in the overlapping range of radio coverage regions of two mesh routers, it chooses an arbitrary mesh router to communicate.

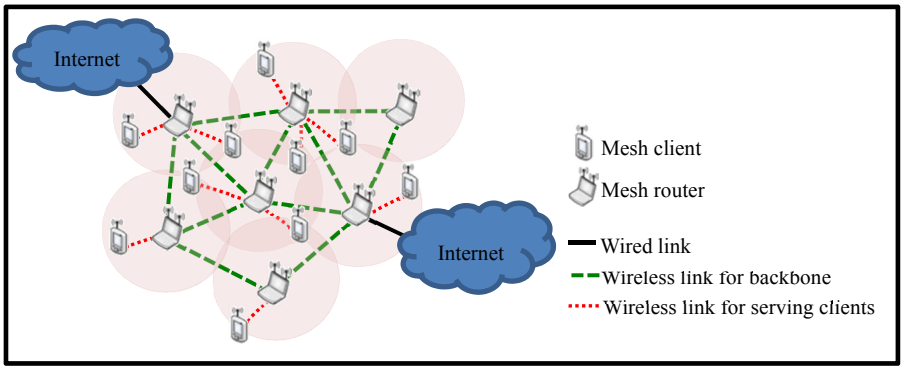


Fig.1. Illustration of a WMN topology.

This paper is concerned with the problem of router node placement (RNP) in WMNs [6], and this line of research has moved from static WMN scenarios [7], [8], [9], [10], [11] to the dynamic WMN scenarios [12]. Note that the RNP problem has been studied in other fields of optimization (e.g., facility planning and logistics), and the geographic concern in wireless networks has as well, e.g., [13]. In recent years, the RNP problem has been applied to wireless networks, e.g., sensor placement in wireless sensor networks [14], gateway placement in WMNs [15], etc. In WMNs, a bad-quality RNP could lead to unavoidable wireless interference as well as load unbalancing, in which some mesh routers have unexpected high service loading while the others have a lower utilization rate [16]. The ob-

jective of the WMN-RNP problem in this paper is to maximize the WMN connectively and the client coverage [17], which are size of the greatest topology subgraph component and number of the covered mesh clients, respectively.

Since the WMN-RNP problem has been shown to be NP-hard [14], [15], [18], it is commonly resolved by metaheuristic algorithms. A lot of previous works existed for static WMN-RNP problems. Oda et al. [7] proposed a genetic algorithm approach, and tested the influences of different mutation and selection schemes on different WMN sizes. Xhafa et al. [8] proposed a simulated annealing approach, and then Xhafa et al. [9] proposed a tabu search approach to avoid the solution search from falling into local optimal solutions. Chang et al. [10] proposed a hill-climbing algorithm. Our previous work in [11] further proposed a simulated annealing approach with momentum terms for the static WMN-RNP problem with mesh client priority constraint. Instead of considering the static WMN scenario, our another previous work in [12] started to investigate the metaheuristic algorithms for dynamic WMN-RNP problems (WMN-dynRNP for short), in which mesh clients and mesh routers have mobility; mesh clients move their positions at different times; mesh routers adjust their deployments accordingly to adapt the WMN topology change. The work in [12] further proposed a particle swarm optimization (PSO) approach for the WMN-dynRNP problem, and derived a theoretical convergence analysis.

This paper focuses on the so-called social-aware dynamic router node placement in WMNs (*social-aware WMN-dynRNP* for short). In practice, some mesh clients may form a social community relationship, in which the mesh clients with similar interest may have a high chance to gather or move together and frequently communicate with each other directly or indirectly, so that they form a social network [19]. As a result, this paper extends our previous work in [12] for the WMN-dynRNP problem to a more practical social network scenario in which some mesh

clients in the WMN form a community structure, so that the mesh clients in the same community gather and move together; additionally, each mesh router is restricted to serve only a restricted number of mesh clients (which were never considered before). Hence, the placement of mesh routers should be aware of the community structure to be adjusted at different times. Some previous works also made their extension in the similar way on different topics. For instance, Kim et al. [20] and Song et al. [21] simulated the movement behavior of real-human communities to conduct the related analysis on their respective problems.

To solve the social-aware WMN-dynRNP problem, if we continue using the original PSO approach in [12], then many mesh clients could not be served, because each single mesh router could not serve dense mesh clients due to their community structure. Additionally, if the placement of mesh routers is not adjusted accordingly in time to fit the community structure, performance could become worse as some communities move to their new locations at the next time step. From the literature, lots of previous works also applied characteristics of social networks to enhance the communication process among nodes, especially used for designing routing protocols and algorithms, e.g., [22], [23], [24], [25], [26]. Therefore, this paper also considers the factor of social community to improve the original PSO approach in [12] to propose the so-called *social-based PSO approach*, which includes an additional social-supporting vector where mesh routers in the same community can communicate with and support each other to make a rapid adjustment for possible community movement behaviors of mesh clients. By doing so, low-loading mesh routers tend to move to be closer to the heavy-loading mesh routers in the same community, so that those unserved mesh clients within the radio coverage regions of heavy-loading mesh routers have higher chance to be served at later time steps. To evaluation performance of the proposed approach, we consider the experimental dataset in [12] with two or three

communities of mesh clients and their moving trajectories. Experimental results show that in the dynamic scenario, the proposed social-based PSO approach is capable of effectively decreasing the total number of unserved mesh clients, and increasing network connectivity.

The organization of this paper is stated as follows. Section 2 describes the social-aware WMN-dynRNP problem in detail. Section 3 proposes a social-based PSO approach for the social-aware WMN-dynRNP problem. Section 4 gives the experimental design and results. Finally, Section 5 gives the conclusion of this work.

2 Problem Description

This section describes the problem of social-aware WMN-dynRNP mathematically.

2.1 Describing the Concerned Problem

For a WMN consisting of mesh routers and mesh clients deployed in a rectangular area, consider a more practical scenario where some of the mesh clients may gather and move together geographically to form a community, so that the WMN with multiple communities constitutes a social network with a community structure. Although a community structure underlies the distribution of mesh clients, it cannot be realized and determined due to its complexity and uncertainty. If the placement of mesh routers can be aware of the community structure of mesh clients to be adjusted dynamically, the performance would be increased. Hence, this paper investigates the so-called social-aware WMN-dynRNP problem, in which the RNP is capable of being aware of the social community movement behavior of mesh clients to dynamically determine their placement at different times, so as to maximize the network connectivity

and the client coverage. Especially, the mesh routers in the same topology subgraph component should be able to support each other to balance the service loading. Hence, the concerned problem is described as follows:

Social-aware dynamic router node placement in WMNs (Social-aware WMN-dynRNP): Consider a WMN deployed in a 2D rectangular area with size $W \times H$, consisting of n mesh routers and m mesh clients. Suppose that each mesh router can serve only a restricted number of mesh clients; and mesh clients form a community structure. At each time step, each mobile mesh client can arbitrarily change its position in the deployment area; some of the mesh clients form a community and the mesh clients of the same community could gather and move together. The problem is to be aware of the social community behavior of mesh clients to determine placement of mesh routers at each time step, and the objective of this problem is to maximize both the network connectivity and the client coverage simultaneously.

2.2 Notations

This paper continues using the notations in [12], except for those used in the service capacity constraint of each mesh router. All the notations used throughout this paper is given in Table 1.

