A Genetic Algorithm for Assigning Cells to Switches in Personal Communication Networks

by Constantin Hedible and Samuel Pierre,
Mobile Computing and Networking Research Laboratory (LARIM)
Ecole Polytechnique of Montreal, Montreal, QC

Abstract

This paper proposes a genetic algorithm to solve the problem of assigning cells to switches in the planning phase of mobile cellular networks. Well-known in the literature as an NP-hard combinatorial optimization problem, this problem requires recourse to heuristic methods in order to obtain good (not necessarily optimal) solutions within a practical amount of time. Computational results obtained from extensive tests confirm the effectiveness of this algorithm in providing good solutions to practical sized problems.

Sommaire

Cet article propose un algorithme génétique pour résoudre le problème d'affectation de cellules aux commutateurs dans la phase de planification des réseaux cellulaires mobiles. Bien connu dans la littérature comme un problème difficile d'optimisation combinatoire, ce problème requiert le recours à des méthodes heuristiques pour obtenir de bonnes solutions, non nécessairement optimales, dans des temps de calcul raisonnables. Les résultats numériques confirment l'efficacité de cet algorithme pour produire de bonnes solutions à des instances du problème de taille pratique.

1.0 Introduction

Personal Communication Network (PCN) is a wireless communication network which integrates various services such as voice, video, electronic mail, accessible from a single mobile terminal. These various services are offered in an area called *coverage zone* which is divided into *cells*. In each cell is installed a *base station* which manages all the communications within the cell. In the cover zone, cells are connected to special units called *switches* which are located in mobile switching centers (MSC). When a user in communication goes from a cell to another, the base station of the new cell has the responsibility to relay this communication by allotting a new radio channel to the user. Supporting the transfer of the communication from a base station to another is called *handoff*. This mechanism, which primarily involves the switches, occurs when the level of signal received by the user reaches a certain threshold. We distinguish two types of handoffs. In the case of Figure 1 for example, when a user moves from cell B to cell A, it refers to *soft handoff* because these two cells are connected to the same switch. The MSC which supervises the two cells remains the same and the induced cost is low. On the other hand, when the user moves from cell B to cell C, there is a *complex handoff.* The induced cost is high because both switches 1 and 2 remain active during the procedure of handoff and the database containing information on subscribers must be updated.

The total operating cost of a cellular network includes two components: the cost of the links between the cells (base station) and the switches to which they are joined, and the cost generated by the handoffs between cells. It appears therefore intuitively more discriminating to join cells B and C to the same switch if the frequency of the handoffs between them is high. The problem of assigning cells to switches essentially consists of finding the configuration that minimizes the total operating cost of the network. The resolution of this problem by an exhaustive search method would entail a combinatorial explosion, and therefore an exponential growth of execution times. This problem belongs to the class of NP-complete problems, well-known especially in operational research. It relates to the problems of warehouse location [1] and graph partitioning [5]. This paper formulates the problem, proposes an algorithm for its solution, then summarizes and analyzes the computational results.

2.0 Formulation Of The Problem

The problem of assigning cells to switches in a cellular mobile network, as described by Merchant and Sengupta [6], can be formulated as follows: Given n cells and m switches, a matrix of the wiring costs between cells and switches, a matrix of handoff costs between cells, minimize the total cost of the network, by choosing the assigning configuration, under constraints of switches' capacity.

