CHAPTER 3
COST OPTIMIZATION IN NON-HIERARCHICAL MOBILE ACCESS NETWORK

3.1 INTRODUCTION

This chapter considers non–hierarchical mobile access networks. The first aim of this chapter is to propose and analyse a cost optimization algorithm for non-hierarchical mobile access network. Second is to compare the proposed algorithm with the existing algorithms. Third is to identify some issues in non-hierarchical mobile access networks. In order to achieve above goals, a mathematical modelling of non-hierarchical mobile access network using mixed integer programming along with technical constraints like capacity and connectivity has been proposed.

An Iterative Local Search (ILS) heuristic algorithm using 2-opt local search technique is proposed as solution for the cost optimization problem in non-hierarchical mobile access network modelled. It also presents experimentation setup for analyzing proposed ILS heuristic algorithm for various network size of mobile access network. For various network instances, the performance of ILS heuristic algorithm is verified for optimal cost and computation time, and it is established that ILS heuristic algorithm outperforms on both counts. A comparative analysis of ILS algorithm with other existing algorithms like simplex algorithm, genetic algorithm and simulated annealing has been carried out and it shows ILS algorithm is well suited to find optimal cost of mobile access network. This chapter starts with mathematical modelling and then proceeds to the performance evaluation of non-hierarchical mobile access network.
3.2 PERFORMANCE EVALUATION OF COST OPTIMIZATION OF NON-HIERARCHICAL MOBILE ACCESS NETWORK

This section presents cost optimization in non-hierarchical mobile access network using mathematical modelling and ILS heuristic algorithm. Next, this section considers performance evaluation based on computation complexity for cost optimization. Then, comparative analysis of ILS heuristic algorithm with CPLEX solver, genetic algorithm and simulated annealing is performed.

3.2.1 MATHEMATICAL MODELING OF COST OPTIMIZATION IN NON-HIERARCHICAL MOBILE ACCESS NETWORK

Previous chapter has discussed architecture of non-hierarchical mobile access network in detail. In short, two characteristics of non-hierarchical mobile access network are (i) non-hierarchical and (ii) single homed. First, non-hierarchical characteristic indicates that, the mobile access network is a basic network with only two network components namely cells and switches [3,5,6,12], there is no further hierarchy of nodes within the network. Unlike advanced mobile access network, there is no concrete distinction between RNC and MSC switches in this basic network. Secondly, single homed characteristic indicates that, the cells are single homed, in the sense, only one network interface is available for cells to connect to exactly one switch. Next, the mathematical formulation of cost optimization is presented.

In order to formulate mathematical modelling for cost optimization problem in non-hierarchical access network, the objective function is defined as optimal assignment of cells to switches, while satisfying technical constraints. Here, the mathematical model has to consider two technical constraints namely (i) connectivity constraint and (ii) capacity constraint. First, the connectivity constraint exhibits the single homed characteristic of each cell, in which each cell connects to exactly one switch. Secondly, the capacity constraint exhibits limit for the allocation of number of cells to each switch. Table 3.1 represents various symbols and their meaning as
used in mathematical formulation of cost optimization in non-hierarchical mobile access network.

**Table 3.1: Symbols and its meaning**

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Total Number of Cells</td>
</tr>
<tr>
<td>$J$</td>
<td>Total Number of Switches</td>
</tr>
<tr>
<td>$\alpha_i^j$</td>
<td>Cable cost of existing link between cells Cell$_i$ and switch Switch$_j$</td>
</tr>
<tr>
<td>$\beta_i^j$</td>
<td>1 if Cell$_i$ and Switch$_j$ are connected; otherwise zero</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>Call servicing capacity of single homed Cell$_i$</td>
</tr>
<tr>
<td>$\omega_j$</td>
<td>Call handling capacity of Switch$_j$</td>
</tr>
</tbody>
</table>

Thus the objective function for basic network connectivity at cell level becomes,

Minimize $\delta = \Sigma_{i=1}^I \Sigma_{j=1}^J \alpha_i^j \beta_i^j$ \hspace{1cm} (3.1)

Subject to:

$\Sigma_{i=1}^I \beta_i^j = 1$ \hspace{1cm} for $j = 1,2,\ldots,J$ \hspace{1cm} (3.2)

$\Sigma_{i=1}^I \lambda_i^j \beta_i^j \leq \omega_j$ \hspace{1cm} for $j = 1,2,\ldots,J$ \hspace{1cm} (3.3)

Equation (3.2) exhibits connectivity constraint with single homing. The single homed cell indicates that each cell connects to one and only one switch, that is, cell $i$ is linked to only one switch $j$ at any time. Equation (3.3) exhibits capacity constraint with single homing. Here, the number of connectivity requests from each cells for a particular switch, must be less than or equal to the remaining call handling capacity $\omega_j$ currently available at switch $j$.

