EA and ACO Algorithms Applied to Optimizing Location of Controllers in Wireless Networks

Dac-Nhuong Le, Hanoi University of Science, Vietnam National University, Vietnam

Optimizing location of controllers in wireless networks is an important problem in the cellular mobile networks designing. In this paper, I present two algorithms based on Evolutionary Algorithm (EA) and Ant Colony Optimization (ACO) to solve it. In the first algorithm, my objective function is determined by the total distance based on finding maximum flow in a bipartite graph using Ford-Fulkerson algorithm. In the second algorithm, I generate pheromone matrix of ants and calculate the pheromone content of the path from controller i to base station j using the neighborhood includes only locations that have not been visited by ant k when it is at controller i. At each step of iterations, I choose good solutions satisfying capacity constraints and update step by step to find the best solution depending on my cost functions. I evaluate the performance of my algorithms to optimize location of controllers in wireless networks by comparing to SA, SA-Greedy, LB-Greedy algorithm. Numerical results show that my algorithms proposed have achieved much better more than other algorithms.

Categories and Subject Descriptors: COMPUTING [**C.2**] COMPUTER-COMMUNICATION NETWORKS

General Terms: Networking

Additional Key Words and Phrases: Terminal Assignment (TA), Optimal Location of Controllers Problem (OLCP), Evolutionary Algorithm (EA), Ant Colony Optimization (ACO), Wireless Networks

ACM Reference Format:

Dac-Nhuong Le **Research on Newsroom Security Challenges**, *International Journal of Computer Communications and Networks (IJCCN)*, 3(2), pp 17–27, 2013

1. INTRODUCTION

In a mobile communication network designing, base stations placement optimization is very important for cheaper and better customer services. This issue is re-lated to the problems of location of devices (Base station (BTS), Multiplexers, Switches, etc.) [Wicker and S. 2003],[Menon and Gupta 2004]. The objective of terminal assignment problem (TA)[F. N. Abuali and Wainwright 1994] 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 and 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 [Networks 1995]. 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. In this case, the objective function is formed by two terms: the sum of the distances from BTSs to the switches, and also there is another term related to handovers, between cells assigned to different switches which must be minimized.

Corresponding Author: Dac-Nhuong Le, Dac-Nhuong Le is with the Hanoi University of Science, Vietnam National University, Hanoi, Vietnam. *or* nhuongld@hus.edu.vn

e-ISSN (online) : 2289-3369, ISSN (printed) : 2289-3350

The optimal location of controller problem (OLCP) [S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández 2006] 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. 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 [Wicker and S. 2003]. S. Salcedo Sanz et al in [S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández 2006] introduced a hybrid heuristic consisting of SA and Greedy algorithm to solve the OLCP problem. In [Pierre and Houéto 2002], [Quintero and Pierre 2003] 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 the latest paper [Le et al. 2012], I have proposed a new algo-rithm based on Particle Swarm Optimization (PSO) [Kennedy and Eberhart 1995]. In this paper, I proposed two algorithms based on EA and ACO algorithms. In the first algorithm, my objective function is determined by the total distance based on finding maximum flow in a transport network using Ford-Fulkerson algorithm. In the second algorithm, I generate pheromone matrix of ants and calculate the pheromone content of the path from controller i to base station j using the neighborhood includes only locations that have not been visited by ant k when it is at controller i. At each step of iterations, I choose good solutions satisfies capacity constraints and update step by step to find the best solution depends on my cost function. The experimental results show that my algorithms proposed have achieved much better performance and easy more than previous algorithms. 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 [S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández 2006]. Section 3 and section 4 present my new algorithm for location of controllers in a mobile communication network based on EA and ACO algorithms. Section 5 presents my simulation and analysis results, and finally, section 6 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 . I 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, I introduce a model for the problem, based on the Terminal Assignment Problem [S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández 2006].

