ADAPTIVE AREA LOCATION PLANNING AND OVERALL NETWORK PLANNING FOR MULTI-CONSTRAINED UMTS NETWORKS

Rahul Bhattacharya[(1)], **P. Venkateswaran**[(2)], **S.K. Sanyal** [(3)], **R. Nandi** [(4)]

[(1)]*Department of Electronics and Tele-Communication Engineering*
Jadavpur University, Kolkata-700032, India.
E-mail: rita_etce_ju@yahoo.com

[(2)]*As (1) above, but E-mail:pvwn@yahoo.co.in*

[(3)]*As (1) above, but E-mail:s_sanyal@ieee.org*

[(4)]*As (1) above, but E-mail:robnon@hotmail.com*

ABSTRACT

This paper proposes a novel constraint-based 3G network-planning model and a novel hybrid approach for optimizing the 3G network-planning problem. Computational results show that the model and the approach are more efficient than the mathematical model and the existing heuristics. Optimal solutions are always obtained for small and medium sized problems. The proposed hybrid approach can be an efficient tool for tackling a wide range of combinatorial NP-hard problems.

I. INTRODUCTION

Effective planning of 3G networks, e.g. UMTS shown in Fig. 1, has a significant impact on costs and QoS [1]. Planning a 3G UMTS network means that we need to determine the number and types of RNCs required, to determine a suitable placement for the RNCs, and to design a homing plan that specifies which node Bs are to be controlled by which RNCs, which defines the RNC borders. Following studies are relevant in a narrower sense. In [2] the cell assignment problem is studied. In [3] the same cell assignment problem has been approached by using tabu search. The novelties of this paper come not only from selecting RNCs and homing node Bs simultaneously considering the handover impact and traffic flow from node B to RNC, but also from the more efficient constraint-based formulation and the novel constraint-based hybrid approach.

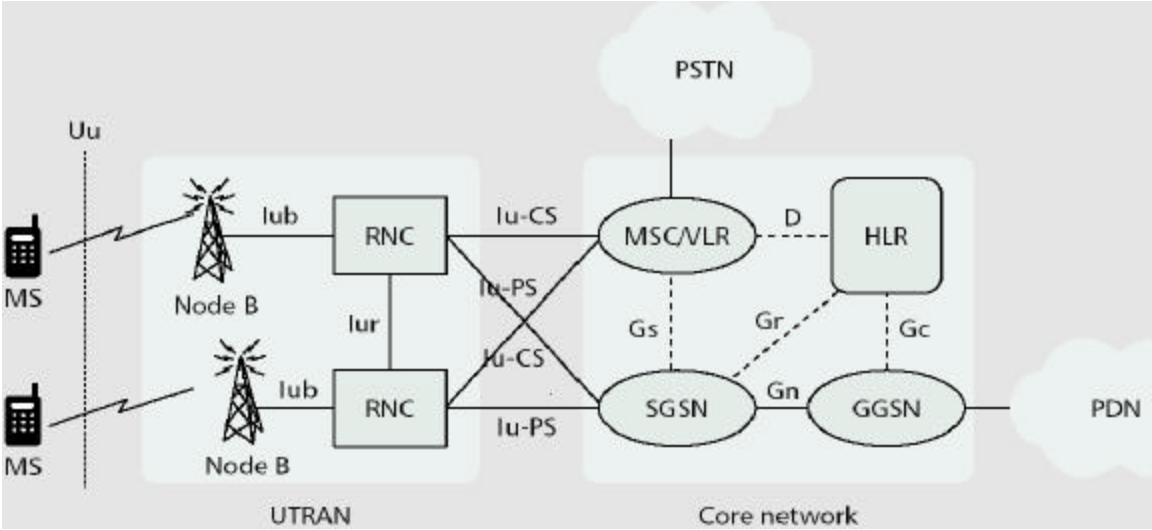

Fig 1: UMTS Architecture

II. CONSTRAINT-BASED MODEL

We create several matrices of boolean variables for the proposed constraint-based model. The constrained variable matrix *connect* is used to store the number of the RNC to which each LA is assigned. The value of a variable from the matrix *connect* indicates the RNC that serves the LA corresponding to that element. The boolean matrix *handover* is used to store the handover cost between cells. The boolean matrix ***setup*** represents the monthly amortization cost for each open RNC. Let *i* be the number of node Bs to be assigned to *m* potential locations of RNCs. Let the intermediate variable C_{ij} in (1) takes the value 1 if node Bs *i* and *j* are both connected to the same RNC and the value 0 if node Bs *i* and *j* are connected to different RNCs.

$$C_{ij} = (\text{connect}[i] = \text{connect}[j]), \quad \text{for, } i=1,2,\ldots.n \ \& \ j=1,2,\ldots.n \tag{1}$$

