On Solving the Tree-Topology Design Problem for Wireless Cellular Networks

Yanick Pomerleau, Steven Chamberland, and Gilles Pesant

Abstract: **In this paper, we study a wireless cellular network design problem. It consists of selecting the location of the base station controllers and mobile service switching centres, selecting their types, designing the network into a tree-topology, and selecting the link types, while considering the location and the demand of base transceiver stations. We propose a constraint programming model and develop a heuristic combining local search and constraint programming techniques to find very good solutions in a reasonable amount of time for this category of problem. Numerical results show that our approach, on average, improves the results from the literature.**

Index Terms: **Cellular wireless networks, constraint programming (CP), local search, topological design.**

I. INTRODUCTION

In a typical wireless cellular network, the area of coverage is geographically divided into cells and the network topology is hierarchically organized in order to reduce costs. Each cell is equipped with a base transceiver station (BTS) that contains the radio transceivers providing the radio interface with mobile stations. One or more BTSs are connected to a base station controller (BSC) that provides a number of functions related to resource and mobility management as well as operation and maintenance for the overall radio network. One or more BSCs are connected to a mobile switching center (MSC) or switch that controls call setup and call routing while performing many other functions provided by a conventional communications switch. An MSC can be connected to other MSCs or networks such as the public switched telephone network (PSTN), in order to provide a larger coverage.

Cellular wireless network service providers dedicate an important proportion of their budget to acquire, install, and maintain the facilities that carry traffic from cell sites to switches and other facilities (Fig. 1). Typically, the design of cellular wireless networks requires:

1. The analysis of radio-wave propagation and/or the field topology to identify a set of possible base station locations.
2. The selection of a least-cost subset of locations (network nodes) as hubs where the traffic is to be aggregated and switched.
3. The assignment of each cell to a switch while taking into account a certain number of constraints including capacity constraints, routing-diversity to assure reliability, handoffs frequency, and so on.

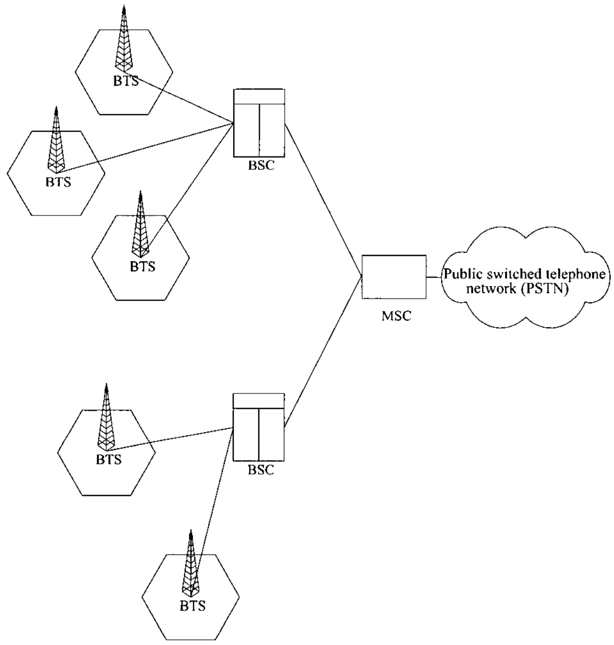


Fig. 1. The network sub-system of a cellular wireless network.

4. The selection of the type of links between the nodes or network elements.

This paper focuses on points 2, 3, and 4. We are interested in the global design problem of such networks. The proposed model deals with: Selecting the location of the BSCs and MSCs; selecting the BSC and MSC types; designing the network topology; selecting the link types.

We make the following assumptions about the organization of the network: (C1) assignment constraints specify that each component is connected to exactly one other, such as a BTS to a BSC and a BSC to an MSC; (C2) uniqueness constraints specify that at most one component is installed at each site; (C3) capacity constraints are applied on links, BSCs, and MSCs; (C4) traffic flow conservation constraints should be satisfied.

The following information is supposed to be known: (A1) the location of the BTSs and their types; (A2) the traffic (in Erlang) between BTSs and between each BTS and the public network; (A3) the location of potential sites to install the BSCs and the MSCs; (A4) the cost and the capacity of the components (BSCs, MSCs, and links) of different types.

The objective is to minimize the cost of the design, including the cost of the equipment and their installation.

Many aspects of the overall design problem correspond to well-known operational research problems, such as graph partitioning [1], [2] or p-fixed hubs location [3]. Since these problems are NP-hard, exact algorithms are inappropriate in practice

Manuscript received October 12, 2003; approved for publication by Wha Sook Jeon, Division III Editor, August 30, 2005.

