

A Novel PSO-Based Algorithm for the Optimal Location of Controllers in Wireless Networks

Dac-Nhuong Le[†], Nhu Gia Nguyen^{††}, and Vinh Trong Le^{†††}.

[†] Faculty of Information Technology, Haiphong University, Vietnam

^{††} Duytan University, Danang, Vietnam

^{†††} Hanoi University of Science, Vietnam National University, Vietnam

Summary

The optimal location of controllers in wireless networks is an important problem in the process of designing cellular mobile networks. In this paper we present a novel algorithm based on Particle Swarm Optimization (PSO) for solving it. Our objective function is determined by the total distance based on finding maximum flow in a transport network using Ford-Fulkerson algorithm satisfies capacity constraints. The experimental results show that our proposed algorithm has achieved much better performance than previous heuristic algorithms.

Key words:

Terminal Assignment (TA), Optimal Location of Controllers Problem (OLCP), Particle Swarm Optimization (PSO), Wireless Networks.

1. Introduction

In the designing of a mobile phone network (cellular network) it is very important to place the base stations optimally for a cheaper and better customer service. This issue is related to the problems of location of devices (Base station (BTS), Multiplexers, Switches, etc.) [1-2].

The objective of terminal assignment problem (TA) [3] involves with determining minimum cost links to form a network by connecting a given collection of terminals to a given collection of concentrators. The capacity requirement of each terminal is known and may vary from one terminal to another. The capacity of concentrators is known. The cost of the link from each terminal to each concentrator is also known. The problem is now to identify for each terminal the concentrator to which it should be assigned, under two constraints: Each terminal must be connected to one and only one of the concentrators, and the aggregate capacity requirement of the terminals connected to any concentrator must not exceed the capacity of that concentrator.

The assignment of BTSs to switches (controllers) problem introduced in [4]. In which it is considered that both the BTSs and controllers of the network are already positioned, and its objective is to assign each BTSs to a controller, in such a way that a capacity constraint has to be fulfilled. The objective function in this case is then

formed by two terms: the sum of the distances from BTSs to the switches must be minimized, and also there is another term related to *handovers*, between cells assigned to different switches which must be minimized. The optimal location of controller problem (OLCP) [5] is selecting N controllers out of M BTSs, in a way that the objective function given by solving the corresponding TA with N concentrators and $M-N$ terminals is minimal.

Both TA and OLCP are NP-hard optimization problems so heuristic approach is a good choice. In [1], a simulated annealing (SA) algorithm tackled the assignment of cells to controller problem. The results obtained are compared with a lower bound for the problem, and the authors show that their approach is able to obtain solutions very close to the problem's lower bound. Authors in [5] have introduced a hybrid heuristic consisting of SA and a Greedy algorithm for solving the OLCP problem. In [6-7], authors proposed a hybrid heuristic based on mixing genetic algorithm (GA), Tabu Search (TS) to solving the BTS-controller assignment problem in such a way that terminal is allocated to the closest concentrator if there is enough capacity to satisfy the requirement of the particular terminal.

In this paper, we propose a novel Particle Swarm Optimization (PSO) algorithm [8] based on Ford-Fulkerson algorithm find maximum flow in networks for the optimal location of controllers in a mobile communication network. Numerical results show that our proposed algorithm is much better than previous studies.

The rest of this paper is organized as follows. Section 2 presents the problem formulation and briefly introduces the main idea of OCLP proposed in [5]. Section 3 presents our new algorithm for location of controllers in a mobile communication network based on Particle Swarm Optimization algorithm. Section 4 presents our simulation and analysis results, and finally, section 5 concludes the paper.

2. Problem Formulation

Let us consider a mobile communication network formed by M nodes (BTSs), where a set of N controllers

must be positioning in order to manage the network traffic. It is always fulfilled that $N < M$, and in the majority of cases $N \ll M$. We start from the premise that the existing BTSs infrastructure must be used to locate the switches, since it saves costs. Thus, the OCLP consists of selecting N nodes out of the M which form the network, in order to locate in them N controllers. To define an objective function for the OCLP, we introduce a model for the problem, based on the Terminal Assignment Problem [5].

