International Journal of Soft Computing 1(3): 166-169, 2006

A Genetic Algorithm Approach for Assigning Mobile Base Stations to Switches in Cellular Mobile Networks

Sakthivel, S. and R. Suresh
Department of Computer Science and Engineering,
Alagappa University Karaikudi, India

Abstract: In this study, we investigate a specific optimization problem involving the optimal assignment of mobile base stations to switches which minimize cost. This assumes that all base stations locations are fixed and known. The cost has two components, one is the cost of handoffs that involves two switches and the other is the cost of cabling. This problem is modeled as a complex integer programming problem and it is an NP-hard problem which, for real size mobile networks, could not be solved using exact methods. 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. In this context, heuristic approaches a Genetic Algorithm (GA) can be used. This study proposes a genetic algorithm to solve this problem. The implementation of this algorithm has been subject to extensive tests. The result obtained confirms the efficiency and the effectiveness of GA to provide good solutions for moderate- and large-sized cellular mobile networks. The goals and the present status of our research are also discussed.

Key words: NP-hard problem, handoffs, base stations, switches, Genetic Algorithm (GA)

INTRODUCTION

During the last decade, there has been a tremendous growth in the deployment of mobile communication systems. The need to exchange information with a user, anywhere and anytime, led to the cellular mobile networks, which are wireless networks integrating several services including electronic mail. The basic system architecture of mobile communication systems is to divide the geographic area to be covered into a collection of smaller areas or hexagonal cell. Each cell contains a Base Station (BS) covering a small geographical area and using an antenna for communications among users with pre-assigned frequencies. The base station is also connected to a switching computer or Switches, which allows it to connect mobile calls to and from the public switched network. A number of cells are chosen to install switches that communicate with each other and serve as relays for communication between cells and provider backbone[1].

An important concept in cellular communication systems is that of the handoff. A handoff occurs when a moving mobile leaves one cell and enters an adjacent one. Then the mobile switches radio communications with the base station of the original cell to radio communications with the base station of the new cell. Thus we have a handoff of the mobile. This handoff is accomplished in a manner that is transparent to the mobile user. Essentially, radio signal strength is measured so that it is clear when a mobile is moving away from the coverage area of the one base station and into the coverage area of another base station. Generally, a single switch may be connected to multiple cells. Thus there are two types of handoff. In one, a mobile moves from one cell to an adjacent cell, both of which are under the control of the same switch. This is a relatively simple handoff in as much as it involves only the one switch and no location updates. In the second type of handoff is complex handoff, a mobile moves from one cell to an adjacent one, each of which is connected to a different switch, This leads to complications in terms of the need to update location databases, a complex protocol between the two switches and the continued routing of the cell through the first switch for billing[2].

In the real world, the assignment depends heavily on the geographical conditions and traffic loads. However, we do not consider in this case any geographical conditions and restrain ourselves to the traffic load constraints. Moreover, even if the problem is not a real-time problem, it cannot be solved exactly for many reasons. The first one is that the search for an exact solution is too much time-consuming. For a network with m switches and n cells, m^n solutions should be examined. With 100 cells and 2 switches, if we take 1 ns (1000 MHZ) per solution, the total running time is about 4×10^{13} years. The second reason is that, as the traffic grows, the operator will need to run again from time to time the

Corresponding Author: S. Sakthivel, Department of Computer Science and Engineering, Alagappa University Karaikudi, India

assignment algorithm. Therefore, he could not afford prohibitive times. For these reasons, only heuristic approaches are used in the literature to solve this problem.

PROBLEM FORMULATION AND RELATED WORKS

The assignment problem of an assigning mobile base stations to switches in a cellular mobile network as described by Merchant and Sengupta[3]. This assignment problem consists of determining a cell assignment pattern, which minimize a certain cost function, while respecting certain constraints, especially those related to limited switch's capacity. An assignment of cells can be carried out according to a single or double's homing. A single homing of cells corresponds to the situation where a cell can be assigned to a single switch. When a cell is related to two switches then that is refers to a double homing. In this paper, only single homing is considered.

In this optimization problem, let N be number of cells to be assigned to M switches. The location of base stations (cells) and switches is fixed and known.
The following variables will be defined:

- h_{ij} Cost per unit time of handoffs from cell i to j. This is propositional to the volume of handoffs between cell i and cell j.
- c_{ik} The cost per time of cabling between cell i and switch k. This cost could be adjusted (for amortization, the cost of money, etc.)
- λ_i No. of calls processed by cell i in unit time
- M_k Call processing capacity of switch k.