At the t-th time step, the WMN with n mesh routers and m mesh clients deployed on a 2D area of size $W \times H$ can be represented as $U_t = R \cup C_t$ in which $R = \{r_1, r_2, \ldots, r_n\}$ is set of mesh routers and $C_t = \{c_1, c_2, \ldots, c_m\}$ is set of mesh clients. For $i \in \{1, 2, .., n\}$, each mesh router r_i is associated with a circular radio coverage region with radius γ_i. Note that the subscript t used in U_t and C_t is because mesh clients could switch off their network access at different

time steps in the dynamic scenario.

Table 1 Notations used in this paper.

Variable	Meaning
W	Width of the rectangular deployment area
H	Length of the rectangular deployment area
t	Index of time step
n	Number of mesh routers
m	Number of mesh clients
r_i	The i-th mesh router, for $i \in \{1, 2, \ldots, n\}$
c_i	The i-th mesh client, for $i \in \{1, 2, \ldots, m\}$
$R = \{r_1, r_2, \ldots, r_n\}$	Set of mesh routers
$C_t = \{c_1, c_2, \ldots, c_m\}$	Set of mesh clients at the t-th time step
$U_t = R \cup C_t$	Set of mesh nodes at the t-th time step
γ_i	Radius of the circular radio coverage range of mesh router r_i
β_i^t	Number of the mesh clients served by mesh router r_i at the t-th time step
$\beta_{\max}$	Upper bound of β_i^t
$D_t(c_i)$	Position of mesh client $c_i \in C_t$ on the deployment area at the t-th time step
$D_t(R) = \{D_t(r_1), \ldots, D_t(r_n)\}$	Set of positions of all mesh routers on the deployment area at the t-th time step
Υ_i^t	The circle centered at $D_t(r_i)$ with radius γ_i
E_t	Set of the links at the t-th time step
$G_t = (U_t, E_t)$	The network topology graph at the t-th time step
$G_t = G_t^1 \cup G_t^2 \cup \ldots \cup G_t^h$	There are h subgraph components $G_t^1, G_t^2, \ldots, G_t^h$ in G_t
$\phi(G_t)$	Network connectivity of topology graph G_t
$\psi(G_t)$	Client coverage of topology graph G_t
$X_k^t = (x_{k1}^t, x_{k2}^t, \ldots, x_{k(2n)}^t)$	Position of particle k on the solution space, for placement at the t-th time step
$V_k^t = (v_{k1}^t, v_{k2}^t, \ldots, v_{k(2n)}^t)$	Velocity of particle k, for placement at the t-th time step
$V_{\max}$	Maximal velocity
ω	Inertia weight to control influence of the velocity at the previous iteration
$f(X_k^t)$	Fitness of particle k, for placement at the t-th time step
$G_{t,k}$	Topology graph according to the placement represented by X_k^t
P_k^t	The best position found by particle k so far, for placement at the t-th time step
P^*	The best position found by all particles so far
$S_i^t = (s_{k1}^t, s_{k2}^t, \ldots, s_{k(2n)}^t)$	Social-supporting vector of particle k at the t-th iteration, for placement at the t-th time step

In what follows, we establish a topology graph based on the mesh nodes in U_t and the following two types of links. First, consider the links between mesh routers and mesh clients. Note that each mesh router has a radio coverage region. Different from the previous works, this paper assumes that each mesh router serves only a restricted

number of mesh clients. Hence, if a mesh client is located within the radio coverage region of some mesh router, and the mesh router does not exceed its restricted number of serving mesh clients, then the mesh client can be linked to the mesh router.

As for communication between two mesh clients, if two mesh clients are linked to the same mesh router, they can communicate with each other via the mesh router. On the other hand, if two mesh clients are linked to two different mesh routers that belong to the same topology subgraph component, they can communicate with each other via multi-hop communication. That is, if two mesh clients belong to two different subgraph components, or not both of them are linked to any mesh routers, they cannot communicate. Additionally, we assume a heterogeneous WMN scenario where the radio coverage region of each mesh router is of a different size.

Second, consider the links between mesh routers. If the radio coverage regions of two mesh routers are overlapped, the two mesh routers are linked and then belong to the same topology subgraph component. Note that the mesh routers in the same topology subgraph component can be aware of existence of each other, and hence, should be able to help and support each other, from the viewpoint of social community cooperation behavior.

One of the main differences of this work from the previous works is to consider that each mesh router serves only a restricted number of mesh clients. Throughout the rest of this paper, let β_i^t denote number of the mesh clients served by mesh router r_i at the t-th time step, and $\beta_{\max}$ denote its upper bound, i.e., $\beta_i^t \le \beta_{\max}$. Additionally, to be consistent with the practice, it is assumed that each mesh router can only be aware of whether $\beta_i^t \le \beta_{\max}$.

2.3 Modeling the Social-aware WMN-dynRNP Problem

After defining notations and definitions of mesh routers and mesh clients, this subsection describes the dynamic scenario of the social-aware WMN-dynRNP problem, in which mesh routers and mesh clients have mobility; mesh clients can change their positions at different times; some of the mesh clients belong to a community, and the mesh clients within the same community could gather and move together. After mesh clients change their positions, the placement of mesh routers in the social-aware WMN-dynRNP problem aims to be aware of the social community behavior of mesh clients to be adjusted periodically. Suppose that the period to determine placement of mesh routers is between two time steps. Hence, at each time step, mesh clients may form a different community structure, and the placement of mesh routers should base on the community structure to be adjusted to maximize the network performance.

The dynamic scenario can be described as follows. At the t-th time step, the position of each mesh client $c_i \in C_t$ on the 2D deployment area is denoted by $D_t(c_i) \in \mathbb{R}^2$. Based on the position distribution of mesh clients at each time step, positions of mesh routers on the deployment area are determined and denoted by $D_t(R) = \{D_t(r_1), D_t(r_2), \ldots, D_t(r_n)\}$. Note that the circle centered at $D_t(r_i)$ with radius γ_i is denoted by Υ_i^t. With the positions of mesh routers and mesh clients, we can establish a topology graph $G_t = (U_t, E_t)$, in which $U_t = R \cup C_t$; E_t is set of the links with two following types: First, for each pair of mesh routers $r_i, r_j \in R$, if $\Upsilon_i^t \cap \Upsilon_j^t = \varnothing$, then $(r_i, r_j) \in E_t$; second, for each mesh client $c_i \in C_t$ and each mesh router $r_j \in R$, if $D_t(c_i) \in \Upsilon_j^t$, then $(c_i, r_j) \in E_t$.

Note that the network topology graph G_t may not be connected, i.e., G_t includes multiple subgraph components. Suppose that h subgraph components $G_t^1, G_t^2, \ldots, G_t^h$ exist in G_t, so that $G_t = G_t^1 \cup G_t^2 \cup \ldots \cup G_t^h$ where $\forall i, j \in \{1,$

2, …, $h\}$, $i \neq j$, $G_t^i \cap G_t^j = \varnothing$. To increase the network connectively of the WMN topology, size of the greatest sub-graph component should be maximized so that it can serve as the backbone of the network to connect all mesh clients. However, a large greatest subgraph component size does not imply that it serves a large number of mesh clients, so we should also consider to serve mesh clients as many as possible.

The social-aware WMN-RNP problem considers to maximize the following two terms: *network connectivity* and *client coverage*, as described in [12]. To make the placement of mesh routers constitute a backbone for the whole network, the *network connectivity* at the t-th time step is defined as the size of the greatest subgraph component among the h components $G_t^1, G_t^2, ..., G_t^h$ in G_t, as expressed as follows:

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

If $\phi(G_t)$ is larger, then more mesh nodes are connected. Specifically, if

$$\phi(G_t) = m + n,$$

then it implies that all mesh nodes are connected.

