Optimal switch location in mobile communication networks using hybrid genetic algorithms

Sancho Salcedo-Sanz* [a] , Jose A. Portilla-Figueras [a], Emilio G. Ortiz-García [a], Angel M. Pérez-Bellido [a], Christopher Thraves [b], Antonio Fernández-Anta [c] and Xin Yao [d]

[a] *Department of Signal Theory and Communications, Universidad de Alcalá.*

[b] *Department of Mathematical Engineering, Universidad de Chile.*

[c] *LADyR, GSyC, Universidad Rey Juan Carlos.*

[d] *Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, the University of Birmingham and Nature Inspired Computation and Applications Laboratory (NICAL), The University of Science and Technology of China, Hefei, Anhui 230027, PR China.*

Abstract

The optimal positioning of switches in a mobile communication network is an important task, which can save costs and improve the performance of the network. In this paper we propose a model for establishing which are the best nodes of the network for allocating the available switches, and several hybrid genetic algorithms to solve the problem. The proposed model is based on the so called capacitated p-median problem, which have been previously tackled in the literature. This problem can be split in two subproblems: the selection of the best set of switches, and a terminal assignment problem to evaluate each selection of switches. The hybrid genetic algorithms for solving the problem are formed by a conventional genetic algorithm, with a restricted search, and several local search heuristics. In this work we also develop novel heuristics for solving the terminal assignment problem in a fast and accurate way. Finally, we show that our novel approaches, hybridized with the genetic algorithm, outperform existing algorithms in the literature for the p-median problem.

Key words: Switch Location Problem, wireless communications networks, hybrid genetic algorithms, heuristics.

* Corresponding author: Sancho Salcedo-Sanz. Department of Signal Theory and Communications, Universidad de Alcalá. 28871 Alcalá de Henares. Madrid. Spain.
Tlph: +34 91 885 6731
Fax: +34 91 885 6699
Email address: sancho@tsc.uc3m.es (Sancho Salcedo-Sanz*).

1 Introduction

In the last decade, mobile telecommunication networks have known an extraordinary development, due to the necessity of information transfer among users. Mobile communication networks can be modelled as formed by hexagonal cells, each corresponding to a different cover zone, and associated to a given Base Station (BTS). The cell is the unit of a cellular communication system. A certain number of cells are chosen to install switches [1], which route calls to another base station or to a public switched telephone network [1].

The design of mobile networks often involves problems of devices location (BTS, multiplexers, switches etc.) [6], [21], [22]. There are works specifically related to the design of the BTS-switch structure, like the BTS location problem [2], in which the objective is to obtain the optimal location of BTSs in a grid, such that the radio coverage of the considered grid is maximum. Another important problem, directly related to the BTS-switch structure in mobile networks, is the assignment of cells to switches problem [1], [3], [4], [5]. This problem considers that the BTSs and switches of the network are already positioned, and its objective is to assign each BTSs to a switch, in such a way that a capacity constraint have to be fulfilled. The objective function is then formed by two terms: first the sum of the distances from the BTSs to the switches must be minimum, and there is another term related to *handovers* between cells assigned to different switches, which must be minimized (see [3] for details). In addition, there are other location problems related to the design of communications network, either mobile networks [7]-[12] and computer networks [13], [14].

Among the algorithms applied to solve the above mentioned problems, there are a wide variety of heuristics and metaheuristics approaches: in [2] several heuristics are presented and applied to solve the BTSs location problem. The performance of Genetic Algorithms (GA), Tabu Search (TS) and Simulated Annealing (SA) approaches are compared with that of random walk and greedy approaches. In [10], a hybrid k-means genetic algorithm is applied to the design of indoor cellular networks. Other hybrid algorithms have been applied to the BTS-switch assignment problem in [4], [5]. Both works deal with the same problem, discussing different approaches to it, based on mixing GA, TS and SA algorithms. SA has 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 mechanism is used to tackle the assignment of cells to switches 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 quite close to the problem's lower bound. Another SA approach is

[1] In the literature switches are also denoted as Base Station Controllers (BSCs).

presented in [7] for the design of telecommunication access networks with reliability constraints. Finally, a SA algorithm is applied to the design of UMTS access network topology in [8]. The SA algorithm in this work is used as global search heuristic, and mixed with a local search technique which searches for tree topologies of radio base stations.

In spite of the huge work carried out on the designing of the access part of mobile communication networks, there are still some problems which have not been completely studied. Specifically, this paper deals with the problem of the location of switches in mobile telecommunication networks. Note that, in the literature revised above, there are several works dealing with the problem of assigning cells to switches, locating BTSs and designing the access topology of mobile networks, but there are very few works tackling the problem of locating switches in such networks. This is mainly because, in the majority of cases, the position of switches is considered a pre-fixed parameter, usually installed within the infrastructure of some BTSs in the network, or considering the most traffic managing BTSs to be the network's switches. On the other hand, a fast and easy model for the location of switches, and the corresponding BTSs' assignment to switches, is applicable in the field of regulatory studies in telecommunications [30], [31].

In this paper we propose a model for the optimal location of switches in a mobile communication network (Switch Location Problem, SLP hereafter), and several hybrid genetic algorithms to solve the problem. Our model starts from the premise that the switches must be located in existing BTSs, in order to use the existing infrastructure, and save costs. We need then to establish which are the optimal BTSs for allocating a given number of switches, taking into account several parameters. First, the distance between switches and their associated BTSs must be minimum, in order to maximize the reliability of the radio link between switches and BTSs. Second, a constraint of capacity must be fulfilled, since a switch is limited on the number of BTSs that it can manage. Thus, we propose to use the *Terminal Assignment Problem* (TA) [15]-[18] as a model to evaluate the selection of a set of BTSs to locate switches. The SLP with the TA for evaluating the set of BTSs, is equivalent to the well known capacitated p-median problem, which has been tackled before in the literature in different fields and applications [19], [23], and it is known to be NP-hard [24]. We propose several hybrid genetic algorithms for solving the SLP, based on the hybridization of a conventional GA and several local search algorithms for the TA. We will show the performance of the local algorithms for solving the TA, comparing their performance with thus of existing approaches. We will also test our approaches for the SLP in several instances, with different number of BTSs and switches available.

The rest of the paper is structured as follows: next section defines the SLP in a mobile communications network. In this section we show that it can be split in two subproblems, the selection of the set of controllers and a terminal assignment problem to evaluate this selection. In Section 3 we present our hybrid genetic algorithms for solving the SLP. In Section 4 we test the proposed algorithms for the TA and for the SLP, by means of several computational experiments, where the performance of our approaches is studied. Finally, Section 5 concludes the paper.

2 Problem definition

Let us consider a mobile communications network formed by N nodes (BTSs), where a set of M switches must be allocated in order to manage the network traffic and other network resources. It is always fulfilled that $M < N$, and in the majority of cases $M \ll N$. We start from the premise that the existing BTSs infrastructure must be used to locate the switches, since it saves costs. Thus, the SLP consists of selecting M nodes out of the N which form the network, in order to locate in them our M switches. Note that there are $\binom{M}{N}$ possibilities of selecting M nodes out of N, and this selection should be optimal with respect to a given objective or cost function. In order to define an objective function for the SLP, a TA must be solved. Thus the SLP can be divided in two subproblems: First, the selection of the M controllers. Second, a TA in the calculation of the cost function for each controllers selection.

