UPCommons

Portal del coneixement obert de la UPC

<http://upcommons.upc.edu/e-prints>

Kolici, V. (2015) Optimization of giant component and number of covered users in Wireless Mesh Networks: a comparison study. *IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, South Korea, March 25-27, 2015: proceedings*. [S.I.]: IEEE, 2015. Pp. 215-222. Doi: <http://dx.doi.org/10.1109/AINA.2015.188>.

© 2015 IEEE. Es permet l'ús personal d'aquest material. S'ha de demanar permís a l'IEEE per a qualsevol altre ús, incloent la reimpressió/reedició amb fins publicitaris o promocionals, la creació de noves obres col·lectives per a la revenda o redistribució en servidors o llistes o la reutilització de parts d'aquest treball amb drets d'autor en altres treballs.

Kolici, V. (2015) Optimization of giant component and number of covered users in Wireless Mesh Networks: a comparison study. *IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, South Korea, March 25-27, 2015: proceedings*. [S.I.]: IEEE, 2015. Pp. 215-222. Doi: <http://dx.doi.org/10.1109/AINA.2015.188>.

(c) 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Optimization of Giant Component and Number of Covered Users in Wireless Mesh Networks: A Comparison Study

Vladi Kolici*, Leonard Barolli†, Shinji Sakamoto‡, Tetsuya Oda‡, Fatos Xhafa§ and Kazunori Uchida†

**Faculty of Information Technologies
Polytechnic University of Tirana*

*Bul. “Dëshmorët e Kombit”, “Mother Theresa” Square, Nr. 4, Tirana, Albania
E-mail: vkolici@fti.edu.al*

*†Department of Information and Communication Engineering
Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
E-mail: barolli@fit.ac.jp, k-uchida@fit.ac.jp*

*‡Graduate School of Engineering
Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
E-mail: shinji.t.sakamoto@gmail.com, oda.tetsuya.fit@gmail.com*

*§Technical University of Catalonia
Department of Languages and Informatics Systems
C/Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: fatos@lsi.upc.edu*

Abstract—One of the key advantages of Wireless Mesh Networks (WMNs) is their importance for providing cost-efficient broadband connectivity. There are issues for achieving the network connectivity and user coverage, which are related with the node placement problem. We already implemented 3 simulation systems (WMN-HC, WMN-SA and WMN-GA) in our previous work. In this work, we present a comparative study by applying these 3 simulation systems for optimal distribution of WMN router nodes in order to provide the best network connectivity and user coverage. From the simulation results, we conclude that for small instances WMN-HC and WMN-SA have better performance than WMN-GA. However, for large instance the performance of WMN-GA simulation system is better.

Keywords—Node Placement, WMNs, Hill Climbing, Genetic Algorithm, Simulated Annealing, Intelligent Algorithms.

I. INTRODUCTION

Wireless Mesh Networks (WMNs) [1]–[3] are important network infrastructure for providing cost-efficient broadband wireless connectivity. They are showing their applicability in deployment of medical, transport and surveillance applications in urban areas, metropolitan, neighbouring communities and municipal area networks. At the heart of WMNs are the issues of achieving network connectivity and stability as well as QoS in terms of user coverage. These issues are very

closely related to the family of node placement problems in WMNs, such as mesh router nodes placement.

Node placement problems have been long investigated in the optimization field due to numerous applications in location science (facility location, logistics, services, etc.) and classification (clustering). Facility location problems are thus showing their usefulness to communication networks, and more especially from WMNs field. WMNs are currently attracting a lot of attention from wireless research and technology community for providing cost-efficient broadband wireless connectivity.

WMNs are based on mesh topology, in which every node (representing a server) is connected to one or more nodes, enabling thus the information transmission in more than one path. The path redundancy is a robust feature of this kind of topology. Compared to other topologies, mesh topology needs not a central node, allowing networks based on such topology to be self-healing. These characteristics of networks with mesh topology make them very reliable and robust networks to potential server node failures. In WMNs mesh routers provide network connectivity services to mesh client nodes. The good performance and operability of WMNs largely depends on placement of mesh routers nodes in the geographical deployment area to achieve network connectivity, stability and user coverage. The objective is

to find an optimal and robust topology of the mesh router nodes to support connectivity services to clients.

