Performance Evaluation of Intelligent Hybrid Systems for Node Placement in Wireless Mesh Networks: A Comparison Study of WMN-PSOHC and WMN-PSOSA

Shinji Sakamoto[1(✉)], Kosuke Ozera[1], Tetsuya Oda[2], Makoto Ikeda[3], and Leonard Barolli[3]

[1] Graduate School of Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
shinji.t.sakamoto@gmail.com, kosuke.o.fit@gmail.com
[2] Department of Information and Computer Engineering, Okayama University of Science, 1-1 Ridai-cho, Kita-Ku, Okayama 700-0005, Japan
oda.tetsuya.fit@gmail.com
[3] Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
makoto.ikd@acm.org, barolli@fit.ac.jp

Abstract. Wireless Mesh Networks (WMNs) have many advantages such as low cost and increased high speed wireless Internet connectivity, therefore WMNs are becoming an important networking infrastructure. In our previous work, we implemented a Particle Swarm Optimization (PSO) based simulation system for node placement in WMNs, called WMN-PSO. Also, we implemented two intelligent hybrid systems for solving node placement problem in WMNs: PSO and Hill Climbing (HC) based system, called WMN-PSOHC, and PSO and Simulated Annealing (SA) based system, called WMN-PSOSA. In this paper, we evaluate two hybrid simulation systems WMN-PSOHC and WMN-PSOSA. We compare WMN-PSOHC with WMN-PSOSA by conducting computer simulations.

1 Introduction

The wireless networks and devises are becoming increasingly popular and they provide users access to information and communication anytime and anywhere [5,7–11]. Wireless Mesh Networks (WMNs) are gaining a lot of attention because of their low cost nature that makes them attractive for providing wireless Internet connectivity. A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among them-selves (creating, in effect, an ad hoc network) [4,13,15]. This feature brings many advantages to WMNs such as low up-front cost, easy

network maintenance, robustness and reliable service coverage [1]. Moreover, such infrastructure can be used to deploy community networks, metropolitan area networks, municipal and corporative networks, and to support applications for urban areas, medical, transport and surveillance systems.

Mesh node placement in WMN can be seen as a family of problems, which are shown (through graph theoretic approaches or placement problems, e.g. [23]) to be computationally hard to solve for most of the formulations [24]. In fact, the node placement problem considered here is even more challenging due to two additional characteristics:

(a) locations of mesh router nodes are not pre-determined, in other wards, any available position in the considered area can be used for deploying the mesh routers.
(b) routers are assumed to have their own radio coverage area.

Here, we consider the version of the mesh router nodes placement problem in which we are given a grid area where to deploy a number of mesh router nodes and a number of mesh client nodes of fixed positions (of an arbitrary distribution) in the grid area. The objective is to find a location assignment for the mesh routers to the cells of the grid area that maximizes the network connectivity and client coverage. Node placement problems are known to be computationally hard to solve [12]. In some previous works, intelligent algorithms have been recently investigated [2,22,25].

In our previous work, we implemented a Particle Swarm Optimization (PSO) based simulation system for node placement in WMNs, called WMN-PSO [16]. Also, we implemented two intelligent hybrid systems for solving node placement problem in WMNs:

1. PSO and Hill Climbing (HC) based system, called WMN-PSOHC;
2. PSO and Simulated Annealing (SA) based system, called WMN-PSOSA.

In this paper, we evaluate two hybrid simulation systems WMN-PSOHC and WMN-PSOSA. We compare WMN-PSOHC with WMN-PSOSA by conducting computer simulations.

The rest of the paper is organized as follows. The mesh router nodes placement problem is defined in Sect. 2. We present our designed and implemented hybrid simulation system in Sect. 3. The simulation results are given in Sect. 4. Finally, we give conclusions and future work in Sect. 5.