The authors are with the Department of Computer Engineering, École Polytechnique de Montréal, Montréal, Canada, email: {yanick.pomerleau, steven.chamberland, gilles.pesant}@polymtl.ca.

1229-2370/06/$10.00

for moderate and large size instances. As a result, heuristic approaches have been largely used for solving these aspects of the design problem [1]–[9].

The literature abounds on network planning and the optimization of cellular wireless networks [10] but few have addressed the design problem as a whole. Merchant and Sengupta [1], [2] studied only the assignment problem. Their algorithm starts from an initial solution, which they attempt to improve through a series of greedy moves, while avoiding to be trapped in a local minimum. The moves used to escape a local minimum explore only a very limited set of options. These moves depend on the initial solution and do not necessarily lead to a good final solution. Other heuristics, strongly inspired by that of Merchant and Sengupta, were proposed by Saha, Mukherjee, and Bhattacharya [11]. These heuristics are based on the formation of cell clusters related to the same switch. The cell where the switch resides is the root of the cluster, then each cluster is extended by judiciously adding other cells. Several versions of the algorithm were proposed. In general, these algorithms improve the results of Merchant and Sengupta but nevertheless remain ineffective for designing large size cellular wireless networks. André, Pesant, and Pierre [4] also studied the assignment problem and used a variable neighborhood search metaheuristic [12] to solve it. The neighborhoods are based on reassignment and redistribution of the BTSs to the switches. The initial solution is found by constraint programming. This algorithm improved the solutions found by the other heuristic methods such as simulated annealing [9] and tabu search [8] in quality and in CPU time. Sohn and Parc [3] studied the p hub location problem that consists of finding the location of p hubs and the cells' assignment so that the total transportation cost is minimized. The authors have proposed a mixed integer formulation to solve the problem. The computational results show that the model works well. Only few instances provided non-integer solutions, less than 0.02%. For those instances, integer solutions are obtained quickly with a branch and bound method.

Cox and Sanchez [7] studied the whole design process and used a tabu search metaheuristic, with embedded knapsack and network flow subproblems to design a least-cost telecommunications network to carry cell site traffic to wireless switches while meeting survivability, capacity, and technical compatibility constraints. In this context, each optimization problem is solved separately while taking into account its impact on the other ones. Because they added some network survivability constraints, some switches must be assigned to at least two other switches. Their results illustrated that optimal solutions can be found. The cost difference between 20% and 100% of switches with survivability constraints is very small (less than 1%). The global design problem defined here was considered by Chamberland and Pierre [6]. The authors proposed a mathematical programming model for this problem, demonstrated that it is NP-hard and proposed a tabu search (TS) algorithm. The TS algorithm determines a set of BSCs and a set of MSCs to move from a solution to another while taking into account tabu moves and the aspiration criterion. The results showed that good solutions can be found with this approach (within 6%, on average, from a proposed lower bound) in under an hour of CPU time.

The lower bound used to evaluate the quality of the solutions obtained by this TS algorithm comes from an exact algorithm applied to a relaxation of the problem. Therefore it is difficult to know what proportion of the gap is an optimality gap. In this context, it is interesting to investigate other avenues to confirm or improve the quality of the solution obtained by the TS algorithm. The tree-topology design problem lends itself well to modeling in constraint programming (CP) [13]. In this paper, we explore such an avenue to solve the design problem of wireless cellular networks. The paper is organized as follows. Section II presents the constraint programming model for the global design problem of cellular wireless networks and, in Section III, a heuristic is proposed. Section IV presents numerical results and we conclude in Section V.

II. THE CONSTRAINT PROGRAMMING MODEL

In this section, we propose a CP model for the global design of the network subsystem of cellular wireless networks. This model is developed for the second generation of cellular networks using, for instance, global system for mobile communications (GSM), code division multiple access (CDMA), or time division multiple access (TDMA) systems [14]. However, it can also be used for the design of third generation networks using, for instance, wideband CDMA (WCDMA) or CDMA2000 systems [14], if a tree-topology architecture is selected for the network subsystem.

The application of CP to the resolution of combinatorial problems rests on the notion of finite domain variables. For each such variable, the (finite) set of all possible values it may take, called its domain, is saved and updated as the computation progresses. Constraints are imposed on the combinations of values these variables may take and are enforced using specialized filtering algorithms, either built-in or user defined, that work by removing inconsistent values from the domains. Ideally, the algorithm should remove all inconsistent values, that is, those for which no feasible combination of values for remaining variables can be found, but this cannot always be achieved with reasonable efficiency. The constraints (or, more precisely, their algorithms) interact through shared variables (or, more precisely, their domains). Accordingly, our model for the problem at hand is composed of such constraints.