For most formulations, node placement problems are shown to be computationally hard to solve to optimality [4]–[7], and therefore heuristic and meta-heuristic approaches are useful approaches to solve the problem for practical purposes. Several heuristic approaches are found in the literature for node placement problems in WMNs [8]–[17].

In this work, we use our implemented WMN-HC, WMN-SA and WMN-GA simulation systems for node placement in WMNs. For simulations, we consider Weibull distribution of mesh clients in a grid area. Then we deploy 16 mesh routers and apply 3 simulation systems to maximize the size of Giant Component (GC) and the Number of Covered Mesh Clients (NCMC).

The rest of the paper is organized as follows. The definition of node placement problem is presented in Section II.

We describe three algorithms we used in Section III. The simulation results are given in Section IV. Finally, concluding remarks and future work are given in Section V.

II. NODE PLACEMENT PROBLEM IN WMNS

In this problem, we are given a grid area arranged in cells 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 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. Network connectivity is measured by the size of the GC 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.

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 (the maximization of the size of the GC) and then the maximization of the NCMC.

III. IMPLEMENTED WMN-HC, WMN-SA AND WMN-GS SIMULATION SYSTEMS

In this section, we introduce three intelligent algorithms: Hill Climbing (HC), Simulated Annealing (SA) and Genetic Algorithm (GA). Then, we present the GUIs of the implemented simulation systems.

A. Hill Climbing

The HC is a 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 neighbouring solutions and if a neighbour 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. There are several variants of the algorithm depending on whether a simple climbing, steepest ascent climbing or stochastic climbing is done:

- Simple climbing: the next neighbour solution is the first that improves current solution.
- Steepest ascent climbing: all neighbour solutions are examined and the best one is chosen as next solution.
- Stochastic climbing: a neighbour is selected at random, and according to yielded improvement of that neighbour is decided whether to choose it as next solution or to examine another neighbour. This kind of climbing has more general forms known as Metropolis and Simulated Annealing algorithms.

Of course, for each climbing method described above, several versions of the algorithm are possible depending on the way a neighbour solution is selected, that is, depending on the neighbourhood structure used.

It should be noted that HC usually ends up in local optima, which can be overcome in some cases by adopting additional techniques such as: (a) getting back to a previous state and exploring another direction; (b) jumping to a new solution, possibly “far away” from current solution; and, (c) considering several search direction in solution space at the same time.

The algorithm first generates a solution which serves as starting point in the search space. Then, the algorithm iteratively selects a movement based on current solution, evaluates the resulting movement in terms of possible improvements with respect to current solution. If the resulting neighbour improves fitness of current solution, the current solution is moved to the new neighbour and so on. In the algorithm the function $\delta(s, m)$ computes the improvement yielded by applying movement m to current solution s (as usually, for maximization, a positive value of δ function means improvement with respect to fitness of current solution). between changing the system to the state s' or staying in the state s . The probabilities are chosen so that the system converges towards lower energy states. Typically this step is repeated until the system reaches a state good enough for the application or when a certain number of iterations is performed.

B. Simulated Annealing

The SA algorithm [18] is inspired by the cooling process of metals by which a material is heated and then cooled

Algorithm 1 : Pseudo-code of SA.

```
t := 0
Initialize T
s0 := Initial_Solution()
v0 := Evaluate(s0)
while (stopping condition not met) do
  while t mod MarkovChainLen = 0 do
    t := t+1
    s1 := Generate(s0,T) //Move
    v1 := Evaluate(s1)
    if Accept(v0,v1,T) then
      s0 := s1
      v0 := v1
    end if
  end while
  T := Update(T)
end while
return s0
```

in a controlled way to increase the size of its crystals and reduce their defects. The heat causes the atoms to leave their initial positions (a local minimum of energy) and move randomly; the slow cooling gives them more likelihood to find configurations with lower energy than the previous one. In each iteration, it considers some neighbors of the current state s , and probabilistically decides.