If only the network connectivity is concerned, the mesh clients covered by other subgraph components may be neglected. Hence, the second term of our problem objective is to maximize the *client coverage* at the t-th time step expressed as follows:

$$\psi(G_t) = \sum_{i \in \{1,...,m\}} d_t(c_i),$$

where $d_t(c_i)$ is defined as follows:

$$d_t(c_i) = \begin{cases} 1, & \text{if mesh client } c_i \text{ at the } t\text{-th time step is served;} \\ 0, & \text{otherwise.} \end{cases}$$

3 The Proposed Social-based PSO Approach for the Social-aware WMN-dynRNP Problem

This section first gives the overview of the proposed social-based PSO approach to the social-aware WMN-dynRNP problem, and then the main components of the approach.

3.1 Algorithm of the Social-based PSO Approach

Particle swarm optimization (PSO) [27], [28] is a metahuristic algorithm that iteratively searches the global solution by imitating a number of particles (agents) that search an optimal solution (in terms of fitness values) on the solution landscape space. Each particle has its own position (a candidate solution on the solution space) at each iteration, and flies to a new position at a velocity at the next iteration. The velocity is updated at each iteration according to the best position found by each particle so far (i.e., the individual best experience of each particle) as well as the best position found by all particles so far (i.e., the global best experience). After a number of iterations, the final solution is generated if almost all particles arrive at the same position (solution).

The proposed social-based PSO approach to the social-aware WMN-dynRNP problem is based on our previous approach to the WMN-dynRNP problem in [12]. The main difference of the proposed approach from [12] is to include a social-supporting vector in the velocity updating formula which makes low-loading mesh routers tend to support the heavy-loading mesh routers in the same topology subgraph component. The algorithm of the proposed social-based PSO approach to determine the placement of mesh routers at the t-th time step is given as follows (see also the flowchart of the algorithm in Figure 2):

1) If time step index $t = 0$, then let each particle k's initial position X_k^0 be a random position (solution) within the solution landscape space, and each particle k's initial velocity V_k^0 be a random velocity within the feasible range; else (i.e., $t > 0$), let each particle k's position X_k^t and velocity V_k^t be their respective values at the previous iteration (i.e., X_k^{t-1} and V_k^{t-1}, respectively). Let the best position P_k^t found by each particle k so far be equal to X_k^t.

2) Evaluate each particle k's fitness value $f(X_k^t)$. According to those fitness values, find the position with the best fitness value, and let it be the best position found by all particles so far, denoted by P^*.

3) Repeat the following steps until the maximal number of iterations is reached.

   a) Use each particle k's social-supporting vector S_k^t to update its velocity V_k^t and position X_k^t according to Equations (1) and (2), respectively. Note that the two vectors must satisfy their respective range constraints.

   b) Evaluate each particle k's fitness value $f(X_k^t)$. According to those new fitness values, update the best positions found by each particle k and all particles so far, i.e., P_k^t and P^*, respectively.

   c) Update each particle's social-supporting vector S_k^t.

The above algorithm is explained as follows. Note that each particle's position represents a candidate solution of the concerned problem, i.e., the placement of mesh routers on the deployment area. At Step 1) of the algorithm, since the concerned problem is to determine the placement of mesh routers at different times, two cases for the initial position of each particle are considered according to the current time step. If the initial time step (i.e., $t = 0$) is con-

cerned, then the initial solution and velocity of each particle are set randomly within the restricted ranges; else (i.e., the positions and velocities at the previous time step exist), the two vectors are set to inherit their respective values at the previous time step.

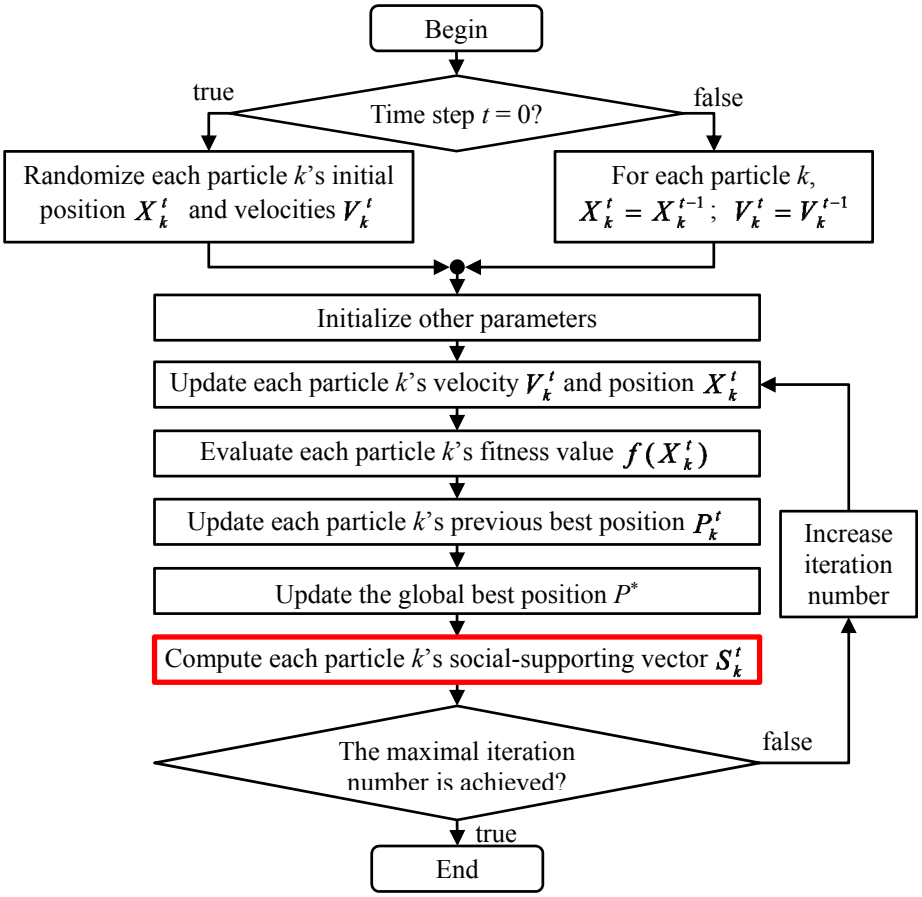


Fig. 2. Flowchart of the proposed social-aware PSO approach.

At Step 2), the fitness value is used to evaluate the performance of each particle according to its position (i.e., placement of mesh routers), and then the best position found by all particles so far is recorded. Next, Step 3) repeats a loop until the maximal number of iterations is achieved. In the loop, Step a) updates each particle's position and ve-

locity according to Equations (1) and (2), which will be explained later. Note that Equation (1) includes a social-supporting vector, which is the main difference from the previous work. Next Steps b) and c) update each particle's best positions respectively found by its own, all particles, and its social-supporting vector.

3.2 Main Components of the Social-based PSO Approach

Based on the social-based PSO algorithm detailed in the previous subsection, this subsection gives the main components of the algorithm.

