Cost–based UMTS Transport Network Topology Optimisation

János Harmatos Alpár Jüttner Áron Szentesi

Traffic Lab
Ericsson Telecommunications Ltd.
Laborc u. 1.,Budapest, Hungary H-1036
{Janos.Harmatos, Alpar.Juttner, Aron.Szentesi}@eth.ericsson.se

April 1, 1999

Abstract

In this paper we propose a heuristic cost–based topological optimisation method for UMTS transport networks. Our method optimizes the number and location of Base Station Controllers jointly with the transmission network topology. The solution is based on the combination of a meta–heuristic called Simulated Annealing and a greedy algorithm. We test the algorithm on different inputs and analyse the optimality of the resulting network configurations. Results show that: a) our proposed algorithm provides solutions very close to the global optimum, b) the computational efficiency of the algorithm makes it usable for network planning tasks of practical sizes.

1 Introduction

The Universal Mobile Telecommunications System (UMTS) will differ from today's GSM networks in several ways. The main difference from the users' point of view is that the new system will incorporate a multitude of services, with special emphasis on data communications. The efficient transport of this new kind of traffic requires different technologies both for the air interface and the fixed transport: W–CDMA and ATM, respectively [2].

UMTS is going to play an important role in future telecommunications: penetration is expected to be very high. Thus, the transport network of UMTS (UTRAN) is going to be a rather complex and high–capacity system. These large and costly systems are going to be deployed in very short timeframes, which emphasizes the importance of a good network design methodology. In fact, the complexity of the system, the high cost factors and the shorter and shorter design timeframes call for the use of *algorithmic network topology optimisation methods.*

Optimisation has been used for network topology design for a long time; for example, it is widely applied in the planning process of GSM networks. However, UMTS differs from legacy networking technologies in many ways; the new transport technology puts differents demands and constraints toward the topology of the network. Therefore, new optimisation methods are needed to consider the new topological constraints.

The task we address here is twofold: first, we want to find the cost–optimal number and location of Base Station Controller nodes in the network. Second, we want to find a cost–optimal interconnection network connecting Base Transceiver Stations to the Base Station Controllers, taking into account specific topological constraints. Our primary goal is to perform the two optimisation steps jointly, thus obtaining a solution closer to the global optimum.

We formulate the task as a discrete optimisation problem. Considering the complexity of the problem, we are restricted to use some kind of quick heuristics here, because scalability is a primary issue. Typically, meta–heuristics can offer the required effectiveness. A technique called *simulated annealing* [4, 5] is widely used and provided good results in many different optimisation areas; we decided to use this method for our network design problem.

The rest of the paper is organised as follows: first, we shortly describe the architecture of UMTS, and the network model we use for our investigations. Following the exact problem definition, we propose a heuristic optimisation method for solving the problem. We conclude the paper with the performance analysis of the proposed algorithm.

2 The UMTS architecture

The terrestrial access part of the UMTS network consists of two types of network elements called BSC (*Base Station Controller*) and BTS (*Base Transciever Station*). The BSC manages the radio channels of BTSs connected to it, concentrates the connections and trunks them to the upper level core network. The BTS handles the radio channels and forwards the traffic of lower level BTSs towards their dedicated BSC, called home BSC. Figure 1 shows the hierarchical architecture of the UMTS terrestrial access network.

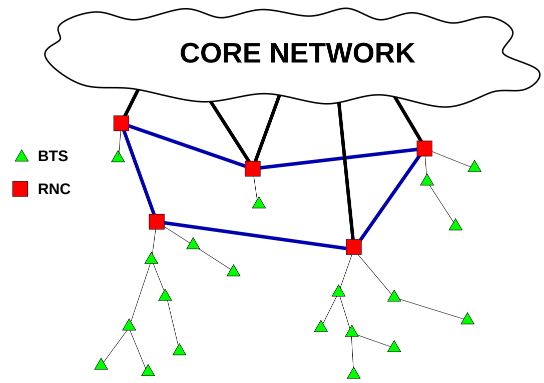

Figure 1: UMTS network

On logical level each BTS is connected directly to its home BSC using ATM PVCs (*Permanent Virtual Connections*), so the logical topology of the UMTS network is a set of star subnetworks. On physical level the BTSs are connected to their home BSC directly as well or through some other BTSs using leased line or microwave interconnections. It means that the physical level network topology is a set of trees.