SA algorithm [19] 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 HC. In fact, it can be seen as a hill climber with an internal mechanism to escape local optima (see pseudo-code in Algorithm 1). 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. This 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 is very slow. The most striking difficulty in applying SA is to choose and tune its parameters such as initial and final temperature, decrement of the temperature (cooling schedule), equilibrium and detection.

C. Genetic Algorithm

The GA [20] has shown their usefulness for the resolution of many computationally hard combinatorial optimization problems. Their main features are briefly described next (see

Algorithm 2 Genetic Algorithm template.

```
Generate the initial population  $P^0$  of size  $\mu$ ;  $t = 0$ .
Evaluate  $P^0$ ;
while not termination-condition do
  Select the parental pool  $T^t$  of size  $\lambda$ ;
   $T^t := \text{Select}(P^t)$ ;
  Perform crossover procedure on pairs of individuals in
   $T^t$  with probability  $p_c$ ;  $P_c^t := \text{Cross}(T^t)$ ;
  Perform mutation procedure on individuals in  $P_c^t$  with
  probability  $p_m$ ;  $P_m^t := \text{Mutate}(P_c^t)$ ;
  Evaluate  $P_m^t$ ;
  Create a new population  $P^{t+1}$  of size  $\mu$  from individ-
  uals in  $P^t$  and/or  $P_m^t$ ;
   $P^{t+1} := \text{Replace}(P^t; P_m^t)$ 
   $t := t + 1$ ;
end while
return Best found individual as solution;
```

Algorithm 2 for a template).

Population of individuals: Unlike local search techniques that construct a path in the solution space jumping from one solution to another one through local perturbations, GA use a population of individuals giving thus the search a larger scope and chances to find better solutions. This feature is also known as “exploration” process in difference to “exploitation” process of local search methods.

Fitness: The determination of an appropriate fitness function, together with the chromosome encoding are crucial to the performance of GA. Ideally we would construct objective functions with “certain regularities”, i.e. objective functions that verify that for any two individuals which are close in the search space, their respective values in the objective functions are similar.

Selection: The selection of individuals to be crossed is another important aspect in GA as it impacts on the convergence of the algorithm. Several selection schemes have been proposed in the literature for selection operators trying to cope with premature convergence of GA.

Crossover operators: Use of crossover operators is one of the most important characteristics. Crossover operator is the means of GA to transmit best genetic features of parents to offsprings during generations of the evolution process.

Mutation operators: These operators intend to improve the individuals of a population by small local perturbations. They aim to provide a component of randomness in the neighbourhood of the individuals of the population.

Escaping from local optima: GA has the ability to avoid falling prematurely into local optima and can eventually escape from them during the search process.

Convergence: The convergence of the algorithm is the mechanism of GA to reach to good solutions. A premature convergence of the algorithm would cause that all individuals of the population be similar in their genetic features and

Figure 1. GUI for WMN-HC system.

Figure 2. GUI for WMN-SA system.

thus the search would result ineffective and the algorithm getting stuck into local optima. Maintaining the diversity of the population is therefore very important to this family of evolutionary algorithms.

D. GUI of Implemented WMN-HC, WMN-SA and WMN-GS Simulation Systems

Based on three intelligent algorithms (HC, SA and GA), we implemented the GUIs for WMN-HC, WMN-SA and WMN-GS simulation systems. The implemented systems can generate instances of the problem using different distributions of clients and mesh routers.

The GUI of WMN-HC, WMN-SA and WMN-GA simulation systems are shown in Fig. 1, Fig. 2 and Fig. 3, respectively. By using these GUI tools, we easily can set-up

Figure 3. GUI for WMN-GA System.

Figure 4. An example of Weibull distribution of clients.

Figure 5. Mesh router coverage in simulation systems.

the network configuration parameters, such as distribution of mesh clients, number of clients, number of mesh routers, grid size, radius of transmission distance and the size of subgrid.

IV. SIMULATION RESULTS

We carried out many simulations to optimize the number of mesh routers using WMN-HC, WMN-SA and WMN-GA simulation systems. In Table I, we show general parameters for the simulation environment. Simulation parameters for WMN-HC, WMN-SA and WMN-GA are shown in Tables II, III and IV, respectively. The distribution of mesh

(a) Giant component

(b) Covered mesh clients

Figure 6. Performance evaluation for WMN-HC.