2   Node Placement Problem in WMNs

For this problem, we have a grid area arranged in cells we want to find where to distribute a number of mesh router nodes and a number of mesh client nodes of fixed positions (of an arbitrary distribution) in the considered area. The objective is to find a location assignment for the mesh routers to the area that maximizes the network connectivity and client coverage. Network connectivity is measured by Size of Giant Component (SGC) of the resulting WMN graph, while the user

coverage is simply the number of mesh client nodes that fall within the radio coverage of at least one mesh router node and is measured by Number of Covered Mesh Clients (NCMC).

An instance of the problem consists as follows.

- N mesh router nodes, each having its own radio coverage, defining thus a vector of routers.
- An area $W \times H$ where to distribute N mesh routers. Positions of mesh routers are not pre-determined and are to be computed.
- M client mesh nodes located in arbitrary points of the considered area, defining a matrix of clients.

It should be noted that network connectivity and user coverage are among most important metrics in WMNs and directly affect the network performance.

In this work, we have considered a bi-objective optimization in which we first maximize the network connectivity of the WMN (through the maximization of the SGC) and then, the maximization of the NCMC.

In fact, we can formalize an instance of the problem by constructing an adjacency matrix of the WMN graph, whose nodes are router nodes and client nodes and whose edges are links between nodes in the mesh network. Each mesh node in the graph is a triple $\boldsymbol{v} = <x, y, r>$ representing the 2D location point and r is the radius of the transmission range. There is an arc between two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$, if $\boldsymbol{v}$ is within the transmission circular area of $\boldsymbol{u}$.

3   Proposed Intelligent Hybrid Systems

3.1   Intelligent Algorithms

3.1.1   Particle Swarm Optimization

In PSO a number of simple entities (the particles) are placed in the search space of some problems or functions and each evaluates the objective function at its current location [14]. The objective function is often minimized and the exploration of the search space is not through evolution. However, following a widespread practice of borrowing from the evolutionary computation field, in this work, we consider the bi-objective function and fitness function interchangeably. Each particle then determines its movement through the search space by combining some aspect of the history of its own current and best (best-fitness) locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function. Each individual in the particle swarm is composed of three $\mathcal{D}$-dimensional vectors, where $\mathcal{D}$ is the dimensionality of the search space. Each particle has the current position $\mathbf{x}_i$, the previous best position $\mathbf{p}_i$ and the velocity $\mathbf{v}_i$.

The particle swarm is more than just a collection of particles. A particle by itself has almost no power to solve any problem; progress occurs only when the
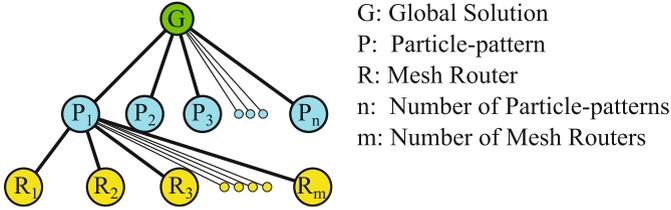
G: Global Solution
P: Particle-pattern
R: Mesh Router
n: Number of Particle-patterns
m: Number of Mesh Routers

Fig. 1. Relationship among global solution, particle-patterns and mesh routers.

particles interact. Problem solving is a population-wide phenomenon, emerging from the individual behaviors of the particles through their interactions. In any case, populations are organized according to some sort of communication structure or topology, often thought of as a social network. The topology typically consists of bidirectional edges connecting pairs of particles, so that if j is in i's neighborhood, i is also in j's. Each particle communicates with some other particles and is affected by the best point found by any member of its topological neighborhood. This is just the vector $\mathbf{p}_i$ for that best neighbor, which we will denote with $\mathbf{p}_g$. The potential kinds of population "social networks" are hugely varied, but in practice certain types have been used more frequently.

In the PSO process, the velocity of each particle is iteratively adjusted so that the particle stochastically oscillates around $\mathbf{p}_i$ and $\mathbf{p}_g$ locations.

Initialization

Our implemented systems (WMN-PSOHC and WMN-PSOHC) starts by generating an initial solution randomly, by *ad hoc* methods [25]. We decide the velocity of particles by a random process considering the area size. For instance, when the area size is $W \times H$, the velocity is decided randomly from $-\sqrt{W^2 + H^2}$ to $\sqrt{W^2 + H^2}$.