This section focuses on mathematical modeling of cost optimization problem in non-hierarchical mobile access network. Next section discusses ILS heuristic algorithm for solving this cost optimization problem.
3.2.2 ITERATIVE LOCAL SEARCH (ILS) HEURISTIC ALGORITHM

In [9], authors propose genetic algorithm (GA) for optimal switch location in mobile communication network. The GA utilizes a standard binary individual representation. This is not very suitable for cost optimization problem. It requires large amount of memory and computation time.

In [6], authors propose simulated annealing (SA) incorporating pricing mechanism for assigning cells to switches in cellular networks. The major issue with price influence simulated annealing is that, it may not generate the best solution in any one run. Here, end of iteration stores the dual price information implicitly from past solutions. This complex procedure has no guarantee to provide good solutions over a series of iterations.

Our goal in this work is to advance a simple and efficient heuristic algorithm for solving the complex mechanism of mobile access network designing. One such heuristic algorithm is Iterative local search algorithm (ILS). In [26], author proposes ILS a simple and powerful metaheuristics among all other heuristic algorithms [6, 8, 9, 11]. This section elaborates operation of ILS heuristic algorithm. Next section of compares the performance of ILS heuristic algorithm with genetic algorithm and simulated annealing.

The proposed ILS algorithm uses efficient local search and avoids struck into local optimal conditions by using effective perturbation operator. Pseudo code for ILS is given in Figure 3.1. ILS algorithm begins with general initial solution P obtained by using random distribution. Next, performs a local search on general initial solution P and produces new local optimal solution P*. Local Search improves the solution in the search space. When the local search is stuck, the locally optimal solution P* is mutated by a move in a neighborhood different from the one already used by the local search. Perturbation kicks current local solution P* and generate an intermediate solution P’. The resulting solution of perturbation P’ is the new starting solution for the local search that takes it again to another new local optimum P*’. Using proximate optimality principle, if P’ is close to P* in Perturbation, then the new local optimal solution generated in local search will also be in proximity of the current local optimal solution.
procedure IterativeLocalSearch

P= Preliminary Solution

P*= LocalSearch(P)

Pbest = P*

until (termination condition not met)

P’ = Pertubation(P*)

P*’=LocalSearch(P’)

if((f(P*)< f(Pbest))then

Pbest= P*’

end-if

P* = AcceptanceCriterion(P*,P*’)

repeat

return Pbest

end-procedure

Figure 3.1: Pseudo code for local search
Finally an acceptance criterion decides which of the two locally optimal solutions, P*' and P*, obtained from previous step is to be selected as a starting point for the next perturbation step. If P*' wins over P* in acceptance criterion, then P*' will replace P* and become current optimal solution, otherwise the search remains at the previous optimal solution P*. The important technique in ILS algorithm lays in carrying out a randomized walk within the search space of the local optima with respect to some local search algorithm. Next section discusses one such local search technique called 2-opt local search [26].

**ILS Algorithm with 2-opt Local Search Technique**

Conventional simplex methods and search techniques provide optimal solution for optimization problems. However, these methods have many disadvantages as discussed further. First disadvantage of these methods is that, they operate by searching through the entire search space of the problem set. Second, these methods will not stop search operation immediately once the optimal solution is found. Rather, they continue searching until end of search space. Third, this uncontrolled mechanism of search operation consumes lot of unnecessary memory space and huge computation time. Fourth, these methods are best suited for small to moderate size problems. In case of large size problems where search space is huge, these methods fail due to computational complexity.

Targeting above disadvantages, the main purpose of local search technique is to overcome increasing complexity of optimality computation in entire search space. Reducing computational complexity involves two best techniques. First, to limit number of neighbors that are needed to be inspected, around the local optimal solution of the given problem. Second, while searching for the better solution, local search should be faster enough to update local optimum, for global search operation. For this purpose, as suggested in [26], 2-opt ILS algorithm is used in this work.