2.1 Cost function calculation in the SLP: the terminal assignment problem

Let us consider a system formed by K terminals and M concentrators. We have a vector $\mathbf{w} = [w_1, \ldots, w_K]$ of terminal weights and a vector $\mathbf{p} = [p_1, \ldots, p_M]$ of concentrator capacities. Finally, we also have an $M \times K$ matrix of distances $\mathbf{D}$, where $d_{ij} \geq 0$ gives the distance or cost of connecting terminal j to concentrator i.

The TA consists of determining the minimum total cost links to form a network, by connecting the terminals to the concentrators, subject to two constraints. First, each terminal must be assigned to one and only one of the concentrators. Second, the capacity of a concentrator can not be smaller than the sum of the weights of the terminals assigned to it.

This problem can be seen as a integer programming problem [15]. Mathematically it can be stated as follows. Let $\mathbf{X}$ be a binary matrix such that element $x_{ij} = 1$ if terminal j has been assigned to concentrator i, and $x_{ij} = 0$ otherwise.

Find $\mathbf{X}$ which minimizes

$$Z(\mathbf{X}) = \sum_{i=1}^{M} \sum_{j=1}^{K} d_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{i=1}^{M} x_{ij} = 1 \quad j = 1, 2, \ldots, K$$

$$\sum_{j=1}^{K} w_j x_{ij} \leq p_i \quad i = 1, 2, \ldots, M$$

$$x_{ij} \in \{0, 1\}$$

2.2 Modelling the SLP as a capacitated p-median problem

Consider now the SLP. Let us suppose that every node of the network can be considered both as a BTSs or as a switch. If we have a particular solution for the SLP, there will be M nodes serving as switches, and $K = N - M$ nodes which are BTSs. For this particular solution, we can associate BTSs in the SLP with terminals in the TA, and switches in the SLP with concentrators in the TA. We have therefore that a particular solution for the SLP is an instance of the TA, which can be solved using any algorithm for the TA existing in the literature. The solution obtained for the TA, has associated an objective function value given by equation (1). The SLP consists then of finding the location of switches into the nodes of the network which makes this objective function minimum. With this definition, the SLP is equivalent to a capacitated p-median problem [24], [19], which has been tackled before in the literature, and applied to different real optimization problems [23].

Mathematically, the formulation of the SLP as a capacitated p-median problem is as follows:

Let $I = \{1, \ldots, N\}$ the set of nodes in the network, M be the number of nodes which will be selected as controllers for the network. Find a binary vector $\mathbf{y}$ such that:

$$S = \{j/y_j = 1\}$$

$$Z = min \left(\sum_{i \in I - S} \sum_{j \in S} d_{ij} x_{ij} \right), \tag{2}$$

subject to:

$$\sum_{i \in I-S} w_i x_{ij} \leq p_j y_j, \quad j \in S, \tag{3}$$

$$\sum_{j \in S} x_{ij} = 1, \quad i \in I - S, \tag{4}$$

$$\mathcal{C}(S) = M, \tag{5}$$

$$y_j \in \{0, 1\}, \ j \in I. \tag{6}$$

$$x_{ij} \in \{0, 1\}, \ i \in I - S, \ j \in S, \tag{7}$$

with $\mathcal{C}(S)$ number of elements of S. Note that this definition includes the resolution of a Terminal Assignment Problem for each value of vector $\mathbf{y}$.

2.3 An example

Consider the SLP defined by the collection of $N = 13$ nodes, with $M = 3$ switches available, shown in Table 1, and displayed in Figure 1 (a). This SLP instance consists of choosing the 3 best nodes of the network for the location of the corresponding switches. Note that there are $\binom{13}{3}$ possibilities of locating 3 switches in a network of 13 nodes. As has been mentioned before, each of the possible placements defines a TA, with $M = 3$ concentrators and $K = 10$ terminals, and an objective function given by equation 1. The best TA solution (the one with minimum value of function 1 after solving the TA) will be considered as the solution for the SLP. In this case, it is easy to show that the best location of switches is given when the TA is defined as in Figure 1 (b), with an objective function value of 185.4.

3 Proposed hybrid genetic algorithms

In this section we present several hybrid genetic algorithms for solving the SLP. Following the split structure of the SLP in two subproblems, our algorithms are based on a global-local search technique. First, we use a GA for choosing which nodes serve as switches, second, a local search heuristics is used to solve the associated TA and obtaining a value of the objective function.

3.1 Global search heuristic

We use a conventional GA with a *restricted search* as global search algorithm. GAs have been successfully applied before to a wide variety of combinatorial optimization problems, and therefore we suppose that the reader is familiarized with its conventional implementation. Readers not familiarized with these technique can consult [25] as basic bibliography. Table 2 shows the algorithmic description of a conventional genetic algorithm: It works by encoding a population of binary strings, representing a possible selection of switches among the nodes which form network. In our case, the length of each individual is equal to this number of nodes of the network. A selection mechanism using the roulette wheel method [25], two-points crossover and flip-type mutation are applied to evolve the population. Also, the best individual in each generation is passed over to the next one, with an operator of elitism.

To deal with the SLP, this conventional implementation of the GA is not appropriate, since it cannot tackle the constraint imposed by the number of 1s in the binary strings (Equation 5). We must introduce then a mechanism for dealing with this SLP constraint. This constraint has been previously applied in the literature. Specifically, the same constraint regarding the number of 1s in GA and SA have been solved by means of the so-called *restricted search* operator in [26] and [27]. The restricted search basically considers one extra operator to be added to the conventional GA, in the following way: after the application of the crossover and mutation operators, the individual $\mathbf{x}$ will have p 1s that, in general, will be different from the desired number of desired 1s in x, M. If $p < M$ the restricted search operator adds $(M - p)$ 1s in random positions, and if $p > M$, the restricted search operator randomly selects $(p - M)$ 1s and removes them from the binary string. This operator can be described in pseudo-code, as follows:

The restricted search operator

```
Let M be the number of 1s in a given individual s of the GA.
For every generation of the GA:
    for every individual of the GA population:
        check the number of 1s p.
            if(p < M)
                Add_ones(M − p);
            else
                Remove_ones(p − M)
            end(if)
    end(individual)
end(generation)
```

The GA using the restricted search will look for the best binary vector $\mathbf{y}$ in terms of the SLP objective function. This objective function is obtained by means of solving the TA associated to a given binary vector $\mathbf{y}$. The following section describes the local search algorithm we have chosen to be hybridized with the global search techniques.

3.2 Local search algorithms

3.2.1 A greedy algorithm for the TA

One of the most important papers on TA was the approach by Abuali et al. [15]. In this article, the authors proposed a greedy algorithm for solving the TA. This greedy approach starts from a random permutation of terminals $\pi(l_K)$ (order in which we assign the terminals to controllers). Then, the cost function to optimize is the Euclidean distance between terminal i and concentrator j. The terminals are assigned to concentrators following the order in $\pi(l_K)$, 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.