Particle-pattern

A particle is a mesh router. A fitness value of a particle-pattern is computed by combination of mesh routers and mesh clients positions. In other words, each particle-pattern is a solution as shown is Fig. 1. Therefore, the number of particle-patterns is a number of solutions.

Fitness function

One of most important thing in PSO algorithm is to decide the determination of an appropriate objective function and its encoding. In our case, each particle-pattern has an own fitness value and compares other particle-pattern's fitness value in order to share information of global solution. The fitness function follows a hierarchical approach in which the main objective is to maximize the SGC in WMN. Also, we tune and evaluate the performance for different coefficient value of fitness function [18]. Then, the fitness function of this scenario is defined as

$$\text{Fitness} = 0.7 \times \text{SGC}(\boldsymbol{x}_{ij}, \boldsymbol{y}_{ij}) + 0.3 \times \text{NCMC}(\boldsymbol{x}_{ij}, \boldsymbol{y}_{ij}).$$

Routers replacement method

A mesh router has x, y positions and velocity. Mesh routers are moved based on velocities. There are many moving methods in PSO field.

- Constriction method (CM) [3,20];
- Random Inertia Weight method (RIWM) [20];
- Linearly Decreasing Inertia Weight method (LDIWM) [20,21];
- Rational Decrement of Vmax Method (RDVM) [17];
- Linearly Decreasing Vmax method (LDVM) [19].

In this scenario, we apply LDVM. In LDVM, PSO parameters are set to unstable region ($\omega = 0.9$, $C_1 = C_2 = 2.0$). A value of V_{max} which is maximum velocity of particles is considered. With increasing of iteration of computations, the V_{max} is kept decreasing linearly [19].

3.1.2   Hill Climbing Algorithm
HC is local search algorithm and is based on incremental improvements of solutions as follows: it starts with a solution (which may be randomly generated or ad hoc computed) considered as the current solution in the search space. The algorithm examines its neighboring solutions and if a neighbor is better than current solution then it can become the current solution; the algorithm keeps moving from one solution to another one in the search space until no further improvements are possible.

3.1.3   Simulated Annealing
SA algorithm [6] is a generalization of the metropolis heuristic. Indeed, SA consists of a sequence of executions of metropolis with a progressive decrement of the temperature starting from a rather high temperature, where almost any move is accepted, to a low temperature, where the search resembles Hill Climbing. In fact, it can be seen as a hill-climber with an internal mechanism to escape local optima. In SA, the solution s' is accepted as the new current solution if $\delta \leq 0$ holds, where $\delta = f(s') - f(s)$. To allow escaping from a local optimum, the movements that increase the energy function are accepted with a decreasing probability $\exp(-\delta/T)$ if $\delta > 0$, where T is a parameter called the "temperature". The decreasing values of T are controlled by a *cooling schedule*, which specifies the temperature values at each stage of the algorithm, what represents an important decision for its application (a typical option is to use a proportional method, like $T_k = \alpha \cdot T_{k-1}$). SA usually gives better results in practice, but uses to be very slow. The most striking difficulty in applying SA is to choose and tune its parameters such as initial and final temperature, decrements of the temperature (cooling schedule), equilibrium and detection.

In our system, cooling schedule (α) will be calculated as

$$\alpha = \left(\frac{SA\ Ending\ temperature}{SA\ Starting\ temperature} \right)^{1.0/Total\ iterations}.$$

The acceptability criteria for newly generated solution is based on the definition of a threshold value (accepting threshold) as follows. We consider a succession t_k such that $t_k > t_{k+1}$, $t_k > 0$ and t_k tends to 0 as k tends to infinity. Then, for any two solutions s_i and s_j, if $fitness(s_j) - fitness(s_i) < t_k$, then accept solution s_j.

