

A Hybrid Greedy-Simulated Annealing algorithm for the optimal location of controllers in wireless networks

S. SALCEDO-SANZ, A. PORTILLA-FIGUERAS, S. JIMÉNEZ-FERNÁNDEZ, J. A. MARTÍNEZ-ROJAS

Department of Signal Theory and Communications

Universidad de Alcalá

Escuela Politécnica, Alcalá de Henares, Madrid

ESPAÑA

Abstract: - This paper presents preliminary work on the location of controllers in wireless communication networks using hybrid heuristics. We propose a model in which N base stations out of the total M of the network must be selected as controllers. The primary design objective is that the distance between all base stations and the N controllers is minimized. There is also a constraint of capacity, which prevents a controller to manage an unlimited number of base stations. We have implemented and tested a greedy-simulated annealing algorithm to solve the problem, comparing its results with that of a lower bound for the problem.

Key-Words: - wireless networks, controllers location, heuristics, simulated annealing.

1 Introduction

Mobile communication networks are usually divided into hexagonal cells, each corresponding to a different cover zone, and associated to a given Base Station (BTS). A certain number of cells can be chosen to install controllers, which route calls to another base station or to a public switched telephone network.

The design of mobile networks often involves problems of location of devices (BTS, multiplexers, switches etc.) [1], [2]. A problem widely tackled in this context is the assignment of BTSs to switches (controllers) [3], 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 have to be fulfilled. The objective function in this case is then formed by two terms: the sum of the distances from the BTSs to the switches must be minimum, and also there is another term related to *handovers* between cells assigned to different switches which must be minimized (see [3] for details).

Hybrid algorithms have been applied to solve the BTS-controller assignment problem [4], [10]. Both papers deal with the same problem, discussing different approaches to it, based on mixing genetic algorithms (GA), Tabu Search (TS) and Simulated Annealing (SA). SA have also been also used in the design of the BTS-Switch structure of a mobile communication network. In [1], a SA algorithm with a pricing mechanisms is used to tackle the assignment

of cells to controllers 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. Note that the works carried out on the assignment of BTSs to controllers, consider that the position of controllers is given in advance. In this paper we propose a model for the optimal location of controllers in a mobile communication network, so it is a previous step to the BTS-controllers assignment problem.

Our model starts from the premise that the switches must be located in existing BTSs, in order to use their infrastructure, and save costs. We propose then a model to establish which ones are the optimal BTSs for allocating a given number N of switches. This model is based on the so-called *Terminal Assignment Problem* (TA) [7], which looks for assignments of BTS to controllers such that the distance between all base stations and the N controllers is minimized.

We propose a hybrid heuristic for solving the Optimal Location of Controllers Problem (OLCP hereafter), based on SA and on an existing local search greedy algorithm for the TA. We test our approach in several switch location problems, with different number of BTSs and switches available, and compare the results obtained with that of a lower bound for the OCLP.

The rest of the paper is structured as follows: next section defines the OCLP in a mobile communications network. In this section we also introduce the model we propose based on the TA. In Section 2 we present the hybrid Greedy-SA algorithm

for solving the OCLP. In Section 3 we test the proposed algorithms by means of several computational experiments, where the performance of the hybrid heuristic approach is studied and compared with a lower bound for the OCLP. Finally, Section 4 concludes the paper giving some remarks.

2 Problem Formulation

Let us consider a mobile communications 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 a well known problem, the Terminal Assignment Problem.

2.1 The Terminal Assignment Problem

The TA can be defined as follows. Given a set of:

- 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 the weight, or capacity requirement of terminal l_i (Note that we consider N concentrators and $M-N$ terminals in our network). The weights and capacities are positive integers and $w_i < \min\{p_1, p_2, \dots, p_{M-N}\}$ for $i=1, 2, \dots, M-N$. The $M-N$ terminals and N concentrators are placed on the Euclidean grid, i.e. l_i has coordinates (l_{i1}, l_{i2}) and r_j is located at (r_{j1}, r_{j2}) . The following definition for the TA, can be found in [7]:

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 j , with \hat{x}_i an integer such that $1 \leq \hat{x}_i \leq N$.

Find \hat{x} which minimize

$$Z(\hat{x}) = \sum_{i=1}^{M-N} cost_{ij}, j=1, 2, \dots, N. \quad (1)$$

subject to:

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

where $cost_{ij}$ is the cost for assigning terminal i to concentrator j (usually, the Euclidean distance between the terminal and its corresponding concentrator), and $R_j = \{i \mid \hat{x}_i = j\}$.e., R_j represents the set of terminals that are assigned to concentrator 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 consist of selecting N controllers out of M nodes, in such a way that the objective function given by solving the corresponding TA with N concentrators and $M-N$ terminals is minimal. Note that we have to solve two problems, first, the selection of the N controllers, second for each election, an associated TA. This process can be seen in Figure 1.

