INDIAN INSTITUTE OF MANAGEMENT CALCUTTA

WORKING PAPER SERIES

WPS No. 701/ July 2012

Dual-homing of RNCs in UMTS Networks

by

Samir K Sadhukhan
SSA, IIM Calcutta, Diamond Harbour Road, Joka P.O., Kolkata 700 104 India

Swarup Mandal
Wipro Technologies Kolkata, India

&

Debashis Saha
Professor, IIM Calcutta, Diamond Harbour Road, Joka P.O., Kolkata 700 104 India

Dual-homing of RNCs in UMTS Networks

Samir K. Sadhukhan
IIM Calcutta, Joka, D. H. Road, Kolkata – 700104, India
samir@iimcal.ac.in

Swarup Mandal
Wipro Technologies, Kolkata, India
Swarup.mandal@wipro.com

Debashis Saha
MIS & Computer Sc group, IIM Calcutta, Joka, D. H. Road, Kolkata – 700104, India
ds@iimcal.ac.in

Abstract- In conventional UMTS cellular networks, during deployment usually a set of NodeBs is assigned to one Radio Network Controller (*RNC*), and a set of *RNC*s to one Serving GPRS Support Node (*SGSN*) for data services, as well as to one Mobile Switching Centre (*MSC*) for voice services. Operators thus far have considered single-homing of *RNCs* to *MSCs/SGSNs* (i.e., many-to-one mapping) with an objective to reduce the total cost over a fixed period of time. However, a single-homing network does not remain cost-effective any more when subscribers later on begin to show specific inter-*MSC/SGSN* mobility patterns (say, diurnality of office goers) over time. This necessitates post-deployment topological extension of the network in terms of dual-homing of *RNCs*, in which some specific *RNCs* are connected to two *MSCs/SGSNs* via direct links resulting in a more complex many-to-two mapping structure in parts of the network. The partial dual-homing attempts to increase link cost minimally and reduce handoff cost maximally, thereby significantly reducing the total cost in a post-deployment optimal extension. In this paper, we formulate the scenario as ILP problem convert into a state space search problem and then solve it using three meta-heuristic techniques, namely Simulated Annealing (SA) , Tabu search (TS) and Ant colony optimization (ACO). The comparative results reveal that, ACO based technique is more efficient among the other meta-heuristic techniques in solving dual-homing problem.

Keywords- Network planning; Cellular network; UMTS; Dual-homing; Optimization; Simulated Annealing; Tabu Search; Ant Colony

1 Introduction

Post-deployment planning plays a key role in optimizing the total cost of operation for modern cellular networks (Demestichas etl, 1999; Saraydar etl, 2000, Demrikol etl, 2001). The dynamic nature of subscriber's profile makes the operation of cellular networks become sub-optimal with the passage of time in terms of the operational cost, and, hence, re-planning of networks needs to be done from time to time, with the existing deployment as a set of constraints (to protect investments).

An operator usually face one of the following three possible scenarios in post-deployment tuning phase: (i) both new call traffic and handoff traffic increase, (ii) new call traffic in-creases but handoff traffic does not, (iii) new call traffic does not increase but handoff traffic does. The first two cases usually occur when subscribers' density increases permanently. This can be addressed in post deployment planning phase by splitting cells (where capital expenditure as well as handoff cost will increase) or by redefining the connectivity of cells and switches. The third case may arise due to a gradual change in mobility pattern of the existing subscriber base over a long period of time (Clayirci & Akyildiz, 2002). This problem can be addressed by regrouping cells into new clusters i.e., by changing the connectivity of *NodeBs* to *RNCs* and *RNCs* to *MSCs/SGSNs* (Quintero & Pierre, 2002; Pierre & Houeto 2002; Diallo etl, 2006; Amzallag etl 2007).

However, this cannot take care of situations where handoff increases with no increase in total traffic due to periodic (temporal) changes of subscribers' locations. If there is a clear pat-tern of this temporal mobility of subscribers, a multi-homing consideration (where many *NodeBs* are connected to one *RNC*, and many *RNCs* are connected to one *MSC/SGSN*) will be a useful strategy in post deployment tuning stage. Obviously, the multi-homing concept can be implemented at two levels, namely, in the first level, multi-homing of *NodeBs*, and, in the second level, multi-homing of *RNCs*. In this paper, we have considered dual-homing of *RNCs*, where some *RNCs* (to be decided optimally) are connected to two *MSCs/SGSNs* (as shown in Fig. 1) to reduce handoff cost, unlike single homing where one *RNC* is connected to one *MSC/SGSN* only. In order to achieve an optimal selection of *RNCs* from the set of potential *RNCs* to be dual-homed, we have proposed three meta-heuristic techniques in this work.

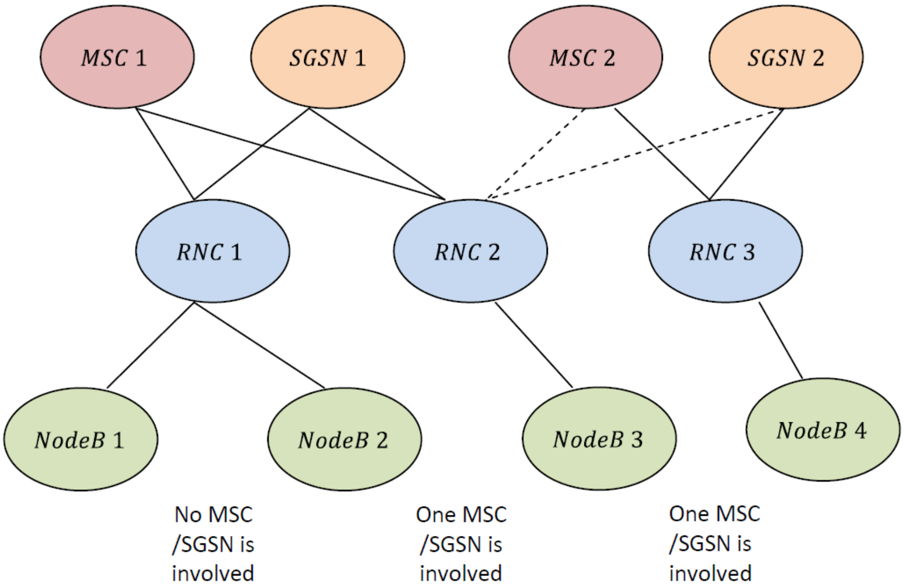


Figure 1. Change in handoff type due to dual-homing of RNC in a UMTS network (only relevant part of the UMTS network is shown; dotted lines indicate additional connections)

We define three types of handoff, namely simplest, simple and complex. We call a handoff simplest when no *MSC/SGSN* is involved in the handoff process. In case of simplest handoff, one RNC handles the whole handoff process. On the contrary, a simple handoff involves one (and only one) *MSC/SGSN* in the handoff process, whereas a complex handoff involves two *MSCs/SGSNs*. Obviously, simplest handoff cost is negligible with respect to complex handoff cost. Dual-homing of *RNC* is attractive for operators because it converts many complex handoffs to simple handoffs. For example, the complex handoff between *NodeB 3* and *NodeB 4* has changed to simple handoff due to dual-homing of *RNC 2* in Fig. 1.

Researchers have traditionally formulated the *NodeB-RNC* (earlier known as cell-switch) assignment problem as a combinatorial optimization problem and have solved it using meta-heuristics (Demrikol etl, 2001;Quintero & Pierre, 2002; Pierre & Houeto 2002; Diallo etl, 2006) or domain-specific heuristics (Merchant & Sengupta, 1995; Saha etl, 2000; Bhattacharjee etl, 2004; Mandal, Saha & Mahanti, 2004, 2005). However, till date they all have considered single homing criteria, and have excluded the multi-homing scenario. Recently, we attempted to solve the dual homing of *RNCs* to *MSCs* in 2.5G networks with a suboptimal greedy algorithm (Sadhukhan etl, 2007) following the approach given in (Din &Tseng, 2002) which, however, deals with ATM networks (not cellular) and employs the meta-heuristic genetic algorithm (GA). The state space formulation of the problem was absent in (Sadhukhan etl, 2007), and, hence, we could not devise any domain specific heuristic there. In the current work, we have extended our earlier work to UMTS networks for dual homing of *RNCs* to *MSCs/SGSNs* and solved it using three separate techniques. We have used simulated annealing (SA) (Demrikol etl, 2001), tabu search (TS) (Pierre & Houeto, 2002) and ant colony optimization (ACO) (Colorni,Dorigo & Maniezzo, 1992) meta-heuristic techniques to solve the dual-homing problem and then compared their solution quality. It is found that ACO based technique is better than SA and TS based techniques.