Pseudo-code of the Greedy algorithm in [15].

Choose a permutation $\pi(l_K)$, at random.

for(each terminal $\pi(l_i)$)

Determine d_{ij}, the distance from

$\pi(l_i)$ to the closest feasible concentrator r_j.

Assign $\pi(l_i)$ to r_j.

end(for)

3.2.2 A Modified greedy approach for the TA

The greedy approach defined in [15] has a major drawback: it is computationally inefficient, since its performance and computational time depend on the number of permutations defined. To solve this point, we propose a modification of the algorithm for reducing its computational time, and trying to improve its performance on the TA. Instead of defining a number of permutations, we define only one permutation π^* which sorts the terminals for their distance to the nearest concentrator. We start assigning terminals to concentrators following the order given by permutation π^*. When a given terminal i cannot be assigned to its nearest concentrator k (due to the capacity constraint), we calculate the distance of all the terminals in k to the second nearest concentrator. If the distance of a terminal j (already assigned to k) to its second closest concentrator is smaller than the distance from terminal i to its second closest concentrator, and $w_j \geq w_i$, then we reassign terminal j to its second closest concentrator, and substitute it by terminal i. In the case that there is not such a terminal or with the requirements of distance or weight, terminal i is assigned to its second closest concentrator. We call this modified greedy approach as the *GreedyExp* algorithm for the TA. It is expected that the GreedyExp approach to be computationally much efficient than the greedy algorithm in [15]. We will show its performance on the TA in Section 4.

3.2.3 Local heuristics based on linear programming relaxation of the TA

The linear programming (LP) relaxation of the TA can be defined as follows. Find $\hat{\mathbf{X}}$ which minimizes

$$Z(\hat{\mathbf{X}}) = \sum_{i=1}^{M} \sum_{j=1}^{K} d_{ij} \hat{x}_{ij} \tag{8}$$

subject to

$$\sum_{i=1}^{M} \hat{x}_{ij} = 1 \quad j = 1, 2, \ldots, K$$

$$\sum_{j=1}^{K} w_j \hat{x}_{ij} \leq p_i \quad i = 1, 2, \ldots, M$$

$$\hat{x}_{ij} \in [0, 1]$$

Note that the solution $\hat{\mathbf{X}}$ of the LP relaxation satisfies that $Z(\hat{\mathbf{X}})$ is a lower bound on the cost of an optimal solution of the TA, i.e $Z(\hat{\mathbf{X}}) \leq Z(\mathbf{X})$.

In this section we present two heuristics for the TA which start from the solution $\hat{\mathbf{X}}$ to the LP relaxation, and use the vectors of capacities $\mathbf{p}$ and weights $\mathbf{w}$ as local information to build a solution of the TA. The output of each heuristic is a result matrix $\mathbf{X}^*$, and its corresponding cost value Z^*.

The first heuristic we propose is called the XWLP algorithm, since it uses an ordering of terminals and concentrators which depends on the product of the elements of $\hat{\mathbf{X}}$ and $\mathbf{w}$ as follows.

XWLP Heuristic:

$x_{ij} \leftarrow 0, \forall i, j$
$x'_{ij} \leftarrow \hat{x}_{ij} w_j, \forall i, j$
while $(\max_{i,j}\{x'_{ij}\} > 0)$
    choose at random n and k such that $x'_{nk} = \max_{i,j}\{x'_{ij}\}$
    if $(p_n \geq w_k)$ then
        $x_{nk} \leftarrow 1$
        $p_n \leftarrow p_n - w_k$
        $x'_{ik} \leftarrow -1, \forall i$
    else
        $x'_{nk} \leftarrow -1$
    end if
end while
if $(\sum_{i,j} x_{ij} = K)$
    $Z^* \leftarrow \sum_{i,j} x_{ij} d_{ij}$
    $X^* \leftarrow X$
else
    - No feasible solution has been found
    $Z^* \leftarrow \infty$

Note that in this heuristic, the terminals corresponding to large elements of matrix $\mathbf{X}'$ are assigned first, breaking ties randomly. If there is enough capacity available, the terminal is assigned to the concentrator, the terminal's weight is subtracted from the remaining concentrator's capacity, and the entire row of matrix $\mathbf{X}'$ is removed from further consideration, by marking it with a value -1. This process ensures that the capacity constraint will not be violated. The resulting solution is unfeasible if not all the terminals are assigned to concentrators. In this case the solution cost is fixed to ∞.

The second heuristic that we propose is called MWFLP (Maximum Weight First Linear Programming) heuristic:

MWFLP Heuristic:

$x_{ij} \leftarrow 0, \forall i, j$
while $(\max_j\{w_j\} \geq 0)$
    choose at random some k such that $w_k = \max_j\{w_j\}$
    if$(\max_i\{\hat{x}_{ik}\} = -1)$ then
        - No feasible solution has been found
        $Z^* \leftarrow \infty$
        exit
    end if
    choose at random some n such that $\hat{x}_{nk} = \max_i\{\hat{x}_{ik}\}$
    if $(p_n \geq w_k)$ then
        $x_{nk} \leftarrow 1$
        $p_n \leftarrow p_n - w_k$
        $\hat{\mathbf{x}}_{ik} \leftarrow -1, \forall i$
        $w_k \leftarrow -1$
    else
        $\hat{x}_{nk} \leftarrow -1$
    end if
end while
$Z^* \leftarrow \sum_{ij} x_{ij} d_{ij}$
$X^* \leftarrow X$

This heuristic selects the terminals to be assigned following the ordering given by the maximum of the vector of weights $\mathbf{w}$. If there is a tie, the heuristic chooses randomly among the unassigned terminals with equal weight. For a chosen terminal, the order in which the concentrators are considered depends on the elements of matrix $\hat{\mathbf{X}}$, breaking ties at random. One terminal is assigned to a concentrator if and only if the concentrator has enough capacity for handle the terminal. If so, the row of the matrix corresponding to the terminal is removed from further consideration by marking it with a value of -1. This process ensures that the capacity constraint is fulfilled. The solution will be unfeasible if any of the terminals has not been assigned. In this case the solution cost is fixed to ∞.

3.2.4 Lower bounds for the SLP

In order to obtain comparison algorithms for assessing the performance of our approaches in the SLP, we consider two Lower Bounds (LB) for the SLP. The first LB is given by the linear programming relaxation of the TA, given in Section 3.2.3. If we hybridize this LB with the GA proposed in Section 3.1, we will obtain an algorithm which provides a LB for the SLP. We call this lower bound as LB_{LP}.

The second LB for the TA has been defined in [5]:

$$LB_{\infty} = \sum_{i=1}^{N-M} \min_{k}(d_{ik}) \tag{9}$$

Note that this LB comes from the solution obtained by assigning each node i to the nearest controller k. 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. The SLP without the capacity constraint is similar to the so-called p-median problem (see [28], [29] for details). We can use then the GA in Section 3.1 hybridized with the LB_{∞} to obtain a LB for the SLP.

4 Computational experiments and results

We divide this section in two major parts. The first part is devoted to test the performance of the proposed heuristics for the TA, presented in Section 3. The second part will show the performance of the hybrid algorithms for the SLP.

