Cell Planning with Capacity Expansion in Mobile Communications: A Tabu Search Approach

Chae Y. Lee and Hyon G. Kang
Department of Industrial Engineering, KAIST
373-1, Kusung Dong, Taejon 305-701, Korea
cylee@heuristic.kaist.ac.kr

Abstract

Cell planning problem with capacity expansion is examined in wireless communications. The problem decides the location and capacity of each new base station to cover expanded and increased traffic demand. The objective is to minimize the cost of new base stations. The coverage by the new and existing base stations is constrained to satisfy a proper portion of traffic demands. The received signal power at the base station also has to meet the receiver sensitivity. The cell planning is formulated as an integer linear programming problem and solved by a Tabu Search algorithm.

In the tabu search intensification by add and drop move is implemented by short-term memory embodied by two tabu lists. Diversification is designed to investigate proper capacities of new base stations and to restart the tabu search from new base station locations.

Computational results show that the proposed tabu search is highly effective. 10% cost reduction is obtained by the diversification strategies. The gap from the optimal solutions is approximately 1~5 % in problems that can be handled in appropriate time limits. The proposed tabu search also outperforms the parallel genetic algorithm. The cost reduction by the tabu search approaches 10~20 % in problems with 2,500 traffic demand areas in CDMA.

1. Introduction

In cell planning for mobile communication systems we need to consider the traffic demand to cover a specific region, availability of base station sites, available channel capacity at each base station, and the service quality at various potential traffic demand areas (TDAs). Selection of good base station sites and channels will result in acceptable coverage performance at base stations both in coverage area and in signal quality. The problem discussed in this paper is to determine the number of base stations and location and capacity of each base station to cover increased traffic demand. The coverage has to satisfy a certain level of total traffic demand and the received signal strength.

Optimal location of transmitters for micro-cellular system is studied by Sherali et al. [5]. The path loss at each subarea is represented as a function of the base station location. A nonlinear programming problem is presented which minimizes a measure of weighted path-losses. Several nonlinear optimization algorithms are investigated to solve the problem. In [6] the radio coverage optimization problem is converted to a maximum independent set problem. The objective is to achieve a large coverage of traffic demand areas with a small number of base stations. A simulation method is employed to examine the relationship between the number of base stations and the relative coverage of traffic demand areas.

Tutschku [12] proposed an automatic cellular network design algorithm without considering the capacity of transmitters. The network design problem is converted to a maximal covering location problem by using demand node concept. The location of transmitter is optimized by minimizing co-channel interference. He [9] also proposed a greedy heuristic to solve the maximal coverage location problem of transmitters. The heuristic takes into account all the RF design objectives as well as the capacity and the network deployment constraints.

Ye et al. [10] solve the cell planning in a CDMA network. They maximize the cell coverage for given traffic and find optimal location configuration by considering nodes covered via soft handoff by two or three base stations.

A genetic algorithm approach is presented by Calegari et al. [7, 8]. The selection of base stations is represented in a bit string. Selection based on fitness value, one-point crossover and mutation operators are employed. The fitness value combines two goals of maximizing the cover rate and minimizing the number of transmitters. To speed up the procedure a parallel genetic algorithm is implemented by using island model. Their computational results show that the solution quality is significantly influenced by the number of islands.

Most of the research in the optimization of the radio coverage in cellular system is restricted to the selection of base station locations. Base stations are considered to start service at the same time. In this paper we consider two types of base stations. Some existing base stations are currently in service for a specified region. The increased traffic demand in the region requires capacity expansion with additional base stations. We thus need to determine the location and the capacity of each new base stations.

The remainder of this paper is organized as follows. In Section 2, we discuss the capacity of a base station in various multiple access technologies and the potential service area of a base station. We also provide a mathematical model for the cell planning problem. Section 3 presents a tabu search procedure to solve the problem. Construction of initial solutions, intensification, and diversification strategies are examined. The performance of various tabu operators and the efficiency of the proposed tabu search procedure are presented and compared with a genetic algorithm and an excellent integer programming algorithm CPLEX [16] in Section 4. Finally, we conclude the paper in Section 5.