Due to various technological reasons there is a constraint on how many BTSs can be connected to a BSC, on logical level further on this limit will be called *BSC degree*. On physical level the maximum number of BTS's connected paralel to an upper level BTS is limited, in this case we say that there is a degree constraint for each BTS. Also there is a constraint on the depth of the BTS subtrees, what means how many other BTS's can be placed between a BSC and a BTS on physical level, in other words how many BTS's can be cascaded in the network. This constraint is called *cascading constraint* further on.

3 Problem definition

A relevant and currently open design task of the UMTS terrestrial access network is to plan a cost optimal physical and logical topology the access network. This task covers the optimisation of the number and location of BSCs and the planning of the physical links connecting BTSs to their corresponding BSCs.

Based on some wave propagation model the radio network planning results in radio cells, so the number and locations of the BTSs and their generated traffic volumes are given as an input to the planning algorithm. Another input is the set of geographical sites which can be optional places of BSCs and the limitations determined by the equipment capabilities. It is also a basic requirement to handle the case if the location of some BSCs are predefined and during the optimization process they can not be moved. Our proposed algorithm suits for both of the above options, and it is also able to continue the design of an existing network.

4 Network model

In this section we will present the formulation of the design problem and the applied mathematical model. The model used is general for these types of optimization problems.

We get the following as input:

- A set S of sites. A site is a geographical point, where a BTS, BSC or both of them can be optionally placed . Let us denote the number of sites by n.
- A set B of BTS's. This is a subset of S.
- Two additional subset of S, the predefined and possible locations of BSC devices, denoted by R_{pre} and R_{pos}.
- The *cascading constraint*, *BTS degree constraint* and *BSC degree constraint* denoted by L, D_{BTS} and D_{BSC}.
- The cost of a BSC device denoted by $Cost_{BSC}$.
- The capacity independent part of the cost of the link between two sites denoted by c_{ij} for all pair i, j of sites. These are tipically in direct proportion to the distance of the sites but they can be arbitrary in our model.
- A *function* $c_{ij}^{cap} : \mathbb{R} \longrightarrow \mathbb{R}$ for all pair of sites which represents the capacity dependent part of the link cost. These are increasing and tipically step-like functions.
- The cost of a BSC device denoted by $Cost_{BSC}$.
- The amount t_i of traffic generated by a BTS for all station.
- A set of predefined links. It must satisfy the constraints on connecting.

With these notations our task is to put some BSCs to the possible places and to find a set of links which connect the BTSs to the BSCs and satisfy the constraints. And, naturally, our main goal is to minimize the total cost of the network.

In our model the total cost of the network consists of the cost of BSCs and the cost of the links. The cost of BSCs is in proportion with their number. The cost a link is the sum of its the capacity dependent and independent cost.

The cost of the BSC subsystem increases linearly with the number of BSCs, because each BSC has a similar, constant cost value. Meanwhile, the cost of BTS access links decreases with the number of BSCs, because the length of the physical links will be smaller when having more BSCs. Figure 2 illustrates a possible function of the two cost components and the their sum, the total network cost as the function of the number of BSCs.

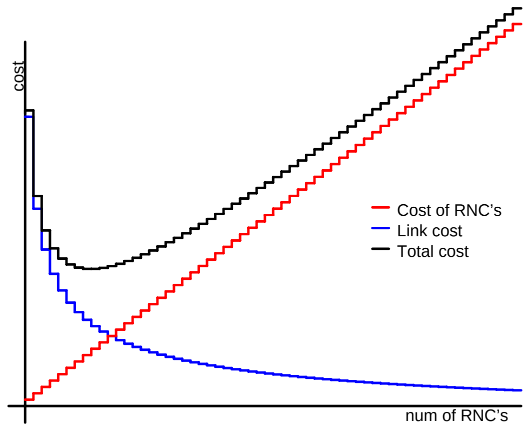

Figure 2: Cost depending on the number of BSC's

If we denote the number of BSC's by N_{BSC} and the set of the links by E then the total cost is the following.

$$\begin{aligned} Cost_{total} &= Cost_{BSC} N_{BSC} + \\ &+ \sum_{ij \in E} \left(c_{ij} + c_{ij}^{cap}(cap_{ij}) \right), \end{aligned}$$

where cap_{ij} is the necessary capacity of the link ij i.e. the sum of the traffic generated by the BTSs which use this link to connect themselves to a BSC.

Let l_j and d_{jBTS} denote the level and the degree of the site j in its connecting tree, and for some BSC i let d_{iBSC} denote the number of BTS's connected to it. With these notations our purpose is to minimize $Cost_{total}$ subject to the following constraints:

$$\begin{aligned} l_j &\leq L && \text{and,} \\ d_{jBSC} &\leq D_{BSC} && \text{for all BTS } j, \\ d_{iBTS} &\leq D_{BTS} && \text{for all BSC } i. \end{aligned}$$

The three equation provides that the BTS subnetwork will conform to the pre-defined requirements.