The logical operator = is used to compare the values of **connect[i]** and **connect[j]**. Let $\mathbf{C_{ij}}$ take 1 if both values are equal and 0 otherwise. The constraint-based model uses variable matrix ***select[j]*** to specify whether an RNC is selected at site ***j***. An important feature here is to use variables or expressions involving variables to index arrays [4]. Thus constraints describing that an RNC must be selected if it serves the cell ***i*** can be expressed as follows:

$$\text{select}(\text{connect}[j]) = 1, \ \text{for } i=1,2,\ldots.n \tag{2}$$

If $\mathbf{\ddot{e}_i}$ denotes the traffic volume per time unit destined to cell ***i***, the limited traffic-handling capacity $æ_k$ and the total number of ports $ç_k$ of $\boldsymbol{k}^{th}$ RNC impose the following constraints (3) and (4) for :

$$\sum_{i}^{n} \boldsymbol{l}i(connect[i] = k) \le \boldsymbol{x}select[k] \tag{3}$$

According to constraints (3), the total load of all cells, which are assigned to $\boldsymbol{k}^{th}$ RNC is less than the traffic handling capacity of the RNC $\mathbf{æ_k}$. The constraints (4) say that the total number of active connections to $\boldsymbol{k}^{th}$ RNC cannot exceed its total number of ports $ç_k$:

$$\sum_{i}^{n} (connect[i] = k) \le \boldsymbol{hk}select[k] \tag{4}$$

Constraint (5) establishes a relationship between the objective function (**Ã**) and other constrained variables. Because our purpose is to minimize the objective function, each modification of this objective function propagates its effects to the other variables, and any modification of a constrained variable propagates to the objective function as well. These two essential processes eliminate parts of the search space that will not produce better solutions [5].

$$\mathbf{\tilde{A}} = \sum select[k] * setup[k] + \sum\sum(1 - Cij) * handover[i, j] \tag{5}$$

III. SOLUTION TECHNIQUES

Our aim is to develop algorithms that take the available constraints, capacity and load information described in Section II as inputs, and find an optimal or near optimal network topology which includes the assignment of cells (Node Bs) to switches (RNCs and MSCs).

III.1 Remarks on the Complexity of the Problem

Since the solution space consists of all possible network topologies, the Node B-to-RNC, RNC-to-MSC and Node B-to-LA assignments determine the complexity of the solution. If there are *n* Node B's, *m* RNCs, *p* MSCs, the number of possible Node B-RNC assignments is calculated as follows. For each Node B, there are *m* possible RNCs to be connected, hence n^m possible connections. Likewise, the number of possible RNC-to- MSC assignments is m^p and because the upper limit of the number of LAs is the number of B's, the number of possible B-to-LA assignments is n^n. Consequently, the size of the solution space is $n^m\ m^p\ n^n$. Although the solution space includes both feasible and infeasible solutions, all solutions must be assessed

whether they are feasible or not. Because of the solution space size, exhaustive methods result in exponential time complexity.

III.2 Simulated Annealing (SA)

Due to the difficulty of the problem described in Section II, it is not possible to guarantee the optimal solution in reasonable running times. Therefore, techniques that give near optimal solutions within acceptable run times are needed. One method that finds a sub-optimal solution without searching the entire solution space is simulated annealing (SA) [7], which approximates the solution of very large combinatorial optimization problems[8].

The SA-based algorithm described here finds a network topology that consists of the LA-Node B assignments, the RNC-MSC topology and the Node B-RNC topology. For a successful SA implementation, two key items that must be defined carefully are the moves that create neighbor solutions and the cooling schedule. A feasible solution is a topology where all network nodes (Node B's, RNCs, MSCs) are connected and the LA borders are specified while satisfying the constraints. Therefore, a (feasible) neighbor solution may be generated by any of the three types of moves:

1) *Changing a Node B-to-RNC assignment*: If constraints are not violated, a new RNC is assigned to a randomly chosen Node B. Then, a random feasible LA is assigned to that BS. If no feasible LA is found, a new LA is created.

2) *Changing a RNC-to-MSC assignment*: First, a RNC and a candidate MSC for new assignment is chosen randomly. If the constraints are not violated, the LAs residing merely in that RNC (i.e., none of their Node B's are connected to another RNC) stay with their Node B connections (i.e., nothing is changed for these LAs), but LAs having some of their Node B's connected to that RNC go through rearrangement. The assignments of Node B's to those LAs are released, and therefore those LAs have no more Node B's assigned to them from that RNC. For those "released" Node B's, an adequate number of new LAs are created (if the capacity of a new LA is full, then another LA is created).

3) *Changing a Node B-to-LA assignment*: Without affecting the Node B-RNC connection, a new LA assignment is done by searching the available LAs residing within the same RNC.

III.3 Proposed Heuristics Algorithm

Since heuristics try to find the best solution without tracing an extensive part of the solution space and instead attempt to build the best solution directly, their time complexity is generally much better than other techniques. However, heuristic solutions tend to offer inferior quality. We can improve them by setting the best objective function (**Ã**) value obtained in a local search algorithm as the upper bound. In this case, the local search algorithm provides a method for pruning the search space used by a follow-up standard constraint optimization search procedure, which is used to improve the solution. When the search procedure concludes until no feasible point is found, the last feasible point can be taken as the optimal solution. If this does not happen for a period of time (or other criteria), an intermediate solution can be obtained as an improvement by interrupting the search process.