In case of 2-opt local search, an initial position is selected at random. Then, the algorithm begins ILS process from that position by exchanging 2-opt constant length portion resulting in generation of new neighbors. All possible 2-opt exchanges in the local optimal solution are investigated for a better solution than previous. To understand better, an example for assignment problem using ILS algorithm with 2-opt local search is further discussed here.
Consider a cellular mobile network of 15 cells as shown in Figure 3.2. Let cell 5 and cell 8 act as switches. It should be noted that each cell connects to its adjacent cells. Let each connecting link be represented by a unit cost link, in real life situation, the link cost is considered directly proportional to Euclidean distance between the cells.

From Figure 3.2, the network connection depicts a grid network with vertices as cells and edges as connecting links. It should be noticed that, in the given example, all the cells are connected to all other cells using link cost equal to distance between two cells. Therefore, the flow matrix is a singular matrix representing that connectivity exists between each cell to all switches. The distance matrix is considered as the sum of shortest unit link costs between cells and switches. For this example, the optimal assignment of cells to switches using Equation (3.1) is achieved at objective cost of 18. Now, the goal of proposed algorithm is to achieve objective cost in proximity to the optimal solution of 18.
Table 3.2: Sample random initial objective value and local search objective value

<table>
<thead>
<tr>
<th>Iterations</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Initial Objective Value</td>
<td>29</td>
<td>25</td>
<td>33</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>Local Search Objective Value</td>
<td>21</td>
<td>18</td>
<td>25</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

For five iterations, Table 3.2 depicts the sample random initial objective value with $P_{\text{best}}$ as 25. In actual, the numbers of iterations will be multiples of hundreds. The objective value obtained during each of the iteration is called neighborhood solution. Then, the local search algorithm is applied to these neighborhood solutions. In case of 2-opt local search, an initial position is selected at random, begins ILS process from that position by exchanging 2-opt constant length portion resulting in generation of new neighbors.

Figure 3.3(a) depicts the initial best random assignment and its initial best objective value ($P^*$) as 25. The duple above the edges indicates flow ($\beta_i^j$) and distance ($\alpha_i^j$) value between cell i and cell j as given in Table 3.1. The dashed lines indicate the actual path of communication between two cells in case the communicating cells are connected to different switches.

On this initial solution, 2-opt local search is applied. In Figure 3.3(b), a random 2-opt assignment exchange is performed between cell 9 and cell 7, which results in the objective value of 22. Further, by exchanging cell 13 and cell 12, the objective value is 18.

During each 2-opt exchange, the local minimum cost assignment is considered. Thus 2-opt ILS algorithm produces objective value ($P^*$') as 18 and the ILS best assignment of cells to switches is reflected in Figure 3.3(c). Now the acceptance criterion is applied on $P^*$=25 and $P^*$'=18 and return $P_{\text{best}}$ as 18. Further, the procedure continues in same way until termination condition is met.
Figure 3.3: 2-Opt Local Search
3.2.3 PERFORMANCE ANALYSIS OF ILS HEURISTIC ALGORITHM

Now, let us analyze how ILS algorithm overcomes the disadvantages of conventional methods like simplex algorithms and search algorithms. First disadvantage of conventional methods is enumeration based search operation. The algorithm performs searching within entire search space of the problem set for optimality computation. That is, every enumerated combination of assignment is computed for objective function and then output is obtained at the end. Whereas, ILS algorithm continues search operation towards optimal solution until it finds first local optimum.

Once the first local optimum is found, the search technique is performed in controlled manner using 2-opt local search technique. The 2-opt local search technique strictly performs neighborhood search around its local optimum. This way, ILS algorithm avoids unnecessary search of entire search space in the given problem.

Second disadvantage of conventional methods, is that, even if the algorithm finds the optimal solution at very early stage, it will not stop its search operation immediately. Rather, it continues searching until the end of search space of problem. Whereas, ILS heuristics algorithm controls termination of search process by using two operations namely the acceptance criterion and the number of iterations.

The acceptance criterion updates the best solution at the end of every optimization step. It also ensures that, the algorithm always produces the best feasible solution as output. Next, the algorithm sets termination limit using number of iterations for performing local search operation. This way, ILS algorithm stops search operation in controlled manner.