2.1 The Terminal Assignment Problem

The TA can be defined as follows [3]:

Problem instance:

Terminals: l_1, l_2, \dots, l_{M-N}

Weights: w_1, w_2, \dots, w_{M-N}

Concentrators: r_1, r_2, \dots, r_N

Capacities: p_1, p_2, \dots, p_N

Where, w_i is weight, or capacity requirement of terminal l_i . The weights and capacity are positive integers and

$$w_i < \min \{ p_1, p_2, \dots, p_N \}, \forall i = 1, 2, \dots, M - N \quad (1)$$

The terminals and concentrator are placed on the Euclidean grid, i.e., l_i has coordinates (l_{i1}, l_{i2}) and r_j has is located at (r_{j1}, r_{j2}) .

Feasible solution: Assign each terminal to one of concentrator such that no concentrator exceeds its capacity. Let $\hat{x} = \{ \hat{x}_1, \hat{x}_2, \dots, \hat{x}_{M-N} \}$ be a vector such that $\hat{x}_i = j$ means that terminal l_i has been assigned to concentrator r_j , with \hat{x} is an integer such that $1 \leq \hat{x} \leq N$.

Capacity of each concentrator must be satisfied:

$$\sum_{i \in R_j} w_i < p_j, j = 1..N \quad (2)$$

where, $R_j = \{ i \mid \hat{x}_i = j \}$, i.e., R_j represents the terminals that are assigned to concentrator r_j .

Objective function: Find \hat{x} that minimizes:

$$F(\hat{x}) = \sum_{i=1}^{M-N} \text{cost } t_{ij} \rightarrow \min, j = 1, 2, \dots, N \quad (3)$$

where, $\text{cost } t_{ij} = \sqrt{(l_{i1} - r_{j1})^2 + (l_{i2} - r_{j2})^2}$, i.e., the result of the distance between terminal l_i and concentrator r_j . It is important to note that in the standard definition of the TA, there is a major objective (the minimization of the distances between terminals and concentrators), and a major constraint (the capacity constraint of concentrators).

2.2 The Optimal Controller Location Problem

The complete OCLP has to deal with two issues, first, the selection of the N controllers in M nodes, second for each election, an associated TA. This process can be seen in Fig.1.

Authors in [5] used a Greedy algorithm to obtain this objective function that terminals are consequently allocated to the closest concentrator if there is enough capacity to satisfy the requirement of a particular terminal. If the concentrator cannot handle the terminal, the algorithm searches for the next closest concentrator and performed the same evaluation. The terminals are assigned to concentrators following the order in $\pi(l_{M-N})$ - a random permutation of terminals. That algorithm is called by *SA-Greedy* algorithm.

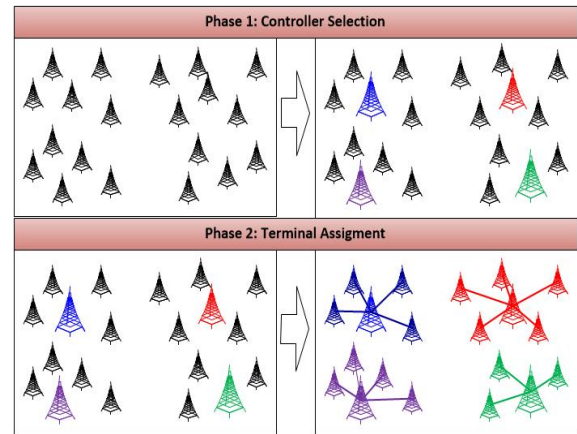


Fig.1. The Optimal Controller Location Problem

In [7], the authors considered the following *Lower Bound* (LB) for the TA, as follows:

$$LB = \sum_{i=1}^{M-N} \min_k (d_{ik}) \quad (4)$$

The Lower Bound comes from the solution obtained by assigning each node i to the nearest controller k . Hybrid Lower Bound- Greedy algorithm is called by *LB-Greedy* algorithm.