Based on the above constraints it is easy to calculate the minimum number of BSC's in the network as follows:

$$\frac{N_{BTS}}{D_{BSC}} \leq N_{BSC}.$$

5 The proposed planning algorithm

The presented optimization task is an NP-complete type problem, therefore we can not present an algorithm which is able to give guarantees for reaching the global optimal cost solution in reasonable time.

A possible solution would be to divide the above task into two independent processes: first, calculate the optimal BSC number and locations, and second, build the BTS trees using two independent optimization methods after each other. This scenario provides a relatively short running time (because of the simplified and smaller state space of the optimization process), but the solution will not be close enough to the global optimum.

A better solution is to do the above two tasks at the same time using a joint optimization process. If we follow this concept, the complexity of the problem will increase a little, but a solution near to the global optimum can be reached with much higher probability. We chose this approach to solve our specified design task and we developed an algorithm which is able to solve the design problem in a joint optimization process. This ensures that the solution will be near to the global optimum, while the running time will not increase significantly.

We propose a two–layer heuristic algorithm for solving the problem. In the higher layer we determine the optimal number and locations of the BSCs using a general local search heuristic method called simulated annealing. In the lower layer, in each step of the simulated annealing we create an access network topology

(BSC number and placement) for the current state taking the constraints into account.

5.1 The optimization process

The simulated annealing optimization method is able to find a solution near to the global optimum even in cases of a great state space with high probability. The advantages of this heuristic are general usability, easy implementation and relatively short running time.

In case of this planning task the size of the state space is 2^K, where K is the number of possible BSC locations. Namely, a state of the simulated annealing process is the current number of BSCs and their locations. The initial state is generated randomly, the algorithm selects some BTSs and places a BSC to each of them. During the optimization each state has some neighbours and in every step one of the neighbour states, called *new state* is also selected randomly. A neighbour state can be reached in the following ways:

- Adding a new BSC into a randomly chosen site.
- Removing a BSC from a randomly chosen site.
- Moving an existing BSC from a randomly chosed site to any other empty possible location.

The proposed optimization process is the following:

Initial step. Place the required number of BSC's at randomly chosen locations, build up the corresponding minimum cost tree and calculate the cost of this topology.

Step 1. Select from the following alternates randomly:

- Move a BSC to another location.
- Place a new BSC into the network.
- Remove a BSC from the network.

Step 2. To determine the cost of access links, build a minimum cost tree using a greedy algorithm which respects the constraints and current BSC positions.

Step 3. Calculate the difference between the cost of current and new network. On the basis of the acceptation criteria the algorithm accept or refuse the new network structure.

Step 4. If step number smaller than the maximum step go to *Step 1*, else finish the optimization.

In *Step 3* the algorithm accept the modification randomly — as usual in the simulated annealing methods — with probability

$$P_{accept} = \min\left\{1, \exp\left(-\frac{Cost_{new} - Cost_{curr}}{T}\right)\right\},$$

where $Cost_{new}$ and $Cost_{curr}$ are the total cost of the network after and before the modification and T is the so called *temperature* which decreases exponentially during the algorithm.

If the cost of new state is lower than the cost of current state the new state is always accepted, but in the other case the acceptation of the new state depends on the above mentioned stochastic criteria. At the beginning of the optimization the probability of the acceptance of a higher cost new state is close to 0.5, while later this probability decreases significantly. We refer the reader to [4] and [5] for more details.

5.2 The greedy algorithm for BTS access link topology design

We need to do a cost comparison in each step of the simulated annealing process to calculate if the new network state is better or not than the actual one. In case of the cost calculation of a network is not enough to know the position of the BSC's, but we have to know the topology of the BTS layer, because the cost of the physical links is a significant part of the total network cost as we have discussed before. One possibility to calculate the network cost would be to use a simple approximation presented in [3] where each BTS will be connected to the closest BSC, resulting in star topologies. We can build up this type of topology in $N_{BTS} \cdot N_{BSC}$ steps but the cost calculated this way is not realistic because it does not take into account the topological constraints. Therefore, it is necessary to calculate the exact physical topology of the BTS access links in each step.

Thus, in each step of the simulated annealing we use a greedy algorithm which gives a closely optimal solution.

This method is based on the well–known minimal spanning tree algorithm but we have to respect the constraint on the connecting trees. This modified spanning tree algorithm does not give the exact solution in all cases, but the solution will be close to the global optimum.