Third disadvantage of conventional methods, is that, they consume lot of unnecessary memory space and require large computation time. Even in case, the algorithm obtains optimal solution at very early stage, the optimality computation is continued on each enumerated combination of search space. This extra computation consumes extra memory space. Further, evaluating objective function for non-optimal variables invests extra computation time. Whereas, ILS
algorithm incorporates search control mechanisms like acceptance criterion and number of iterations. This way, ILS algorithm avoids consumes less memory space and computation time.

Fourth, conventional methods are best suited for small to moderate sized problems. In case of large sized problems where search space is huge, these methods fail due to computational complexity. However, the ILS algorithm performs well in large sized optimization problems using effective local search mechanism.

In this section, the performance analysis of ILS heuristic algorithm has been discussed. In next section, based on experimentation, comparative analysis of ILS heuristic algorithm with other algorithms is presented.

3.3 EXPERIMENTAL ANALYSIS OF ILS HEURISTIC ALGORITHM WITH OTHER ALGORITHMS

Even though conventional methods like simplex algorithm and search algorithms generate optimal solution, they perform well only for small sized problems as discussed above. Therefore, it is necessary to show that, how proposed ILS heuristic algorithm overcomes those issues highlighted in conventional methods.

In this aspect, this section discusses empirical analysis of conventional methods and proposed ILS heuristic algorithm. The analysis is conducted in two sections. First, comparative analysis of ILS with Integer programming methods is carried out. Next, comparative analysis of ILS with genetic algorithm and simulated annealing is presented.

3.3.1 CONVENTIONAL INTEGER PROGRAMMING VERSUS HEURISTIC ALGORITHMS

Genuine Intel® CPU T2250 @ 1.73GHz, 1.00 GB of RAM machine is used for experimentation. CPLEX solver is used for evaluating integer programming. The C++ language is used for ILS algorithm. Here, ILS algorithm is compared with conventional Integer Programming method. Further, cells are equally distributed on the grid and switches are uniformly distributed at the center of cells.
Distance matrix and flow matrix are two input matrices to the algorithm. The cable cost of link between cells and switches are considered as proportional to the geometric distance between cells and switches. In binary flow matrix, the value one indicates when there exists a link between cell and switch. The value zero indicates non-existence of link between cell and switch.

The two sets of test instances are generated for analysis. First set is based on constant number of switches and varying number of cells. Second set is based on varying number of cells and varying number of switches. For test cases, the number of switches are varied between 2 and 5 and number of cells are varied between 15 and 240. Each instance was run, for 100 iterations.

Initially when comparing conventional Iterative Local Search algorithm with Integer Programming methods, both produced optimal results for number of cells within 30. For both cases of varying number of switches as well as constant number of switches, Integer Programming method and conventional Iterative Local Search algorithm showed linear increase in the computational time.

Beyond 30 cells, the integer programming method failed to produce any feasible results as depicted in Table 3.3. In Integer programming method, the total computational data set exceeded the memory available in case the numbers of cells were more than 30. Thus the computation time became too large and infeasible to obtain optimal solution, indicated as asterisk (*) in Table 3.3.
<table>
<thead>
<tr>
<th>No. of Cells</th>
<th>No. of switches</th>
<th>% Feasible solution</th>
<th>%CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IP</td>
<td>ILS</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>*</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>*</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>*</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>*</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>*</td>
<td>100</td>
</tr>
</tbody>
</table>

*indicates infeasible.
3.3.2 ILS VERSES GA AND SIMULATED ANNEALING

Genuine Intel® CPU T2250 @ 1.73GHz, 1.00 GB of RAM machine is used for experimentation. Here, cells and switches are uniformly distributed on the grid. Two matrixes namely distance matrix and flow matrix are input to the algorithm. For distance matrix, geometric distance between cells and switches is considered. flow matrix has binary entries with value one when there exists a link between the cell and the switch. If there no link between the cell and the switch then matrix entries are made zero.

The two sets of test instances are considered for experimentation. First set is based on constant number of switches and varying number of cells. Second set is based on varying number of cells and varying number of switches. For test cases, the number of switches varied between 2 and 7 and number of cells varied between 15 and 200. We ran each instance, for 100 iterations.