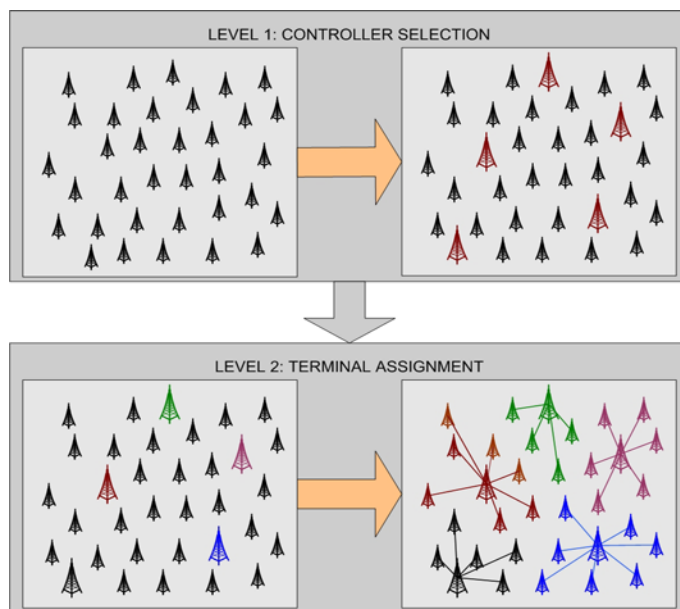


Figure 1. Structure of the Optimal Controller Location Problem.

3 A hybrid Greedy-SA algorithm for solving the OCLP

3.2 The simulated annealing algorithm

SA has been widely applied to solve combinatorial optimization problems [8], [9]. It is inspired by the physical process of heating a substance and then cooling it slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches to an equilibrium point, and no more changes happen. Each stage of the process consists in changing the configuration several times, until a thermal equilibrium is reached, and a new stage starts, with a lower temperature. The solution of the problem is the configuration obtained in the last stage. In the standard SA, the changes in the configuration are performed in the following way: A new configuration is built by a random displacement of the current one. If the new configuration is better, then it replaces the current one, and if not, it may replace the current one probabilistically. This probability of replacement is high in the beginning of the algorithm, and decreases in every stage. This procedure allows the system to move toward the best configuration. Although SA is not guaranteed to find the global optima, it is still better than others algorithms in escaping from local optima.

In this paper we consider that configurations in the SA are sets of N nodes which will be evaluated as controllers for the network. The encoding of the configurations is by means of binary strings, in such a way that a 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 serve as BTS. Due to we must select N nodes to be the controllers of the network, our SA searches for binary strings with exactly N 1s on them. Standard SA cannot manage the constraint of fixed number of 1s, and an extra operator has to be added to the standard SA, in the following way: after the application of the mutation in the SA, the individual x will have p 1s that, in general, will be different from the desired number of desired 1s in x , N . If $p < N$ the restricted search operator adds $(N-p)$ 1s in random positions, and if $p > N$, the restricted search operator randomly selects $(p-N)$ 1s and removes them from the binary string. With this procedure we ensure that all the

binary strings managed by the SA has exactly N 1s, representing N controllers. This procedure is called *restricted search*.

Standard Simulated Annealing

```

BEGIN
  Set initial temperature of the system,  $T_0$ ;
  Generate initial state  $s(T_0)$  randomly,  $i \leftarrow 0$ ;
  REPEAT
     $T_i = \frac{T_0}{i+1}$ ;
    REPEAT
      Apply mutation to the current state  $s(T_i)$ ;
      Calculate fitness of the current ( $s(T_i)$ ) and mutated ( $s_{mut}(T_i)$ ) solutions.
      If  $s_{mut}(T_i)$  is better than  $s(T_i)$ , then  $s(T_i) = s_{mut}(T_i)$ .
      If  $s_{mut}(T_i)$  is worse than  $s(T_i)$ , then
        generate random number  $u$ ,  $u \in [0, 1]$ .
        if  $e^{(-\frac{\Delta}{T_i})} > u$  then  $s(T_i) = s_{mut}(T_i)$ , else discard mutated solution  $s_{mut}(T_i)$ .
    UNTIL Max_mutations;
     $i = i + 1$ 
  UNTIL  $i = \text{Max\_temp\_changes}$ ;
END