Next, a mathematical program is setup in the form of an integer linear programming problem. Let us define the two following classes of decision variables:

$$x_{ij} = \begin{cases} 1 & \text{Cell i assigned to switch k.} \\ 0 & \text{otherwise} \end{cases}$$

There is a constraint on the call processing ability of each switch:

$$\sum_{i=I}^{N} \lambda_i x_{ik} \le M_k \quad k=1,2......M$$

A second constraint forbids each cell to be assigned to more than one switch:

$$\sum_{k=I}^{M} x_{ik} = M_k \quad i=1,2......N$$

Now the objective function has two additive components.
One is the cost of cabling switches to cells:

$$Z_{cabling} = \sum_{i=I}^{N} \sum_{k=I}^{M} c_{ik} x_{ik}$$

The individual c_{ik} may be proportional to distance and/or take into account local cabling costs.
The second cost component is the cost of handoffs occurring between two distinct switches. To account for this, first define Z_{ijk} with $i \neq j$.

$$Z_{ijk} = x_{ik} x_{jk} \text{ for } i, j = 1,2,....N \text{ and } k = 1,2,....M$$

One can seen that Z_{ijk} is equal to 1 if cells i and j, with $i \neq j$, are both connected to the same switch k, otherwise Z_{ijk} is equal to 0.
Next, define y_{ij}:

$$y_{ij} = \sum_{k=I}^{M} Z_{ijk} \quad \text{for } i, j=1,2,.....N \text{ and } i \neq j.$$

where y_{ij} takes the value 1 if cells i and j are both connected to the same switches and the value 0 if cells i and j are connected to the different switches.
The cost of handoffs (again, per unit time) is:

$$Z_{handoff} = \sum_{i=I}^{N} \sum_{j=I}^{N} h_{ij} (1 - y_{ij})$$

It can be seen that h_{ij}, cost of per unit of time of handoffs between cell i and cell j, makes a contribution towards $Z_{handoff}$ only if cell i and cell j are connected to different switches.
The overall cost function is Z = Z and$_{cabling}$ + $Z_{handoff}$

$$Z = \sum_{i=I}^{N} \sum_{k=I}^{M} c_{ik} x_{ik} + \sum_{i-I}^{N} \sum_{j=I}^{N} h_{ij} (1 - y_{ij})$$

GENETIC ALGORITHM

As computers become more and more powerful and omnipresent, they are expected to solve increasingly difficult problems. Since the Assignment problems defined in the previous section as a special case, they turn out to be NP-hard. The development and use of optimization models is well established. Our research in this area is primarily concerned with using Genetic Algorithms to solve the assignment problem[4]. Genetic

Algorithm (GA) is evolutionary optimization approaches which are an alternative to the traditional optimization methods[5]. GAs is a fairly easy and effective method ofcomputing an nondeterministic problem.

A Genetic Algorithm (GA) is an adaptive search technique based on the principles of natural evolution[6]. GAs iteratively applies genetic operators for selection, crossover, mutation and reproduction on a given population to improve its average fitness generation by generation. GAs has been successfully used to solve many combinatorial optimization problems such as Travelling Salesman Problem (TSP), the Assignment Problem and the Knapsack Problem.

In 1975 J.Holland introduced Genetic Algorithms which are a computational method of simulating nature's evolutionary methods in an attempt to solve some of our own optimization problems[7]. In order to introduce the concept of GAs, Fig. 1 provides an informal example of a simple GA

To use a genetic algorithm, we must represent a solution to or problem as a genome (or chromosome). The GA then creates a population of solutions and applies genetic operators such as Crossover and Mutation to evolve the solutions in order to find the best one(s). A GAs can be used to find a solution in much less time. Although it probably will not find the best solution, it can find a near perfect solution in less than a minute. The three most important aspects of using GAs are i) definition of the objective function. ii) definition and implementation of the genetic representation iii) definition and implementation of the genetic operator. Once these three have been defined, the generic genetic algorithm should work fairly well. As GAs works on a population, or a collection of several alterative solutions to the given problem.

The GA consists of four main stages: evaluation, selection, crossover and mutation.

The evaluation procedure (fitness function) measures the fitness of each individual solution in the population and assigns it a relative value based on the defining optimization (or search criteria). Typically in a non-linear programming scenario, this measure will reflect the objective value of the given model. It is user-defined and problem-specific. The selection procedure randomly selects individuals of the current population (pair of chromosomes) for the development of next generation. It ensures survival of fittest.

The crossover and mutation are two basic operators of GA that are commonly used to generate offspring in the next generation. Performance of operators depends on the encoding and also on the problem. There are many ways

1. Randomly generate an initial population.
2. Calculate fitness for each individual in the population.
3. Probalistically selects surviving members of the current population based on their relative fitness's.
4. Generate the next generation by applying genetic operators (Crossover and Mutation) to the surviving individuals.
5. Repeat steps 2 through step 4 indefinitely or until a halting condition is met (for example the desired maximum number of generations is exceeded.