2.1 The Terminal Assignment Problem

The TA [F. N. Abuali and Wainwright 1994] can be defined as follows:

Problem instance	
Terminals:	$l_1, l_2, ..., l_{M-N}$
Weights:	$w_1, w_2, ..., w_{M-N}$
Concentrators:	$t_1, t_2, ..., t_N$
Capacities:	$p_1, p_2, ..., p_N$

where, w_i is weight, or capacity requirement of terminal l_i. The weights and capacities are positive integers satisfied:

$$w_i < min\{p_1, p_2, ..., p_N\}, \forall_i = 1, 2, ..., M - N \tag{1}$$

The terminals and concentrator are placed in the Eu-clidean 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. I use $\hat{x} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{M-N}\}$ is 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:

$$\sum_{i \in R_j} w_i < p_i, j = 1, ..., N \tag{2}$$

where,$R_j = \{i|\hat{x}_i = j\}$,i.e., R_j represents the terminals that are assigned to concentrator r_j.
Objective function: Find $\hat{x}$ that minimizes:

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

where, $cost\ t_{ij} = \sqrt{(l_{i1} - r_{j1})^2 + (l_{i2} - r_{j2})^2}$ is 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 OCLP complete has to deal with two issues. The first is selection of the N controllers in M nodes, and the second for each election, an associated TA. This process can be seen in Fig. 1. Authors in [S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández 2006] 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. A. Quintero and S. Pierre considered the following *Lower Bound* (LB) for the TA [Quintero and Pierre 2003], as follows:

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

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. EVOLUTIONARY ALGORITHM FOR THE OCLP

3.1 Initialization

I consider configurations in the EA algorithm are sets of N nodes which will be evaluated as controller for the network. The encoding of the configuration use a binary string with length M, say $x = \{x_1, x_2, ...x_M\}$ where $x_i = 1$ in the binary string means that the corresponding node has been selected to be a controller, whereas $x_i = 0$ in the binary string means that the corresponding node is not

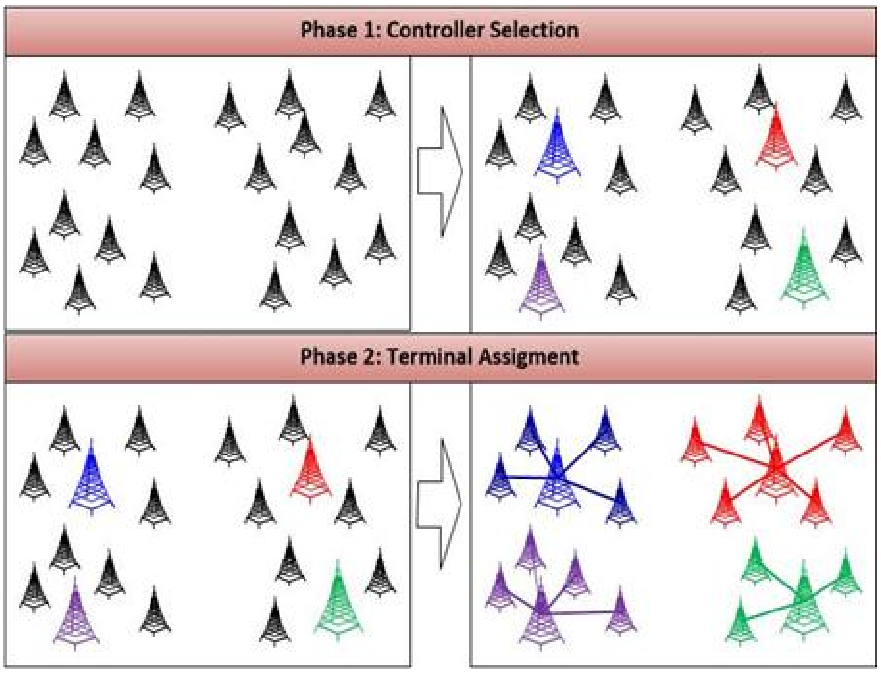


Fig. 1. The Optimal Controller Location Problem

a controller, but serve as BTS. I must select N nodes to be the controllers of the network.
I use fully random initialization in order to initialize the individuals. After the mutation, the individual x will have p 1s. To ensure that all binary strings in individuals have exactly N 1s representing N controllers, I present *Repair Individual function* as follows:

Repair Individual Function Algorithm	
INPUT	The $individual$ $x = \{x_1, x_2, ...x_M\}$ has p 1s
OUTPUT	The $individual$ 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
ENDIF	