Locations of cells and switches are known. c_{ik} denotes the cost of wiring cell i to switch k, λ_i the call rate generated in cell i, and M_k the capacity (in number of calls) of the switch k. The problem of assigning cells to switches may be regarded as an integer programming one. Let's define the variable:

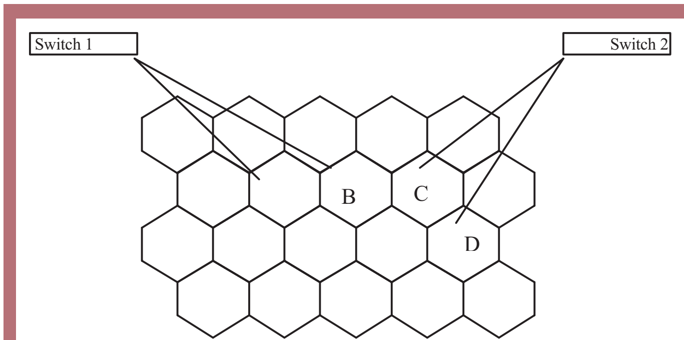

Figure 1: Geographical carving of the cover zone

$$x_{ik} = \begin{cases} 1 \text{ if cell i is assigned to switch k,} \\ 0\text{, otherwise.} \end{cases}$$

Considering that a given cell can be assigned to only one switch, we have the following constraint:

$$\sum_{k=1}^{m} x_{ik} = 1, i = 1, \ldots, n\, , i = 1, \ldots, n \qquad (1)$$

The constraint of capacity on the switches are expressed as follows:

$$\sum_{k=1}^{m} \lambda_i x_{ik} \le M_k, k = 1, \ldots, m\, , k = 1, \ldots, m \qquad (2)$$

On the other hand, the wiring cost is:

$$\sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik} x_{ik} \qquad (3)$$

Let's assume that H_{ij} and H'_{ij} are respectively the handoff cost if cells i and j are assigned to the same switch, and the handoff cost if they are assigned to different switches. These costs are more difficult to handle. We define therefore the additional variables:

$$z_{ijk} = x_{ik} \cdot x_{jk},\ i, j = 1, \ldots, n, \text{ and } k = 1, \ldots, m \qquad (4)$$

$$\text{and } y_{ij} = \sum_{k=1}^{m} z_{ijk} \quad i, j = 1, \ldots, n \tag{5}$$

From these definitions, the handoff cost by time unit is given by:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij} y_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij}(1 - y_{ij}) \tag{6}$$

The total cost to be minimized is the sum of the costs of links between cells and switches and those of handoffs between cells. It can be written as follows:

$$F = \sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik} x_{ik} + \sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij} y_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij}(1 - y_{ij}) \tag{7}$$

3.0 The Proposed Genetic Algorithm

Genetic algorithms (GA) are robust search techniques based on natural selection and genetic production mechanisms. GAs perform a search by evolving a population of candidate solutions through non-deterministic operators and by incrementally improving the individual solutions forming the population using mechanisms inspired from natural genetics and heredity (e.g., selection, crossover and mutation). In many cases, especially with problems characterized by many local optima (graph coloring, travelling salesman, network design problems, etc.), traditional optimization techniques fail to find high quality solutions. GAs can be considered as an efficient and interesting option.

GAs [3] are composed of three phases: a phase of creation of an initial population, a phase of alteration of this population by applying various genetic operators on its elements, and finally a phase of evaluation of this population during a certain number of generations. Each generation is supposed to provide new elements that are better than those of the preceding generation. Intuitively, the larger the number of generations, the more refined the solution. It is hoped that the last generation will provide a good solution, but this solution is not necessarily the optimum.

In our adaptation, we opted for a non-binary representation of the chromosomes [4]. As shown in Figure 2, the genes (squares) represent the cells, and the integers they contain represent the switch to which the cell i (gene of the i^{th} position) is assigned. Our chromosomes have therefore a length equal to the number of cells in the network, and the maximal value that a gene can take is equal to the number of switches.

3.1 Initial population formation

The first element of the initial population is the one obtained when all cells are assigned to the nearest switch. This first chromosome is created therefore in a deterministic way. The creation of other chromosomes of the population is probabilistic and follows the strategy of population without doubles. This strategy ensures the diversity of the population and a good coverage of the search space. All chromosomes of the population verify the unique assignment constraint, but not necessarily the switch capacity constraint. The maximum size of this initial population cannot exceed m^n in order to avoid duplicates. Various operators and functions are then applied to this population.