3.2.1 Solution Representation

The solution representation in this work is the same as [12]. Position of each particle represents a placement of mesh routers (i.e., (x, y)-coordinates of n mesh routers) on a 2D deployment area of size $W \times H$, whose bottom-left corner is placed at the origin of the xy plane. Hence, for each particle k, position of particle k at the t-th time step is encoded as $X_k^t = (x_{k1}^t, x_{k2}^t, ..., x_{k(2n)}^t)$, in which $(x_{k(2i-1)}^t, x_{k(2i)}^t)$ is the (x, y)-coordinate of mesh router r_i, $\forall i \in \{1, 2, \ldots, n\}$; $0 \leq x_{k(2i-1)}^t \leq W$; $0 \leq x_{k(2i)}^t \leq H$. Let P_k^t and P^* denote the best positions found of each particle k and all particles so far, respectively. Also let $f(X)$ be the fitness value of position X. Hence, $f(X_k^t)$, $f(P_k^t)$, and $f(P^*)$ are the fitness values of X_k^t, P_k^t, and P^*, respectively.

3.2.2 Updating the Particle Velocity and Position

At each iteration, each particle k at the t-th time step moves its position X_k^t at a velocity $V_k^t = (v_{k1}^t, v_{k2}^t, ..., v_{k(2n)}^t)$ with $-V_{\max} \leq v_i^t \leq V_{\max}$, $\forall i \in \{1, 2, \ldots, 2n\}$, where $V_{max} \leq \max\{W, H\}$. Before movement, velocity V_i^t is updated according to the following formula:

$$V_k^{t'} = \omega \cdot V_k^t + w_1 \cdot e_1 \cdot (P_k^t - X_k^t) + w_2 \cdot e_2 \cdot (P^* - X_k^t) + w_3 \cdot e_3 \cdot S_k^t , \quad (1)$$

where $V_k^{t'}$ denotes the updated velocity with four sources: the original velocity V_k^t, the direction toward the best position P_k^t found by particle k so far, the direction toward the best position P^* found by all particles so far, and the social-supporting vector S_k^t of particle k, which will be explained later; parameter ω is the inertia weight to control the influence of V_k^t; parameters w_1, w_2, and w_3 are used to control the scales of the later three terms in this formula; e_1, e_2, and e_3 are three random floating numbers between 0 and 1. Note that ω was proposed first in [29], and in general, it decreases linearly from 0.9 to 0.4 as number of iterations grows. Then, the work in [30] proposed a random inertia weight. Since the dynamical positions of mesh clients cannot be predicted in the concerned problem, the random inertia coefficient is more suitable for our approach.

After updating the velocity $V_k^{t'}$, the new position $X_k^{t'}$ of particle k at the t-th time step can be obtained by the following formula:

$$X_k^{t'} = X_k^t + V_k^{t'} . \quad (2)$$

3.2.3 Fitness Evaluation

Fitness value is used to evaluate performance of a particle's position, and the PSO approach aims to iteratively improve the fitness value of each particle. The fitness evaluation in this work is the same with [12], which aims to simultaneously maximize the network connectivity and the client coverage as follows:

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

where $f(X_k^t)$ is the fitness value of the position X_k^t of particle k at the t-th time step; $G_{t,k}$ is the topology graph according to the placement of mesh routers represented by X_k^t; $\phi(G_{t,k})$ and $\psi(G_{t,k})$ are the network connectivity and the client coverage based on $G_{t,k}$, respectively; the denominator of each term is used for normalization; λ is a floating number between 0 and 1, which is used for balancing the two terms. Note that the previous works in [7], [8], [9], [10] were two-stage methods for static RNP problems, which maximize the first term and then the second term. The concerned problem focuses on a dynamic scenario, which is generally resolved by simultaneously maximizing the two terms [11], [12].

3.2.4 Social-Supporting Vector

In the social-aware WMN-dynRNP problem, each mesh router serves only a restricted number of mesh clients. Recall that β_i^t denotes the number of mesh clients served by mesh router r_i at the t-th time step, and β_{max} denotes the maximal number of mesh clients that each mesh router can serve. Hence, we have $\beta_i^t \le \beta_{\max}$.

Given a placement of mesh routers, consider the case where a lot of mesh clients are located densely within the radio coverage region of a mesh router (i.e., one dense community may exist in this region), and the total number of those mesh clients is greater than β_{max}. In this case, the mesh router is aware that a community of mesh clients exists in the neighborhood of the coverage region, but has no enough capacity to serve all of those mesh clients. To make use of the awareness, the proposed social-based PSO approach includes a social-supporting vector S_i^t in the velocity updating formula of V_i^t in Equation (1), which makes the mesh routers in the same topology subgraph component

tend to support the mesh router that is aware of a dense community. Note that the reason why we only make use of the mesh routers in the same topology subgraph component is that the other components cannot communicate with the concerned component and hence cannot support it.

Let $S_i^t = (s_{k1}^t, s_{k2}^t, ..., s_{k(2n)}^t)$ where $(s_{k(2i-1)}^t, s_{k(2i)}^t)$ is the movement of mesh router r_i for supporting other mesh routers for each $i \in \{1, 2, \ldots, n\}$. The algorithm of calculating S_k^t is stated as follows:

1) Consider each mesh router, say r_i, that does not serve any mesh client.

2) Arbitrarily select a mesh router r_j that serves no less than β_{max} mesh clients in the same topology subgraph component with mesh router r_i (i.e., it is aware that a community may exist in its neighborhood).

3) The entries $(s_{k(2i-1)}^t, s_{k(2i)}^t)$ for mesh router r_i in the social-supporting vector S_k^t are calculated as follows:

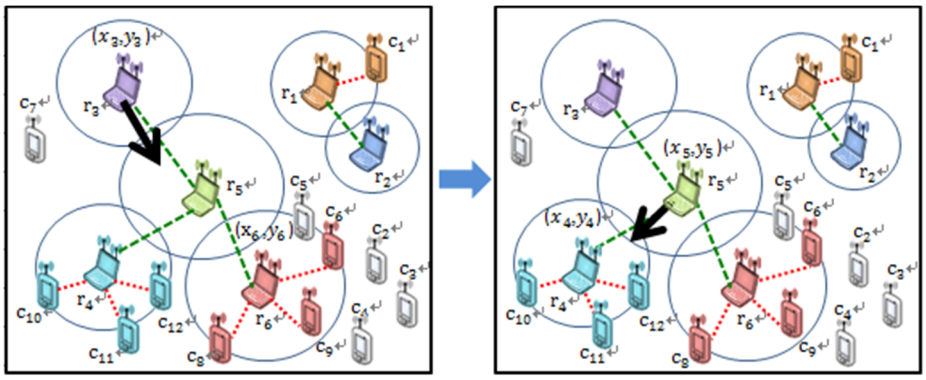
   $(s_{k(2i-1)}^t, s_{k(2i)}^t) = (x_{k(2j-1)}^t, x_{k(2j)}^t) - (x_{k(2i-1)}^t, x_{k(2i)}^t)$.

   That is, mesh router r_i will tend to move closer to the position of mesh router r_j.

4) After considering all the mesh routers at Step 1), the final social-supporting vector S_k^t is obtained. Note that each pair (0, 0) in this vector means that its corresponding mesh router does not support any mesh router.

Note that in Step 1) of the above algorithm, if there is no mesh router that does not serve any mesh client, then no social support is generated from the remaining steps of the algorithm. However, it is hard to decide whether to move some mesh router that has served some number of mesh clients, because those served mesh clients may lose connection of the mesh router. Hence, it would be of interest to design a cost function that evaluates whether to move a mesh router that has served some number of mesh clients.