```

Figure 2. Pseudo-code of the standard SA

Figure 2 shows the pseudo-code of the standard SA implemented. The state $s(T)$ stands for the binary string with N 1s, and the mutated state $s_{mut}(T)$ stands for a binary string after the application of a flip mutation operator and the restricted search operator. The variables `Max_temp_changes` and `Max_mutations` stand for the maximum temperature changes and maximum number of mutations in each temperature, respectively. The initial temperature of the system T_0 is chosen in such a way that the initial probability of acceptance worse solutions is 0.8, a standard value for the SA. Note that this probability will decrease with the temperature of the system. The states $s(T)$ and $s_{mut}(T)$ will be better or worse than the other on the basis of an objective function, which in this case is obtained solving a TA, i.e. we consider equation (1) as the objective function to the OCLP. We use a Greedy algorithm to obtain this objective function.

3.3 The Greedy Algorithm

One of the most important papers on TA was the approach by Abuali et al. [7]. In this article the authors proposed a Greedy algorithm for solving the TA. This greedy approach uses the notation given in Section 2, and starts from a random permutation of

terminals $\pi(l_{M-N})$. Then, the cost function $cost_{ij}$ is the Euclidean distance between terminal i and concentrator j . The terminals are assigned to concentrators following the order in $\pi(l_{M-N})$, in such a way that a terminal is allocated to the closest concentrator if there is enough capacity to satisfy the requirement of the particular terminal. If the concentrator cannot handle the terminal, the algorithm searches for the next closest concentrator and performed the same evaluation. This process is repeated until an available concentrator is found, and the algorithm is continued to assign the remaining terminals, if there are any. In the case that no concentrator can accommodate the required capacity of a given terminal, the search is considered failed, and the solution provided by the greedy algorithm is not feasible.

Greedy Heuristic used

For the binary string x of the SA:
 Select the N 1s to be concentrators.
 Select the $M-N$ 0s to be terminals.
 Choose a permutation $\pi(l_{M-N})$ at random.
For (each terminal $\pi(l_i)$):
 Determine $cost_{ij}=d_{ij}$ (distance from $\pi(l_i)$ to the closest feasible concentrator r_j).
 Assign $\pi(l_i)$ to r_j .
 Calculate $Z(x)=\sum cost_{ij}$
Endfor

4 Experiments and results

In order to test our approach, we have tackled several OCLP instances of different difficulty. Table 1 shows the main characteristics of the instances. There are 6 OCLP instances, with different values for N and M . We have small size networks (Instances 1 and 2), medium size networks (3 and 4) and large size networks (5 and 6). Instances 1, 2 and 3 have been randomly generated in a 100x100 grid, and Instances 4, 5 and 6 have been generated over a 200x200 grid. The capacities of all the nodes have been randomly generated between 17 and 22, and the weight of the nodes have also been randomly generated between 1 and 5. It is expected that the difficulty of the

instances increases with the number of nodes in each instance.

Table 1. Main characteristics of the problems tackled.

Problem #	Nodes (M)	Controllers (N)	Grid
1	13	3	100x100
2	20	4	100x100
3	40	6	100x100
4	60	8	100x100
5	80	10	200x200
6	100	12	200x200

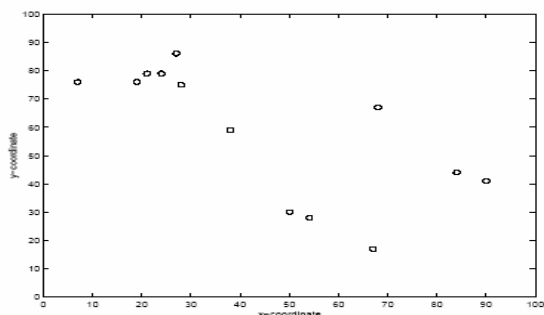
In order to obtain a comparison algorithm for assessing the performance of our approach in the OCLP, we consider the following Lower Bound (LB) for the TA, which have been defined in [4]:

$$LB = \sum_{i=1}^{M-N} \min_k (d_{ik}) \tag{3}$$

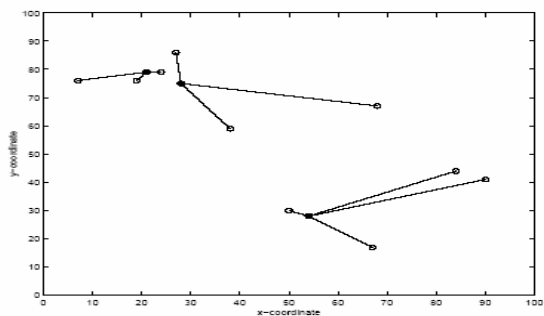
Note that this Lower Bound comes from the solution obtained by assigning each node i to the nearest controller k . Since it is a LB for the TA, the resulting algorithm for comparison is formed by the standard GA with the restricted research described in Section 3, hybridized with this LB instead of the greedy algorithm. It is important to see that this LB is equivalent to have controllers with infinite capacity, in such a way that they can handle any number of nodes. This LB provides then the best possible assignment if no capacity constraint is considered.