This filtering out of inconsistent values will not be sufficient in general to obtain a solution directly. Search is necessary, usually taking the form of an enumeration tree whose branches correspond to fixing a variable to a value in its current domain. Many strategies for choosing the next variable to fix and the next value to try have been devised and are often problem dependent. At each node of the tree, more filtering can occur since at least one of the domains has shrunk. For more details about constraint programming see [13].

The following notation is used throughout the paper.

Sets:

- I, the set of BTSs;
  - α_i, the capacity (in circuits) of the BTS $i \in I$;
  - η_i, the number of links necessary to connect it to a BSC;
- J, the set of BSC sites;
- K, the set of MSC sites;
- L, the set of links types;

  - β_ℓ, the capacity (in circuits) of a link of type $\ell \in L$;
- S, the set of BSC types;
  - α_s, the capacity (in circuits) of a BSC of type $s \in S$;
  - η_s^I, the maximum number of BTS interfaces of a BSC of type $s \in S$;
  - η_s^T, the maximum number of MSC interfaces of a BSC of type $s \in S$;
- T, the set of MSCs types;
  - α_t, the capacity (in circuits) of a MSC of type $t \in T$;
  - η_t^S, the maximum number of BSC interfaces of a MSC of type $t \in T$.

Note that all of these sets are disjoint.

Bandwidth parameters:
- $g_{ii'}$, the average number of communications per hour from BTS $i \in I$ to BTS $i' \in I$;
- g_{iP}, the average number of communications per hour from BTS $i \in I$ to the public network;
- g_{Pi}, the average number of communications per hour from the public network to BTS $i \in I$.

Cost parameters:
- a_{ij}, the link and interface card costs (including the installation cost) for connecting BTS $i \in I$ to a BSC at site $j \in J$;
- $b_{\ell jk}$, the link and interface card costs (including the installation cost) for connecting a BSC at site $j \in J$ to an MSC at site $k \in K$ through a link and interfaces of type $\ell \in L$;
- c_j^s, the cost of a BSC of type $s \in S$ and installing it at site $j \in J$;
- d_k^t, the cost of an MSC of type $t \in T$ and installing it at site $k \in K$.

Decision variables:
- v_i, the BSC site assigned to the BTS $i \in I$;
- w_j, the MSC site assigned to the BSC at site $j \in J$;
- x_j, the BSC type installed at site $j \in J$;
- y_k, the MSC type installed at site $k \in K$;
- $z_{j\ell}$, the number of links of type $\ell \in L$ installed from the BSC site $j \in J$ to a MSC.

Traffic variables:
- t_i, the traffic (in Erlang) from BTS $i \in I$ to a BSC;
- t_j, the traffic (in Erlang) from BSC $j \in J$ to a MSC.

A. The Model

Constraints (C1) and (C2) are already enforced by the choices of decision variables. For example, the v_i variables ensure that the BTS i is connected to exactly one BSC. Indeed, the variables can only be assigned to one value of their domain. The other constraints are explicitly enforced in the model.

The constraint programming model for the design problem of wireless cellular networks, noted design problem for wireless cellular networks (DPWCN), can now be given by

$$\min\left(\sum_{i\in I} a_{iv_i} + \sum_{j\in J}\sum_{\ell\in L} z_{j\ell} b_{\ell j w_j} + \sum_{j\in J} c_j^{x_j} + \sum_{k\in K} d_k^{y_k}\right) \quad (1)$$

where a_{iv_i} is the link and interface card costs (including the installation cost) for connecting BTS $i \in I$ to a BSC at site $v_i \in J$, $b_{\ell j w_j}$ is the link and interface card costs (including the installation cost) for connecting a BSC at site $j \in J$ to an MSC at site $w_j \in K$ through a link and interfaces of type $\ell \in L$, $c_j^{x_j}$ is the cost of a BSC of type $x_j \in S$ and of installing it at site $j \in J$ and $d_k^{y_k}$, the cost of an MSC of type $y_k \in T$ and of installing it at site $k \in K$.

Subject to

BSC capacity constraint (BTS interface level)

$$\sum_{i\in I}((v_i = j)\eta_i) \le \eta_{x_j}^I, \qquad j \in J \quad (2)$$

where $(v_i = j)$ is a 0–1 expression such that $(v_i = j)$ is equal to 1 if and only if $v_i = j$.