For the SA, t_k values are taken as accepting threshold but the criterion for acceptance is probabilistic:

- If $fitness(s_j) - fitness(s_i) \leq 0$ then s_j is accepted.
- If $fitness(s_j) - fitness(s_i) > 0$ then s_j is accepted with probability $\exp[(fitness(s_j) - fitness(s_i))/t_k]$ (at iteration k the algorithm generates a random number $R \in (0,1)$ and s_j is accepted if $R < \exp[(fitness(s_j) - fitness(s_i))/t_k]$).

In this case, each neighbour of a solution has a positive probability of replacing the current solution. The t_k values are chosen in way that solutions with large increase in the cost of the solutions are less likely to be accepted (but there is still a positive probability of accepting them).

Algorithm 1. Pseudo code of PSO-HC.

```
/* Generate the initial solutions and parameters */
Computation maxtime:= T_max, t := 0;
Number of particle-patterns:= m, 2 ≤ m ∈ N¹;
Particle-patterns initial solution:= P⁰_i;
Global initial solution:= G⁰;
Particle-patterns initial position:= x⁰_ij;
Particles initial velocity:= v⁰_ij;
PSO parameter:= ω, 0 < ω ∈ R¹;
PSO parameter:= C₁, 0 < C₁ ∈ R¹;
PSO parameter:= C₂, 0 < C₂ ∈ R¹;
/* Start PSO-HC */
Evaluate(G⁰, P⁰);
while t < T_max do
    /* Update velocities and positions */
    v^{t+1}_ij = ω · v^t_ij
            +C₁ · rand() · (best(P^t_ij) − x^t_ij)
            +C₂ · rand() · (best(G^t) − x^t_ij);
    x^{t+1}_ij = x^t_ij + v^{t+1}_ij;
    /* if fitness value is increased, a new solution will be accepted. */
    if Evaluate(G^{(t+1)}, P^{(t+1)}) >= Evaluate(G^{(t)}, P^{(t)}) then
        Update_Solutions(G^t, P^t);
        Evaluate(G^{(t+1)}, P^{(t+1)});
    else
        /* "Reupdate_Solutions" makes particle back to previous position */
        Reupdate_Solutions(G^{(t+1)}, P^{(t+1)});
    end if
    t = t + 1;
end while
Update_Solutions(G^t, P^t);
return Best found pattern of particles as solution;
```

3.2   Proposed and Implemented Simulation Systems

We show the PSO-HC hybrid algorithm in Algorithm 1. WMN-PSOHC system uses PSO and HC mechanisms. We call this simulator WMN-PSOHC.

We show the PSO-SA hybrid algorithm in Algorithm 2. The WMN-PSOSA system considers PSO and SA mechanisms. We call this simulator WMN-PSOSA.

Algorithm 2. Pseudo code of PSO-SA.

/* Generate the initial solutions and parameters */
Computation maxtime:= T_{max}, $t := 0$;
Number of particle-patterns:= m, $2 \leq m \in \boldsymbol{N}^1$;
Starting SA temperature:= $Temp$;
Decreasing speed of SA temperature:= T_d;
Particle-patterns initial solution:= $\boldsymbol{P}_i^0$;
Global initial solution:= $\boldsymbol{G}^0$;
Particle-patterns initial position:= $\boldsymbol{x}_{ij}^0$;
Particles initial velocity:= $\boldsymbol{v}_{ij}^0$;
PSO parameter:= ω, $0 < \omega \in \boldsymbol{R}^1$;
PSO parameter:= C_1, $0 < C_1 \in \boldsymbol{R}^1$;
PSO parameter:= C_2, $0 < C_2 \in \boldsymbol{R}^1$;
/* Start PSO-SA */
Evaluate($\boldsymbol{G}^0, \boldsymbol{P}^0$);
while $t < T_{max}$ **do**
   /* Update velocities and positions */
   $\boldsymbol{v}_{ij}^{t+1} = \omega \cdot \boldsymbol{v}_{ij}^t$
        $+C_1 \cdot \text{rand}() \cdot (best(P_{ij}^t) - x_{ij}^t)$
        $+C_2 \cdot \text{rand}() \cdot (best(G^t) - x_{ij}^t)$;
   $\boldsymbol{x}_{ij}^{t+1} = \boldsymbol{x}_{ij}^t + \boldsymbol{v}_{ij}^{t+1}$;
   /* if fitness value is increased, a new solution will be accepted. */
   if Evaluate($\boldsymbol{G}^{(t+1)}, \boldsymbol{P}^{(t+1)}$) $>=$ Evaluate($\boldsymbol{G}^{(t)}, \boldsymbol{P}^{(t)}$) **then**
      Update_Solutions($\boldsymbol{G}^t, \boldsymbol{P}^t$);
      Evaluate($\boldsymbol{G}^{(t+1)}, \boldsymbol{P}^{(t+1)}$);
   else
      /* a new solution will be accepted, if condition is true. */
      if Random() $> e^{\left(\frac{Evaluate(\boldsymbol{G}^{(t+1)}, \boldsymbol{P}^{(t+1)}) - Evaluate(\boldsymbol{G}^{(t)}, \boldsymbol{P}^{(t)})}{Temp} \right)}$ **then**
         /* "Reupdate_Solutions" makes particle back to previous position */
         Reupdate_Solutions($\boldsymbol{G}^{t+1}, \boldsymbol{P}^{t+1}$);
      end if
   end if
   $Temp = Temp \times t_d$;
   $t = t + 1$;
end while
Update_Solutions($\boldsymbol{G}^t, \boldsymbol{P}^t$);
return Best found pattern of particles as solution;