3.2 Evaluation Function

After the mutation, each individual x has exactly N 1s representing N controllers. I will construct a bipartite graph $G = (I, J, E)$ corresponding individual x, where $I = \{1, 2, ..., N\}$ is the set of controllers, $J = \{1, 2, ..., M - N\}$ is the set of BTSs and E is the set of edge connection between the controller r_i and the BTS l_j.
I find the maximum flow *(max-flow)* of the bipartite graph 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 is $c(S, r_i) = p_i,\ (i = 1, ..., N)$.
—The edges from BTS l_j to vertex D is weight of l_j is $c(l_j, D) = w_j,\ (j = 1, ..., M - N)$.
—The edges from the controllers r_i to the BTSs l_j is $c(r_i, l_j) = w_i,\ ((i, j) \in E)$.

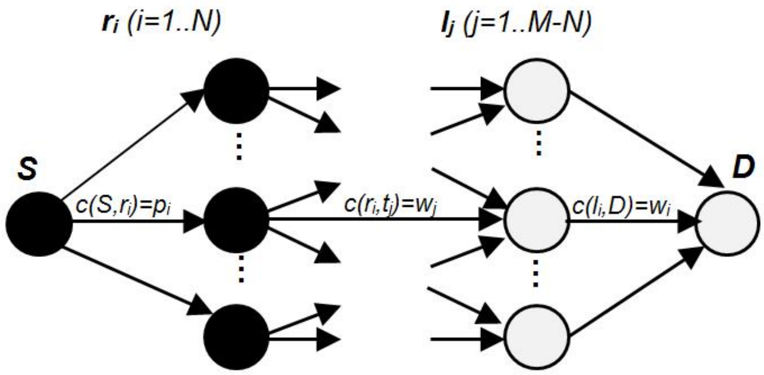


Fig. 2. The bipartite graph $G = (I, J, E)$ corresponding $individualx$

I find the max-flow satisfies capacity constraints given by the formula 4 in graph G based on Ford-Fulkerson algorithm [L.R. Ford and Fulkerson 1962].
The objective function is determined by the max-flow based on the total distance is given by:

$$F(x) = \sum_{i=1}^{N} \sum_{j=1}^{M-N} \sqrt{(l_{i1} - r_{j1})^2 + (l_{i2} - r_{j2})^2} \tag{5}$$

3.3 My Evolutionary Algorithm

The pseudo-code of the EA algorithm using Ford-Fulkerson algorithm for the optimal location of controllers in a mobile communication network, as follows:

EVALUTIONARY ALGORITHM (EA)

BEGIN
INITIALISE *population* with random candidate solutions;
REPAIR FUNCTION (candidate);
EVALUATE FUNCTION each candidate;
REPEAT
1. *SELECT* parents;
2. *RECOMBINE* pairs of parents;
3. *MUTATION* the resulting offspring;
4. REPAIR FUNCTION (candidates);
5. BUILD BIPARTITE GRAPH $G = (I, J, E)$;
6. FIND MAX_FLOW (G);
7. *EVALUATE FUNCTION* new candidates;
8. *SELECT* individuals for the next generation;
UNTIL (*TERMINATION CONDICTION* is satisfied)
END

4. ANT COLONY OPTIMIZATION FOR THE OCLP

4.1 Ant Colony Optimization

The ACO algorithm is originated from ant behavior in the food searching. When an ant travels through paths, from nest food location, it drops pheromone. According to the pheromone concentration the other ants choose appropriate path. The paths with the greatest pheromone concentration are the shortest ways to the food. The optimization algorithm can be developed from such ant behavior. The first ACO algorithm was the Ant System [Dorigo et al. 1996], and after then, other implementations of the algorithm have been developed [Dorigo et al. 2006].

4.2 Solving the OCLP based on ACO

Similar EA algorithm above, I use a binary string to en-coding the ant k configuration, say $k = \{x_1, x_2, ...x_M\}$ where $x_i = 1$ in the binary string means that the corresponding node has been selected to be a controller, whereas $x_i = 0$ in the binary string means that the corresponding node is not a controller, but serve as BTS. I must select N nodes to be the controllers of the network.
I use fully random initialization in order to initialize the ant population. After that, the ant k will have p 1s. I present *Ant_Repair* function to ensure that all binary strings in ants have exactly N 1s representing N controllers. In my case, the pheromone matrix is generated with matrix elements that represent a location for ant movement, and in the same time it is possible receiver location. Each ant k has exactly N 1s representing N controllers is associated to one matrix.