The paper is organized in five sections. Following Section 1 that introduces the dual homing assignment problem, Section 2 presents an integer linear programming (ILP) formulation of the problem. Section 3, Section 4 and Section 5 discuss the SA, TS and ACO based solution methodologies. Section 6 contains the experimental results with discussion, and Section 7 concludes the paper.

2 Mathematical Formulation

Let us consider that, in the UMTS network of a mobile telecom service provider (MTSP), there are n *NodeBs,* r *RNCs*, m *MSCs* and s *SGSNs*, whose locations are known. Let $I = \{1,2,\cdots,n\}$ denote the set of *NodeBs*, $J = \{1,2,\cdots,r\}$ denote the set of *RNCs*, $K = \{1,2,\cdots,m\}$ denote the set of *MSCs* and $L = \{1,2,\cdots,s\}$ denote the set of *SGSNs*. From the existing single homing network, the initial assignments of *NodeBs* to *RNCs* to *MSCs* and *SGSNs* are known a priori. Throughout this formulation, we use a small letter to denote a member of the set represented by the corresponding capital letter; for example, $i \in I$, $j \in J$. Moreover, we assume *NodeB* i and *NodeB* i' are different ($i \neq i'$). Similarly we assume $j \neq j'$, $k \neq k'$, $l \neq l'$.

Let us now consider the following notations:

$x'_{jk} = 1$, if *RNC* j is assigned to *MSC* k (new link) in dual homed network, 0, otherwise

$x''_{jl} = 1$, if *RNC* j is assigned to *SGSN* l (new link) in dual homed network, 0, otherwise

$d_{ij} = 1$, if *NodeB* i is assigned to *RNC* j (old link) in single homed network, 0, otherwise

$d'_{jk} = 1$, if *RNC* j is assigned to *MSC* k (old link) in single homed network, 0, otherwise

$d''_{jl} = 1$, if *RNC* j is assigned to *SGSN* l (old link) in single homed network, 0, otherwise

c'_{jk} is the amortized cost of the link between *RNC* j and *MSC* k

c''_{jl} is the amortized cost of the link between *RNC* j and *SGSN* l

cap_j and cap'_j	are the capacity in circuit switching (number of calls per unit of time) and capacity in packet switching (bits per second) of *RNC* j.
cap''_k and cap'''_l	are the capacity in circuit switching (number of calls per unit of time) and capacity in packet switching (bits per second) of *MSC* k and *SGSN* l respectively.
f^{voice}_{jk}	is the amount of voice traffic produced by *RNC* j and destined to *MSC* k
f^{data}_{jl}	is the amount of data traffic produced by *RNC* j and destined to *SGSN* l
$H^{voice}_{ii'}$	is the cost per unit time for complex handoff between *NodeB* i and *NodeB* i' involving two *MSCs*
$H^{data}_{ii'}$	is the cost per unit time for complex handoff between *NodeB* i and *NodeB* i' involving two *SGSNs*
μ_1	Ratio of cost of a complex voice handoff and cost of a simple voice handoff
λ_1	Ratio of cost of a complex data handoff and cost of a simplest data handoff

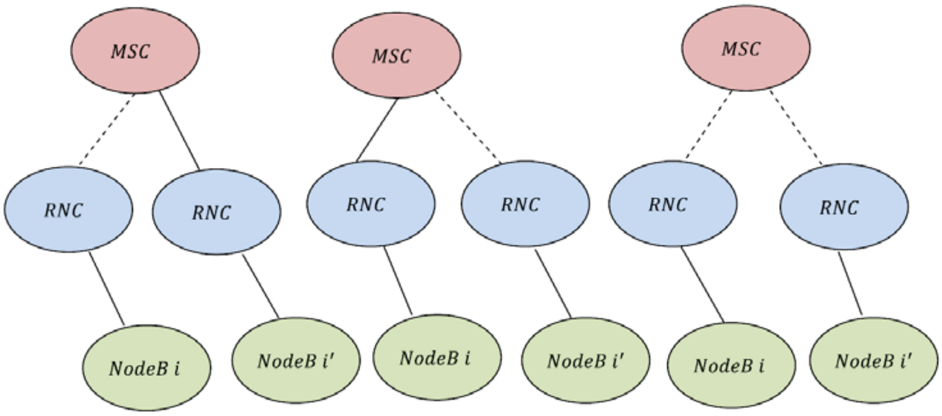


Figure 2: Dual-homing of RNC which converts complex handoff to simple handoff

Figure 2 shows the conversion of the complex handoff occurring between *NodeB* i and *NodeB* i' to simple handoff due to the new link between *RNC* and *MSC*. Let us define the following composite variables: $p_{ijk} = d_{ij}d'_{jk}$, $u_{ik} = \sum_{j\epsilon J} p_{ijk}$ Then u_{ik} is equal to 1 if there is an old path from *NodeB* i to *MSC* k. Similarly, we define, $\hat{p}_{ijk} = d_{ij}x'_{jk}$, $\hat{u}_{ik} = \sum_{j\epsilon J} \hat{p}_{ijk}$. Then $\hat{u}_{ik}$ is equal to 1 if there is a path *NodeB* i to *MSC* k such that there is an old link between *NodeB* i-*RNC* and new link between *RNC*- *MSC* k.

Then, $(\hat{u}_{ik}\, u_{i'k} \vee u_{ik}\, \hat{u}_{i'k} \vee \hat{u}_{ik}\, \hat{u}_{i'k})$ is equal to 1 if *NodeB* i and *NodeB* i' are under *MSC* k using at least one new path. Moreover, $\vee_{k\epsilon K}(\hat{u}_{ik}\, u_{i'k} \vee u_{ik}\, \hat{u}_{i'k} \vee \hat{u}_{ik}\, \hat{u}_{i'k})$ is equal to 1 if *NodeB* i and *NodeB* i' are under one *MSC* using at least one new path. The operator $\vee$ stands for logical OR.

Let us define the following variables: $q_{ijl} = d_{ij}d''_{jl}$, $v_{il} = \sum_{j\epsilon J} q_{ijl}$ Then v_{il} is equal to 1 if there is a path from *NodeB* i to *SGSN* l. Similarly, we define, $\hat{q}_{ijl} = d_{ij}x''_{jl}$, $\hat{v}_{il} = \sum_{j\epsilon J} \hat{q}_{ijl}$ .Then $\hat{v}_{il}$ is equal to 1 if there is a path *NodeB* i to *SGSN* l such that there is an old link between *NodeB* i-*RNC* and new link between *RNC*-*SGSN* l.

Then, $(\hat{v}_{il}\, v_{i'l} \vee v_{ik}\, \hat{v}_{i'l} \vee \hat{v}_{il}\, \hat{v}_{i'l})$ is equal to 1 if *NodeB* i and *NodeB* i' are under *SGSN* l using at least one new path. Moreover, $\vee_{l\epsilon l}(\hat{v}_{il}\, v_{i'l} \vee v_{ik}\, \hat{v}_{i'l} \vee \hat{v}_{il}\, \hat{v}_{i'l})$ is equal to 1 if *NodeB* i and *NodeB* i' are under one *SGSN* using at least one new path.