To clarify the above algorithm, a simple numerical example is given as follows. Consider the topology graph G_t at the t-th time step as illustrated in Figure 3(a), in which there are 6 mesh routers r_1, r_2, …, r_6 and 12 mesh clients c_1, c_2, …, c_{12}; the links between mesh routers and mesh clients are marked by red dotted line segments; the links between mesh routers are marked by green dashed line segments. Suppose that each mesh router serves at most three mesh clients (i.e., $\beta_{\max} = 3$). Hence, mesh routers r_4 and r_6 in Figure 3(a) serve no less than 3 mesh clients, i.e., each of them is aware that a community may exist in its neighborhood.



(a) (b)

Fig. 3. A numerical example for illustrating the social-supporting vector.

Step 1) of the above algorithm finds the mesh routers that do not serve any mesh clients, i.e., r_2, r_3, and r_5. Since there is no heavy-loading mesh router in the topology subgraph component that mesh router r_2 belong to, it suffices to consider mesh routers r_3 and r_5. Suppose that Step 2) of the above algorithm selects mesh routers r_6 and r_4 respectively for mesh routers r_3 and r_5. Let (x_i, y_i) denote the (x, y)-coordinate of mesh router r_i for each $i \in \{1, 2, \ldots, 6\}$. Hence, Step 3) of the above algorithm calculates the entries for mesh r_3 and r_5 in the social-supporting vector as

$(x_6 - x_3, y_6 - y_3)$ and $(x_4 - x_5, y_4 - y_5)$, respectively. Hence, Step 4) of the above algorithm obtains the final social-supporting vector as follows:

$$S_k^t = (0, 0, 0, 0, x_6 - x_3, y_6 - y_3, 0, 0, x_4 - x_5, y_4 - y_5, 0, 0).$$

3.3 Complexity analysis of the proposed algorithm

In order to analyze the space and time complexity of the proposed algorithm, some notations are concerned as follows. Recall that n and m denote numbers of mesh routers and mesh clients, respectively. Let η be number of particles applied in the social-based PSO algorithm, and τ be number of iterations of the main loop of the algorithm.

Consider to analyze the space complexity of the proposed social-based PSO algorithm. The proposed algorithm works on η particles. Each particle keeps a position vector X_k^t, a velocity vector V_k^t, its previous best positions P_k^t, and its social-supporting vector S_i^t; and the whole particle swarm keeps the global best solution P^*. Also, each vector is of length $2n$. Hence, the space complexity of recording all particles' data structures is $O(\eta \cdot (4 \cdot 2n) + 2n) = O(\eta \cdot n)$. To evaluate fitness of each particle, the graph topology of each particle (Subsection 2.3) is recorded for convenience of computation. Each graph topology consists of links between mesh routers and mesh clients (stored in $O(m \cdot n)$ space) and links between mesh routers (stored in $O(n^2)$). Hence, the total space complexity of the algorithm is $O(\eta \cdot n + \eta (m \cdot n + n^2)) = O(\eta (m + n) n)$.

Consider to analyze the time complexity of the main algorithm (Subsection 3.1). Initialization of all parameters in Step 1) is done in time $O(\eta \cdot n)$ because length of each parameter vector is at most $2n$ and some parameters are set at most η times for all particles. At Step 2), to evaluate fitness of each particle, a topology graph corresponded to the

position of each particle is established in $O((m + n)\ n)$ time as analyzed in the above space complexity analysis. Since network connectivity $\phi(G_{t,k})$ and client coverage $\psi(G_{t,k})$ can be computed linearly in side of the topology (Subsection 2.3), the fitness value of a particle can be computed in $O((m + n)\ n)$ time. Therefore, Step 2) is done in $O(\eta\ (m + n)\ n)$ time.

Before analyzing Step 3) of the algorithm in Subsection 2.3, consider to analyze the algorithm of calculating a social-based supporting vector in Subsubsection 3.2.4. Step 1) considers each mesh router r_i, and hence, this algorithm has n iterations. For each iteration, Step 2) finds the neighboring mesh router r_j in $O(n)$ time, because the data structure of the graph topology has recorded the adjacency relationship of links during establishing topology. Step 3) is done in $O(1)$ time. Step 4) just outputs the final result. Hence, the total time complexity of the algorithm of calculating a social-based supporting vector in Subsubsection 3.2.4 is $O(n^2)$.

Now consider Step 3) of the main algorithm in Subsection 2.3. Substep a) updates all particles' velocities and positions in $O(\eta \cdot n)$ time. Substep b) is done in $O(\eta\ (m + n)\ n)$ time as the analysis for Step 2). Since a social supporting vector is calculated in $O(n^2)$ time, Substep c) calculated all particles' social supporting vectors in $O(\eta\ n^2)$ time. As a result, Step 3) of the main algorithm considers τ iterations, so is done in $O(\tau\ (\eta\ n + \eta\ (m + n)\ n + \eta\ n^2)) = O(\tau\ \eta\ (m + n)\ n)$ time.

With the above time complexity analysis, the proposed social-based PSO algorithm is computed in $O(\tau\ \eta\ (m + n)\ n)$ time.

4 Experimental Design and Results

Performance of the proposed social-based PSO approach to the social-aware WMN-dynRNP problem is evaluated in this section, and compared with the previous PSO approach in [12]. This section first describes the experimental data and the experimental design on dynamic scenarios, and then gives the experimental results under various scenarios.

4.1 Experimental Data

We continue using the experimental datasets in [12], including the following three different-scale network cases, each of which has 10 instances:

- Small-scale case: Consider a 32 × 32 deployment area (i.e., $W = H = 32$). The problem is to place 16 mesh routers with circular radio coverage regions with radii ~ $U(3, 6)$ to serve 48 mesh clients.
- Middle-scale case: Consider a 64 × 64 deployment area. The problem is to place 32 mesh routers with circular radio coverage regions with radii ~ $U(4\sqrt{2}-2, 8\sqrt{2}-2)$ to serve 96 mesh clients.
- Large-scale case: Consider a 128 × 128 deployment area. The problem is to place 64 mesh routers with circular radio coverage regions with radii ~ $U(7, 14)$ to serve 192 mesh clients.

The simulation is implemented in C++ programming language, and the parameter settings are given in Table 2, in which most of the parameter values continue using those in [12].

Note that the setting of the λ value has been discussed in our previous work in [12], which was decided by not only lots of experimental trials but also visualization of the WMN configurations (see also Fig. 4 in [12]). The work

in [12] found that a too large λ value leads to a deployment where mesh routers are too dense in some regions so that number of the served clients is small; while a too small λ value cannot lead to a large topology component. Hence, the λ value is set to 0.3 finally. Also note that number of the iterations between two time steps in the dynamic scenario is 30, i.e., mesh clients change at each time step (at each 30 iterations), and the placement of mesh routers is adjusted to adapt to this change.

Table 2 Parameter setting.

Parameter	Value
Number of iterations in the static scenario	90
Number of iterations between two time steps in the dynamic scenario	30
The total number of iterations in the dynamic scenario for small-scale network case	$30 \cdot 16$
The total number of iterations in the dynamic scenario for middle-scale network case	$30 \cdot 31$
The total number of iterations in the dynamic scenario for large-scale network case	$30 \cdot 55$
The farthermost distance to which a mesh client can move in the dynamic scenario	3
The maximal number of mesh clients that a mesh router can serve (i.e., $\beta_{\max}$)	3
The weighting parameter that controls the two terms in the objective λ	0.3
Parameter w_1 in Equation (1)	2
Parameter w_2 in Equation (1)	2
Parameter w_3 in Equation (1)	1
The maximal velocity $V_{\max}$ of each particle	$0.1\ W$
The total number of particles	100

4.2 Experimental Design of Dynamic Scenarios for Social Communities

To test whether the proposed approach can cope with the dynamic behavior of social communities of mesh clients, we design three dynamic scenarios with social community behavior in the experiments: *simplified dynamic scenario*, *generalized static scenario*, and *generalized dynamic scenario*, in which the term *simplified* means that each mesh client belongs to a community; the term *generalized* means that except for those mesh clients that belong to communities, the others are scattered on the deployment area; the term *static* means that all mesh clients are static, while the

term *dynamic* means that each mesh client could move its position or switch its network access as time goes. For simplicity, we only consider the three scenarios, but they suffice to analyze performance of the proposed approach.