BSC capacity constraint (MSC interface level)

$$\sum_{\ell\in L} z_{j\ell} \le \eta_{x_j}^T, \qquad j \in J. \quad (3)$$

BSC capacity constraint (switch fabric level)

$$\sum_{i\in I}((v_i = j)\alpha_i) \le \alpha_{x_j}, \qquad j \in J. \quad (4)$$

MSC capacity constraint (BSC interface level)

$$\sum_{j\in J}((w_j = k)\sum_{\ell\in L} z_{j\ell}) \le \eta_t^S y_k, \qquad k \in K. \quad (5)$$

MSC capacity constraint (switch fabric level)

$$\sum_{j\in J}((w_j = k)\sum_{\ell\in L} z_{j\ell}\beta_\ell) \le \alpha_{y_k}, \qquad k \in K. \quad (6)$$

Link capacity constraint

$$t_i \le \alpha_i, \qquad i \in I \quad (7)$$

$$t_j \le \sum_{\ell\in L} z_{j\ell}\beta_\ell, \qquad j \in J. \quad (8)$$

Traffic flow conservation constraint

$$t_j = \sum_{i\in I}(v_i = j)\sum_{o\in I}((v_o \ne j)(g_{io} + g_{oi})) + \sum_{i\in I}((v_i = j)(g_{iP} + g_{Pi})), \qquad j \in J \quad (9)$$

$$t_i = \sum_{o\in I}(g_{io} + g_{iP}) + \sum_{o\in I} g_{oi} + g_{Pi}, \qquad i \in I. \quad (10)$$

Domains of the variables

$$v_i \in J\ (i \in I),\ w_j \in K\ (j \in J), x_j \in S\ (j \in J), y_k \in T\ (k \in K), z_{j\ell} \in N\ (j \in J, \ell \in L). \quad (11)$$

The cost function (1) of DPWCN, representing the total cost of the network subsystem, is composed of the cost of the links and interface cards and the cost of the BSCs and MSCs (including the installation costs). Constraint (2) require that the total number of BTS to BSC links connected at site $j \in J$ be less than or equal to the maximum number of BTS interfaces that can be installed in the BSC type set up at that site. Constraint (3) require that the total number of BSC to MSC links connected at site $j \in J$ be less than or equal to the maximum number of

MSC interfaces that can be installed in the BSC type set up at that site and constraint (4) impose that the sum of the BTS rates connected to the BSC type installed at site $j \in J$ be less than or equal to its switch fabric capacity. Constraint (5) require that the total number of BSC to MSC links connected to site $k \in K$ be less than or equal to the maximum number of BSC interfaces that can be installed in the MSC type set up at that site and constraint (6) impose that the sum of the rates of the BSC to MSC links connected to the MSC type installed at site $k \in K$ be less than or equal to its switch fabric capacity. Constraints (7) and (8) are respectively BTS to BSC and BSC to MSC link capacity constraints and constraints (9) and (10) are the traffic flow conservation constraints.

III. THE HEURISTIC METHOD

In this section, we present a local search heuristic for DP-WCN, denoted constraint programming local search (CPLS). The main idea behind CPLS is to find a good subset of BSC sites to use by quickly estimating the cost of such subsets.

A. The Initial Solution

Algorithm 1 initial solution algorithm

$o \leftarrow$ number of BSC sites used by the model
$p \leftarrow$ number of BTSs
$q \leftarrow$ number of sites to permute
sort J in increasing order of sum of distances with p nearest BTSs
$J' \leftarrow$ the first o sites from J
launch the probe on J' to determine the cost
repeat
    permute q sites of J' with q sites not in J'
    launch the probe on J' to determine the cost
until $\nexists$ some possible distinct permutation
return J' of better cost

It begins by looking for an initial solution. The initial subset is selected by considering the closest BSC site to the BTSs. We order the BSC sites according to the sum of distances with the p nearest BTS (Algorithm 1). We choose the o BSC sites for which p BTSs will be assigned with the lower cost. Then, we replace q BSC sites of this subset by q others to evaluate the impact of each BSC site. We keep the best subset as the initial solution. The parameters used to search for the initial solution were obtained by experimentation. Algorithm 1 mentions launching a probe—we defer the discussion of this topic until Section III-C.