3. Particle Swarm Optimization for the OCLP

3.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy [8], inspired by social behavior of bird flocking or fish schooling. It shares many similarities with other evolutionary computation techniques such as genetic algorithms (GA). The algorithm is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, the PSO algorithm has no evolution operators such as the crossover and the mutation operator.

In the PSO algorithm, the potential solutions, called particles, fly through the problem space by following the

current optimum particle. By observing bird flocking or fish schooling, we found that their searching progress has three important properties. First, each particle tries to move away from its neighbors if they are too close. Second, each particle steers towards the average heading of its neighbors. And the third, each particle tries to go towards the average position of its neighbors. Kennedy and Eberhart generalized these properties to be an optimization technique as below.

Consider the optimization problem P . First, we randomly initiate a set of feasible solutions; each of single solution is a "bird" in search space and called "particle". All of particles have *fitness values* which are evaluated by the *fitness function* to be optimized, and have *velocities* which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. The better solutions are found by updating particle's *position*. In iterations, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*.

After finding the two best values, the particle updates its velocity and positions with following equation: (5.a) (which use global best *gbest*) or (5.b) (which use local best *lbest*) and (6).

$$v[i] = v[i] + c_1 * rand() * (pbest[i] - present[i]) + c_2 * rand() * (gbest[i] - present[i]) \quad (5.a)$$

$$v[i] = v[i] + c_1 * rand() * (pbest[i] - present[i]) + c_2 * rand() * (lbest[i] - present[i]) \quad (5.b)$$

$$present[i] = present[i] + v[i] \quad (6)$$

In those above equation, *rand()* is a random number between 0 and 1; c_1 and c_2 are cognitive parameter and social parameter respectively.

PSO Algorithm

{FOR each particle

 Initialize particle

ENDFOR

DO

 FOR each particle

 Calculate fitness value

 IF the fitness value is better than the best fitness value (*pBest*) in history

 Set current value as the new *pBest*

 ENDIF

 ENDFOR

 Choose the particle with the best fitness value of all the particles as the *gBest* (or Choose the particle with the best fitness value of all the neighbors particles as the *lBest*)

 For each particle

 Calculate particle velocity according to (5.a) (or (5.b))

 Update particle position according to (6)

 ENDFOR

WHILE (stop condition is true)

The stop condition mentioned in the above algorithm can be the maximum number of interaction is not reached or the minimum error criteria are not attained.

3.2 Solving the OCLP based on PSO

In this section, we present application of PSO technique for the OCLP problem. Our novel algorithm is described as follows.

We consider that configurations in the evolution algorithm are sets of N nodes which will be evaluated as controllers for the network.

1) *Represent and decode a particle*: The encoding of the configuration is by means of binary string of length M , say $x = \{x_1, x_2, \dots, x_M\}$ where $x_i = 1$ in the binary string means that the corresponding node has been selected to be a controller, whereas a 0 in the binary string means that the corresponding node is not a controller, but serves as BTS. We must select N nodes to be the controllers of the network.

2) *Initiate population*: We use fully random initialization in order to initialize the population. After that, the particle x will have p 1s.

We present *Repair function* to ensure that all binary strings in the particles have exactly N 1s representing N controllers.

Repair function Algorithm ($x = \{x_1, x_2, \dots, x_M\}$)

Input: The particle x has p 1s

Output: The particle x will have exactly N 1s

IF $p < N$ THEN

 Adds $(N-p)$ 1s in random positions

ELSE

 Select $(p-N)$ 1s randomly and removes them from the binary string

3) *Fitness function*: Each particle x has exactly N 1s representing N controllers. We construct a transport network $G = (I, J, E)$ corresponding particle x , where $I = \{1, 2, \dots, N\}$ is the set of controllers, $J = \{1, 2, \dots, M - N\}$ is the set of BTSs and E is the set of edge connections between controller r_i and the BTS l_j .

We find the maximum flow (max-flow) of the transport network G by adding two vertices S (Source) and D (Destination) is shown in Fig.2.