Table 1. WMN-PSOHC parameters.

Parameters	Values
Clients distribution	Normal distribution
Grid size	32×32
Number of mesh routers	16
Number of mesh clients	48
Total iterations	800
Iteration per phase	4
Number of particle-patterns	9
Radius of a mesh router	2.0
Replacement method	LDVM

Table 2. WMN-PSOSA parameters.

Parameters	Values
Clients distribution	Normal distribution
Grid size	32×32
Number of mesh routers	16
Number of mesh clients	48
Total iterations	800
Iteration per phase	4
SA Starting temperature	10
SA Ending temperature	0.1
Decreasing speed of SA temperature (α)	0.994260
Number of particle-patterns	9
Radius of a mesh router	2.0
Replacement method	LDVM

4   Simulation Results

In this section, we show simulation results using WMN-PSOHC and WMN-PSOSA systems. In this work, we set the same parameters in order to compare two simulation systems. We consider normal distribution of mesh clients. The number of mesh routers is considered 16 and the number of mesh clients 48. The total number of iterations is considered 800 and the iterations per phase is considered 4. We consider the number of particle-patterns 9. We conducted simulations 10 times, in order to avoid the effect of randomness and create a general view of results. For WMN-PSOSA, the temperature value was set to 10. The temperature value decreases by multiplying 0.99426 per a iteration. At the end of simulation, the temperature value will be 0.1.
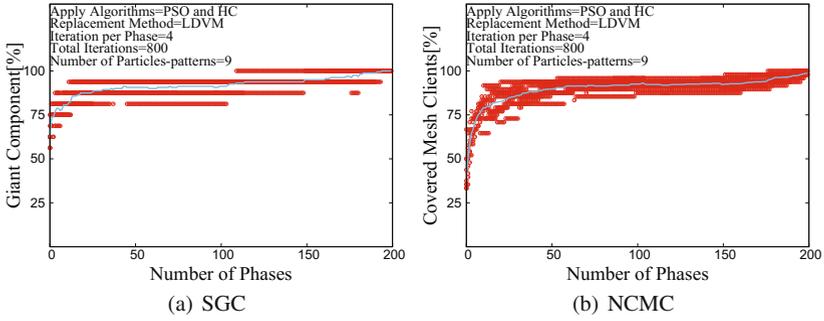
(a) SGC

(b) NCMC

Fig. 2. Simulation results for WMN-PSOHC.

(a) SGC

(b) NCMC

Fig. 3. Simulation results for WMN-PSOSA.