B. The Local Search

Next, we perform a local search on the initial solution found, trying to remove the BSC sites having the lowest number of BTS assigned to them while keeping those with many BTS assigned to them since this is good for the optimization of the cost of BTS to BSC links. The local search procedure (Algorithm 2) begins by ordering the BSC sites in the current solution according to the number of BTS assigned to each one. After that, at each iteration of the local search, we select the number of BSC sites,

Algorithm 2 local search algorithm

$o \leftarrow$ number of BSC sites used by the model
$J \leftarrow$ set of possible BSC sites
$n \leftarrow$ number of BSC sites considered for removal
$m \leftarrow$ number of BSC sites considered for addition
repeat
    sort J in increasing order of the number of BTS assigned
    $J' \leftarrow$ the first o sites from J
    $J'' \leftarrow J - J'$
    for i in $1 \cdots n$ **do**
        remove the i-th site of J'
        launch the probe to estimate the neighbor's cost
        for j in $1 \cdots m$ **do**
            replace the i-th site of J' by the j-th of J''
            launch the probe to estimate the neighbor's cost
        end for
    end for
    compare the probes and keep the least expensive solution
until $\nexists x \in N(x_k)$ **such that** $f(x) \leq f(x_k)$

denoted n, used in the current solution having the lowest number of BTS assigned to them. To visit the neighborhood $N(x_k)$ of the current solution x_k at the iteration k, we evaluate the cost f_x of solutions obtained by changing one of the n sites previously identified ($i \in n$) by another not used in the current solution as well as the solution obtained by removing that site. The best solution is selected as the current solution. The local search continues while the cost of the current solution decreases.

C. Probing

Probing consists of estimating the cost of a solution that uses only a given subset of BSC sites by an incomplete exploration of the search space reporting the least expensive solution found within a fixed computation time. In order to do this, we decompose our CP model into two subproblems. The purpose of the first subproblem is to assign the BTSs to the BSCs and to select the BSC types at each BSC site for a given subset of BSC sites to use. The purpose of the second subproblem is to assign the BSCs to the MSCs, to select the MSC types and the BSC to MSC link types and to ensure traffic flow conservation. For a given subset of the BSC sites, the subproblems are solved one after the other using a constraint programming software (e.g., ILOG OPL Studio [15]). The purpose of this decomposition is to avoid propagation (domain reduction) through the traffic flow conservation constraints during the assignment of v_i's. Such a propagation would be too time consuming because of a double summation in (9) that involves v_i's. Indeed, at each assignment of a v_i variable, (9) will be used to reduce the domain of t_j. To accomplish this task we would have to find the BTSs that are assigned to the same BSC site by calculating the double summation.

In the first subproblem, corresponding to the section between BTS and BSC, the assignment heuristic first fixes each BTS to a BSC site (v_i) (Fig. 2). In the figures the "forall" keyword is used to specify the order in which variables are assigned and the "tryall" keyword is used to specify the order in which the values are tried. The BTS variables are ordered by decreasing

Strategy 1: BTS to BSC site assignment
forall ($i \in I$ ordered by decreasing η_i)
tryall ($j \in J$ ordered by increasing distance between i and j)
$v_i = j$

Fig. 2. The strategy of the first subproblem.

Table 1. Features of the BTS types.

	Type A	Type B	Type C
Capacity (circuits)	96	288	576
Num. of BTS DS-1 interfaces	1	3	6

Table 2. Costs of the BSC types (including the installation costs).

	Type A	Type B	Type C
Capacity (circuits)	5,000	10,000	15,000
Max. num. of BTS interfaces	15	30	60
Max. num. of MSC interfaces	15	30	60
Cost ($)	50,000	90,000	120,000

Table 3. Costs of the MSC types (including the installation costs).

	Type A	Type B	Type C
Capacity (circuits)	100,000	200,000	300,000
Max. num. of BSC interfaces	50	100	150
Cost ($)	200,000	350,000	500,000

Table 4. Costs of the interface types (including the installation costs).

Interface type	Capacity (circuits)	Cost ($)
DS-1	96	500
DS-3	2,688	2,500

Table 5. Costs of the BTS to BSC links (including the installation costs).

BSC type	Num. of DS-1	Capacity (circuits)	Cost($)
A	1	96	2,000
B	3	288	3,000
C	6	596	4,000

Table 6. Costs of the BSC to MSC links (including the installation costs).

Link type	Capacity (circuits)	Cost ($/km)
DS-1	96	2,000
DS-3	2,688	4,000

Table 7. Initial heuristic parameters.

$\|I\|$	o	p	q
50	4	12	2
100	4	18	2
150	4	24	3
200	4	28	3

number of links used, to assign first the BTS with more links and reduce the total link cost. The values in the domain of each BTS variable v_i are ordered by increasing distance with each BSC site. Hence, the BTSs are first assigned to the nearest BSC site available. Next, we can determine the BSC type to install at each site. Knowing the BTS assignment it is easy to install the least cost BSC type at each site (x_j). For each BSC site, we compute the minimum cost equipment to install by adding the demands of the BTSs assigned to the BSC site.