(a) Giant component

(b) Covered mesh clients

Figure 7. Performance evaluation for WMN-SA.

Table I
SIMULATION SETTINGS.

Parameters	Values
Client distribution	Weibull
Area size	32×32
Number of mesh routers	16
Number of mesh clients	48
Communication distance	2

Table II
SIMULATION SETTINGS FOR WMN-HC.

Parameters	Values
Iteration per phase	512
Total iteration	1024000
Replacement method	Combination

clients is Weibull distribution (See Fig. 4). In Fig. 5 is shown the radius of mesh router nodes, which is set to 2. The number of mesh clients is considered to be 48 and the number of mesh routers 16.

For each phase, the WMN-HC and WMN-SA run for 512

iterations. For each generation, the GA population is 512. We show simulation results for WMN-HC in Fig. 6. The result shows that for 100 phases the WMN-HC has the maximum size of GC. While the NCMC is maximized for 110 phases.

In Fig. 7, we show the results for WMN-SA. For WMN-SA, the size of GC is maximum for 63 phases, however there are some oscillations and the average value GC is not the

(a) Giant component

(b) Covered mesh clients

Figure 8. Performance evaluation for WMN-GA.

(a) Giant component

(b) Covered mesh clients

Figure 9. Performance evaluation for WMN-HC.

Table III
SIMULATION SETTINGS FOR WMN-SA.

Parameters	Values
SA temperature	1
Iteration per phase	512
Total iteration	1024000
Replacement method	Combination

Table IV
SIMULATION SETTINGS FOR WMN-GA.

Parameters	Values
Population Size	512
Number of Generation	2000
Crossover Rate	0.8
Mutate Rate	0.2
Selection Method	Liner Ranking
Initial Method	Hotspot

maximum (16). For the maximization of NCMC are needed 145 phases.

In Fig. 8, we show results of WMN-GA system. For WMN-GA, the size of GC is maximized for 18 generations. However, the average value of NCMC is not maximised until 200 generations. This shows that the convergence of WMN-

GA is slower compared with other simulation systems.

In Fig. 9, Fig. 10 and Fig. 11, we increase the values of number of phases and populations from 200 to 2000. We can see that all simulation systems have the maximum values for

(a) Giant component

(b) Covered mesh clients

Figure 10. Performance evaluation for WMN-SA.

(a) Giant component

(b) Covered mesh clients

Figure 11. Performance evaluation for WMN-GA.

both the size of GC and NCMC, but the GA is faster for getting the maximum value of GC.

V. CONCLUSIONS

In this paper, we conducted simulations with our implemented WMN-HC, WMN-SA and WMN-GA simulation systems, where we deployed 16 mesh routers and 48 mesh clients. Using our simulation systems, we optimized the size of GC and then the NCMC.

From the simulation results, we conclude as following.

- Convergence of WMN-HC is the fastest. On the other hand, WMN-GA is the slowest among 3 algorithms.
- All algorithms had the maximum size of GC and NCMC. However, the convergence time is different.

- For 200 iterations and populations, the local search algorithms are better, thus WMN-HC, WMN-SA have better performance than WMN-GA.
- For 2000 iterations and populations, all simulation systems have the maximum values for both the size of GC and NCMC, but the GA is faster for getting the maximum value of GC.

In our future work, we would like to make evaluations for different scenarios. Moreover, we would like to implement other search optimization algorithms in our simulation system.

REFERENCES

- [1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey", *Computer Networks*, Vol. 47, No. 4, pp. 445-487, 2005.