Table 2. Results obtained in the OCLP instances tackled.

problem	SA-greedy (50000)	SA-greedy (30000)	SA-greedy (15000)	LB-greedy
1	185.4	185.4	185.4	185.4
2	357.0	357.0	357.0	357.0
3	492.8	496.3	499.8	462.6
4	1476.4	1489.3	1491.8	1385.0
5	1767.6	1796.1	1829.2	1573.5
6	2132.7	2184.6	2196.3	1940.7



(a)

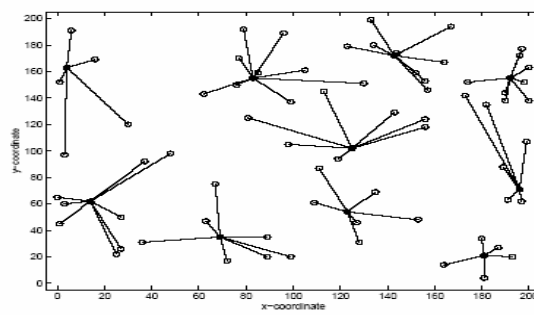


(b)

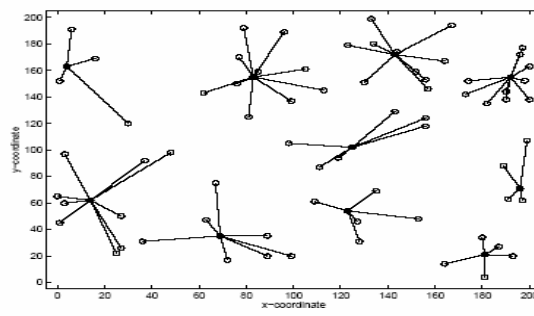
Figure 2. Distribution of nodes and optimal solution for the problem #1.

Figure 2 shows the distribution of nodes and the optimal solution for the problem #1. This can be considered an easy instance, with only 13 nodes and 3 controllers to install. The LB for this problem is very close to the optimal solution obtained by the SA-Greedy algorithm, as can be seen in Table 2.

Figure 3 shows the best solution found by the SA-Greedy algorithm compared with the solution of the LB-Greedy. Note that both algorithms choose the same set of nodes to be controllers, in spite of the LB-Greedy provides the solution without capacity constraint. The final assignment provided by the TA is different due to that constraint, but the differences are small.



(a)



(b)

Figure 3. Solution for the problem #5 (a) solution given by the SA-Greedy algorithm, (b) solution of the LB-GA.

5 Conclusions

In this paper we have presented preliminary work carried out on the Optimal Controller Location Problem, which arises in the process of designing cellular mobile networks. We have introduced a hybrid algorithm consisting of a Simulated Annealing, and a Greedy algorithm which provides the objective function. We have tested our hybrid approach in several OCLP instances of different difficulty, comparing the results obtained with that of a lower bound for the problem.

References:

- [1] 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.
- [2] 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.
- [3] 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.
- [4] A. Quintero and S. Pierre, Assigning cells to switches in cellular mobile networks: a comparative study, *Computers Commun.*, vol. 26, pp. 950-960, 2003.
- [5] H. Pirkul, S. Narasimhan and P. De, Locating concentrators for primary and secondary coverage in a computer communications network, *IEEE Trans. Commun*, vol 36, no. 4, pp. 450-457, 1988.
- [6] S. Pierre and F. Houeto, Assigning cells to switches in cellular mobile networks using taboo search, *IEEE Trans. Syst. Man Cyben. B*, vol. 32, no. 3, pp. 351-357, Jun. 2002.
- [7] F. N. Abuali, D. A. Schoenefeld and R. L. Wainwright, Terminal assignment in a communications network using genetic algorithms, In *Proc. 22sd Annual ACM Computer Science Conference*, pp. 74-81, ACM press, 1994.
- [8] S. Kirpatrick, C. D. Gerlatt and M. P. Vecchi, Optimization by simulated annealing, *Science*, vol. 220, pp. 671-680, 1983.
- [9] S. Kirpatrick, Optimization by simulated annealing--Quantitative studies, *J. Stat. Phys.*, vol. 34, pp. 975-986, 1984.
- [10] 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.