2. Cell Planning with Capacity Expansion

The cell planning we are interested in is to decide the location and capacity of each new base station to cover increased traffic demand. It causes cell splitting in urban area and requires new cell sites in suburban area. Thus, some of the existing TDAs are served by the new base stations due to the cell splitting. Here, we consider the increased traffic demand, the capacity of each new base station to locate and the coverage of the base station.

2.1 Capacity of a Base Station

The capacity of a base station has experienced a great improvement due to the digital modulation, multiple access schemes and other technological development. Advanced Mobile Phone Service (AMPS) employs the frequency division multiple access (FDMA). It utilizes 50 MHz of spectrum in the 800 MHz band. Each channel band of AMPS is 30 kHz. Assuming two competing carriers in the market, a carrier has 416 channels [1]. Twenty one channels are used for control and the rest for traffic channels.

When we assume 7-cell reuse pattern, the number of traffic channels available in a cell becomes 56. It corresponds to a base station capacity of 46 Erlangs (Erlang B) when the blocking probability is 2%.

Global System for Mobile (GSM) is a typical standard of the time division multiple access (TDMA). GSM utilizes two bands of 25 MHz for forward and reverse links. The frequency band is divided into 200 kHz wide channels called ARFCNs (Absolute Radio Frequency Channel Numbers). Each channel is time shared between as many as eight subscribers using TDMA. Since each radio channel consists of 8 time slots, there are thus a total of 1000 traffic channels within GSM. In practical implementations, a guard band of 100 kHz is provided at the upper and lower ends of GSM spectrum, and only 124 channels are implemented [1]. By assuming two companies as in AMPS, each carrier has 62 channels. Assuming the 4-cell reuse pattern, a BS can use 120 time slots. This corresponds to 107.4 Erlangs when the blocking probability is 2 %.

Interrim Standard 95 (IS-95) [14] is the standard of the code division multiple access (CDMA) and offers some advantages over TDMA and FDMA. Each CDMA channel which is called the frequency assign (FA) occupies 1.25 MHz of spectrum. Assuming 25 MHz for both links, the number of available channels becomes 10 FAs. Since the reuse pattern in IS-95 is one, one cell use up to 10 FAs. However, normally 3-4 FAs are used practically in a cell. Assuming 4 FAs and 36 traffic channels [15] per 1 FA, 144 traffic channels are available in a cell. If the system uses 120-degree directional antenna, then the capacity is increased approximately 2.5 times, which corresponds to 360 traffic channels. Assuming 2% blocking probability, the capacity becomes 345.7 Erlangs at each base station.

2.2 Potential Service Area of a Base Station

Potential service area of a base station represents the TDAs that can be served with sufficient quality by the base station. In this study, we are interested in a general propagation path-loss formula in a general mobile radio environment. By using the path loss model the received signal power can be estimated as a function of transmitted power, distance between the transmitter and receiver, processing gains, and antenna heights. If we ignore fading, the following propagation model [2] may well be used in computing potential service areas. In the model the path loss exponent is assumed to be four.

$$P_r = P_t + G_r + G_t + 20 \log h_r + 20 \log h_t + L - 40 \log r$$

P_r: received power
P_t: transmitted power
G_r, G_t: processing gains of receiver and transmitter
h_r, h_t: antenna heights of receiver and transmitter
L: buffer for fading
r: distance between transmitter and receiver

From the above model, the radius of a cell site can be computed for a given receiver sensitivity. In other words, the TDAs that can be covered by a base station are determined by comparing the received power and the receiver sensitivity. Since we are interested in the location and the capacity of each new base station, we consider the received power at a base station from TDAs. We also assume that co-channel and adjacent channel interferences are negligible in the uplink analysis.

2.3 Problem Formulation

Suppose that mobile users are distributed over a designated region composed of N TDAs. Each traffic demand area TDA_i has a traffic demand d_i, $i=1, ..., N$.

Assume that the region has K_1 existing base stations each of which is denoted by BS_k, $k=1, \ldots, K_1$. To satisfy increased traffic demands K_2 candidate cell sites are considered. It is assumed that the potential location of each candidate base station BS_k, $k=K_1+1, \ldots, K_1+ K_2$ is known. Let c_k and M_k be respectively the cost and capacity of each new base station k. Note that the cost and capacity are mainly dependent on the way of multiple access, number of user channels, and sectorization. We assume that the base station cost c_k is linear to the capacity M_k.