3.2 Crossover operator

This operator creates two new "child" chromosomes by crossing the parent chromosomes (taking genes 1 … i of one parent and genes (i+1) … n of the other parent for some randomly chosen i). We randomly choose a pair of chromosomes from the population, then either create two new chromosomes by applying the crossover operator, or apply the *inversion* operator to modify the parent chromosomes by reversing the order of their genes. The decision of which operator to apply is governed by a parameter called the crossover probability. In either case, the parent chromosomes are kept in the new population.

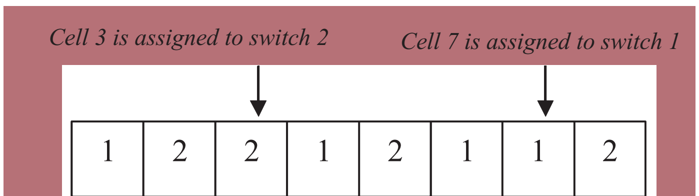

Figure 2: A chromosome representation for a network of 2 switches and 8 cells.

3.3 Mutation operator

This operator randomly modifies (with a certain probability) one or more genes of a chromosome. This is necessary to bring back the genetic material that would have been forgotten during the generations. Note that some of the new chromosomes may violate the switch capacity constraint - such chromosomes will be discarded later. By applying the mutation operator to each element of the population, we create a new population with twice the number of elements (the mutated chromosomes plus the original ones). This new population is evaluated, then sorted.

3.4 Evaluation function

One of the key elements in a GA is the evaluation function which determines how well the chromosomes suit the needs of the problem domain. In the first stage of evaluation, we compute the cost associated with each chromosome and then sort them in ascending order of cost. The second stage of evaluation checks if the chromosomes violate the capacity constraint on the switches. We keep solutions that violate the capacity constraint by 10% or less in a list which will be checked later, since a small modification could make them feasible.

3.5 Selection operator

To select the elements of the new generation, we used the method of the *casino caster*. As the problem that we have to solve is a problem of minimization, we applied the caster to the inverses of the cost values of the chromosomes of the population. We recover then in the new selected population either chromosomes that verify the constraint on the capacity of the switches or those that violate it. The number of generations is fixed at the beginning of the execution. We inserted into our adaptation the concept of *cycle* - each cycle runs several successive genetic processes. At every cycle, a new initial population is created. Figure 3 shows the flow chart of the genetic process.

For i:=1 to Nb_Cycle
Generate a new population
For j:=1 to Nb_Generation

- Apply crossover operator
- Apply mutation operator
- Evaluate the obtained population
- Sort the population
- Verify if the new chromosomes violate the capacity constraint
- Save the best chromosomes
- Apply selection operator

Figure 3: Genetic process flow chart.

4.0 Computational Results

To implement the proposed adaptation, we designed a program (in C++) which has a complexity of mn^2 (m and n being respectively the number of switches and the number of cells). Two files essentially constitute the input data for this program. The program was run on a 450 MHz Pentium III PC running Linux. To verify the performance of our algorithm, we performed some tests on networks of different sizes ranging from 15 cells and 2 switches to 200 cells and 7 switches. Each test was performed 5 times and we report the average costs. We repeated the experiments with various parameter values in order to see which values worked best.

Figure 4 illustrates the effect of the population size on the obtained results. It shows, for 4 input files representing different networks of the