Therefore the total reduction of complex handoff to simple handoff in the dual home network will be

$$HC = \sum_{i\epsilon I}\sum_{i'\epsilon I} \vee_{k\epsilon K}(\hat{u}_{ik}\, u_{i'k} \vee u_{ik}\, \hat{u}_{i'k} \vee \hat{u}_{ik}\, \hat{u}_{i'k})(1 - 1/\mu_1)\; H_{ii'}^{voice}$$

$$+ \sum_{i\epsilon I}\sum_{i'\epsilon I} \vee_{k\epsilon K}(\hat{v}_{il}\, v_{i'l} \vee v_{ik}\, \hat{v}_{i'l} \vee \hat{v}_{il}\, \hat{v}_{i'l})\; (1 - 1/\lambda_1)\; H_{ii'}^{data}$$

The total cost of the new links in the dual-home network will be

$$LC = \sum_{j\epsilon J}\sum_{k\epsilon K} x'_{jk}c'_{jk} + \sum_{j\epsilon J}\sum_{l\epsilon L} x''_{jl}c''_{jl}$$

The link constraints are

$$x'_{jk} + d'_{jk} \leq 1$$

The link constraint signifies that there could be at most one link (either an old link or a new link or no link) between *RNC* j and *MSC* k.

Similarly, $x''_{jl} + d''_{jl} \leq 1$

$$\sum_{k \epsilon K} x'_{jk} \leq 1$$

The above constraint signifies that *RNC* j can be connected to at most one *MSC* using a new link.

Similarly,

$$\sum_{l \epsilon L} x''_{jl} \leq 1$$

The capacity constraints can be considered in two ways:

i. Worst case capacity constraints (when the capacity of *RNCs* are considered) are

$$\sum_{j \epsilon J} cap_j (x'_{jk} + d'_{jk}) \leq cap''_k$$
$$\sum_{j \epsilon J} cap'_j (x''_{jl} + d''_{jl}) \leq cap'''_l$$

ii. Best case capacity constraints (when the capacity utilization of *RNCs* are considered) are

$$\sum_{j \epsilon J} f_{jk}^{voice} (x'_{jk} + d'_{jk}) \leq cap''_k$$
$$\sum_{j \epsilon J} f_{jl}^{data} (x''_{jl} + d''_{jl}) \leq cap'''_l$$

Therefore, the dual-homing problem can be formulated as

Maximize $HC - LC$	$(i \neq i')$	(A)
Where		
$p_{ijk} = d_{ij} d'_{jk}$	$i \in I, j \in J, k \in K$	(1)
$u_{ik} = \sum_{j \epsilon J} p_{ijk}$	$i \in I, k \in K$	(2)
$\hat{p}_{ijk} = d_{ij} x'_{jk}$	$i \in I, j \in J, k \in K$	(3)
$\hat{u}_{ik} = \sum_{j \epsilon J} \hat{p}_{ijk}$	$i \in I, k \in K$	(4)
$q_{ijl} = d_{ij} d''_{jl}$	$i \in I, j \in J, l \in L$	(5)
$v_{il} = \sum_{j \epsilon J} q_{ijl}$	$i \in I, l \in L$	(6)
$\hat{q}_{ijl} = d_{ij} x''_{jl}$	$i \in I, j \in J, l \in L$	(7)
$\hat{v}_{il} = \sum_{j \epsilon J} \hat{q}_{ijl}$	$i \in I, l \in L$	(8)

Subject to

$x'_{jk} + d'_{jk} \leq 1$	$j \in J, k \in K$	(1)
$x''_{jl} + d''_{jl} \leq 1$	$j \in J, l \in L$	(2)

$$\sum_{k\epsilon K} x'_{jk} \leq 1 \qquad j \in J \qquad (3)$$
$$\sum_{l\epsilon L} x''_{jl} \leq 1 \qquad j \in J \qquad (4)$$
$$\sum_{j\epsilon J} cap_j (x'_{jk} + d'_{jk}) \leq cap''_k \qquad k \in K \qquad (5)$$
$$\sum_{j\epsilon J} cap'_j (x''_{jl} + d''_{jl}) \leq cap'''_l \qquad l \in L \qquad (6)$$

So, here problem is to find the joint dual homing assignment matrix, $[x'_{jk}]$ and $[x''_{jl}]$ while maximizing the reduction of total cost given by the expression, (A) subject to the constraints indicated in the equations (1) – (6).

The objective function, formulated above, is not linear and contains boolean operator 'OR' and product of binary variables. The boolean operator 'OR' can be converted to arithmetic addition as follows:

$X \vee Y = X + Y - XY$, where X and Y are binary variable.

The nonlinear term XY (i.e., the product of two binary variables) can be converted to linear form using additional variable $Z = XY$ (Z is binary too) and by adding the following type of constraints in the above formulation.

$$Z \leq X$$
$$Z \leq Y$$
$$Z \geq X + Y - 1$$

Similarly, product of more than two binary variables can be converted to linear form using additional binary variables and constraints. Thus, the above formulation of the problem can be converted to a 0-1 integer linear programming (ILP) problem. An exhaustive enumeration technique for assigning *RNCs* in dual homing problem requires checking of $(m + s)^r$ combinations to solve. When the problem size is large, meta-heuristic techniques may be suitable to solve such problems.

A state S is defined by a set of connections, $S = \{C_1, C_2, C_3, C_4 \dots \dots . .\}$ represented in the form of a matrix. A connection C is an ordered pair $< j\text{-}k >$ which implies that *RNC* j has been assigned (connected) to *MSC* k. The solution state space of the problem is the entire set of sets of connections generated by all possible and feasible combinations of $< j\text{-}k >$ pairs. To illustrate the concept, let us show a toy UMTS network of three *MSCs* and six *RNCs* (*NodeBs* are omitted for the sake of simplicity) in Fig. 3. Spare capacities of *MSCs* are shown in square boxes next to them in Fig. 3.

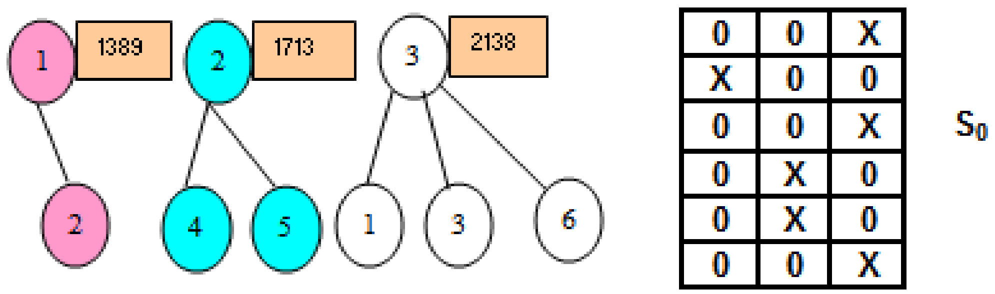


Figure 3. Initial single-home network and its state representation

S_0 (Fig. 3) represents the initial state for the network; we assume that a state matrix represents the network in terms of a connectivity matrix whose rows indicate *RNCs* and columns indicate *MSCs*. State representation of a network is explained in more details in Section 6. Let us consider that S_0 is the initial single-homed network that needs post-deployment tuning now. We shall use this toy network in explaining our proposed algorithms in subsequent sections. For simplicity, we assume only voice network and cost of a simple handoff is negligible compare to complex handoff. Table 1 shows the capacities of *RNCs*, and Table 2 shows the complex handoff costs between *RNCs* for S_0. Table 3 shows the amortized cost of the high speed links from *RNCs* to *MSCs*.

TABLE 1. RNC capacity

RNC	1	2	3	4	5	6
Capacity	480	1158	318	1188	240	984

TABLE 2. Handoff Cost for S_0

RNC \ RNC	1	2	3	4	5	6
1	-	-	-	9	14	-
2	-	-	19	4	-	24
3	-	19	-	5	6	-
4	9	4	5	-	-	-
5	14	-	6	-	-	-
6	-	24	-	-	-	-

TABLE 3. Link Cost

RNC \ MSC	1	2	3
1	36	36	-
2	-	36	36
3	20	26	17
4	36	-	36
5	26	17	20
6	35	44	10

The sum of handoff cost and link cost of the initial state S_0 is 206.