To formulate the problem, we introduce two types of variables. Let y_{ik} be the wireless connection between TDA_i and BS_k such that

$$y_{ik} = \begin{cases} 1, & \text{if } \mathrm{TDA}_i \text{ is covered by } \mathrm{BS}_k \\ 0, & \text{otherwise} \end{cases}$$

Also, let z_k be the selection variable of BS_k, $k=1, \ldots, K_1+K_2$, such that

$$z_k = \begin{cases} 1, & \text{if } \mathrm{BS}_k \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

The objective of our cell planning problem is to minimize the cost of newly installed base stations. Thus the objective function of the problem is

$$\text{Minimize} \quad \sum_{k=K_1+1}^{K_1+K_2} c_k z_k$$

Now, TDA_i can be covered either by a new base station or by an existing base station. It can be covered by a new base station only if it has been selected. Thus, we have

$$y_{ik} \le z_k \qquad \text{for } i=1, \dots, N \text{ and } k=1, \dots, K_1+K_2$$

In the constraint above, the existing base stations are assumed selected, i.e.,

$$z_k = 1 \qquad \text{for } k=1, \dots, K_1.$$

Note that due to the increased traffic demand from TDAs and cell splitting the coverage of existing base station may change.

In cell planning it is important to satisfy the coverage limit of the total traffic demand in the region. Specifically, the problem is handled with the minimum portion that has to be covered by a wireless carrier in the specified region. The minimum portion is given either by the area or by the traffic demand. In this study, we employ the minimum portion of traffic demand. In other words, at least α ($0 \le \alpha \le 1$) of the total expected traffic demand has to be covered by a set of base stations in the region. Thus, we have the following constraint:

$$\sum_{k=1}^{K_1+K_2} \sum_{i=1}^{N} d_i y_{ik} \ge \alpha \sum_{i=1}^{N} d_i$$

Now, consider TDAs that are covered by a base station. Clearly, the total traffic demand covered by the base station cannot exceed the capacity. The capacity constraint is given as follows:

$$\sum_{i=1}^{N} d_i y_{ik} \le M_k z_k \quad \text{for } k=1, \dots, K_1+K_2$$

Finally, we take into account the received signal power strength at BS_k which is transmitted from TDAs. Due to the path-loss of radio propagation, if the received power from a TDA does not exceed the receiver sensitivity at BS_k, then the TDA cannot be covered by the BS_k. Let $P(i,k)$ denotes the received power at BS_k which is transmitted from the center of TDA_i. Also let QoS be the minimum required power level at each base station. Then we have the following path-loss constraint:

$$P(i,k) \geq QoS \times y_{ik} \qquad \text{for } i=1, ..., N \text{ and } k=1, ... , K_1+K_2$$

From the above, the cell planning problem can be formulated as the following linear integer problem.

$$\text{Minimize} \qquad \sum_{k=K_1+1}^{K_1+K_2} c_k z_k$$

$$\text{s.t.} \quad z_k = 1 \qquad \text{for } k=1, ... , K_1 \tag{1}$$

$$y_{ik} \leq z_k \qquad \text{for } i=1, ..., N \text{ and } k=1, ... , K_1+K_2 \tag{2}$$

$$\sum_{k=1}^{K_1+K_2} \sum_{i=1}^{N} d_i y_{ik} \geq \alpha \sum_{i=1}^{N} d_i \tag{3}$$

$$\sum_{i=1}^{N} d_i y_{ik} \leq M_k z_k \qquad \text{for } k=1, ... , K_1+K_2 \tag{4}$$

$$P(i,k) \geq QoS \times y_{ik} \qquad \text{for } i=1, ..., N \text{ and } k=1, ... , K_1+K_2 \tag{5}$$

$$y_{ik}, z_k \in \{0,1\}$$

The above cell planning problem is equivalent to the set covering problem [6] when the coverage factor $\alpha=1.0$. The problem seeks a set of base stations that covers the traffic demand areas in a specified region. When $\alpha=1.0$, all TDAs are covered by the base stations. Note that the set covering which is a special case of the above cell planning problem is a well-known NP-complete problem [11]. We thus propose a tabu search procedure to solve the problem.