4.1 Experiments in the TA

First we would like to test the performance of the algorithms we have proposed for the TA, as it is a key part for accurately solve the SLP. To do it, we have run a set of experiments, where the performance and computational time of the different algorithms for the TA are evaluated.

To test the performance of the algorithms, we use a 300×300 grid, where a set of $K = 200$ terminals and $M = 10$ concentrators will be placed. First of all, we randomly and uniformly set the concentrators' coordinates (r_{i1}, r_{i2}), for $i = 1, \cdots, M$, in the grid. Then, the terminals' coordinates (l_{j1}, l_{j2}), for

$j = 1, \cdots, K$, are obtained starting from those of the concentrators. For each terminal j, a concentrator k is randomly and uniformly chosen, and the terminal coordinates are obtained as:

$$l_{j1} = r_{k1} + N(0, \sigma) \tag{10}$$

$$l_{j2} = r_{k2} + N(0, \sigma), \tag{11}$$

where $N(0, \sigma)$ is a normally distributed one-dimensional random number, of mean 0 and variance σ. This framework allows different cases in the distribution of terminals and concentrators, depending on the parameter σ. If this variance σ is small, the terminals will be placed in the surroundings of the concentrators. Figure 2 show an example of this, using a variance $\sigma = 10$. On the other hand, for large values of σ, the terminals will be almost uniformly spread in the grid. This can be seen in Figure 3, where a value of $\sigma = 200$ has been used. The testing of the heuristics has been carried out in this framework, by varying the parameter σ from 5 to 200, in steps of 5, and running 20 experiments for each value of σ.

The weights of the terminals have been randomly chosen, with values between 2 and 6. The capacities of the concentrators have been calculated from the terminal weights, as follows:

$$p_i = round\left(\frac{\sum_{j=1}^{K} w_j}{M}\right) + (1 + round(U(0, 4))); \tag{12}$$

where $U(0, 4)$ is a uniformly distributed number between 0 and 4. Note that, with this definition $\sum_{j=1}^{K} w_j < \sum_{i=1}^{M} p_i$. Finally, the distance matrix has been computed as the Euclidean distance between each pair of terminal-concentrator.

Note that XWLP and MWFLP algorithms proposed in this paper make random choices when ties are encountered. This may influence on the chances of finding a feasible solution and its quality. Empirically, we have found that in these test cases, the random choices do not influence much the performance of the XWLP algorithm. However, the quality of the solutions obtained with the MWFLP algorithm can vary from one run to another. For these reasons, in the evaluation performed we only execute the XWLP algorithm once for each problem, while we repeat the execution of the MWFLP algorithm 200 times, and keep the best solution found. We have tested the greedy approach in [15] by running it with 15000, 30000 and 50000 permutations.

The computational time of the compared algorithms has also been analyzed by means of computational experiments. To do this, we have created a new set of experiments, fixing the parameter σ and varying the number of terminals K from 50 to 200. We have run 30 experiments with each value of K, obtaining the CPU time employed by each heuristic. All the experiments have been carried out in a Pentium IV processor (2.5 GHz).

Figure 4 shows the results obtained by the different algorithms tested. The figure represents the parameter σ in the x-axis, and the cost of each algorithm divided by the cost of the LB_{LP} in the y-axis. Each point of the curves was obtained as an average of the 30 experiments in each variance. Note that the best results are obtained using the MWFLP algorithm, followed by the XWLP. Both heuristics improve the results obtained by the GreedyExp and the greedy approach with 15000, 30000 and 50000 permutations. Note also that the GreedyExp algorithm obtains better results than the greedy algorithm.

The dashed line in Figure 4 represents the LB_{LP}. It is interesting that the differences between this LB and the result obtained by the MWFLP and XWLP heuristics are small. There are, however, some differences depending on the value of σ. It seems that in the cases where the terminals surround the concentrators (small value of σ), the differences between the proposed heuristics and the LB_{LP} is larger than in the experiments where the terminals are almost randomly distributed in the grid (large value of σ). These differences are small, about 2% in the best cases.

Regarding the computational time of the compared heuristics, Figure 5 shows the CPU time in seconds of each algorithm in TA examples of different size. Note that the GreedyExp approach is the algorithm with less computational cost. XWLP heuristic also has a good computational time, and solves all the instances in less than 1 second. The MWFLP heuristic solves all the instances in less than 5 seconds, and obtains better results than the rest of the algorithms, as was shown in Figure 4. The greedy heuristic with 15000, 30000 and 50000 permutations are more time-consuming algorithms than the other heuristics, as can be seen in the figure.

4.2 Experiments in the SLP

In order to test the hybrid algorithms in the SLP, we have proposed several SLP instances of different difficulty. Table 3 shows the main characteristics of the instances tackled. There are 6 SLP 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). Instance 1 is the small example of Section 2.3. Instances 2, 3 have been randomly generated in a 100×100 grid, and Instances 4, 5 and 6 have been generated over a 200×200 grid. The capacities of all the nodes have been randomly generated between values of 15 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. The parameters of the conventional GA used in the simulations are population of 50 individuals, 200 generations, crossover probability $P_c = 0.6$ and mutation probability $P_m =$

0.01. We compare the performance of the GA hybridized with the MWFLP, GreedyExp algorithm and greedy algorithm in [15] with 50000 permutations. We also include a comparison with the results obtained by a GA proposed in [19] for the capacitated capacitated p-median problem.

Table 4 shows the results obtained by the hybrid algorithms tested. We have run each algorithm 30 times, keeping the values of the best, mean and standard deviation. We also include the lower bounds for the considered instances defined in Section 3.2.4. Note that the bound obtained with the bound LB_{LP} and the GA is tighter than the bound obtained with the GA plus the LB_∞.

The best results obtained with the approaches compared in this paper are achieved with the hybrid approach GA_MWFLP. This hybrid algorithm obtained the minimum value of the objective function Z in all instances tested. The algorithm GA_GreedyExp obtains similar results in Instances 1 and 2, but the results obtained in the rest of instances seem to be worse. The hybrid approach GA_Greedy50000 obtains the worst results among the heuristics tested. Note also that our approaches GA_MWFLP and GA_GreedyExp outperform the results obtained by the GA proposed in [19]. In order to statistically corroborate these results, Table 5 shows the results of a t-test performed over the data obtained by the compared algorithms. The t-test shows that the GA_MWFLP performs statistically better than the rest of algorithms compared in Instances 3, 4 and 5. In the smallest instances 1 and 2, and in Instance 6, GA_MWFLP and GA_GreedyExp performs in a similar way. Both approaches outperform statistically the GA proposed in [19]. Regarding the comparison between GA_MWFLP and GA_Greedy50000 algorithm, it is easy to see that the GA_MWFLP improves the performance of the GA_Greedy approach in all Instances, and this improvement is statistically significant in Instances 2, 3, 4 and 5.