Additionally, two or three communities are considered in each scenario. Hence, considering 3 network scales (i.e., small, middle, large), 2 possible community numbers (i.e., 2 or 3), and 3 scenarios (i.e., simplified dynamic, generalized static, and generalized dynamic scenarios), there are $3 \times 2 \times 3 = 18$ combinations of experimental settings in total. The number of mesh clients in each community in different-scale network cases is given in Table 3.

Table 3 The number of mesh clients in each community in different-scale network cases.

Small-scale network case				
Num. of Communities	2	2 (simplified)	3	3 (simplified)
Num. of mesh clients in Community #1	16	24	12	16
Num. of mesh clients in Community #2	16	24	12	16
Num. of mesh clients in Community #3	0	0	12	16
Other scattered mesh clients	16	0	12	0
Total mesh clients	48	48	48	48
Middle-scale network case				
Num. of Communities	2	2 (simplified)	3	3 (simplified)
Num. of mesh clients in Community #1	32	48	24	32
Num. of mesh clients in Community #2	32	48	24	32
Num. of mesh clients in Community #3	0	0	24	32
Other scattered mesh clients	32	0	24	0
Total mesh clients	96	96	96	96
Large-scale network case				
Num. of Communities	2	2 (simplified)	3	3 (simplified)
Num. of mesh clients in Community #1	64	96	48	64
Num. of mesh clients in Community #2	64	96	48	64
Num. of mesh clients in Community #3	0	0	48	64
Other scattered mesh clients	64	0	48	0
Total mesh clients	192	192	192	192

In what follows, the dynamic design of communities is explained in detail. Consider two types of mesh clients: the mesh client of type 1 must belong to some community and hence move together with the community; the mesh client of type 2 does not belong to any community and hence can move arbitrarily on its own. Hence, the *simplified* scenario is defined to include only the first type of mesh clients, while the *generalized* scenario is defined to include

both of the two types of mesh clients. For example, a generalized scenario is illustrated in Figure 4, which includes two communities and some scattered mesh clients on the deployment area.

Next, consider the community movement in the dynamic scenario. Take Figure 4 for an example in the generalized dynamic scenario with two communities, in which the moving trajectory of each community is shown.

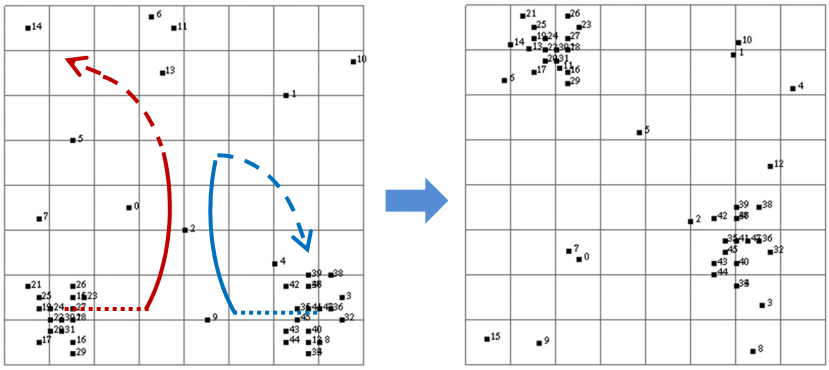


Fig. 4. The moving trajectories of two communities.

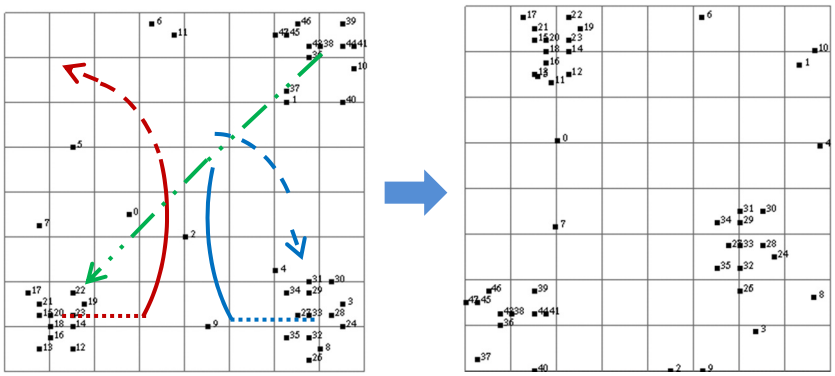


Fig. 5. The moving trajectories of three communities.

Our dynamic design for this example includes four social community behaviors: first, the two communities move to gather together and hence are merged into a single super community (i.e., the trajectories shown by dotted

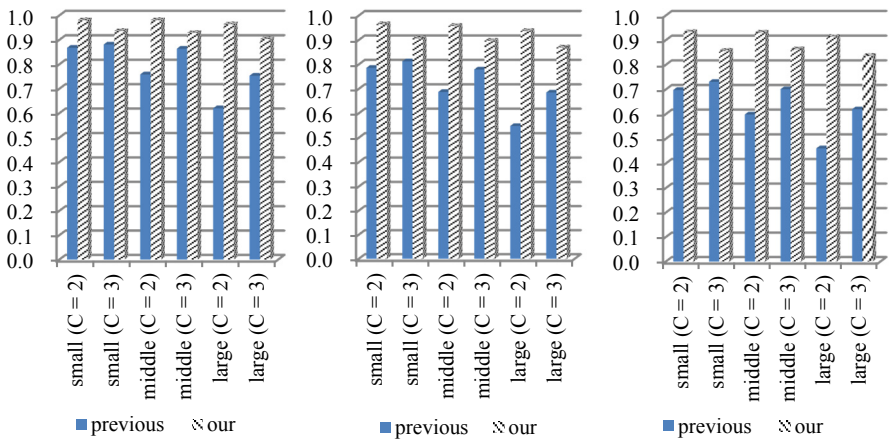
curves in Figure 4); then, the super community moves together for a while (i.e., the trajectories shown by solid curves in Figure 4); then, the super community is divided into the original two communities that keep moving on their own (i.e., the trajectories shown by dashed curves in Figure 4); finally, all the mesh clients in the two respective communities are disbanded (i.e., they do so at the arrow marks in Figure 4). Those social community behaviors are obtained by imitating the human behaviors in real world. Figure 5 is an example in the generalized dynamic scenario with three communities, in which the third community moves from the upper-right corner to the lower-left corner and finally is disbanded.

Although our dynamic design for community movement is artificial, some notable works on social networks (e.g., see [31]) also applied artificial dynamics in their dynamic design. In fact, investigating real-world dynamic traces in the dynamic design suffices to form another full paper (including user survey and behavior validation) and is complicated. It is out of the scope of this paper. Hence, a future line of the research is to investigate whether the proposed approach can handle real-world dynamic traces in the dynamic design in the future work.