3. SIMULATED ANNEALING TECHNIQUE

Simulated Annealing (SA) is a global optimization algorithm that can be applied to solve various combinatorial optimization problems. Annealing is a metallurgical process. It is basically heating followed by a slow cooling down process of a molten material in order to increase the crystal size and hence decrease their structural defects. Heat causes the atoms to get energized from their initial positions (a local minimum), wander randomly in states of higher energy, and then slow cooling gives the atoms chance to find configurations with lower internal energy than the initial state. SA-based techniques follow this approach for combinatorial optimization. By analogy with metallurgical annealing, SA randomly finds a neighbour of a current state, and then replaces the current state with this neighbour based on an acceptance probability which depends on the costs of the two states and the temperature of the system. When the temperature of the system is high (i.e., in the beginning), the probability of an 'uphill' move (i.e., acceptance of a worse solution) is more. This prevents the search process from being stuck at a local optimum. However, as the temperature of the system decreases slowly over time, the probability of accepting a worse solution decreases, and, at considerably low temperatures, almost all moves are 'down-hill'. When temperature becomes 'zero', the system freezes giving the lowest cost state as output.

The algorithm we have used here is not a pure SA algorithm. A pure SA algorithm does not keep track of the best solutions found in previous iterations. Thus, a pure algorithm, at the end of the iterations, always returns the current state (i.e., the state obtained at the last iteration) to be the solution. In contrast, our modified SA algorithm returns the best result of all the states visited during the iterations. We have found that, in several cases, this can be a significant improvement over the solution returned by the pure SA.

In SA, we start with an initial solution marked as the current state. Then, at each iteration, a child node is selected as the neighbour (the leftmost feasible child of the current state). We shall define neighbour below. If the acceptance probability of the selected state is greater than some threshold, the selected state is accepted for further exploration and is made the current state. If the selected state is not accepted, the next child is taken as the neighbour (i.e., the one to the right of the previous child). In this way the search continues. Typically, in this formulation, all children of a state are 'neighbours' of the state.

Neighbourhood Function: We define neighbourhood function of a state as a function which produces feasible children of the state. Neighbourhood generation of one example initial state S_1 (for a toy network of *4 RNCs* and *3 MSCs*) is shown in Fig. 4. All children of a state may not be feasible because some child states may correspond to infeasible network operation. We need to ignore them. So only the feasible children (without repetitions) of a parent state are its neighbour at the next level.

In the example shown in Fig. 4, S_2, S_3, S_4, S_5 and S_6 are children (neighbours) of the current initial state S_1. Neighbours of a node are generated by changing one connection of the node at a time, thus generating all permutations which differ from the parent state in exactly one *RNC*. In the shown example, S_2 differs from the parent at *RNC 1*, S_3 differs in *RNC 2*, S_4differs in *RNC 3*, S_5 and S_6 differ from the parent at *RNC 4* and so on. Hence, all pairs of nodes that differ from each other in exactly one *RNC* are neighbours of each other also.

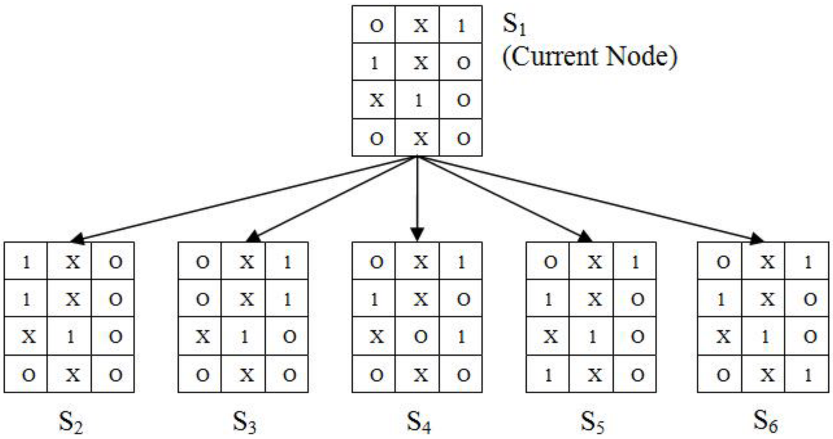


Figure 4. Neighbourhood nodes (states) of an initial node (state)

Reachability of Neighbourhood Function: The neighbourhood function we are using is exhaustive i.e., all possible combinations of *RNC*-to-*MSC* assignments are generated in this search tree. Hence, every node can be reached from every other node and the reachability criterion is satisfied.

Selection Criteria for new current state: If the cost C' of any randomly chosen neighbour S' of the current state S is less than the cost C of current state S, then the neighbour S' is selected as the new current state for the next iteration. Otherwise, a random number $\mathrm{R}(0,1)$ between 0 and 1 is chosen from a uniform distribution. If $\mathrm{P}(\mathrm{C}, \mathrm{C}', \mathrm{T}) > R(0,1)$, the neighbor S' is selected as the new current state (T is the current temperature, and function P is defined below). Otherwise, the present current state is propagated to the next iteration as the new current state. $\mathrm{P}(\mathrm{C}, \mathrm{C}', \mathrm{T})$ is the acceptance probability and is defined by the Boltzmann probability factor as follows:

$$\mathrm{P}(\mathrm{C}, \mathrm{C}', \mathrm{T}) = 1, \text{ if } \mathrm{C}' < C$$
$$= \mathrm{e}^{-\Delta\mathrm{C}/\mathrm{T}} \text{ where } \Delta\mathrm{C} = \mathrm{C}' - \mathrm{C}, \text{ otherwise}$$

Annealing Schedule: Typically, update of the temperate T is done by using the relation $\mathrm{T} = \mathrm{T} * \alpha$ (α is known as the cooling rate) after every V iterations, starting with the initial temperature $\mathrm{T} = \mathrm{T}_0$.

Thus, the various parameters for the SA algorithm are:

- α, the cooling rate i.e., the rate at which the temperature decreases $(0 < \alpha < 1)$
- T_0, the initial temperature
- V, the maximum number of iterations at a particular temperature
- γ, the maximum number of consecutive acceptance of worse solutions
- I, the maximum number of iterations, $\mathrm{I} \gg \mathrm{V}$

Initial Feasible Solution: First an initial solution has to be generated which becomes the current state S for the very first iteration. To generate an initial feasible solution, we take the following heuristic approach (algorithm is given below). We maintain internally a table which stores the possible handoff reduction that can be achieved by connecting an *RNC* to an *MSC*. We connect that *RNC* to the *MSC* only if (i) the *RNC* is not already dual-homed, (ii) the reduction achieved is greater than the link cost, and (iii) the capacity of *MSC* is sufficient to accommodate the capacity of the *RNC*. Since we are designing a dual-homing approach, an *RNC* cannot have more than 2 links i.e., an *RNC* cannot be connected to more than two *MSCs*. After connecting an *RNC* to an *MSC*, the capacity of the *MSC* is reduced by an amount equal to the capacity of the *RNC*. We simply try to connect *RNCs* to *MSCs* one after another in this greedy fashion provided the capacity constraint and link constraint are satisfied for each new connection. This ultimately results in a complete solution after which no more connections are possible. This is then taken as the initial feasible solution. Then the main algorithm sets out to improving this initial solution.

<u>Algorithm GENERATE_INITIAL_SOLUTION:</u>
for $k = 0$ to *MAX_NO_OF_MSCS*
{ for $j = 0$ to *MAX_NO_OF_RNCS*
{ if *RNC* j is connectable to *MSC* k (i.e. link cost from *RNC* j to *MSC* k < handoff cost reduction

& spare capacity of *MSC* k is $\geq$ capacity of *RNC* j & number of links of *RNC* j is $= 1$)
{ connect j to k
decrement the spare capacity of *MSC* k by the capacity of *RNC* j
increment the number of links of *RNC* j by 1
update the tables that store information about the handoff cost and handoff cost reduction
}
}
}

The cost of the initial solution is also computed. The initial solution becomes the current solution for the first iteration. The iteration steps are performed in the main annealing process.