ANT_REPAIR FUNCTION ALGORITHM
Input: The ant $k = \{x_1, x_2, ...x_M\} has\ p$ 1s
Output: *The ant k 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*
ENDIF

I use real encoding to express an element of matrix A_{m*n} (where n is the number of controllers, m is number of BTSs). Each ant can move to any location according to the transition probability defined by:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]\beta}{\sum\limits_{l \epsilon N_i^k} [\tau_{ij}]\alpha[\eta_{ij}]\beta} \tag{6}$$

where,τ_{jj} is the pheromone content of the path from controller i to BTS j, N_i^k is the neighborhood includes only locations that have not been visited by ant k when it is at controller i, η_{ij} is the desirability of BTS j, and it depends of optimization goal so it can be my cost function.
The influence of the pheromone concentration to the probability value is presented by the constant α, while constant β do the same for the desirability. These constants are determined empirically and my values are $\alpha = 1$, $\beta = 10$. The ants deposit pheromone on the locations they visited according to the relation.

$$\tau_j^{new} = \tau_j^{current} + \Delta\tau_j^k \tag{7}$$

where $\Delta\tau_j^k$ is the amount of pheromone that ant k exudes to the BTS j when it is going from controller i to BTS j.

This additional amount of pheromone is defined by:

$$\Delta\tau_j^k = \frac{1}{d_{ij}} \tag{8}$$

In which, d_{ij} is the distance between controller i to BTS j is given by:

$$d_{ij} = \sqrt{(r_{i1} - l_{j1})^2 + (r_{i2} - l_{j2})^2} \tag{9}$$

The cost function for the ant k is the total distance between controllers to BTSs is given by:

$$f_k = \sum_{i=1}^{N} \sum_{j=1}^{M-N} \sqrt{(r_{i1} - l_{j1})^2 + (r_{i2} - l_{j2})^2} \tag{10}$$

The stop condition I used in this paper is defined as the maximum number of interaction N$_{max}$ (N$_{max}$ is also a designed parameter).

4.3 My ACO Algorithm

The pseudo-code of ACO algorithm to solving OCLP as follows:

```
ACO ALGORITHM
INITIALIZATION:
 Algorithm parameters:α = 1, β = 10
 Ant population size: K.
 Maximum number of iteration: N_Max.
GENERATION:
 Ant_Repair function (k) : ∀k ∈ K.
 Generating the pheromone matrix for the ant k.
 Update the pheromone values and set x∗ = k;
 i = 1.
REPEAT
 FOR k = 1 TO K DO
    Computing the cost function for the ant k by the formula (10)
    Computing probability move of ant individual by the formula (6)
    IF f(k) < f(x∗) THEN
      Update the pheromone values by the formula (7)
      Set x∗ = k.
    ENDIF
 ENDFOR
UNTIL i > N_Max
```

5. EXPERIMENTS AND RESULTS

5.1 The problems tackled

In my experiments, I 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 I.

Table I. Main Characteristic Of The Problems Tackled

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

Table II. Evolutionary Algorithm Specifications

Representation	Binary strings of length m
Recombination	One point crossover
Recombination probability	70%
Mutation	Each value inverted with independent probability p_m per position
Mutation probability p_m	1/m
Parent selection	Best out of random two
Survival selection	Generational
Population size	500
Number of offspring	500
Initialization	Random
Termination condition	No improvement in last 100 generations

Table III. The Aco Algorithm Specifications

Ant Population size	K = 100
Maximum number of interaction	N_{Max} = 500
Parameter	$\alpha = 1, \beta = 10$