The weight of the edges on the graph is defined as follows:

- The edges from vertex S to the controllers r_i is capacity of r_i , denoted as $c(S, r_i) = p_i$, ($i=1..N$).
- The edges from BTS l_j to vertex D is weight of l_j , denoted as $c(l_j, D) = w_j$, ($j=1..M-N$).
- The edges from the controllers r_i to the BTSs l_j is denoted as $c(r_i, l_j) = w_{ij}$, ($(i, j) \in E$).

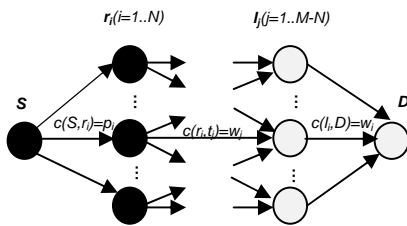


Fig. 2: The transport network $G = (I, J, E)$ corresponding particle x

From the transport network G , we find the max-flow satisfies capacity constraints given by the formula (4) based on Ford-Fulkerson algorithm [9].

The fitness value of this particle is computed with the max-flow based on the total distance is given by:

$$F(x) = \sum_{i=1}^N \sum_{j=1}^{M-N} \sqrt{(r_{i1} - l_{j1})^2 + (r_{i2} - l_{j2})^2} \quad (7)$$

4) *Stop condition*: The stop condition used in this paper is defined as the maximum number of interaction N_{gen} (N_{gen} is also a designated parameter).

4. Experiments and Results

For the experiments, we have tackled several OCLP instances of different difficulty levels. There are 10 OCLP instances with different values for N and M , and size networks shown in Table 1.

Table 1: Main characteristic of the problems tackled

Problem #	Nodes (M)	Controllers (N)	Grid size
1	10	2	100x100
2	15	3	100x100
3	20	4	100x100
4	40	6	200x200
5	60	8	200x200
6	80	10	400x400
7	100	15	600x600
8	120	20	800x800
9	150	25	1000x1000
10	200	50	1500x1500

In our experiments, we have already defined parameters for the PSO as below:

- Population size $P = 1000$
- Maximum number of interaction $N_{gen} = 500$
- Cognitive parameter $c_1 = 1$
- Social parameter $c_2 = 1$
- Update population according to (5.b) and (6) with number of neighbor is $K = 3$

The experimental results of our algorithm in Table 2 show the objective function has achieved a much better performance than other algorithms. The results show that problems with the small grid size and small number of nodes such as problem #1, #2 and #3, all algorithms has approximate results.

Table 2: Results obtained in the OCLP instances tackled

Prob.#	SA	SA-Greedy	LB-Greedy	PSO
1	187.4	187.4	187.4	187.4
2	315.0	315.0	315.0	315.0
3	428.3	427.2	419.6	412.7
4	1784.7	1798.5	1658.2	1615.3
5	2135.9	2215.1	1976.3	1911.9
6	4863.2	4863.2	4627.5	4503.4
7	7955.6	8027.2	7371.9	7137.1
8	12863.7	13753.8	10863.7	9563.6
9	23638.6	26624.3	19569.2	16861.3
10	157894.2	168253.7	143665.4	141235.8

However, when the problem size is large, the experimental results are considerable different such as problem #6, #7, #8, #9 and #10. In some cases, *LB-Greedy*, *SA-Greedy* and *PSO* algorithms choose the same set of nodes to be controllers, but the objective function of *PSO* is much better, as can be seen in Fig.3.

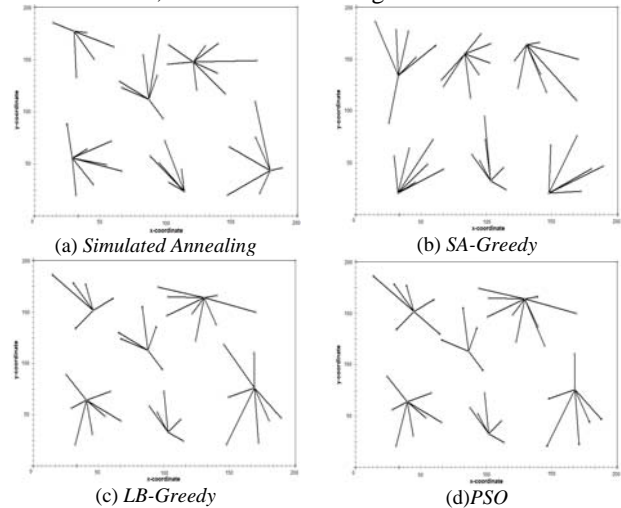


Fig. 3: Solutions for the problem #4 given by SA, SA-Greedy, LB-Greedy and PSO algorithms