The computational resource requirements of this method will be essentially not more than the presented in [3], but it gives much more exact cost values for the current network and all constraints are taken into account. The operation of the greedy algorithm is as follows:

During the algorithm the set of sites containing BTS or BSC are divided to two disjoint sets B and U, namely the sites which are already connected to a BSC (or it contains a BSC itself) are in B and the others are in U. At the beginning the algorithm the sites containing a BSC are in set B and the others are in set U.

In each step for every BTS $u \in U$ the algorithm determines the $b_u \in B$ such that u can be connected to b_u without violating the constraints and the cost of connection is minimal. Then it chooses the BTS $u_{min} \in U$ with minimal cost of the connection and connect it to the corresponding element in B, and repeat this step while the set U is not empty.

To perform these steps efficiently we maintain a database of the actual best candidate for connection, its cost and its home BSC for all element of U. This data structure can be set up in $N_{BSC}N_{BTS}$ steps and, after connecting a BTS, it can be updated in $O(N_{BTS})$ steps in almost always as follows.

If the level of a newly connected BTS is less than L and its home BSC has less than D_{BSC} connected BTS then if it is closer to some BTS's which are in set U than their neighbour then we update its data.

If some BTS or BSC reaches its degree constraint then we have to update the data structure of all BTS's in U for which its neighbour is this site or is in this connecting tree, accordindly. This updating requires $O((N_{BTS})^2)$ steps in the worst cases, but usually it is much more less.

5.3 Cost calculation methods

In the cost calculation we can use three type of scenarios:

- In the first case the cost depends on just the distance of nodes and we ignore the capacity dependent cost parts. This approach may be called an uncapacitated topological design problem.
- Another approach if we use the previous concept, but after a tree was built up we calculate the capacity dependent costs for each link of trees, therefore the total network cost will be contains both distance and capacity dependent cost. The most important advantage of this method is

that it will eventually force the simulated annealing to move the BSC towards the high capacity BTS's, or maybe co–locate them, because in this way the capacity dependent link cost will be reduced. In this case the running time does not increase considerably, because just an additional calculation step is required after building up each tree.

- In case of the last scenario we take the capacity dependent cost into account in each step of the tree building. The node connection depends on both the distance and the generated capacity increasing in the tree by the new connected node. Because we need to make a cost calculation in case of connecting each node, the required steps of the optimization will increase significantly.

6 Performance of the algorithm

First, we have tested the computational efficiency of our algorithm. We found that it runs for several hundred nodes in practically feasible times. However, the most substantial disadvantage of simulated annealing is that, we can not guarantee that the global optimum will be found.

In this section we examine the accuracy of our algorithm by calculating lower bounds for the capacity independent model on smaller examples.

6.1 Calculation of the global optimum

In this section we give a method to compute the exact optimal solution by formulating the problem as an integer program.

We are looking for

$$\min\left(\sum_{i,j} c_{ij}x_{ij} + Cost_{BSC}\sum_{i} r_i\right)$$

with respect to the inequalities

$$\begin{array}{rcll}
0 \leq x_{ij} & \leq & 1 & \forall i,j, \\
0 \leq v_{ij} & & & \forall i,j, \\
0 \leq l_i & \leq & L & \forall i, \\
0 \leq r_i & \leq & 1 & \forall i, \\
0 \leq d_i & & & \forall i, \\
l_i - l_j + Nx_{ij} & \leq & 1 + N & \forall i,j, \\
l_i - l_j - Nx_{ij} & \geq & 1 - N & \forall i,j, \\
\frac{1}{L} l_i \leq r_i & \leq & l_i & \forall i, \\
\sum_j x_{ij} & \geq & r_i & \forall i, \\
\sum_i x_{ij} + Nr_j & \leq & D_{BTS} + N & \forall j, \\
v_{ij} & \geq & d_i - N + Nx_{ij} & \forall i,j, \\
d_j & \geq & 1 + \sum_i v_{ij} & \forall j, \\
d_j & \geq & D_{BSC} & \forall j,
\end{array}$$

where x_{ij}, l_i, r_i are integer variables for any $i,j \in \{1,\ldots,n\}$ and $N = n^2$. It's straightforward to verify that this system really corresponds to our task. So we can compute the real optimum using some integer programming software package, but unfortunately it works only on very small input. Comparison with the results of our proposed heuristic algorithm shows practically no difference, i. e. the simulated annealing finds the global optimum for small problems.