5.2 Parameters for the EA algorithm

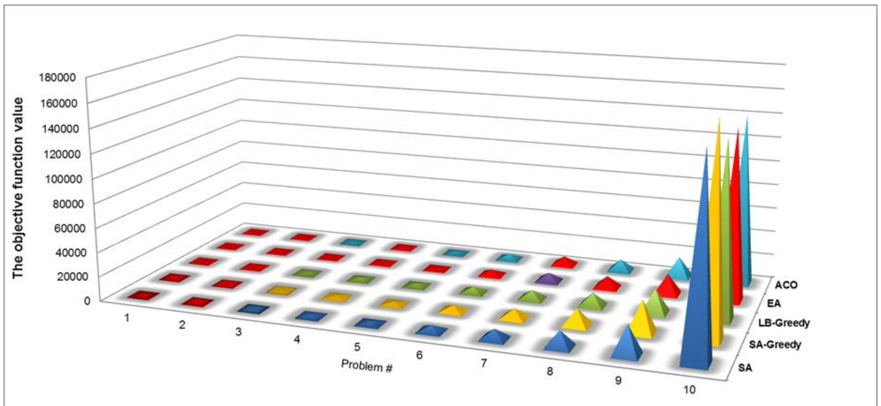
I have already defined my crossover probability as 0.7, I will work with a population size of 500 and a mutation rate of $p_m = 1/m$.
My evolutionary algorithm to tackle these problems can be specified as below in Table II.

5.3 Parameters for the ACO algorithm

In my experiments, I have already defined parameters for the ACO algorithm shown in Table III:

5.4 Numerical Analysis

I evaluate the performance of my algorithms to optimize location of controllers in wireless networks by comparing to SA, SA-Greedy, LB-Greedy algorithm. The experimental objective function results show in Fig 3.
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. 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-Greed*, EA and ACO algorithms choose the same set of nodes to be controllers, but the objective function results of EA or ACO are much better. The results show that ACO has better properties compared to EA algorithm. Fig.4 shows the results of the simulator of solutions for the problem #4 given by SA, SA-Greedy, LB-Greedy, EA and ACO algorithms.



Problem #	1	2	3	4	5	6	7	8	9	10
SA	187.4	315	428.3	1784.7	2135.9	4863.2	7955.6	12863.7	23638.6	157894.2
SA-Greedy	187.4	315	427.2	1798.5	2215.1	4863.2	8027.2	13753.8	26624.3	168253.7
LB-Greedy	187.4	315	419.6	1658.2	1976.3	4627.5	7371.9	10863.7	19569.2	143665.4
EA	187.4	315	415.4	1615.3	1910.6	4507.8	7144.1	9584.3	16896.7	141276.9
ACO	187.4	315	418.7	1615.3	1916.6	4518.1	7136.5	9578.4	16874.7	141257.2

Fig. 3. Results obtained in the OCLP instances tackled

6. CONCLUSION AND FUTURE WORK

In this paper, I have proposed two new algorithms based on Evolutionary Algorithm and Ant Colony Optimization for the optimal location of controllers in wireless networks, which is an important problem in the process of mobile communication network designing. In the first algorithm, my objective function is determined by the total distance based on finding maximum flow in a bipartite graph using Ford-Fulkerson algorithm. In the second algorithm, I generate pheromone matrix of ants and calculate the pheromone content of the path from controller i to base station j using the neighborhood includes only locations that have not been visited by ant k when it is at controller i. At each step of iterations, I choose good solutions satisfies capacity constraints and update step by step to find the best solution depends on my cost functions.

I evaluate the performance of my algorithms to optimize location of controllers in wireless networks by comparing to SA, SA-Greedy, LB-Greedy algorithm. Numerical results show that my algorithms proposed have achieved much better more than other algorithms. Optimizing location of controllers in wireless networks with profit, coverage area and throughput maximization is my next research goal.

REFERENCES

Marco Dorigo, Mauro Birattari, and Thomas Stutzle. 2006. Ant colony optimization. *IEEE Computational Intelligence Magazine* 1, 4 (Nov. 2006), 28–39. DOI:http://dx.doi.org/10.1109/MCI.2006.329691

Marco Dorigo, Vittorio Maniezzo, and Albert Colorni. 1996. Ant System : Optimization by a Colony of Cooperating Agents. 26, 1 (1996).

D. A. Schoenefeld F. N. Abuali and R. L. Wainwright. 1994. Terminal assignment in a communications network using genetic algorithms. *In Proc. 22sd Annual ACM Computer Science Conference* (1994), pp.74–81.

James Kennedy and Russell Eberhart. 1995. Particle Swarm Optimization. (1995), 1942–1948.