Algorithm SA
Step 1. (initialization) $iteration\_count\_ = 0$, $0 < \alpha < 1$ (cooling_rate),) $max\_iteration\_count\_$, T, V
Step 2. Find an initial feasible solution S. C = cost of S. Set $best\_state = S$, $best\_cost = C$.
Step 3. Set S as the current state.
Step 4.(termination) IF $iteration\_count\_ > max\_iteration\_count\_$, exit with output node $best\_state$ and cost $best\_cost$.
Step 5. Randomly select a neighbour of state S.
Step 6. Call the selected neighbor state S'. C' = cost of S'.
Step 7. Check if S' is the best solution found so far; if yes, store it as $best\_state$ and its cost C' as $best\_cost$.
Step 8. Generate a random number R between 0 and 1. Compute the acceptance probability as follows:

$$P(C, C', T) = 1, \text{ if } C' < C$$
$$= e^{(C-C')/T} \quad \text{otherwise}$$

Step 9. IF $P(C, C', T) > R$, select S' to be the next state S and C = cost of S.
Step 10. $iteration\_count\_ = iteration\_count\_ + 1$
Step 11. IF $iteration\_count\_ \ mod\ V = 0$ then $T = T * \alpha$
Step 12. GOTO Step 4.

Example: For the network of Fig. 3, we take SA parameters as T_0=3000, V=50, α =0.5, I =1000. Link cost of single home network = 17+17+10 = 44. Total handoff cost of single home network = 9+14+19+4+24+19+5+6+9+4+5+14+6+24 = 162. Total cost of single home network = Link cost + Handoff cost = 44+162 = 206. The state representation of the toy single home network is also given in Fig. 3 as S_0. To generate the initial solution, connections are attempted one after another in a directed manner i.e., in the order 1-to-1, 2-to-1, 3-to-1, 4-to-1, 5-to-1, 6-to-1, 1-to-2, 2-to-2, 3-to-2 and so on. The connections already present in S_0 are ignored. If a connection is found to be feasible (i.e., capacity and link constraints are satisfied) then it is included. The initial solution thus generated and the *RNC*-*RNC* handoff costs for that solution are given in Fig. 5(a) and Fig 5(b).

RNC \ MSC	1	2	3
1	0	1	X
2	X	0	0
3	1	0	X
4	0	X	0
5	0	X	0
6	1	0	X

Figure 5. (a) Intial solution

RNC \ RNC	1	2	3	4	5	6
1	-	-	-	9	14	-
2	-	-	-	4	-	-
3	-	-	-	5	6	-
4	9	4	5	-	-	-
5	14	-	6	-	-	-
6	-	-	-	-	-	-

(b) Handoff costs

Total cost of this solution = Single home link cost + Dual home link cost + Handoff cost=44+ (36+20+35) + (4+5+6+ 4+5+6) = 165. Hence, cost reduction = Single home cost – dual home cost = 206 – 165 = 41 (assuming cost of a simple handoff is negligible).

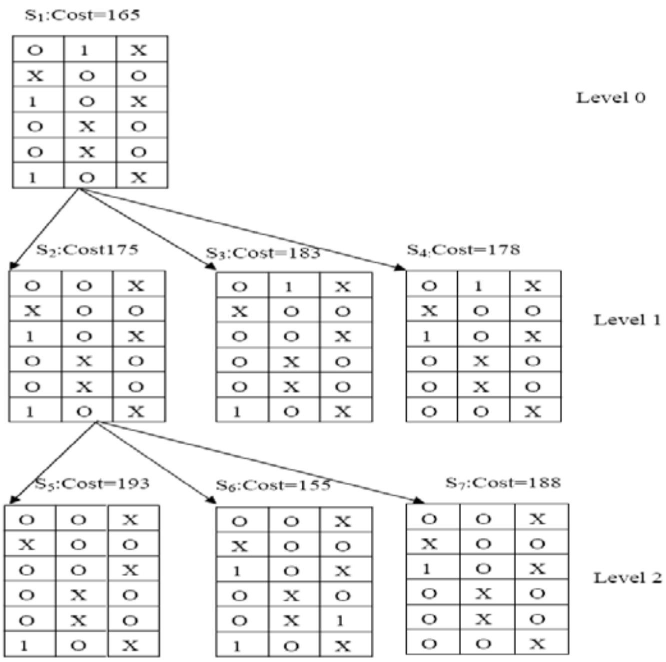


Figure 6. Neighbour generation. Only feasible nodes are shown.

Now, taking this initial solution as the root node (current node), the tree traversal starts in a breadth first manner. Children are accepted probabilistically and new solutions are checked for improvements over the previously found best solution. The leftmost feasible child of the current node and the corresponding handoff matrix are shown in Fig. 6.

Cost of this child (neighbour) = Single home link cost + Dual home link cost + Handoff cost = 44+(20+35)+(4+5+6+9+14+4+5+6+9+14) = 175. Hence, cost reduction=Single home cost – dual home cost = 206 – 175 = 31. Also, acceptance probability = $e^{-\Delta C/T}$ = $e^{-(175-165)/3000}$= 0.996672. Thus, at initial high temperature, we see the probability of accepting a worse solution is very high, and we accept it. The tree traversal continues in this fashion, and each child is checked for acceptance. After every 50 iterations, the temperature of the system is halved (α =0.5). We stop when the iteration count reaches the a priori set value of I.

RNC \ MSC	1	2	3
1	0	0	X
2	X	0	0
3	1	0	X
4	0	X	0
5	0	X	0
6	1	0	X

Figure 7. (a) New solution

RNC \ RNC	1	2	3	4	5	6
1	-	-	-	-	-	-
2	-	-	-	4	-	-
3	-	-	-	5	6	-
4	-	4	5	-	-	-
5	-	-	6	-	-	-
6	-	-	-	-	-	-

(b) Handoff costs

4. TABU SEARCH TECHNIQUE

This process is very similar to SA. The word “tabu” describes a sacred place or object. Things that are tabu must be left alone and should not be visited or touched. Tabu Search extends hill climbing by this concept– it declares solution candidates which have already been visited as tabu. Hence, they must not be visited again, and the optimization process is less likely to get stuck to a local optimum. The simplest realization of this approach is to use a list of tabu, which stores all solution states that have already been visited. If a newly generated neighbour can be found in this list, it is not accepted but rejected right away. Of course, the list cannot grow infinitely but has a finite maximum length N.

The tabu search (TS) algorithm we are using is straightforward. We start with an initial solution as current solution (say S) which can be found randomly. In every iteration of TS, we find a new solution by making local movements on this current solution. The neighbourhood function is used to find all neighbours of the current state S i.e., the neighbourhood set of S. The next solution state is the best solution among all the possible neighbours of S which are not already in the tabu list. Here, we use the same state space formulation and neighbourhood function as those used in SA.

Typically there are two kinds of tabu lists- (i) a long term memory maintaining the history through all the exploration process as a whole, and (ii) a short term memory to keep the most recently visited tabu movements. The first approach can actually produce the optimal solution if the state space search algorithm is exhaustive and the reachability criterion is satisfied. We use the second approach for faster solution. The tabu list has a finite length which is a parameter for the process. Whenever a case arises in which we have to add a solution to the tabu list and the list is full, we first remove the oldest solution from the list and then add the new one.

Neighbourhood Function: The neighbourhood generation used here is similar to that used in SA (Fig. 6). However, unlike a single neighbour in SA, all neighbours of the current node are generated. Then, the best child (neighbour), not already in the tabu list, is selected as the current node and search continues in the same manner. The various parameters needed for the TS algorithm are:

- N, the length of the tabu list i.e., how many recent solutions are to be remembered
- I, the maximum number of iterations

Initial Feasible Solution: Same approach is used as it is in the case of SA.