6.2 Relaxing the constraints

Another type of lower bound is relaxing the constraints on the tree connecting BTSs to the BSCs. This is a lower bound and it's easy to compute the optimum, by reducing this relaxed problem to finding a minimal–cost spanning tree on a modified graph as follows. We take a complete graph on S as its vertices, and the cost of the edges are simply the values of the original cost function. Moreover, we take an additional node, add an edge from this node to each of the other nodes with cost $Cost_{BSC}$. Deleting the additional node from the minimal–cost spanning tree we get a solution without respecting restrictions. Unfortunately, this way we end up in a quite different problem which doesn't provide us with a sharp lower bound.

#Nodes	Star version		Error
	exact opt.	SimAnn	
10	68.7538	68.7538	Opt.!
20	117.815	118.039	0.16%
30	150.129	150.38	0.17%
40	185.212	185.212	Opt.!
50	220.642	222.071	0.65%

Table 1: Results I: Simulated Annealing

6.3 Efficiency of the layers of the algorithm

The third way of testing was to examine the two layers of the algorithms separately. For testing the simulated annealing we examine the special case when the maximum depth of connecting tree is one, that is, only stars are enabled as a connecting tree. Moreover, we suppose that there is no degree constraint on the BSC's. This is very similar to the general case from the point of view of simulated annealing but the optimum solution can be found for larger inputs by solving the following mixed–integer linear program.

$$\min(\sum_{i,j} c_{ij}x_{ij} + Cost_{BSC}\sum_i r_i)$$

with respect to the inequalities

$$\begin{array}{rcll} 0 \leq x_{ij} & \leq & 1 & \forall i,j, \\ 0 \leq \ r_i & \leq & 1 & \forall i, \\ \sum_j x_{ij} + r_i & \geq & 1 & \forall i, \\ \sum_i x_{ij} & \leq & Nr_j & \forall j, \end{array}$$

where r_i are integer variables for any $i \in \{1,\ldots,n\}$ and N is same as above. It's easy to see that there always exists a solution of the above problem where every variable x_{ij} is integer, so this inequalities equivalent to our task.

The running result are given in Table 1. The simulated annealing method gives very good result. The error is almost equal to zero. Moreover these were not too small instances, so we can say that the simulated annealing works very well.

6.3.1 The greedy method

To examine the efficiency of our greedy method we fix a tcertain configuration of BSCs and find a minimal cost connecting opology without any restrictions. We can also solve this problem *exactly* by the spanning tree method and it gives us a lower bound for connecting with the restrictions with respect to this fixed set of BSC's.

#Nodes	SimAnn+ Greedy	MinSpan. (fixed BSC's)	Error
100	274.164	252.537	$< 8.6\%$
200	399.547	359.420	$< 11.1\%$
300	150.129	150.38	$< 10.3\%$
400	567.729	509.484	$< 11.4\%$
500	631.082	577.184	$< 9.3\%$
1000	914.95	807.546	$< 13.3\%$

Table 2: Result II: The Greedy Method

Some running results on expected network-sizes are shown in Table 2. These lower bounds give us that the error was at most about 10% on this network.

7 Conclusion

We have proposed a heuristic optimisation algorithm for the problem of planning cost–optimal UMTS accesss networks. Our algorithm jointly optimizes the number and location of Base Station Controllers and the access link topology connecting BTSs to their corresponding BSCs. The algorithm runs efficiently on large inputs, such as several times of 100 BTS's. We have tested the efficiency of our method by calculating either the exact optimum for smaller–size examples or calculating lower bounds for the solution. We have found that our algorithm gives results very close to the global optimum.

References

[1] Mark S. Daskin, *Network and Discrete Location*, 1995, Jon Wiley & Sons, Inc.

[2] Tero Ojanperä, Ramjee Prasad, *Wideband CDMA for Third Generation Mobile Communication*, 1998, Artech House Publishers.

[3] Paul Kallenberg, *Optimization of the Fixed Part GSM Networks Using Simulated Annealing*, Networks 98, Sorrento, October 1998.

[4] Bruce Hajek, *A Tutorial Survey of Theory and Applications of Simulated Annealing*, Proceedings of 24th Conference on Decision and Control, Ft. Lauderdale, Fl., December 1985.

[5] Bruce Hajek, *Cooling Schedules for Optimal Annealing, Mathematics of Operation Research*, Vol 13, No.2, May 1988

[6] Brian G. Marchent, *Third Generation Mobile Systems for the Support of Mobile Multimedia based on ATM Transport*, tutorial presentation, 5th IFIP Workshop, Ilkley, July 1997.

[7] Joseph C. S. Cheung, Mark A. Beach, Joseph P. McGeehan, *Network Planning for Third-Generaion Mobile Radio Sustems*, IEEE Communication Magazine, November 1994