When comparing ILS with Genetic Algorithm (GA) and Simulated Annealing (SA), all three heuristic algorithms always produced objective cost values in proximity of the optimal solution. Whereas, ILS showed better results than the other two heuristics GA and SA by producing improved objective cost function. After executing 2-opt ILS Algorithm in both cases, namely, with constant number of switches and with varying number of switches, ILS outperformed GA and SA in terms of computation time.

The improvement factor in case ILS verses GA was around 1% as given in Figure 3.4(a) and ILS verses SA around 2% as given in Figure 3.4(b). In practice, even this small improvement in the objective cost function would reflect a huge reduction in overall cost of the assignment problem in cellular Mobile networks.
Figure 3.4: Objective Cost comparison

(a) Varying Number of switches

(b) Constant Number of switches

Figure 3.4: Objective Cost comparison
3.4 ISSUES IN COST OPTIMIZATION OF NON-HIERARCHICAL MOBILE ACCESS NETWORK

Previous sections highlight the importance of cost optimization in mobile access network. However, there are some issues in cost optimization of non-hierarchical mobile access network. This section brings out, some of the issues in cost optimization in non-hierarchical mobile access network. First, this section discusses, various issues in non-hierarchical mobile access network, and then later discusses various issues in cost optimization of non-hierarchical mobile access network.

Issues in non-hierarchical mobile access network

- **No concrete levels at access layer:** As discussed in [2,3], earlier mobile network and till 2G mobile networks are of non-hierarchical architecture. Here, there are no further distinguishable levels of access network. However, in 2.5G and above generation of mobile networks, the access network has multiple levels like NodeB level, RNC level and MSC level. Literature review section has already discussed the detail of these levels.

- **Less number of network components has been involved:** It is to be noted that, access layer of non-hierarchical network has no concrete distinction in switches [2]. Hence, very few network components are considered. These components are cells, switches, and transmission links interconnecting these components. However, in modern age, advanced mobile networks involve several other components like NodeB, RNC, MSC, transmission links. Therefore, while designing mobile networks, these components are also to be considered.

Issues in cost optimization in non-hierarchical mobile access network

- **Cost optimization on limited number of network components:** The major objective of earlier non-hierarchical mobile networks is only voice communication. In addition, the cost optimization of such non-hierarchical network involves few numbers of components. However, the modern era mobile network involve several objectives like voice communication, data communication and multimedia communication. These advancements
involve several components in mobile access networks. Therefore, while designing such advanced mobile network for cost optimization has to consider several network components and technical constraints of network.

- **NP hard problem and need for better heuristic algorithm:** The cost optimization in mobile access network is NP hard. The heuristic algorithm proposed for non-hierarchical access network is not sufficient for complex advanced mobile access network.

### 3.5 CONCLUDING REMARKS

This chapter describes mathematical integer programming for cost optimization in non-hierarchical mobile access network. This involves technical constraints like capacity and connectivity. For solving cost optimization problem, an efficient ILS heuristic algorithm has been proposed.

Next, this chapter compares iterative local search heuristic algorithm with integer programming using experimentation. CPLEX solver is used for formulating integer programming. The experimentation considers various mobile access network scenarios depicting both smaller and larger network sizes. The experimental outcome shows that ILS algorithm outperforms CPLEX solver by computation time. The experimental results for larger network exhibits that, the ILS algorithm consumes computation time within few seconds, where as CPLEX solver fails to converge to final solution within limited time. Although, result for computation time consumed in smaller network shows that ILS algorithm performs almost same as CPLEX solver.

Further, this work compares ILS heuristic algorithm with Genetic algorithm and Simulated Annealing. The improvement factor in case ILS verses GA was around 1% and ILS verses SA around 2%. In practice, even this small improvement in the objective cost function would reflect a huge reduction in overall cost of the assignment problem in mobile access network. In summary, the cost optimization in non-hierarchical mobile access network using heuristic algorithm exhibits efficient performance.
Next, this chapter shows cost optimization problem is NP hard, by reducing into set partitioning problem. Finally, discusses, the issues involved in non-hierarchical mobile access network. In summary, this chapter discusses cost optimization in non-hierarchical mobile access network. However, in reality, mobile network has hierarchical topology with various levels. Next chapter discusses issues in cost optimization of hierarchical mobile access network.