The case of Instance 2 is interesting. Note that in this instance, the LB_{LP} is equal to the solution obtained by the hybrid algorithms. This means that the solution obtained by the linear relaxation of the TA is also a solution for the TA. Figure 6 shows Instance 5 nodes distribution (Figure 6 (a)), the solution obtained by the GA_MWFLP hybrid algorithm (Figure 6 (b)) and the solution given by the LB_{LP} (Figure 6 (c)). Note the differences between Figures 6 (b) and 6 (c) due to the capacity constraint of the problem.

4.3 The SLP in a real case

We conclude the experiment part of this paper by showing the resolution of a SLP in a real case: the SLP of in the wireless network of Alcalá de Henares, Madrid, Spain. Alcalá de Henares is a medium size city, with 180000 citizens, sited on the north-east of Madrid. In the last few years there have been a massive growth of the wireless telephony use within the city, and its 2G wireless network is now completely deployed, having 33 BTS of different types for covering the city. Figure 7 shows the situation of the BTSs in a map of the city. The parameters of the different BTSs (Table 6) has been obtained from the mobile network operators. The weights of the TA are equivalent to the number of transceivers (Trx) of each BTS. All the BTSs have the same maximum capacity if it is considered as a switch. This table also provides the position of the BTSs in Universal Transverse Mercator (UTM) coordinates.

We have applied our best approach, the GA_MWFLP algorithm to obtain the switches location. Note that, in this case, the minimum number of switches to cover the network capacity requirements is 5. Figure 8 shows the solution found by our algorithm, with a maximum cost of 26059.9 meters. This solution is also given in Table 7.

5 Conclusions

In this paper we have presented several hybrid genetic algorithms for solving the Switch Location Problem (SLP) in a wireless communication network. We have defined a model for the problem, based on the capacitated p-median problem, and following this model, we have constructed several hybrid genetic approaches. All the heuristics proposed consist of a conventional genetic algorithm for choosing which nodes of the networks contains a switch, hybridized with local heuristics for providing the associated objective function (solution of the associated terminal assignment). The experiments carried out have shown the good performance of the hybrid genetic heuristics for solving the SLP. The heuristic developed in this paper, and the model in which they are based, can also be applied in different cost models for regulatory studies in telecommunications.

Acknowledgement

The authors would like to thank anonymous reviewers for their interesting comments and suggestions to improve the paper. This work has been partially supported by a Universidad de Alcalá project number UAH-PI2005/019. X. Yao's work has been partially supported by a National Natural Science Foundation of China grant, number 60428202.

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", in *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] S. Pierre and F. Houéto, "Assigning cells to switches in cellular mobile networks using taboo search," *IEEE Trans. Syst. Man Cyben. B*, vol. 32, no. 3, pp. 351-357, 2002.

[5] A. Quintero and S. Pierre, "Assigning cells to switches in cellular mobile networks: a comparative study," *Computers Commun.*, vol. 26, pp. 950-960, 2003.

[6] M. Riis, "Deployment of mobile switching centers in a telecommunications network: a stochastic programming approach," *Telecomm. Systems*, vol. 26, no. 1, pp. 93-109, 2004.

[7] S. Soni, S. Narasimhan and L. LeBlanc "Telecommunication access network design with reliability constraints," *IEEE Trans. Reliability*, vol 53, no. 4, pp. 532-541, 2004.

[8] A. Jüttner, A. Orbán and Z. Fiala, "Two new algorithms for UMTS access network topology design," *European Journal of Operational Research*, vol. 164, pp. 456-474, 2005.

[9] Y. Pomerleau, S. Chamberland and G. Pesant, "A constraint programming approach for the design problem of cellular wireless networks," In *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 881-884, Canada, 2003.

[10] B. Abolhassani, J. E. Salt, D. Dodds, "A two-phase genetic K-means algorithm for placement of radioports in cellular networks," *IEEE Trans. Syst. Man Cybern. B*, vol 34, no. 1, pp. 533-538, 2004.

[11] S. Chamberland, "A efficient heuristic for the expansion problem of cellular wireless networks," *Computers & Operations Research*, vol. 31, pp. 1769-1791, 2004.

[12] C. H. Chu, G. Premkumar and H. Chou, "Digital data networks design using genetic algorithms," *European J. of Operational Research*, vol. 127, pp. 140-158, 2000.

[13] A. Kershembaum, "Telecommunications Network Design Algorithms", McGraw-Hill, 1993.

[14] 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.

[15] 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.

[16] S. Khuri and T. Chiu, "Heuristic algorithms for the terminal assignment problem," In *Proc. of the 1997 ACM symposium on Applied computing*, pp. 247-251, ACM press, 1997.

[17] S. Salcedo-Sanz and Xin Yao, "A hybrid Hopfield network-genetic algorithm for the terminal assignment problem," *IEEE Trans. Systems Man Cybern. B*, vol. 34, no. 6, pp. 2343-2353, 2004.

[18] X. Yao, F. Wang, K. Padmanabham and S. Salcedo-Sanz, *Hybrid evolutionary approaches to terminal assignment in communication networks*, In *Recent Advances in Memetic Algorithms and Related Search Technologies*, W. Hart, N. Krasnogor and J. E. Smith (eds), pp. 129-159, Springer-Verlag, 2004.

[19] E. Santos-Correa, M. T. Steiner, A. A. Freitas and C. Carnieri, "A genetic algorithm for solving a capacitated P-median problem," *Numerical Algorithms*, vol. 35, no. 4, pp. 373-388, 2004.

[20] G. Mateus and R. V. Fanqueira, "Model and heuristic for a generalizated access network design problem," *Telecomm. Systems*, vol. 15, pp. 257-271, 2000.

[21] P. Cortés, J. Larrañeta, L. Onieva, J. M. García and M. S. Caraballo, "Genetic algorithm for planning cable telecommunication networks," *Applied Soft Computing*, vol. 1, no. 1, pp. 21-33, 2001.

[22] E. Rosenberg, "Dual ascent for uncapacitated telecommunications network design with access, backbone, and switch costs," *Telecomm. Systems*, vol. 16, no. 3, pp. 423-435, 2001.

[23] N. D. Pizzolato, F. Broseghini-Barcelos and L. A. Nogueira-Lorena, "School location methodology in urban areas of developing countries," *International Transactions in Operational Research*, vol. 11, pp. 667-681, 2004.

[24] V. Maniezzo, A. Mingozzi and R. Badacci, "A bionomic approach to the capacitated *p*-median problem," *Journal of Heuristics*, vol. 4, no. 3, pp. 263-280, 1998.

[25] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading:MA, 1989.

[26] S. Salcedo-Sanz, G. Camps-Valls, F. Pérez-Cruz, J. Sepúlveda-Sanchís and C. Bousoño-Calzón and, "Enhancing genetic feature selection through restricted search and Walsh analysis," *IEEE Trans. Syst. Man Cybern.*, vol. 34, no. 4, pp. 398-406, 2004.

[27] S. Salcedo-Sanz and J. Su, "Improving metaheuristics convergence properties in inductive query by example using two strategies for reducing the search space," *Computers & Operations Research*, vol. 34, no. 1, pp. 91-106, 2007.

[28] J. A. Pacheco and S. Casado, "Solving two location models with few facilities by using a hybrid heuristic: a real health resources case," *Computers & Operations Research*, vol. 32, no. 12, pp. 3075-3091, 2005.