Algorithm TS

Step 1. (initialization) $iteration\_count = 0$, $max\_iteration\_count$, N (size of tabu_list)
Step 2. Find an initial feasible solution S. from the state space. C = cost of S. Set $best\_state = S$, $best\_cost = C$.
Step 3. Set S. as the current state.
Step 4. (termination) IF $iteration\_count > max\_iteration\_count$, exit with output node $best\_state$ and cost $best\_cost$.
Step 5. Generate the feasible neighbour set $N(S)$ of the state S.
Step 6. Select the best state from $N(S)$ which is not in the tabu_list. Call this state S'. C' = cost of S'.
Step 7. Check if S' is the best solution found so far; if yes, store it as $best\_state$ and its cost C' as $best\_cost$.
Step 8. Add S' to the tabu_list.
Step 9. Select S' to be the next state S and C = cost of S.
Step 10. $iteration\_count = iteration\_count + 1$
Step 11. GOTO Step 4.

Example: For the network of Fig. 3, we take parameters I=3, N=2 (size of tabu list). Initial solution is generated similar to the one found in SA and hence is as given in Fig. 5(a). Total cost of this solution is 165 and cost reduction is 41. This solution (S_1) is put into the tabu list as the first entry (Fig. 8(a)).

Tabu Search Process: As shown in Fig. 6, the initial solution is made as the root node. Of the three feasible children in level 1, the costs (from left to right respectively) are 175, 183 and 178. Hence, the best child is the leftmost child (S_2), and it is not in the tabu list. It is accepted, made the current node (to be explored next) and entered into the tabu list. The list now contains S_1 and S_2 (Fig. 8(b)).

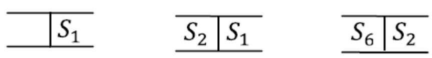


Figure 8(a) (b) (c)

At level 2 (iteration 2), there are three feasible children. The costs are 193, 155 and 188 respectively (from left to right). Of these, the best child is accepted i.e., S_6, the middle one (cost=155) which is not already in the tabu list and entered into the tabu list. The tabu list now contains S_2 and S_6 (Fig. 8(c)). S_6 is then made the current node and search continues in this way.

5 ANT COLONY OPTIMIZATION TECHNIQUE

This meta-heuristic is based on real-world behaviour of ants (Colorni, Dorigo, & Maniezzo, 1992). When ants set out to search for food, they initially wander randomly. However, as the ants move from colony to food source and back, they deposit a special type of chemical called 'pheromone' on their trail which helps inform subsequent ants about the path to the food. Pheromones are volatile and evaporate with time. Thus, a short path, which takes less time to traverse, can accumulate more pheromone on it than a longer path which takes more time. Ants travel probabilistically on paths having more pheromone. By a positive feedback mechanism, the shorter paths thus accumulate more pheromone, and more and more ants travel on those paths, ultimately leading to all ants moving on the shortest path i.e., leading to convergence.

ACO Design Approach: We create a specialized graph for mapping the *RNC-MSC* assignment problem for the solution. Each ant starts its tour at a *RNC* wherefrom it can move to any of the *MSCs*. A *RNC* to *MSC* move by the ant results in the assignment of that *RNC* to that particular *MSC*.

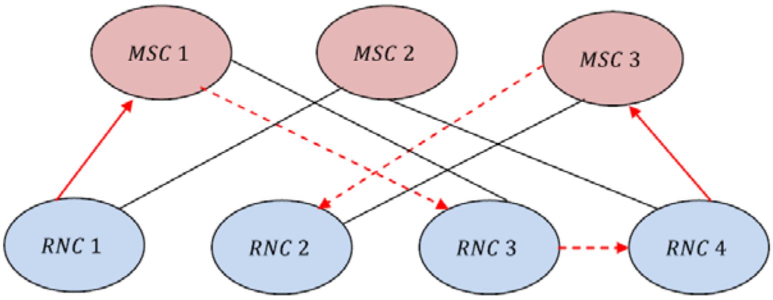


Figure. 9: Traversal of an ant

In Fig. 9, the black lines show the initial single home connections. The tour of the ant (shown by red lines) starts from *RNC 1*, and then successively continues through *MSC 1*, *RNC 3*, *RNC 4*, *MSC 3* and stops at *RNC 2*. The solid red lines show those parts of the tour which actually results in a *RNC* to *MSC* assignment while the dotted red lines depict those parts of the tour which are not actually significant for the assignment. Thus in this case the dual homed connections are *RNC 1* to *MSC 1* and *RNC 4* to *MSC 3*. Pheromone is deposited only on paths *RNC 1* to *MSC 1* and *RNC 4* to *MSC 3*. Subsequent ants travel similarly starting from *RNCs* 2, 3, 4, 1, 2, 3… and so on.

Transition rule: The probability of a move by the ant from RNC j to MSC k (path j-k) is given by the function:

$$P_{jk} = \frac{(\phi_{jk} h_{jk})}{\sum_m \phi_{jm} h_{jm}} \qquad \text{assuming } \alpha = \beta = 1$$

where,

ϕ_{jk} denotes the amount of pheromone on path (j-k)
α is a parameter to control the influence of ϕ_{jk}
h_{jk} is the desirability of path (j-k)
β is a parameter to control the influence of h_{jk}
m is the set of all *MSCs* for which the path (j-m) is feasible

The desirability of path j-k is taken as the handoff reduction possible by assigning *RNC* j to *MSC* k. The more the handoff reduction, the more desirable is the path. After an ant a has moved from *RNC* j to *MSC* k or has checked all *MSCs* for moves from *RNC* j and has not moved, we next bring the ant to another *RNC* to prepare for the next move. For this we create a list of all those *RNCs* which are yet to be unexplored and then randomly choose a *RNC* from that list, and then bring the ant back to the randomly chosen *RNC* for the next move.

Pheromone Update: This process of path assignments and bringing back ant to unexplored *RNC* continues until all *RNCs* have been traversed. Then we stop the process, analyze this tour to compute how much cost reduction we have achieved and then update the pheromone concentration on the paths by the following formula:

$\phi_{jk} = (1-\rho)\phi_{jk} + \rho \vartriangle \phi_{jk}$ where,

ϕ_{jk} is the amount of pheromone on a given path j-k

ρ is the rate of pheromone evaporation (the pheromone evaporation coefficient)

$\Delta\ \phi_{jk}$ is the amount of pheromone deposited by the ant a, typically given by $\Delta\ \phi_{jk}(a) = C_a$, if ant a had traversed the path j-k, 0, otherwise; where C_a is the cost reduction achieved by the a^{th} ant's tour

Parameters for Ant Colony Optimization: The various parameters for the ACO algorithm are:

- ρ , the pheromone evaporation co-efficient ($0 < \rho < 1$)
- α & β, the controlling weightage parameters for the path probability function. (Typically equal weightages are given for α and β equal to 1.0).
- I, the maximum number of iterations
- $\phi_{initial}$, the initial pheromone concentration on all paths

.

<u>*Algorithm ACO*</u>

```
Initialize: Set ITERATIONS = 0.
   Set MAX_IT = max. iteration count.
   Set Φinitial = initial pheromone concentration.
   Set ρ = pheromone evaporation co-efficient.
   Set best_state = single home state S.
   Set best_cost = Cost of S.
   Set initial pheromone concentration on all paths = Φinitial.
        Mark all RNCs and MSCs as 'unexplored'.
WHILE ITERATIONS ≤ MAX_IT
   Select an unexplored RNC as current RNC j.
   WHILE (true)
      WHILE there exists an unexplored MSC k
         Calculate Pjk = (φjk.hjk)/ ∑m( φjm .hjm)
         Generate a random number R in (0,1).
         IF R > Pjk
            Move ant from RNC i to MSC j.
            Assign RNC j to MSC k.
            Break.
         ELSE
            Mark MSC k as 'explored'.
         END IF
      END WHILE
      Mark the current RNC j as 'explored'.
      IF there exists another unexplored RNC j
         Bring ant to RNC j.
         Mark all MSCs as 'unexplored'.
      ELSE
         Break.
      END IF
   END WHILE
   Construct the solution S' from ant's tour.
   Set C' = cost of S'.
   IF C' < best_cost
      Set best_state = S'.
      Set best_cost = C'.
   END IF
   Update pheromone concentration on all paths.
   Set φjk = (1-ρ) φjk + ρ Δφjk
   ITERATIONS = ITERATIONS + 1
END WHILE
```