Fig. 1: A basic genetic algorithm in high level pseudo code

how to perform crossover and mutation. The crossover procedure takes two selected individuals and combines them about a crossover point thereby creating two new individuals. Simple (asexual) reproduction can also occur which replicates a single individual into the new population. It is used for reproduction.

The mutation procedure randomly modifies the genes of an individual subject to a small mutation factor, introducing further randomness into the population. Sometimes a mutation can lead to a better solution that a crossover would not have found. Due to the complexity of the encoding method, no mutation is done in our method.

This iterative process continues until one of the possible termination criteria is met: if a known optimal or acceptable solution level is attained; or if a maximum number of generations have been performed; or if a given number of generations without fitness improvement occur. Generally, the last of these criteria applies as convergence slows to the optimal solution.

RESULTS

In order to evaluate its performance, we have implemented the algorithm and applied it to solve problems that were randomly generated. The results of these experiments are reported below. In all the experiments, the implementation was conducted in C. and all the experiments were run on a Personal Computer (PC) with a AMD 2400+ 266 MHz CPU and 256 MB RAM.

The performance of the algorithm was often evaluated based on the speed of improving the fitness of its solution and based on its complexity. To verify the performance of our algorithm, we performed some tests on networks of different sizes ranging from 5 cells and 3 switches to 100 cells and 7 switches. Each test was performed 5 times and we report the average costs.

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 HB heuristic proposed by

Table 1: Comparative results for GA, SA and HB (Variable numbers of Switches)

# of cells	# of switches	GA	SA	HB
5	3	47	54	58
15	2	124	136	142
30	3	357	407	498
50	4	558	651	667
100	7	1145	1530	1320

Table 2: Comparative results for GA, SA and HB (Fixed numbers of Switches)

# of cells	# of switches	GA	SA	HB
5	3	47	54	58
15	3	203	245	227
30	3	357	407	498
50	3	524	576	612
100	3	1390	1472	1206

Beaubrun *et al.*[8] 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[9,10] and we use the same sets of data to achieve the comparison. The results are reported in Tables 1 and 2. These represent the costs of different networks and all reported solutions are feasible.

The results of this comparative survey, our GA provides better results than the other methods for moderate and large-sized networks. 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.*[8].

RESEARCH STATUS

The research we propose is the problem is modeled as a complex integer programming problem and it is an NP-hard problem. Well-known in the literature as an NP-hard combinatorial optimization problem, this problem requires recourse to heuristic methods in order to good (not necessarily optimal) solutions within a practical amount of time. We hope that this study will serve as a starting point for further research and development.

CONCLUSION

In this study, we have investigated a multi-population GA to solve of assigning cells to switches in cellular mobile networks. Experiments have been conducted to measure the quality of solutions provided by this algorithm. The result obtained confirms the efficiency and the effectiveness of GA to provide good solutions for large-sized cellular mobile networks. This heuristic can be used to solve NP-hard problems, like designing and planning, in the next-generation mobile networks. A version of the algorithm taking in consideration the double homing problem could also be designed. Further, new techniques are also under investigation.

REFERENCES

1. Er-Rong Din, D. and S.S. Tseng, 1999. A solution model of integrating cells of PCS to switches in wireless ATM network. Technical Report, TR-WATM-9903, Department of computer and information Science, Taiwan, R.O.C.
2. Robertazzi, T.G., 1999. Planning Telecommunication Networks: State University of New York at Stony Brook: IEEE, Inc., (USA).
3. Merchant, A. and B. Sengupta, 1995. Assignment of Cells to Switches in PCS Networks. IEEE/ACM Transactions on Networking, 3: 521-526.
4. Sakthivel, S. and R. Suresh, 2004. A Genetic Algorithm approach to solve Mobile base station location problem. UGC sponsored National conference on Networking and Multi-agent systems, pp: 24.
5. Davis, L., (Ed)., 1991. Handbook of Genetic algorithms: Van Nostrand Reinhold, New York, NY, U.S.A.
6. Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and machine learning: Reading MA, USA: Addison Wesley.
7. Mitchell, M., 1999. An Introduction to Genetic Algorithms: Cambridge USA, London UK: MIT Press.
8. Beaubrun, R., S. Pierre and J. Conan, 1999. An efficient method for optimizing the assignment of the Cells to MSCs in PCS Networks. Proceedings of the 11th International conference on Wireless Comm. wireless 99, Calgary(AB), 1: 259-265.
9. Pierre, S. and F. Hoeuto, 2002. A tabu search approach for assigning cells to switches in cellular mobile networks. Computer Communication, 25: 464-467.
10. Quintero, A. and S. Pierre, 2002. A Memetic algorithm for assigning cells to switches in cellular mobile networks. IEEE Communications letters, 6: 11.