4.3 Experimental Results in the Simplified Dynamic Scenario

The purpose to conduct the experiments in the simplified dynamic scenario (i.e., each mesh client belongs to a community) is to obtain the basic test for whether our proposed social-based PSO approach can make more efficient and effective adjustment for such a community structure than the previous PSO approach in [12]. Note that there is no previous work that is directly related to this work except for [12], and hence, only the work in [12] is compared experimentally with this work. We conduct the experiments of three different-scale datasets in the simplified dynamic

scenario with two or three communities by using our proposed approach and the previous approach in [12]. Note that each dataset contains 10 instances, and each statistical value is obtained by averaging 20 times of running the experiments on the 10 instances. After comparing all experimental statistics (the best fitness value, the average fitness value, the worst fitness value, and the standard deviation of fitness values), our proposed approach always performs better than the previous approach. Comparisons of the best, average, and worst fitness values using our and the previous approaches in the simplified dynamic scenario are given in Figure 6(a)–(c), respectively, in which the horizontal axis lists all different-scale cases and two possible community numbers (denoted by *C*). From Figure 6, our proposed approach performs better than the previous approach in terms of the three statistics of fitness values in all cases.



(a) Best fitness (b) Average fitness (c) Worst fitness

Fig. 6. Comparison of the fitness values using our and the previous approaches in the simplified dynamic scenario.

To analyze the reason why our proposed approach performs better in the simplified dynamic scenario, we find that the previous approach in [12] does not utilize the social relationship of mesh routers in the same topology sub-

graph component, and hence requires more iterations to adjust its placement of mesh routers to adapt to the community structure of mesh clients; furthermore, before finishing the adjustment to adapt to the community structure, some communities may move to their next positions at the next time step in the dynamic scenario, and hence, the performance of the previous approach becomes worse. On the other hand, our proposed new social-based approach includes a social-supporting vector to make effective adjustment to adapt to the community structure.

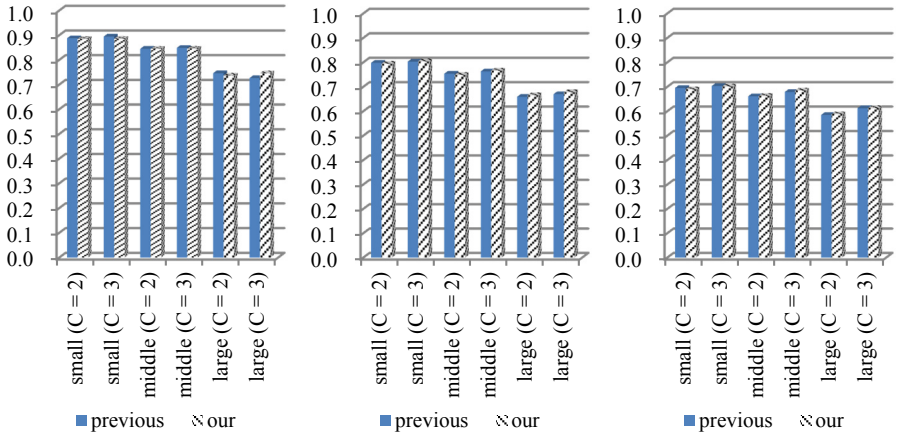
4.4 Experimental Results in the Generalized Static Scenario

The experimental results in the previous subsection have shown that our proposed approach performs better in the simplified dynamic scenario. However, in real world, aside from communities, scattered mesh clients exist on the deployment area (i.e., they do not belong to any community). Hence, this paper further analyzes the experimental results in a generalized scenario. Note that although this paper focuses on the dynamic scenario, it is of interest in this subsection to first analyze whether our proposed approach also performs better in the generalized static scenario.

The comparisons of the best, average, and worst fitness values using our and the previous approaches in the generalized static scenario are given in Figure 7(a)–(c), respectively.

From Figure 7, the difference of all fitness statistics using our and the previous approaches looks almost no different. Hence, the independent-sample t-test is conducted to test whether performance of the two approach has remarkable difference. In the test, we select the experimental setting with the most representative average fitness value and the most complex dataset (i.e., the large-scale network case and three communities). Under the 95% confidence interval, the p-value of the t-test is $0.197 > 0.05$. That is, the average fitness values using the two approaches

do not have remarkable difference. As a result, we conclude that our social-based PSO approach mainly focuses on the dynamic scenario, and hence, the experimental results in this subsection confirm that even if scattered mesh clients exist on the deployment area, our approach still keeps as good performance as the original PSO approach, so that we can further analyze the experimental results in the generalized dynamic scenario.



(a) Best fitness (b) Average fitness (c) Worst fitness

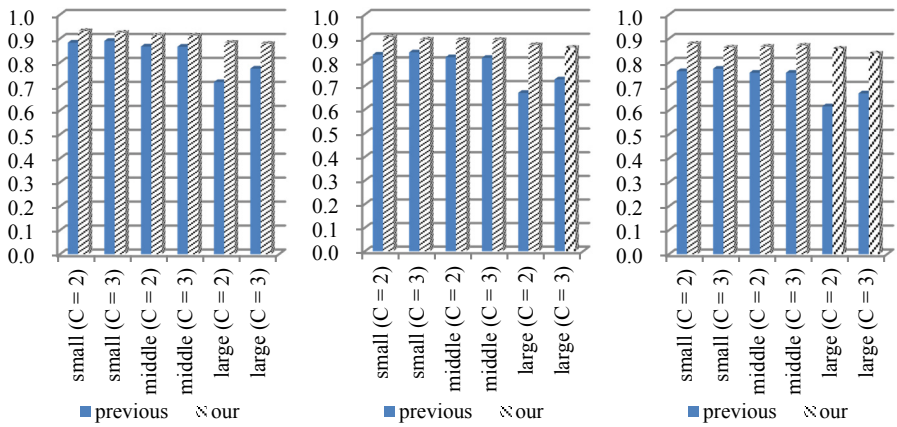
Fig. 7. Comparison of the fitness values using our and the previous approaches in the generalized static scenario.

4.5 Experimental Results in the Generalized Dynamic Scenario

From the experimental results in the two previous subsections, we have known that our approach performs better in the simplified dynamic scenario, and performs similar to the previous approach in the generalized static scenario. This subsection further analyzes the performance in the generalized dynamic scenario.

Comparisons of the best, average, and worst fitness values using our and the previous approaches in the generalized dynamic scenario are given in Figure 8(a)–(c), respectively. From Figure 8, it can be observed that our pro-

posed approach performs better than the previous approach in terms of all fitness statistics. To analyze this, we find that since each of the majority of the mesh clients in the generalized dynamic scenario belong to a certain community, mesh clients can support each other by using our social-based approach, and hence an efficient and effective readjustment can be achieved.



(a) Best fitness (b) Average fitness (c) Worst fitness

Fig. 8. Comparison of the fitness values using our and the previous approaches in the generalized dynamic scenario.

5 Conclusion and Future Work

This paper focuses to include the social-awareness of mesh routers to the placement of dynamic router node placement in WMNs (WMN-dynRNP), in which mesh clients may belong to a community, and hence move and gather together; additionally, each mesh router can serve only a restricted number of mesh clients. To cope with this problem, this paper has proposed a social-based PSO approach, which includes a social-supporting vector in the formula of updating the velocity of each particle. This vector makes low-loading mesh routers tend to support the heavy-loading mesh routers in the same topology subgraph component, and hence, the mesh clients within the radio coverage re-

gions of those heavy-loading mesh routers may have higher chance to be served by other mesh routers.