3. Tabu Search for the Cell Planning Problem

Tabu Search incorporates three general components [4]: 1) short-term and long-term memory structures, 2) tabu restrictions and aspiration criteria, and 3) intensification and diversification strategies. Intensification strategies utilize short-term memory function to integrate features or environments of good solutions as a basis for generating still better solutions. Such strategies focus on aggressively searching for a best solution within a strategically restricted region. Diversification strategies, which typically employ a long-term memory function, redirect the search to unvisited regions of the solution space.

In our case the short-term memory is implemented by means of tabu lists and aspiration criteria. A tabu list records attributes of solutions (or moves) to forbid moves that lead to solutions that share attributes in common with solutions recently visited. A move remains tabu during a certain periods (or tabu size) to help aggressive search for better solutions. Aspiration criteria enable the tabu status of a move to be overridden, thus allowing the move to be performed, provided the move is good enough.

3.1 Initial Base Stations and Covering

To obtain an initial solution two strategies are adopted; "All Candidate Base Stations" and "Random Feasible Base Stations". In the method of All Candidate Base Stations, base stations are located at every candidate cell site. For feasibility, sufficiently many candidate cell sites are initially prepared to satisfy the capacity and path loss constraints. Each TDA is assigned to the nearest base station as far as the capacity is satisfied. In the method of Random Feasible Base Stations, base stations are selected in nonincreasing order of the capacity from candidate cell sites. Each TDA is covered by the nearest base station within the capacity limit. The selection of base stations is terminated when all TDAs are covered.

3.2 Intensification with Short-term Memory Function

We first define two types of moves. They are "Drop move" and "Add move". Drop move makes a currently active base station inactive. In other words, a base station which is dropped can no more cover TDAs until it is selected again. This Drop move is implemented by moving the base station from Active_List to Candidate_List. Add move is the opposite of Drop move. Add move selects a base station to cover TDAs. Thus the base station is moved from the Candidate_List to the Active_List.

The short-term memory function, embodied in the two tabu lists, is implemented as an array *Tabu_Time(k)* which records the earliest iteration that BS_k is allowed to move: either to Candidate_List or to Active_List. To prevent moving back to previously investigated solutions, we define two different tabu times T_1 and T_2 as the time that must elapse before a base station is permitted to move from Candidate_List and Active_List respectively. Both tabu times are measured in number of iterations. If by an

Add move BS_k is moved from Candidate_List to Active_List, then the *Tabu_Time(k)* = *Current_Iteration* + T_1. If by a Drop move BS_k is moved from Active_List to Candidate_List, then the *Tabu_Time(k)* = *Current_Iteration* + T_2. Thus BS_k is tabu if *Current_Iteration* ≤ *Tabu_Time(k)*. The choice of tabu times is important to the Tabu search algorithm. We will empirically select the tabu times (T_1, T_2) which lead to reasonably good solutions.

In a Drop move a base station is selected to drop from the Active_List by considering base station cost c_k and normalized residual capacity *NRC(k)*. The residual capacity is normalized such that the total capacity is equal to the base station cost. A base station whose sum of the two costs is the maximum is selected to drop.

In an Add move a base station is selected from Candidate_List by comparing the coverage and the base station cost. A base station that maximizes the number of TDAs covered with minimum cost is selected to add.

The above two moves are explained in Step 3 and 6 of the Procedure Tabu Search. Aspiration by default is applied when the coverage of TDAs is infeasible due to a Drop move. In this case the tabu status is overridden and the base station with the least *Tabu_Time(k)* - *Current_Iteration* from Candidate_List is added to the solution.

3.3 Reassignment of TDAs

In the intensification process reassignment is performed after each move. After a Drop move each TDA which was covered by the dropped base station need to be covered by another base station. Each TDA is now covered by a base station in the Active_List such that the received signal from the base station is the strongest. The same is true after an Add move. When a base station is added to satisfy feasibility of the covering problem, TDAs are selected which will be covered by the newly added base station. Each TDA whose received power from the added base station is stronger than that from the current base station is assigned to the new base station as far as the capacity is satisfied. These two reassignments are explained in Step 4 and 7 in the tabu search procedure.

3.4 Diversification with Long-term Memory Function

The diversification strategy is helpful to explore new unvisited regions of the solution space. It enables the search process to escape from local optimality. In our procedure the diversification is performed when no solution improvement results consecutively for *Nmax* iterations in the intensification process. Also, a path is defined as the iterations between any two consecutive diversifications. Two diversification strategies are employed: capacity diversification and coverage diversification.