5. Conclusion and Future Work

In this paper we have presented a novel particle swarm optimization algorithm for the optimal location of controllers in wireless networks, which is an important problem in the process of designing cellular mobile networks. Our objective function is determined by the total distance based on finding max-flow in a transport network using Ford-Fulkerson algorithm satisfying capacity constraints.

The experimental results show that our proposed algorithm has achieved a much better performance than previous studies. It is also proved to be a cost-effective solution. Optimizing location of controllers in wireless networks with profit, coverage area and throughput maximization is our next research goal.

Acknowledgments

This research is partly supported by research projects of Vietnam National University, Hanoi.

References

- [1] B. Krishnamachari and S. Wicker, *Base station location optimization in cellular wireless networks using heuristic search algorithms*, Soft Computing in Communications, L. Wang (Edt), Springer, 2003.
- [2] S. Menon and R. Gupta, Assigning cells to switches in cellular networks by incorporating a pricing mechanism into simulated annealing, *IEEE Trans. Syst. Man Cybern. B*, vol. 34, no. 1, pp. 558-565, 2004.
- [3] F. N. Abuali, D. A. Schoenefeld and R. L. Wainwright, Terminal assignment in a communications network using genetic algorithms, *In Proc. 22nd Annual ACM Computer Science Conference*, pp. 74-81, ACM press, 1994.
- [4] A. Merchant and B. Sengupta, *Assignment of cells to switches in PCS networks*, *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp.521-521, 1995.
- [5] S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández, J. A. Martínez-Rojas, *A Hybrid Greedy-Simulated Annealing algorithm for the optimal location of controllers in wireless networks*, *Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Databases*, Madrid, Spain, pp.159-164, 2006.
- [6] F. Houeto and S. Pierre, *A Tabu search approach for assigning cells to switches in cellular mobile networks*, *Computer Commun.*, vol. 25, no. 3, pp. 464-477, 2002.
- [7] A. Quintero and S. Pierre, *Assigning cells to switches in cellular mobile networks: a comparative study*, *Computers Commun.*, vol. 26, pp. 950-960, 2003.
- [8] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publisher Inc, 2001.
- [9] L.R. Ford, Jr. and D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, New Jersey , 1962.
- [10] Vinh Trong Le, Nhu Gia Nguyen, Dinh Huu Nghia, "A Novel PSO-Based Algorithm for Gateway Placement in Wireless Mesh Networks", in *Proc. 3rd IEEE International Conference on Communication Software and Networks (ICCSN)*, China, 2011, pp. 41-46.



and networks security.

Dac-Nhuong Le received the BSc degree in computer science and the MSc degree in information technology from College of Technology, Vietnam National University, Vietnam, in 2005 and 2009, respectively. He is currently a lecturer at the Faculty of information technology in Haiphong University, Vietnam. His research interests include algorithm theory, computer network



Nhu Gia Nguyen received the BSc degree in computer science and the MSc degree in information technology from Danang University, Vietnam, in 1998 and 2006, respectively. He currently works in Duytan University, Danang, Vietnam. His research interests include algorithm theory, network and wireless security.



network and wireless security.

Vinh Trong Le received the MSc degree in information technology from Faculty of Mathematics, Mechanics and Informatics, Hanoi University of Science, Vietnam National University in 1997, PhD degrees in computer science from Japan Advanced Institute of Science and Technology in 2006, respectively. He is currently a lecturer at the Faculty of Mathematics, Mechanics and Informatics, Hanoi University of Science, Vietnam National University. His research interests include algorithm theory, network and wireless security.