[29] O. Alp, E. Erkut and Z. Drezner, "An efficient genetic algorithm for the *p*-median problem," *Annals of Operations Research*, vol. 122, pp. 21-41, 2003.

[30] K. Hackbarth, A. Portilla-Figueras and C. Díaz, "Cost models for telecommunication networks and their application to GSM sytems," *Encyclopedia of Multimedia Technology and Networking*, Idea Group, 2005.

[31] K. Hackbarth, A. Portilla-Figueras, G. Kulenkampf and F. González, "Cost and network models and their applications in telecommunications issues," In *Proc. of the 13th Biennial International Telecommunication Society Conference*, Madrid, Spain, 2002.

List of Tables

List of Figures

Table 1
Nodes weight, capacity and coordinates for the problem in Section 2.3.

Node #	Weights	Capacities	Coordinates
1	4	12	(19,76)
2	3	14	(50,30)
3	2	13	(21,79)
4	5	12	(54,28)
5	4	13	(28,75)
6	4	13	(84,44)
7	2	12	(67,17)
8	3	12	(90,41)
9	1	14	(68,67)
10	3	14	(24,79)
11	4	13	(38,59)
12	5	12	(27,86)
13	4	13	(07,76)

Table 2
Algorithmic description of a conventional genetic algorithm

Conventional Genetic Algorithm

BEGIN
    Generate initial population $P(0)$ randomly,
    $i \leftarrow 0$;
    REPEAT
        Calculate fitness for all the individuals in P(i),
        Select the parents from $P(i)$ based on their fitness in $P(i)$;
        Apply crossover to the parents;
        Apply mutation to the individuals and replace the population;
        $i = i + 1$
    UNTIL i =Max_gen;
END

Table 3
Main characteristics of the SLP instances tackled.

Instance #	Nodes	Controllers	Grid
1	13	3	100×100
2	20	4	100×100
3	40	6	100×100
4	60	8	200×200
5	80	10	200×200
6	100	12	200×200

Table 4
Results obtained (best/avg/std. dev.) by the different approaches studied, in the SLP instances considered.

P #	GA_MWFLP	GA_greedy50000	GA_GreedyExp	GA in [19]	LB_∞	LB_{LP}
1	185.4/185.4/0.0	185.4/185.4/0.0	185.4/185.4/0.0	185.4/185.4/0.0	120.0	181.7
2	357.0/357.0/0.0	357.0/359.6/4.0	357.0/357.0/0.0	357.0/358.4/3.5	357.0	357.0
3	468.1/471.7/2.3	492.8/501.4/6.15	471.5/474.3/1.6	485.3/497.3/6.8	449.4	462.6
4	1420.3/1444.9/13.9	1476.4/1553.5/42.5	1425.8/1452.7/16.6	1457.2/1503.6/32.7	1346.4	1385.0
5	1699.2/1748.4/20.2	1767.5/1827.9/29.9	1704.0/1753.6/28.3	1726.9/1794.3/38.6	1573.5	1618.8
6	2105.2/2198.1/50.6	2132.7/2221.9/43.5	2127.5/2213.8/44.6	2143.8/2206.3/40.8	1940.0	2031.7

Table 5
t values obtained by a two-tailed t-test for the SLP instances tackled. † stands for values of t with 29 degrees of freedom which are significant at $\alpha = 0.05$.

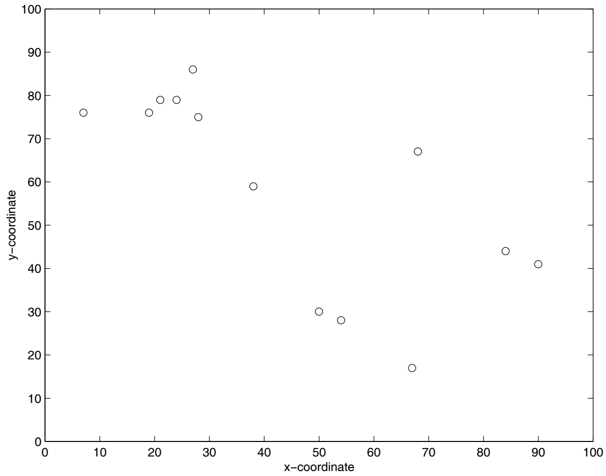
P #	GA_MWFLP-GA_GreedyExp	GA_MWFLP-GA_Greedy50000	GA_MWFLP-GA [19]
1	0.0	0.0	0.0
2	0.0	-3.5†	-2.8†
3	-4.85†	-28.57†	-17.63†
4	-2.06†	-15.75†	-12.48†
5	-0.98	-15.74†	-12.74†
6	-1.21	-1.94	-1.90

Table 6
Weights and capacities of the BTSs in Alcalá de Henares.

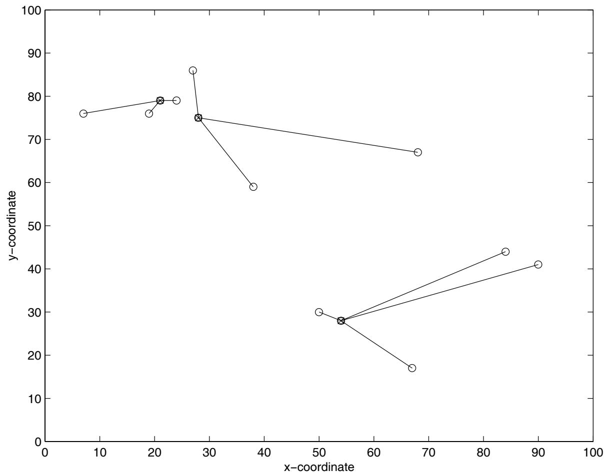
Node #	Weight (Trx)	Capacity	x-coordinate (UTM)	y-coordinate (UTM)
1	3	48	467092	4481354
2	3	48	466105	4480876
3	3	48	468067	4480731
4	6	48	465630	4485016
5	6	48	468816	4481043
6	6	48	469296	4480356
7	6	48	468647	4483846
8	6	48	469815	4482204
9	3	48	470204	4481771
10	3	48	470274	4482279
11	3	48	471468	4483752
12	9	48	466147	4483155
13	9	48	468168	4482323
14	9	48	467872	4481516
15	9	48	468806	4482498
16	12	48	469012	4481755
17	12	48	469455	4482025
18	12	48	469262	4481140
19	12	48	469243	4480275
20	9	48	468914	4484249
21	9	48	469539	4482412
22	9	48	470508	4482427
23	9	48	470272	4481806
24	9	48	471322	4483141
25	6	48	468319	4480835
26	6	48	466265	4484883
27	6	48	466249	4481399
28	9	48	469147	4481880
29	9	48	469692	4482710
30	6	48	469713	4482094
31	6	48	471650	4483967
32	3	48	470633	4482783
33	3	48	470943	4483039

Table 7
Best solution obtained for the SLP problem in the wireless network of Alcalá de Henares. We show the nodes which serves as switches and their assigned nodes.