IV. EXPERIMENTAL RESULTS

Experiments are carried out on 5 sets of data, each containing 30 test samples, with a number of cells varying between 30 and 300 and a number of potential sites varying between 20 and 80. These five test cases are generated by a *MATLAB* program by supposing that the cells are arranged on a hexagonal grid of almost equal length and width. The locations of node Bs and the potential locations of RNCs are generated randomly with a uniform distribution. It is assumed that the RNCs have a single cabinet supporting up to 180 node Bs, 310 Mps of raw data throughput. Each node B is assumed to have an *Iub* interface bandwidth [1] of 1.7 Mb/s. The cluster system named ASMA (Advanced system for multi-computer applications) is employed in the experiments, as linear programming optimizer.

Each test case is run 30 times for 10000 solution evaluations for each sample test run. To evaluate the performance of the hybrid approach, we first compare its solutions with those obtained by simulated annealing and greedy algorithm. The initial solutions for these local search algorithms are generated randomly with a uniform distribution and without a post optimization process, while the hybrid approach uses the constraint satisfaction method to obtain an initial feasible solution as the input for the immediate tabu search, followed by a post optimization procedure. Fig. 2 illustrates the average results over the sample test runs on all five test cases, and finds that the hybrid approach consistently offers the best performance.

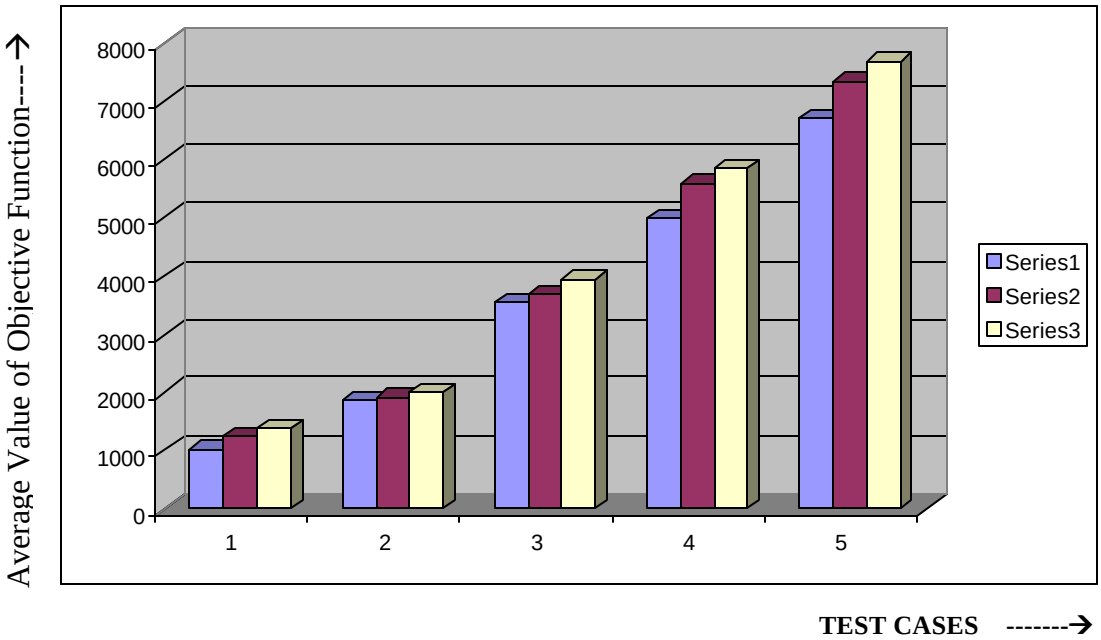

Fig 2: Comparison of Average value of Objective Function for 5 Test Cases, using Simulated Annealing (pink bars) and greedy algorithm(yellow bars) compared to proposed one (blue bars)

REFERENCES

[1] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, 2nd ed. New York:Wiley, 2002.

[2] A. Merchant and B. Sengupta, "Assignment of cells to switches in PCS networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 521–526, Oct. 1995.

[3] S. Pierre and F. Houeto, "A tabu search approach for assigning cells to switches in cellular mobile networks," *Computer Commun.*, vol. 25, pp. 464–477, 2002.

[4] P. Van Hentenryck, "Search and strategies in OPL," *ACM Trans. Computational Logic*, vol. 1, no. 2, pp. 282–315, Oct. 2000.

[5] M. Wallace, *Survey: Practical Applications of Constraint Programming*. London, U.K.:William Penney Lab., Imperial College London, Sept. 1995.

[6] E. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*. Chichester, U.K.: Wiley, 1997.

[7] G. Pesant and M. Gendreau, "A view of local search in constraint programming," *Constraint Programming*, pp. 353–366, 1997.

[8] E. H. Aarts, J. Korst, and P. J. van Laarhoven, *Simulated Annealing, Local Search in Combinatorial Optimization*, E. Aarts and J. K. Lenstra, Eds. New York: Wiley, 1997, [9] E. Rich and R. Knight, *Artificial Intelligence*. New York: McGraw-Hill, 1991.