(a) WMN-PSOHC

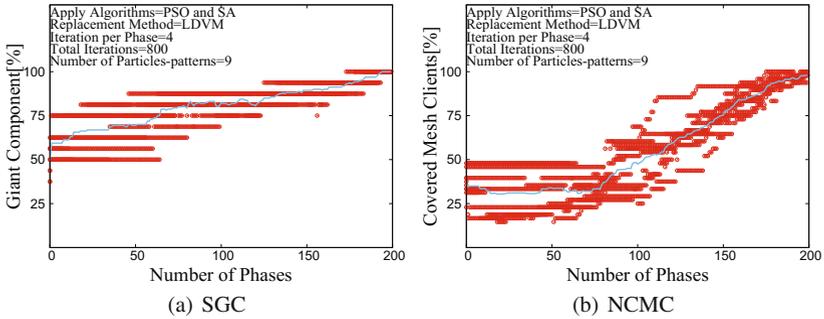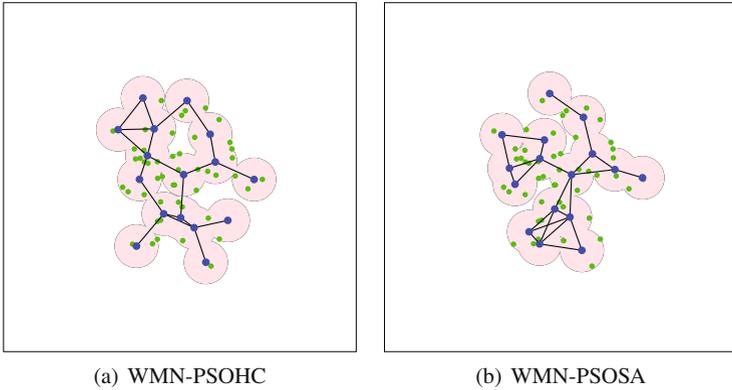(b) WMN-PSOSA

Fig. 4. Visualized image of simulation results.

We show the parameter setting for both WMN-PSOHC and WMN-PSOSA in Tables 1 and 2, respectively.

We show the simulation results from Figs. 2 to 4. In Fig. 2, we show simulations results for WMN-PSOHC. We see that solutions converge very fast. The

NCMC also converges soon and all solution reaches maximum value. Figure 3 shows that the SGC and NCMC of WMN-PSOSA increases gradually and all solution reaches maximum value. We show the visualized results for WMN-PSOHC and WMN-PSOSA in Fig. 4.

We see that all mesh routers are connected and all mesh clients are covered for both systems. The WMN-PSOHC converges faster than WMN-PSOSA.

5   Conclusions

In this work, we compared two hybrid simulation systems: PSO and HC based intelligent hybrid system (called WMN-PSOHC) and PSO and SA based intelligent hybrid system (called WMN-PSOSA).

From the simulation results, we conclude that all mesh routers are connected and all mesh clients are covered for both systems. Comparing two intelligent hybrid systems, the WMN-PSOHC converges faster than WMN-PSOSA.

In our future work, we would like to evaluate the performance of the proposed systems for different parameters and patterns.

References

1. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: a survey. Comput. Netw. **47**(4), 445–487 (2005)
2. Behnamian, J., Ghomi, S.F.: Development of a PSO-SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. Expert Syst. Appl. **37**(2), 974–984 (2010)
3. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
4. Gupta, B.K., Patnaik, S., Mallick, M.K., Nayak, A.K.: Dynamic routing algorithm in wireless mesh network. Int. J. Grid Util. Comput. **8**(1), 53–60 (2017)
5. Hiyama, M., Sakamoto, S., Kulla, E., Ikeda, M., Barolli, L.: Experimental results of a MANET testbed for different settings of HELLO packets of OLSR protocol. J. Mob. Multimedia **9**(1–2), 27–38 (2013)
6. Hwang, C.R.: Simulated annealing: theory and applications. Acta Applicandae Mathematicae **12**(1), 108–111 (1988)
7. Inaba, T., Elmazi, D., Liu, Y., Sakamoto, S., Barolli, L., Uchida, K.: Integrating wireless cellular and ad-hoc networks using fuzzy logic considering node mobility and security. In: 29th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA-2015), pp. 54–60 (2015). doi:10. 1109/WAINA.2015.116
8. Inaba, T., Elmazi, D., Sakamoto, S., Oda, T., Ikeda, M., Barolli, L.: A secure-aware call admission control scheme for wireless cellular networks using fuzzy logic and its performance evaluation. J. Mob. Multimedia **11**(3&4), 213–222 (2015)