The second subproblem, the network between the BSCs and the MSCs, has small instances, allowing us to evaluate each possibility. The solutions found for this section are optimal according to the choices made in the first subproblem. The search procedure first determines the MSC type to install at each site (y_k) considering the demand of each BSC (Strategy 2, Fig. 3). The variables are assigned in simple lexicographic order because we evaluate all the possibilities so we don't need to begin with a particular variable. As mentioned before, by ordering the values by increasing cost, the first solution obtained for this variable is the least expensive one. Finally, we determine the number of links between the BSC and MSC sites (z_{jl}) (Strategy 3, Fig. 3). The variables are ordered by increasing link cost and the values are ordered by increasing number of links to install, consequently the first feasible solution found is the least expensive one. The z_{jl} assignment will also fix the BSC assignment (w_j): The BSC site will be assigned to the MSC site where there are links between them.

IV. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed heuristic. The heuristic was implemented in the ILOG OPL script language [15] on a AMD Athlon PC (1.6 GHz and 512 MB of RAM). For the tests, three BTS types, three BSC types and three MSC types are used. Their features are presented respectively in Tables 1–3. Moreover, DS-1 links are used to connect the BTSs to the BSCs and DS-1 and DS-3 links are used to connect the BSCs to the MSCs. The interface costs are presented in Table 4 and the link costs in Tables 5 and 6. Table 7 presents the parameters used to find the initial solution.

For the tests, 28 instances were generated as follows. $|I|$ points corresponding to BTSs' locations, $|J|$ points corresponding to the BSC sites and $|K|$ points corresponding to the MSC sites were generated in a 100 km by 100 km grid following a uniform distribution. The type of each BTS was selected randomly among the three BTS types considered (Table 1). Finally, the demand between each pair of BTSs and between the BTSs and the public network was generated randomly in the interval [0,0.2] Erlang, following a uniform distribution. All test results were compared to the results obtained on the same instances with a tabu search algorithm, denoted TS, proposed by Chamberland

Strategy 2: MSC type to MSC site assignment
  forall ($k \in K$)
    tryall ($t \in T$ ordered by increasing cost of an MSC of type t)
      $y_k = t$
Strategy 3: Number of links of each type between the BSC and the MSC assignment
  forall ($j \in J$)
    forall ($\ell \in L$ ordered by increasing cost of a link of type ℓ)
      tryall ($n \in 0 \cdots \min\{\eta^T_{y_k}, \alpha_{y_k}/\alpha_\ell\}$)
        $z_{j\,\ell} = n$

Fig. 3. The strategies of the second subproblem.

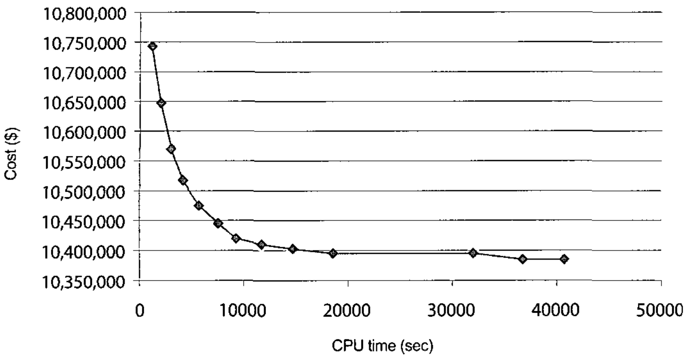


Fig. 4. Typical evolution of the cost function (200 BTS, 30 BSC, 10 MSC).

and Pierre [6].

The results are presented in Table 8. In this table, columns 1 to 3 present respectively the number of BTSs, the number of BSC sites and the number of MSC sites. Column 4 presents a lower bound, denoted LB, obtained by solving a relaxed version of the model which doesn't consider the traffic flow conservation constraints (see [6] for more detail concerning the lower bound). Columns 5 and 6 present respectively the results obtained by the TS algorithm and the corresponding CPU time. Columns 6 to 10 present the value of the solution found by algorithm CPLS, the CPU time to find it and the GAP that indicates the percentage of improvement of the solution value compared to the value of LB and TS. These results show that CPLS confirm and improve, on average, the solutions found by TS. Indeed, the average improvement is 0.17% and TS has found only eight better solutions than CPLS. Moreover, the average percentage gap between the solution values found by CPLS and the lower bound values is 5.13%.

As we can see in Table 8, the CPU time used by the CPLS approach is higher than the one used by TS algorithm. In this context of a problem at the strategic level, the quality of the solution found is more important than the speed of the algorithm. In any case, Fig. 5 illustrates that the initial solutions quickly compare to the final solutions.