- [2] N. Nandiraju, D. Nandiraju, L. Santhanama, B. He, J. Wang, and D. Agrawal, "Wireless Mesh Networks: Current Challenges and Future Direction of Web-in-the-Sky", *IEEE Wireless Communications*, pp. 79-89, 2007.
- [3] Ch. Chen and Ch. Chekuri. "Urban Wireless Mesh Network Planning: The Case of Directional Antennas", Tech Report No. UIUCDCS-R-2007-2874, Department of Computer Science, University of Illinois at Urbana-Champaign, 2007.
- [4] M.R. Garey and D.S. Johnson, "Computers and Intractability -A Guide to the Theory of NP-Completeness", Freeman, San Francisco, 1979.
- [5] A. Lim, B. Rodrigues, F. Wang and Zh. Xua, " k -Center Problems with Minimum Coverage", *Theoretical Computer Science*, Vol. 332, No. 1-3, pp. 1-17, 2005.
- [6] E. Amaldi, A. Capone, M. Cesana, I. Filippini, F. Malucelli. "Optimization Models and Methods for Planning Wireless Mesh Networks", *Computer Networks*, Vol. 52, pp. 2159-2171, 2008.
- [7] J. Wang, B. Xie, K. Cai and D.P. Agrawal, "Efficient Mesh Router Placement in Wireless Mesh Networks", *MASS-2007*, Pisa, Italy, pp. 9-11, 2007.
- [8] S. N. Muthaiah and C. Rosenberg, "Single Gateway Placement in Wireless Mesh Networks", In *Proc. of 8th International IEEE Symposium on Computer Networks*, Turkey, pp. 4754-4759, 2008.
- [9] P. Zhou, B.S. Manoj and R. Rao, "A Gateway Placement Algorithm in Wireless Mesh Networks", *Proc. of the Third Annual International Wireless Internet Conference (WICON-2007)*, pp. 1-9, October 2007.
- [10] M. Tang, "Gateways Placement in Backbone Wireless Mesh Networks", *International Journal of Communications, Network and System Sciences*, Vol. 2, No.1, pp. 45-50, 2009.
- [11] A. Antony Franklin and C. Siva Ram Murthy, "Node Placement Algorithm for Deployment of Two-Tier Wireless Mesh Networks", In *Proc. of IEEE GLOBECOM-2007*, Washington, USA, pp. 4823-4827, 2007.
- [12] T. Vanhatupa, M. Hännikäinen and T.D. Hämäläinen, "Genetic Algorithm to Optimize Node Placement and Configuration for WLAN Planning", In *Proc. of 4th International Symposium on Wireless Communication Systems*, pp. 612-616, 2007.
- [13] S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli, F. Xhafa, "Performance Evaluation Considering Iterations per Phase and SA Temperature in WMN-SA System", *International Journal of Mobile Information Systems (MIS)*, Vol. 10, No. 3, pp.321-330, November 2013.
- [14] T. Oda, A. Barolli, E. Spaho, F. Xhafa, L. Barolli and M. Takizawa, "Evaluation of WMN-GA for Different Mutation Operators", *International Journal of Space-Based and Situated Computing*, Vol. 2, No. 3, pp. 149-157, 2012.
- [15] S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli, F. Xhafa, "A Comparison Study of Simulated Annealing and Genetic Algorithm for Node Placement Problem in Wireless Mesh Networks". *Journal of Mobile Multimedia (JMM)*, Vol. 9, No. 1&2, pp. 101-110, November 2013.
- [16] T. Oda, A. Barolli, F. Xhafa, L. Barolli, M. Ikeda and M. Takizawa, "Performance Evaluation of WMN-GA for Different Mutation and Crossover Rates Considering Number of Covered Users Parameter", *Mobile Information Systems*, Vol. 8, No. 1, pp. 1-16, 2012. DOI:10.3233/MIS-2011-0128
- [17] Y. S. Lee, J. W. Park and L. Barolli, "A Localization Algorithm Based on AOA for Ad-hoc Sensor Networks", *Mobile Information Systems*, Vol. 8, No. 1, pp. 61-72, 2012. DOI:10.3233/MIS-2012-0131
- [18] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", *Journal of Science*, Vol. 220, pp. 671-680, 1983.
- [19] F. Xhafa, C. Sanchez, L. Barolli, R. Miho, "An Annealing Approach to Router Nodes Placement Problem in Wireless Mesh Networks", In *Proc. of CISIS-2010*, pp. 245-252, 2010.
- [20] F. Xhafa, C. Sanchez, L. Barolli, "Genetic Algorithms for Efficient Placement of Router Nodes in Wireless Mesh Networks", *Proc. of AINA-2010*, pp. 465-472, 2010.