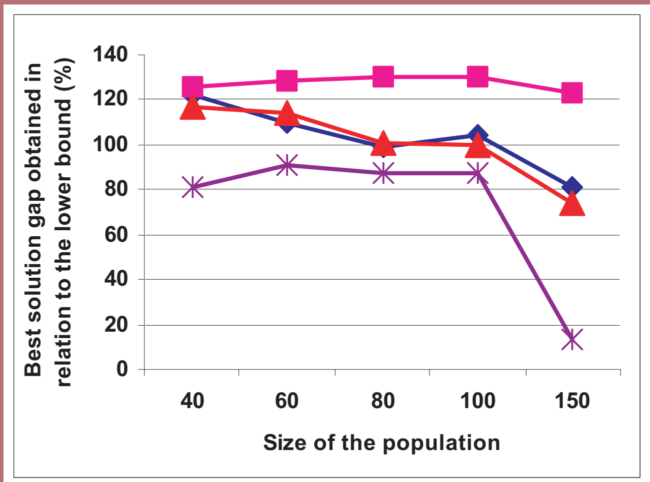

Figure 4: Effect of the population size.

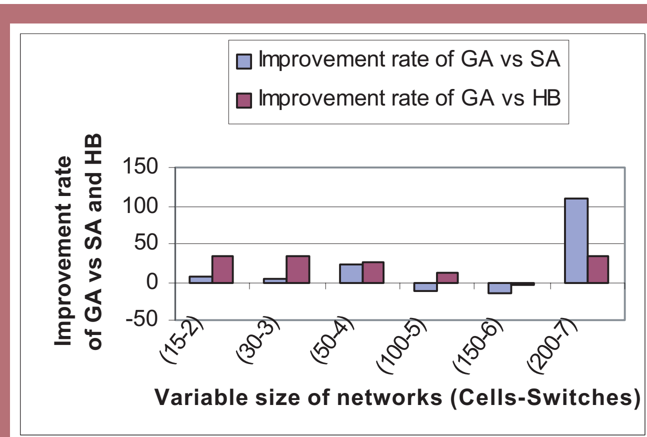

Figure 5: Comparative results obtained for networks of variable size.

same size (7 switches and 200 cells), the gap between the obtained solutions and the lower bound obtained by assigning each cell to the nearest switch. We executed our algorithm on these networks with population sizes of 40, 60, 80, 100 and 150 elements. The other parameter values were:

- Number of generations: 40
- Number of cycles: 20
- Crossover probability: 0.9
- Mutation probability: 0.08

4.1 Network of 200 cells and 7 switches

The obtained solutions are very near or identical to the optimal ones. Nevertheless, we generally note, for moderate- or large-sized networks, that as the size of the population increases, the gap between the obtained solutions and the lower bound decreases. One could conclude that a larger population gives better results because it introduces a high diversity in the population and permits good coverage of the search space.

We compared the results obtained with our algorithm (GA) with those obtained by application of two other methods that have all been designed to solve the problem of assigning cells to switches in cellular mobile networks. Those methods are the HB heuristic proposed by Beaubrun et al. [2], and the method of simulated annealing (SA). We performed the tests on two series of data. The first set related to a variable number of switches, and the second to a fixed number of switches. These methods have been coded by the authors and we use the same sets of data to achieve the comparison. The results are reported in tables 1 and 2. These results represent the costs of different networks and all the reported solutions are feasible.

The results of this comparative survey are summarized in figures 5 and 6. Our genetic algorithm provides better results than the other methods for small- and moderate-sized networks. As shown in Figure 5, our algorithm gets results that are about 30% better for networks up to 50 cells and 4 switches (small and moderate-sized networks).

The results provided by GA are always better than those provided by HB, as shown in figures 5 and 6. The improvement rate is in general higher than 20% for the networks having 3 switches and a number of cells lower or equal to 60. And in general, GA provides better results than simulated annealing. Sometimes solutions provided by GA are less good than those generated by SA, especially for networks of 100 and 150 cells.

In summary, considering the overall performance of these different heuristics, the proposed GA generally gives better results than simulated annealing and the heuristic HB proposed by Beaubrun et al. [2].

5.0 Conclusion

In this paper, we have proposed an adaptation of the genetic algorithm to solve the problem of assigning cells to switches in Personal Communication Networks. Computational experiments show that our method compares favorably with 2 other methods (simulated annealing and heuristic HB [2]) when applied to the same problem.