Node serving as switch	Assigned nodes
14	1, 2, 3, 13, 15, 25, 27
33	7, 10, 11, 20, 22, 24, 31, 32
26	4, 12
18	5, 6, 16, 19, 28
30	8, 9, 17, 21, 23, 29



(a)



(b)

Fig. 1. (a) Example of SLP; (b) Optimal solution.

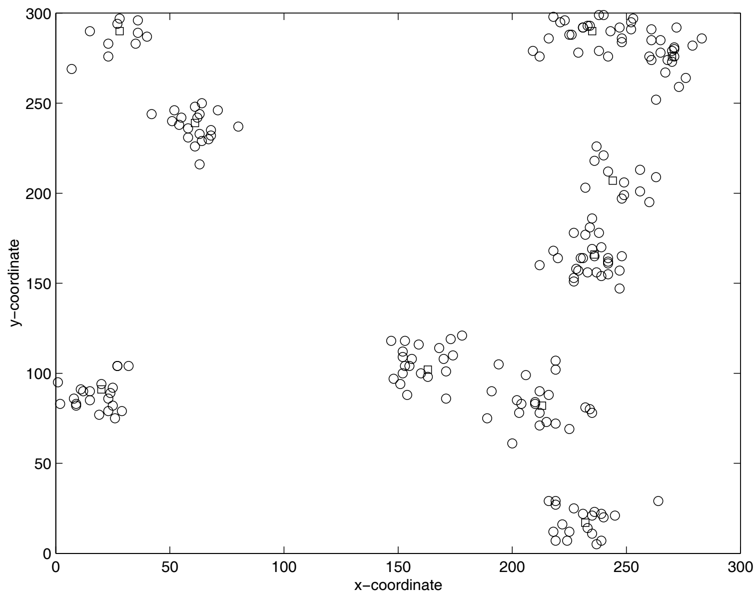


Fig. 2. Example of TA test problem, $M = 10$, $K = 200$ and $\sigma = 10$. The squares and circles stand for concentrators and terminals, respectively.

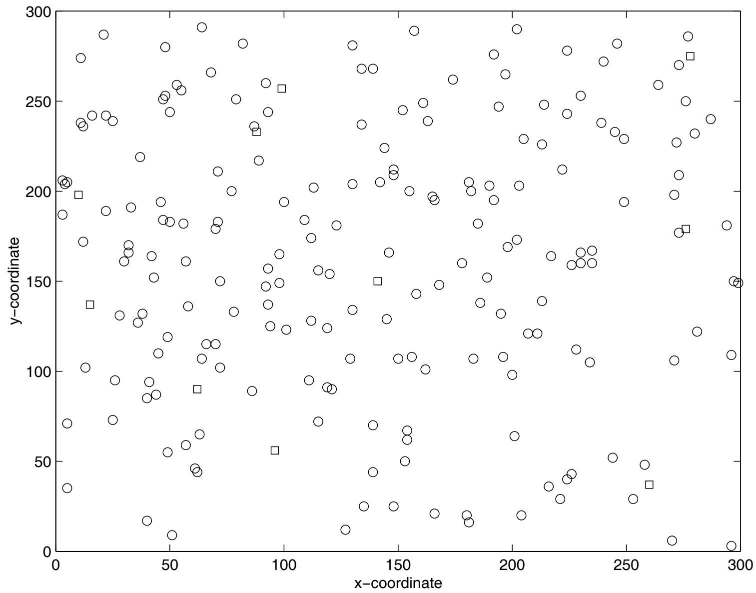


Fig. 3. Example of TA test problem, $M = 10$, $K = 200$ and $\sigma = 200$. The squares and circles stand for concentrators and terminals, respectively.

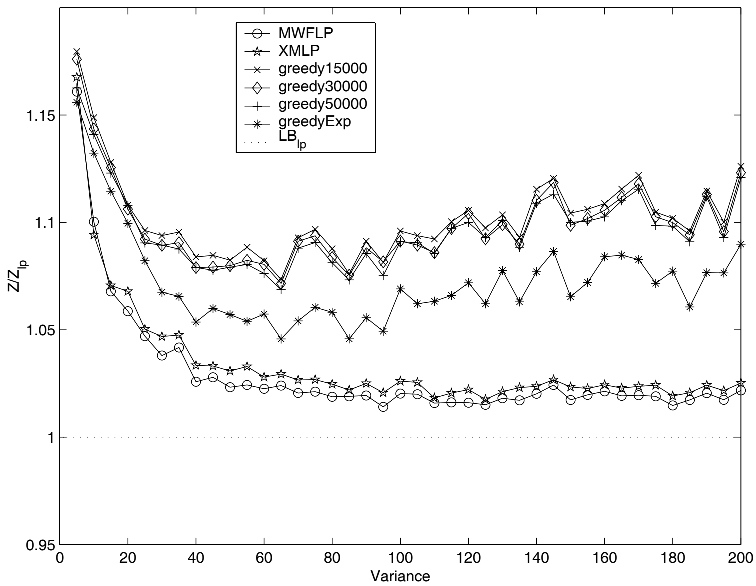


Fig. 4. Results obtained by the heuristics considered in the TA simulations performed.

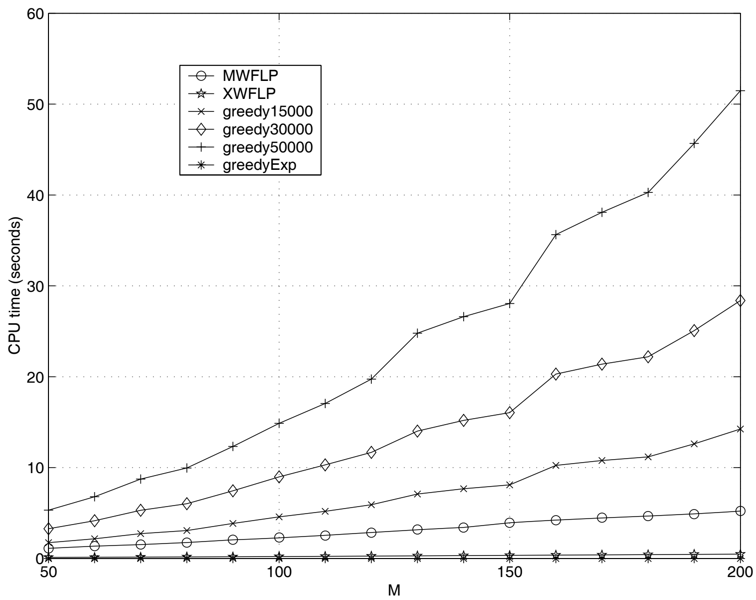
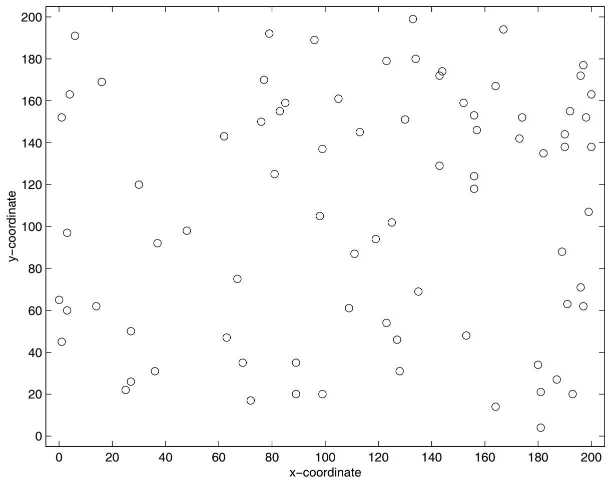
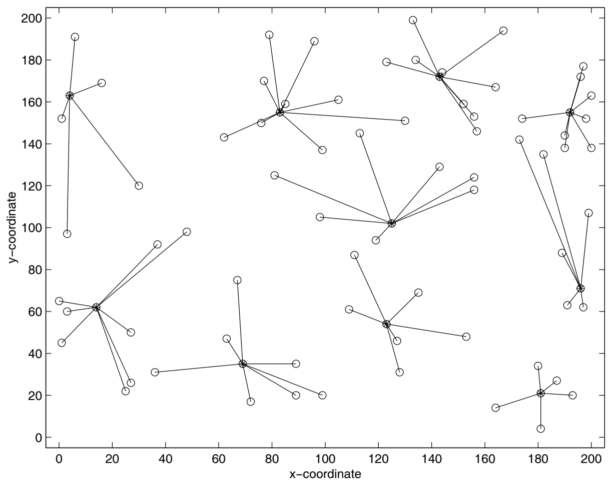