Now let us solve the example of Fig. 3. We are taking initial pheromone concentration on all paths to be 1.0 and pheromone evaporation co-efficient ρ = 0.5. α = β =1.0. From Table 3 it is found that the link cost of single home network = 17+17+10 = 44 and from Table 2 it is found that the total handoff cost of single home network = Sum of all handoffs = 162. Hence, the cost of single home network = link cost + handoff cost = 44+162 = 206. The initial single home network and initial pheromone concentration are shown in Fig. 10(a) and Fig. 10(b):

MSC / RNC	1	2	3
1	0	0	X
2	X	0	0
3	0	0	X
4	0	X	0
5	0	X	0
6	0	0	X

Figure. 10 (a) Initial state

MSC / RNC	1	2	3
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0
4	1.0	1.0	1.0
5	1.0	1.0	1.0
6	1.0	1.0	1.0

(b) Initial Pheromone Concentration

Iteration 1: The first *RNC* i.e., *RNC 1* is chosen as the starting position of the ant. From *RNC 1*, ant can move to *MSCs 1* and *2* resulting in handoff reductions of 0 and 46 (9+9+14+14), respectively, as it is evident from Table 2. The move from *RNC 1* to *MSC 1* is thus not feasible. The amortized link cost between *RNC 1* and *MSC 2* is 36.00 (from Table 3) which is less than the handoff reduction i.e., 46 (cost constraint), and capacity of *RNC 1* i.e., 480 is less than spare capacity of *MSC 2* i.e., 1713 (capacity constraint). Hence the move from *RNC 1* to *MSC 2* is feasible. Being the only feasible move, the move from *RNC 1* to *MSC 2* has a probability 1. So ant moves from *RNC 1* to *MSC 2* i.e., *RNC 1* is assigned to *MSC 2* and becomes dual-homed. Spare capacity of *MSC 2* is now reduced to 1713-480 = 1233. Now mark *RNC 1* as explored. The updated handoff costs and *MSC* spare capacity after this assignment are shown in Fig. 11(a) and Fig. 11(b).

RNC / RNC	1	2	3	4	5	6
1	-	-	-	-	-	-
2	-	-	19	4	-	24
3	-	19	-	5	6	-
4	-	4	5	-	-	-
5	-	-	6	-	-	-
6	-	24	-	-	-	-

Figure 11(a). Handoff Costs

MSC	1	2	3
Capacity	1389	1233	2138

(b) Spare capacity of MSCs

The current partial solution (Fig. 11(c)) is (Ex signifies that a *RNC* has been explored):

MSC / RNC	1	2	3
1 (Ex)	0	1	X
2	X	0	0
3	0	0	X
4	0	X	0
5	0	X	0
6	0	0	X

Figure 11 (c). Partial Solution

From *MSC 2*, we now bring the ant back to one of the unexplored *RNCs*, say *RNC 3*. From *RNC 3*, ant can move to *MSCs 1* and *2* resulting in handoff reductions of 38 (19+19) and 22 (5+5+6+6), respectively, as it is evident from Fig. 5(a). Also, the amortized link cost between *RNC 3* and *MSC 1* is 20 (from Table 3) which is less than the handoff reduction i.e., 38, and capacity of *RNC 3* (i.e., 318) is less than spare capacity of *MSC 1* (i.e. 1389). So the move from *RNC 3* to *MSC 1* is feasible. The amortized link cost between *RNC 3* and *MSC 2* is 26.00 (from Table 3) that is greater than the handoff reduction i.e., 22. So move from *RNC 3* to *MSC 2* is not feasible.

The move from *RNC 3* to *MSC 1* has probability 1 being the only feasible move. So ant moves from *RNC 3* to *MSC 1* i.e. *RNC 3* is now dual-homed and assigned to *MSC 1*. Spare capacity of *MSC 1* is now reduced to 1071 (1389-318=1071). *RNC 3* is marked explored. The updated handoff costs and *MSC* spare capacities after this assignment are as follows (Fig 12(a) to Fg. 12(c))

RNC \ RNC	1	2	3	4	5	6
1	-	-	-	-	-	-
2	-	-	-	4	-	24
3	-	-	-	5	6	-
4	-	4	5	-	-	-
5	-	-	6	-	-	-
6	-	24	-	-	-	-

Figure 12(a). Handoff Costs

MSC	1	2	3
Capacity	1071	1233	2138

(b) Spare capacity of MSCs

The current partial solution is:

RNC \ MSC	1	2	3
1 (Ex)	0	1	X
2	X	0	0
3 (Ex)	1	0	X
4	0	X	0
5	0	X	0
6	0	0	X

Figure 12 (c). Partial Solution

Similarly the ant will move from other unexplored *RNCs* to *MSCs* to complete the first iteration. Cost of this solution = Link cost + Handoff cost = 136+(4+5+6+4+5+6)=166

Hence cost reduction=Cost of Single home network – Cost of Dual home network = 206 – 166 = 40

After first iteration, pheromone concentrations on all paths are updated. Pheromone is deposited on paths *RNC 1* to *MSC 2*, *RNC 2* to *MSC 3* and *RNC 3* to *MSC 1*. Amount of pheromone deposited is $\rho C_a = 0.5*40 = 20$. Previous pheromone on all paths become $(1-\rho)*1.0=0.5$.

Here the best solution (Fig. 13(a)) and pheromone concentration (Fig. 13(b)) found after the first iteration are shown below:

RNC \ MSC	1	2	3
1 (Ex)	0	1	X
2 (Ex)	X	0	1
3 (Ex)	1	0	X
4 (Ex)	0	X	0
5 (Ex)	0	X	0
6 (Ex)	0	0	X

Figure 13 (a). Partial Solution

RNC \ MSC	1	2	3
1	0.5	20.5	0.5
2	0.5	0.5	20.5
3	20.5	0.5	0.5
4	0.5	0.5	0.5
5	0.5	0.5	0.5
6	0.5	0.5	0.5

(b) Pheromone concentration

Subsequent iterations: In iterations 2, 3, 4, 5, 6 and so on the starting *RNC* of ant is taken 2, 3, 4, 5, 6, 1, 2, 3… and in this way the iterations continue. Costs of all solutions are calculated and checked if they are better than the previously found best solution. However, we see that the solution produced in iteration 2 (Fig. 14) is the best solution. Pheromone concentrations on the good paths like *RNC 2* to *MSC 3* and *RNC 5* to *MSC 3* increase over time as more and more ants follow these paths.

MSC / RNC	1	2	3
1 (Ex)	0	0	X
2 (Ex)	X	0	1
3 (Ex)	0	0	X
4 (Ex)	0	X	0
5 (Ex)	0	X	1
6 (Ex)	0	0	X

Figure 14. Solution in iteration 2

7. RESULTS AND DISCUSSIONS

We divide a rectangular area into multiple hexagonal *NodeBs* created using a non orthogonal Cartesian system inclined at 60^0. Each *NodeB* has exactly six neighbouring *NodeBs* except the boundary *NodeBs* that have less than six neighbouring *NodeBs*. A specified set of *RNCs* and *MSCs* are placed randomly in some *NodeBs* such that an *MSC* is co-located with an *RNC*. Subsequently, *NodeBs* are assigned to *RNCs,* and *RNCs* are assigned to *MSCs* using nearest neighbour distance.

After the creation of the above synthetic UMTS single home network architecture, voice handoff costs for neighbouring *NodeBs* are generated from direction based mobility of the users (Sadhukhan etl, 2010). Amortized cost for a link from NodeB to *RNC* and *RNC* to *MSC* is taken proportional to the physical distance. Complex voice handoff costs between *RNCs* are calculated based on handoff cost between *NodeBs*. Spare capacities of *MSCs* and capacities of *RNCs* of the network are generated using capacities of *NodeBs* which are generated from a uniform distribution.