6.0 References

[1]. Abdinnour-Helm S. 1998. "A hybrid heuristic for the uncapacited hub location problem", European Journal of Operational Research, Vol. 106, pp. 489-499.

[2]. Beaubrun R., Pierre S. and Conan J. "An efficient method for Optimizing the Assignment of Cells to MSCs in PCS Networks", Proceedings of the 11th International Conference on Wireless Comm. Wireless 99, Vol. 1, July 1999, Calgary (AB), pp. 259-265.

[3]. Goldberg D.E., 1989. Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison-Wesley, Don Mills.

Table 1: Comparative results for GA, SA and HB (variable numbers of switches)

# of Cells	# of Switches	GA	SA	HB
15	2	114	123	153
30	3	394	405	524
50	4	697	851	873
100	5	2265	1999	2511
150	6	4980	4271	4807
200	7	3721	7801	4963

Table 2: Comparative results for GA, SA and HB (fixed numbers of switches)

# of Cells	# of Switches	GA	SA	HB
15	3	133	139	169
20	3	238	189	292
30	3	395	369	498
40	3	424	611	581
50	3	600	748	816
60	3	917	832	1071

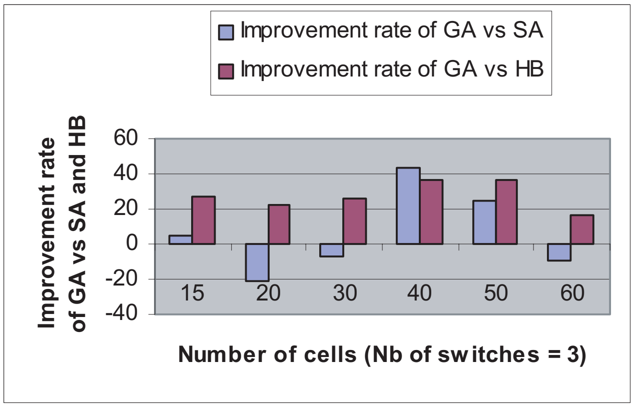

Figure 6: Comparative results obtained for networks of 3 switches and variable number of cells ( ).

[4]. Gondim, R.L.P. "Genetic algorithms and the location area partitioning problem in cellular networks". Proceedings of the Vehicular Technology Conference '96, Atlanta, VA April 29-30, May 1, 1996, pp. 1835-1838.

[5]. Krishnamurthy B. 1984. "An improved min-cut algorithm for partitioning VLSI networks", IEEE Transactions on Computers, Vol. C-33, pp. 438-446.

[6]. Merchant A. and Sengupta B. 1995. "Assignment of Cells to Switches in PCS Networks", IEEE/ACM Transactions on Networking, Vol. 3, No 5, pp. 521-526.

About the authors

Constantin Hedible received a B.Eng degree in electrical engineering in 1997 from Université Nationale du Bénin and an M.Sc.A. in Electrical Engineering from École Polytechnique de Montréal in 2000. From 1999 to 2001, he has been a research assistant in the Mobile Computing and Networking Research Laboratory (LARIM), working in optimization of telecommunications networks. Since 2002, he has been working as network manager in a leading company in telecommunications.

Samuel Pierre is a Professor of Computer Engineering at École Polytechnique de Montréal where he is Director of the Mobile Computing and Networking Research Laboratory (LARIM) and NSERC/Ericsson Industrial Research Chair in Next-generation Mobile Networking Systems. He is the author of four books, co-author of two books and six book chapters, as well as over 200 other technical publications including journal and proceedings papers. His research interests include wireline and wireless networks, mobile computing, performance evaluation, artificial intelligence, and electronic learning. He is a Fellow of the Engineering Institute of Canada, a senior member of IEEE, a member of ACM and IEEE Communications Society. He is an Associate Editor of IEEE Communications Letters and IEEE Canadian Review, and he serves on the editorial board of *Telematics and Informatics* published by Elsevier Science.