V. CONCLUSIONS

In this paper, we studied the tree-topology design problem for wireless cellular networks that consists of selecting the location of the network nodes (BSC and MSC) and their types, designing the network tree-topology and selecting the link types. A constraint programming model has been proposed combined with a local search heuristic.

As mentioned before, the purpose of the heuristic strategies is to estimate quickly the cost of a subset of BSC sites. In Strategy 1, it is possible that some reassignment reduces the cost of the solution. This is explained by the fact that some BTSs could not be assigned to the nearest BSC site because of the capacity limit of a BSC. This possible reassignment provides the biggest uncertainty of our estimation method. However, in a real network, the BTS and the BSC sites are geographically distributed. So, it is possible that some BSC site has a demand exceeding their capacity but not in an excessive way. The BTSs that were not assigned to the nearest BSC site will be assigned to the second nearest one. The possible gain for a real network is small. The other strategies find the optimal solution according to the choices made by the first strategy. So the estimation is good for a real network.

Experimental results show that the local search heuristic produces, on average, better solutions than the best previously known algorithm for this problem. Since our local search approach is essentially a simple descent and therefore runs the risk of getting trapped in a local minimum, improvements could possibly be achieved through standard mechanisms to escape such minima [12], [16], [17]. A first attempt with a multi-start approach (multiple runs of the algorithm with a different initial solution) improved fewer than 10% of the solutions and by only 0.5% on average, at the expense of a higher computational cost. Similarly, preliminary experiments with variable neighborhood search only improved some of the solutions and by less than 1%.

REFERENCES

[1] A. Merchant and B. Sengupta, "Multiway graph partitioning with applications to PCS networks," in *Proc. IEEE INFOCOM'94*, Toronto, Canada, June 1994, pp. 593–600.
[2] A. Merchant and B. Sengupta, "Assignment of cells to switches in PCS networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp. 521–526, 1995.
[3] J. Sohn and S. Park, "Efficient solution procedure and reduced size formulations for p-hub location problems," *European J. Operational Research*, vol. 108, pp. 118–126, 1998.
[4] M. André, G. Pesant, and S. Pierre, "A variable neighborhood search algorithm for assigning cells to switches in wireless networks," *J. Computer Sci.*, vol. 1, no. 2, pp. 175–181, 2005.
[5] G. Amoussou, M. André, G. Pesant, and S. Pierre, "A constraint programming approach to assign cells to switches in personnal communication networks," *Annals of Telecommun.*, vol. 58, no. 3, pp. 584–604, 2003.
[6] S. Chamberland and S. Pierre, "On the design problem of cellular wireless networks," *Wireless Networks*, vol. 11, no. 4, pp. 489–496, 2005.
[7] L. A. Cox and J. R. Sanchez, "Designing least-cost survivable wireless backhaul networks," *J. Heuristics*, vol. 6, no. 4, pp. 525–540, 2000.

Table 8. Computational results.

$\lvert I\rvert$	$\lvert J\rvert$	$\lvert K\rvert$	LB	TS		CPLS			
			Cost	Cost	CPU	Cost	CPU	GAP (%)	
			($)	($)	(sec)	($)	(sec)	vs. LB	vs. TS
50	10	10	4,051,500	4,083,500	91	4,051,500	360	0.00	−0.78
		20	4,148,000	4,218,000	159	4,177,700	681	0.72	−0.96
	20	10	3,316,600	3,321,100	126	3,321,100	2033	0.14	0.00
		20	3,557,400	3,583,200	182	3,559,100	2445	0.05	−0.67
	30	10	3,518,300	3,525,200	176	3,518,300	6147	0.00	−0.20
		20	3,583,800	3,591,700	245	3,591,700	4092	0.22	0.00
	40	10	3,267,900	3,304,200	254	3,284,500	9968	0.51	−0.60
		20	3,318,800	3,366,100	363	3,337,600	12076	0.57	−0.85
100	10	10	6,865,300	7,215,800	200	7,136,300	718	3.95	−1.10
		20	6,573,100	6,944,900	320	6,861,000	517	4.38	−1.21
	20	10	6,244,200	6,599,400	220	6,576,900	4485	5.33	−0.34
		20	6,934,900	7,150,900	417	7,112,800	4730	2.57	−0.53
	30	10	6,114,500	6,217,300	381	6,157,900	14049	0.71	−0.96
		20	5,606,100	5,804,100	614	5,737,900	11756	2.35	−1.14
	40	10	5,399,700	5,607,300	573	5,586,600	16400	3.46	−0.37
		20	5,249,700	5,377,700	731	5,364,800	18807	2.20	−0.24
150	20	10	8,571,800	9,346,300	370	9,330,100	5374	8.85	−0.17
		20	8,208,200	8,913,100	774	8,924,900	6210	8.73	0.13
	30	10	7,725,000	8,294,500	593	8,289,400	13903	7.31	−0.06
		20	7,551,400	8,092,900	759	8,109,300	17120	7.39	0.20
	40	10	8,234,000	8,817,500	885	8,739,400	60179	4.86	−0.89
		20	7,360,400	8,037,900	1086	8,064,900	41513	9.57	0.34
200	20	10	10,608,200	11,843,700	492	11,811,400	13827	11.34	−0.27
		20	11,106,300	12,231,600	862	12,295,700	7533	10.71	0.52
	30	10	9,281,200	10,216,800	709	10,384,600	40764	11.89	1.64
		20	10,304,100	11,070,600	1030	11,312,000	25234	9.78	2.18
	40	10	9,111,700	10,030,000	1158	10,123,400	87318	11.10	0.93
		20	9,598,300	10,467,400	1489	10,549,000	55555	9.90	0.78