Dac-nhuong Le, Nhu Gia Nguyen, and Vinh Trong Le. 2012. A Novel PSO-Based Algorithm for the Optimal Location of Controllers in Wireless Networks. 12, 8 (2012), 23–27.

Jr. L.R. Ford and D.R. Fulkerson. 1962. Flows in Networks. *Princeton University Press, Princeton* (1962).

Syam Menon and Rakesh Gupta. 2004. Assigning Cells to Switches in Cellular Networks by Incorporating a Pricing Mechanism Into. 34, 1 (2004), 558–565.

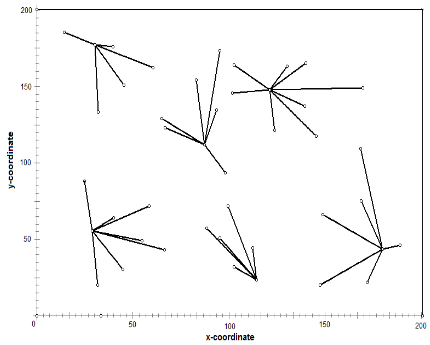
P C S Networks. 1995. Assignment of Cells to Switches in PCS Networks. 3, 5 (1995), 521–526.

Samuel Pierre and Fabien Houéto. 2002. A tabu search approach for assigning cells to switches in cellular mobile networks. *Computer Communications* 25, 5 (March 2002), 464–477. DOI:http://dx.doi.org/10.1016/S0140-3664(01)00371-1

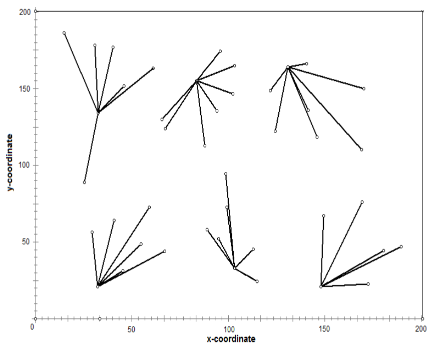
Alejandro Quintero and Samuel Pierre. 2003. Assigning cells to switches in cellular mobile networks : a comparative study. 26 (2003), 950–960.

J. A. Martínez-Rojas S. Salcedo-Sanz, A. Portilla-Figueras, S. Jiménez-Fernández. 2006. 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* (2006), pp.159–164.

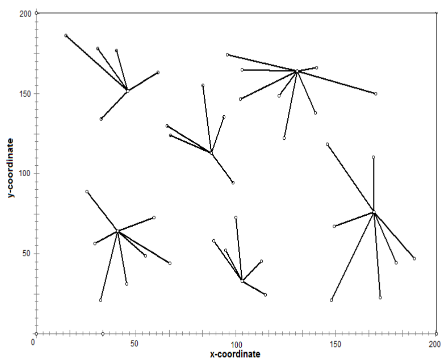
B. Krishnamachari Wicker and S. 2003. Base station location optimization in cellular wireless networks using heuristic search algorithms. *Soft Computing in Communications L. Wang (Edt), Springe* (2003).



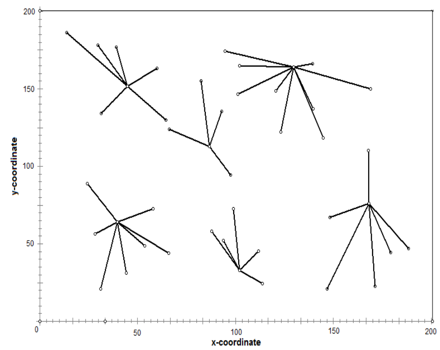
(a) *Simulated Annealing algorithm*



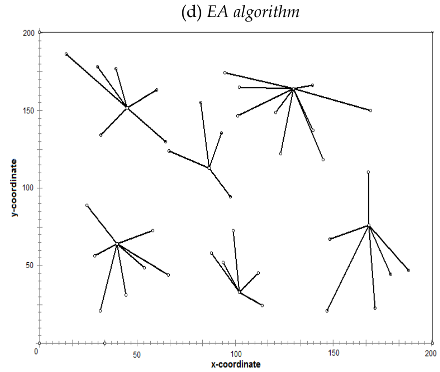
(b) SA-Greedy algorithm



(c) *LB-Greedy algorithm*



(d) *EA algorithm*



(e) *ACO algorithm*