9. Inaba, T., Obukata, R., Sakamoto, S., Oda, T., Ikeda, M., Barolli, L.: Performance evaluation of a QoS-aware fuzzy-based CAC for LAN access. Int. J. Space Based Situated Comput. **6**(4), 228–238 (2016)
10. Inaba, T., Sakamoto, S., Kulla, E., Caballe, S., Ikeda, M., Barolli, L.: An integrated system for wireless cellular and ad-hoc networks using fuzzy logic. In: International Conference on Intelligent Networking and Collaborative Systems (INCoS-2014), pp. 157–162 (2014)
11. Inaba, T., Sakamoto, S., Oda, T., Ikeda, M., Barolli, L.: A testbed for admission control in WLAN: a fuzzy approach and its performance evaluation. In: International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 559–571. Springer (2016)
12. Maolin, T., et al.: Gateways placement in backbone wireless mesh networks. Int. J. Commun. Netw. Syst. Sci. **2**(1), 44 (2009)
13. Niewiadomska-Szynkiewicz, E., Sikora, A.: Simulation-based design of self-organising and cooperative networks. Int. J. Space Based Situated Comput. **1**(1), 68–75 (2011)
14. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intell. **1**(1), 33–57 (2007)
15. Puzar, M., Plagemann, T.: Data sharing in mobile ad-hoc networks-a study of replication and performance in the midas data space. Int. J. Space Based Situated Comput. **1**(2–3), 137–150 (2011)
16. Sakamoto, S., Oda, T., Ikeda, M., Barolli, L., Xhafa, F.: Implementation and evaluation of a simulation system based on particle swarm optimisation for node placement problem in wireless mesh networks. Int. J. Commun. Netw. Distrib. Syst. **17**(1), 1–13 (2016)
17. Sakamoto, S., Oda, T., Ikeda, M., Barolli, L., Xhafa, F.: Implementation of a new replacement method in WMN-PSO simulation system and its performance evaluation. In: 30th IEEE International Conference on Advanced Information Networking and Applications (AINA-2016), pp. 206–211 (2016). doi:10.1109/AINA.2016.42
18. Sakamoto, S., Oda, T., Ikeda, M., Barolli, L., Xhafa, F., Woungang, I.: Investigation of fitness function weight-coefficients for optimization in WMN-PSO simulation system. In: 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2016), pp. 224–229 (2016)
19. Schutte, J.F., Groenwold, A.A.: A study of global optimization using particle swarms. J. Global Optim. **31**(1), 93–108 (2005)
20. Shi, Y.: Particle swarm optimization. IEEE Connections **2**(1), 8–13 (2004)
21. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Evolutionary Programming VII, pp. 591–600 (1998)
22. Singh, L., Singh, S.: Score-based genetic algorithm for scheduling workflow applications in clouds. Int. J. Grid Util. Comput. **7**(4), 272–284 (2016)
23. Tan, L., Chen, Y., Yang, M., Hu, J., Lian, J.: Connecting priority algorithm for node deployment in directional sensor networks. Int. J. Grid Util. Comput. **8**(1), 29–37 (2017)
24. Vanhatupa, T., Hannikainen, M., Hamalainen, T.: Genetic algorithm to optimize node placement and configuration for WLAN planning. In: Proceedings of 4th IEEE International Symposium on Wireless Communication Systems, pp. 612–616 (2007)
25. Xhafa, F., Sanchez, C., Barolli, L.: Ad hoc and neighborhood search methods for placement of mesh routers in wireless mesh networks. In: Proceedings of 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS-2009), pp. 400–405 (2009)