[8] S. Pierre and F. Houeto, "Assigning cells to switches in cellular mobile networks using taboo search," *IEEE/ACM Trans. Syst., Man, Cybernetics*, vol. 32, no. 3, pp. 465–478, 2002.

[9] S. Pierre and A. Quintero, "Assigning cells to switches in cellular mobile networks: A comparative study," *Computer Commun.*, vol. 26, no. 9, pp. 950–960, 2003.

[10] P. S. Bhattacharjee, D. Saha, and A. Mukherjee, "Heuristics for assignment of cells to switches in a PCSN: A comparative study," in *Proc. IEEE PWC'99*, Jaipur, India, Feb. 1999, pp. 331–334.

[11] D. Saha, A. Mukherjee, and P. Bhattacharya, "A simple heuristic for assignment of cell to switches in a PCS network," *Wireless Pers. Commun.*, vol. 12, pp. 209–224, 2000.

[12] P. Hansen, N. Mladenovic, and D. Perez-Brito, "Variable neighborhood decomposition search," *J. Heuristics*, vol. 7, no. 4, pp. 335–350, 2001.

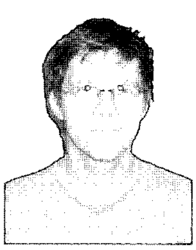
[13] K. Marriott and P. J. Stuckey, *Programming with Constraints: An Introduction*, The MIT Press, 1998.

[14] M. D. Yacoub, *Wireless Technology: Protocols, Standards, and Techniques*, CRC Press, 2002.

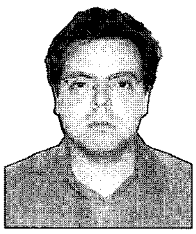
[15] ILOG, "ILOG OPL STUDIO 3.5—The Optimisation Language," 1999.

[16] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publisher, 1997.

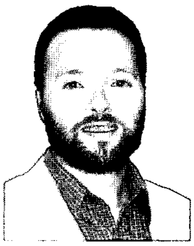
[17] S. Kirpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.



Yanick Pomerleau received the Computer Engineering degree from the École Polytechnique de Montréal, Canada, in 2002 and the master degree in Computer Engineering in 2003 also from the École Polytechnique. He is now working for Softimage in Montréal. His research interests include constraints programming and wireless networks planning and design.



Steven Chamberland received the Electrical Engineering degree from the École Polytechnique de Montréal, Canada, in 1994 and the Ph.D. degree in Operations Research in 1998 also from the École Polytechnique. From 1998 to 1999, he was with the network architecture group of Bell Canada and from 1999 to 2001, he was a professor at the École de technologie supérieure. Since 2001, he has been an associate professor of Computer Engineering at the École Polytechnique de Montréal where he is director of the Bell Canada Research Chair in Network Architecture. His research interests include telecommunication network planning and design and network architecture. Dr. Chamberland is a member of IEEE and ACM. He also serves on the Editorial Board of *Computers and Operations Research* and as an associate editor for *Telecommunication Systems* and *Information Systems and Operational Research* (INFOR).



Gilles Pesant is currently an associate professor in the Department of Computer Engineering at the École Polytechnique de Montréal, Canada, after postgraduate studies at McGill, U. Montréal, and the Centre for Research on Transportation. Over the last twelve years, his research interests have focused on Constraint Programming and its application to workforce scheduling, transportation logistics and telecommunications. He has authored more than thirty refereed papers in the area and has given several invited tutorials on Constraint Programming. He has been involved in the scientific programming of a dozen events and serves as an associate editor for *Journal of Heuristics* and *Information Systems and Operational Research* (INFOR).