Algorithms SA, TS, ACO are run for 100 instances, and average solution cost is recorded against the triplet (*MSCs*, *RNCs* and *NodeBs*). Fig. 15 compares the algorithms with respect to average solution costs. The initial cost is taken as the cost of the single home network. For each dual-homed architecture solution, the reduction in handoff cost for the network is calculated, considering the cost of amortized link, subject to the capacity constraints of the *MSCs* and the *RNCs*. For TS, we have taken the tabu list size is equal to 10 and maximum number of iteration is equal to 1000. Average solution costs obtained by ACO are better than SA and TS with increase in problem size.

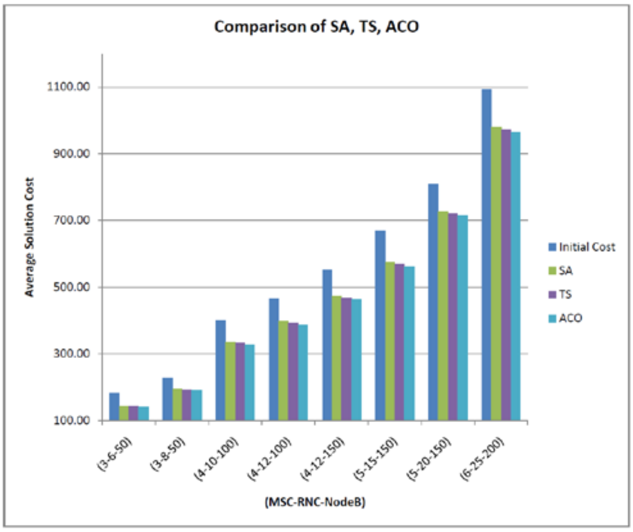


Figure 15. Dual-home solution costs

8. CONCLUSION

We have formulated the selective dual homing scenario of *RNCs* for post-deployment tuning of an existing single home UMTS networks as an ILP problem. Dual-homing of *RNCs* reduces the handoff cost but additional capacities are required at the *MSCs/SGSNs* in the network. The problem is difficult to solve in polynomial time as it falls into NP-Complete category (Garey & Johnson 1979). We have then mapped the dual-homing problem into a state space and solved the problem using SA,TS and ACO (meta-heuristic) algorithms. It is found that ACO technique is capable of finding good quality solutions which are better than those obtained by SA or TS. The technique will be helpful to the operators because the proposed solution can be implemented with minor changes in the protocol stack of the existing standards.

References

[1] Amzallag D., Naor J., Raz D., (2007) Algorithmic aspects of access networks design in B3g/4g cellular networks, Proc. *IEEE INFOCOM,* pp:991-999.

[2] Bhattacharjee P. S., Saha D., Mukherjee A.,(2004) An approach for location area planning in a Personal Communication Services Network (PCSN), *IEEE Trans on Wireless Communication*

[3] Colorni A., Dorigo M., and Maniezzo V., (1992) An Investigation of Some Properties of an Ant Algorithm, *International Conference on Evolutionary Computation - The Second Conference on Parallel Problem Solving from Nature,* North-Holland, Amsterdam, The Netherlands, pp 509-520

[4] Colorni A., Dorigo M., and Maniezzo V., (1992) Distributed Optimization by Ant Colonies, Towards a Practice of Autonomous Systems: *Proceedings of the First European Conference on Artificial Life,*. MIT Press, Cambridge, MA, pp 134-142.

[5] Clayirci E., Akyildiz I. F., (2002) User mobility pattern scheme for location update and paging in wireless systems, *Mobile Computing, IEEE Transactions,Vol.1 , Issue: 3* , pp:236 – 247,.

[6] Demrikol I., Ersoy C., Caglayan M. U., Delic H.,(2001) Location area planning in cellular networks using Simulated Annealing, *Proc. IEEE INFOCOM*,.

[7] Demestichas P. et. al. (1999), *Control* of the location update and paging signaling load in cellular systems by means of planning tools, *Proc. Fall. IEEE VTC, Vol. 4,* pp: 2119 -2123.

[8] Diallo M. M., Pierre S., Beaubrun R. (2006) An Approach for Cell Assignment in UMTS Networks *VTC,* pp: 1-5,.

[9] Din D. R., Tseng S. S., (2002) A genetic algorithm for solving dual-homing cell assignment problem of the two-level wireless ATM network, *Computer Communications, 25* , pp 1536–1547

[10] Garey M. R. , Johnson D. S. (1979). *Computers and Intractability: A guide to the theory of NP-Completeness*, New York: W. H. Freeman.

[11] Mandal S., Saha D., and Mahanti A.(2004) Heuristic search techniques for cell to switch assignment in location area planning for cellular networks, *Proc. IEEE International Conference on Communications (ICC)*, pp: 4307-4311.

[12] Mandal S., Saha D., and Mahanti A.(2005) A technique to support dynamic pricing strategy for differented cellular mobile services, *Proc. IEEE Globecom*., pp: 3388-3392

[13] Merchant A., Sengupta B.,(1995) Assignment of cells to switches in PCS networks, *IEEE/ACM Trans. on Networking, Vol.3*,pp:521–526

[14] Pierre S., Houeto F, (2002) Assigning cells to switches in cellular mobile networks using Taboo Search, IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 32 , Issue: 3, pp:351 – 356

[15] Quintero A., Pierre S. A. (2002) Memetic algorithm for assigning cells to switches in cellular mobile networks, IEEE Communications Letters, Vol. 6, Issue:11, pp: 484 –486

[16] Sadhukhan S.. K., Mandal S., and Saha D.,(2007) A heuristic technique for solving dual-homing assignment problem of 2.5G cellular networks, *Proc. IEEE International Conference on Computing: Theory and Applications*, pp: 66-71

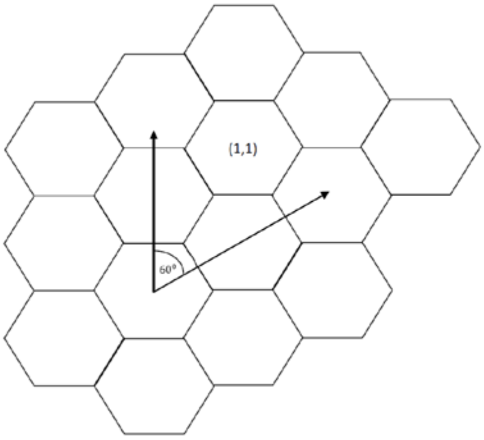
[17] Sadhukhan S.. K., Mandal S., Bhaumik P, and Saha D.,(2010) A Novel Direction-Based Diurnal Mobility Model for Handoff Estimation in Cellular Networks, *Proc. IEEE INDICON.* pp 1-5.

[18] Saha D., Mukherjee A., Bhattacharya P. S.,(2000) A simple heuristic for assignment of cells to switches in a PCS network, *Proc. Wireless Personal Communications*, pp: 209- 224.

[19] Saraydar C. U., Kelly O.E., Rose C. (2000) One-dimensional location area design*, IEEE Transactions on Vehicular Technology, Vol. 49, Issue: 5*, pp: 1626 – 1632

Appendix:

A. Non-orthogonal Cartesian System:



The axes are inclined at 60^0. The distance between two cells (i,j) and (k,l) is

$$d_{(i,j)(k,l)} = \sqrt{(i-k)^2 + (j-l)^2 + (i-k)(j-l)}$$
$$if\ d_{(i,j)(k,l)} \leq 1\ then\ cell\ (i,j) and\ cell\ (k,l)\ are\ neighbours$$
$$if\ (p-1) < d_{(i,j)(k,l)} \leq p\ \ then\ cell\ (i,j) and\ cell\ (k,l)\ are\ p - distance\ neighbours$$

B. Calculation of amortized cost of cable:

Let the *NPV* of the cable is $\$x$ per unit length and its life is y time unit. The total number of handoff during the life of the cable is $y * MHR$, where MHR is the mean handoff rate (i.e, expected number of handoff per time unit). So the amortized cost per unit of cable per handoff is $\frac{\$x}{y*MHR}$. If the total cable required is l unit and n be the number of handoffs per time unit then the amortized cost of cable per time unit is $\frac{\$lnx}{y*MHR}$.