The experimental results are compared with our previous approach to the basic WMN-dynRNP problem in [12]. First, in the simplified dynamic scenario (in which each mesh client belongs to a community, i.e., there is no scattered mesh clients), our proposed approach obviously performs better than the previous approach to make effective adjustment to adapt the dynamic change of communities. Next, we consider the generalized static scenario in which scattered mesh clients are added, and we find that our approach performs no significantly different from the previous approach, but our approach still can ensure to keep the solution quality. Last, the experimental results in the generalized dynamic scenario show that our proposed social-based approach can perform better to adjust the mesh router placement according to the social community behavior of mesh clients, under different problem scales.

In the future work, more extension on the social behaviors of mesh clients could be made, and the data in a real-world deployment area can be applied. Especially, it is of interest to investigate whether the proposed approach can handle real traces of communities in the dynamic design. Additionally, detailed parameter analysis of the proposed algorithm would be also interesting. For the solution solvability, a cost function may be considered during the process of calculating the social-supporting vector, so that mesh routers could move to the direction with the most benefit to support the other mesh routers so as to achieve better solutions.

Acknowledgements

The authors thank the anonymous referees for comments that improved the content as well as the presentation of this paper. This work has been supported in part by MOST 104-2221-E-009-134-MY2 and NSC

101-2221-E-018-004-MY3, Taiwan.

References

[1] Wang X (2008) Wireless mesh networks. Journal of Telemedicine and Telecare 14(8): 401-403.

[2] Pathak PH, Dutta R (2011) A survey of network design problems and joint design approaches in wireless mesh networks. IEEE Communications Surveys & Tutorials 13(3): 396-428.

[3] Shu L, Zhang Y, Zhou Z, Hauswirth M, Yu Z, Hynes G (2008) Transmitting and gathering streaming data in wireless multimedia sensor networks within expected network lifetime. ACM/Springer Mobile Networks and Applications 13(3-4): 306-322.

[4] Shu L, Zhou Z, Hauswirth M, Phuoc D, Yu P, Zhang L (2007) Transmitting streaming data in wireless multimedia sensor networks with holes. In: Proc. of the 6th International Conference on Mobile Ubiquitous Multimedia (MUM 2007), ACM Press, pp. 24-33.

[5] Akyildiz IF, Wang X, Wang W (2005) Wireless mesh networks: A survey. Journal of Computer Networks 47(4): 445-487.

[6] Franklin AA, Murthy CSR (2007) Node placement algorithm for deployment of two-tier wireless mesh networks. In: Proc. of IEEE Global Telecommunications Conference (GLOBECOM'07), IEEE Press, pp. 4823-4827.

[7] Oda T, Barolli A, Xhafa F, Barolli L, Ikeda M, Takizawa M (2013) WMN–GA: A simulation system for WMNs and its evaluation considering selection operators. Journal of Ambient Intelligence and Humanized

Computing 4(3): 323-330.

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

[9] Xhafa F, Sánchez C, Barolli A, Takizawa M (2011) A tabu search algorithm for efficient node placement in wireless mesh networks. In: Proc. of Third International Conference on Intelligent Networking and Collaborative Systems (INCoS 2011), IEEE Press, pp. 53-59.

[10] Chang X, Oda T, Spaho E, Ikeda M, Barolli L, Xhafa F (2013) Performance evaluation of WMNs using hill climbing algorithm considering giant component and different distributions. In: J.J. Park et al. (eds.), Information Technology Convergence, vol. 253 of Lecture Notes in Electrical Engineering, pp. 161-167.

[11] Lin CC, Shu L, Deng DJ (2014) Router node placement with service priority in wireless mesh networks using simulated annealing with momentum terms. To appear in IEEE Systems Journal.

[12] Lin CC (2013) Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis. Information Sciences 232: 294-308.

[13] Shu L, Zhang Y, Yang L, Wang Y, Hauswirth M, Xiong N (2010) TPGF: Geographic routing in wireless multimedia sensor networks. Telecommunication Systems 44(1-2): 79-95.

[14] Younis M, Akkaya K (2008) Strategies and techniques for node placement in wireless sensor networks: A survey. Ad Hoc Networks 6(4): 621-655.

[15] Li F, Wang Y, Li XY, Nusairat A, Wu Y (2008) Gateway placement for throughput optimization in wireless mesh networks. Mobile Networks and Applications 13(1-2): 198-211.

[16] Wang J, Xie B, Cai K, Agrawal DP (2007) Efficient mesh router placement in wireless mesh networks. In: Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2007), IEEE Press, pp. 1-9.

[17] Oda T, Barolli A, Spaho E, Barolli L, Xhafa F, Younas M (2014) Effects of population size for location-aware node placement in WMNs: Evaluation by a genetic algorithm–based approach. Personal and Ubiquitous Computing 18(2): 261-269.

[18] Barolli A, Xhafa F, Takizawa M (2011) Optimization problems and resolution methods for node placement in wireless mesh networks. In: Proc. of 14th IEEE International Conference on Network-Based Information Systems (NbiS 2011), IEEE Press, pp. 7-9.

[19] Kas M, Appala S, Wang C, Carley KM, Carley LR, Tonguz OK (2012) What if wireless routers were social? Approaching wireless mesh networks from a social networks perspective. IEEE Wireless Communications Magazine 19(6): 36-43.

[20] Kim S, Lin MC, Manocha D (2014) Simulating crowd interactions in virtual environments (doctoral consortium). In: Proc. of IEEE Virtual Reality 2014 (VR 2014), IEEE Press, pp. 135-136.

[21] Song Y, Gong J, Niu L, Li Y, Jiang Y, Zhang W, Cui T (2013) A grid-based spatial data model for the simulation and analysis of individual behaviours in micro-spatial environments. Simulation Modelling Practice and Theory 38: 58-68.

[22] Li F, Wu J (2009) Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks. In Proc. of 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc

Communications and Networks (SECON'09), IEEE Press, pp. 1-9.

[23] Basurra SS, Ji Y, De Vos M, Padget J, Lewis T, Armour S (2012) Social-aware routing for wireless mesh networks. In: Proc. of 2012 IEEE Vehicular Technology Conference (VTC Fall 2012), IEEE Press, pp. 1-5.

[24] Wong GK, Jia X (2013) A novel socially-aware opportunistic routing algorithm in mobile social networks. In: Proc. of 2013 International Conference on Computing, Networking and Communications (ICNC 2013), IEEE Press, pp. 514-518.

[25] Wei K, Zeng D, Guo S, Xu K (2013) Social-aware relay node selection in delay tolerant networks. In: Proc. of 22nd International Conference on Computer Communications and Networks (ICCCN 2013), IEEE Press, pp. 1-7.

[26] Wei K, Liang X, Xu K (2014) A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues. IEEE Communications Surveys & Tutorials 16(1): 556-578.

[27] Eberhart RC, Kennedy J (1995) A new optimizer using particles swarm theory. In: Proc. of the 6th International Symposium on Micro Machine and Human Science (MHS'95), IEEE Press, pp. 39-43.

[28] Kennedy J (1997) The particle swarm: Social adaptation of knowledge. In: Proc. of IEEE International Conference on Evolutionary Computation, IEEE Press, pp. 303-308.

[29] Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proc. of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, pp. 69-73.

[30] Eberhart RC, Shi Y (2001) Tracking and optimizing dynamic systems with particle swarms. In: Proc. of the 2001 Congress on Evolutionary Computation, IEEE Press, vol. 1, pp. 94-100.

[31] Hsiao KJ, Kulesza A, Hero AO (2014) Social collaborative retrieval. IEEE Journal of Selected Topics in Signal Processing 8(4): 680-689.