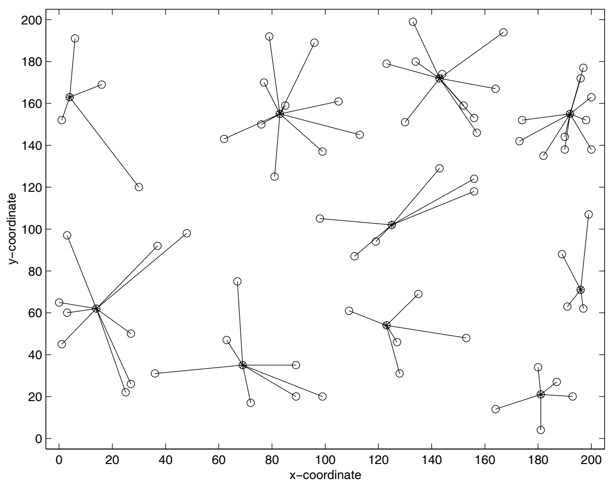
Fig. 5. CPU time in seconds in the TA simulations performed for the different heuristics considered.



(a)



(b)



(c)

Fig. 6. (a) Distribution of nodes in Instance 5; (b) Best solution found by the GA_MWFLP algorithm in Instance 5.; (c) Best solution found by the GA_LB_∞ algorithm in Instance 5.

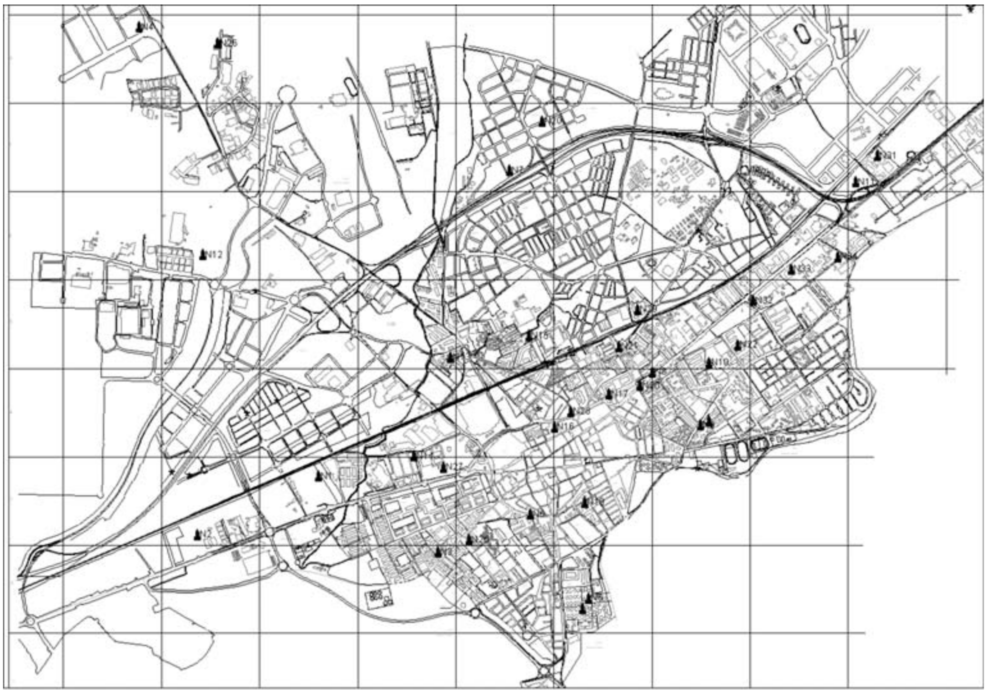


Fig. 7. BTSs locations in Alcalá de Henares (Madrid).

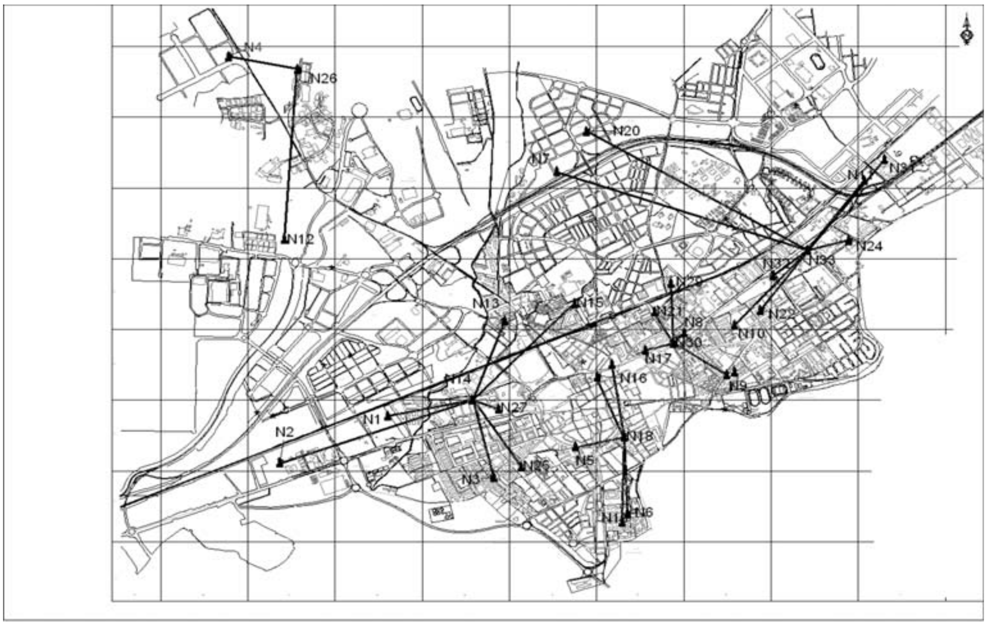


Fig. 8. Best solution obtained for the SLP problem in the wireless network of Alcalá de Henares.