The capacity diversification is the process of determining an appropriate capacity at each new base station. It is implemented by examining the best solution in a path. When a BS_k has unused residual capacity *RC(k)* which is greater than the capacity variation unit ΔM, then the capacity is reduced by ΔM. When the capacity of a base station is fully employed to cover TDAs, then the capacity is increased by one unit, i.e., ΔM.

The coverage diversification is performed by using *Active_Freq(k)* and *Move_Freq(k)*. *Active_Freq(k)* represents the number of iterations BS_k was in solution in the previous path, while *Move_Freq(k)* represents total number of Add and Drop Moves performed on BS_k. In the coverage diversification the preference is given to the base stations with low *Active_Freq(k)* and *Move_Freq(k)*. Base stations with relatively lower *Active_Freq(k)* + *Move_Freq(k)* are selected until all required traffic demands are covered. This diversification strategy has the effect of restarting the tabu search from a solution that is far away from the solutions obtained in the intensification procedure.

The above two diversification strategies described in Step 9 of Procedure Tabu Search are designed to investigate proper capacities of base stations and better coverage of the TDAs.

Procedure Tabu Search

Step 1. Initial Solution Method

Obtain Initial feasible solution by one of the following two methods.

Method 1: All Candidate Base Stations

Method 2: Random Feasible Base Stations

Step 2. Starting Set up

Current_Iteration :=0, NoImprove := 0, MaxNoImprove := Nmax, Diversification := 0;

MaxDiversification := Dmax, Tabu_Time(k) := -1, Move_Freq(k) := 0, Active_Freq(k) :=0;

Best_Solution_Value := ∞*, Record_Solution_Value :=* ∞*;*

T_1 := Tabu Time in Active_List, T_2 := Tabu Time in Candidate_List;

N := Number of TDAs, d_i := Traffic Demand of TDA_i, α := Coverage Factor;

Total_Traffic_Demand = $\sum_{i=1}^{N} d_i$;

M_k := Capacity of BS_k, c_k := Construction Cost of BS_k, *RC(k)* := Residual Capacity of BS_k;

NRC(k) := Normalized residual capacity of BS_k, i.e., $NRC(k) = \frac{RC(k)}{M_k} \times c_k$;

P(i,k) := Received power at BS_k which is transmitted from the center of TDA_i;

Step 3. Drop Move

(a) Evaluate the current solution of BSs in Active_List.

$Eval_1(k) = c_k + NRC(k).$

(b) Do in nonincreasing order of the evaluation value $Eval_1(k)$ until a BS is selected.

For BS_k

If *Tabu_Time(k) < Current_Iteration,* select the BS_k.

Else if the BS_k satisfies the default aspiration criterion then select the BS_k.

Otherwise repeat for the next BS in the order.

(c) Move the selected BS_k from Active_List to Candidate_List.

Set *Tabu_Time(k) := Current_Iteration +* T_2, *Move_Freq(k) := Move_Freq(k)+1*.

(d) Go to Step 4.

Step 4. Reassignment after Drop Move

(a) For each TDA that is not covered

(i) Find BSs that can cover the TDA in Active_List.

(ii) Sort BSs in nonincreasing order of the received power *P(i,k)*.

(iii) If $d_i < RC(k)$, TDA_i is covered by BS_k (i.e. $y_{ik} := 1$) and set $RC(k) = RC(k) - d_i$.

Otherwise repeat for the next BS in the order.

(b) Go to Step 5.

Step 5. Feasibility Check

(a) Compute Covered_Traffic = $\sum_{k=1}^{K_1+K_2} (M_k - RC(k))$.

(b) If *Covered_Traffic* $< \alpha \times$ *Total_Traffic_Demand*, go to Step 6.

Otherwise go to Step 8.

Step 6. Add Move

(a) Evaluate the current solution of BSs in Candidate_List.

$Eval_2(k) = Number\ of\ TDAs\ which\ can\ be\ covered\ by\ BS_k / c_k$

$T\_Asp(k) = \sum d_i / M_k$ for TDA_i which can be covered by BS_k

(b) Do in nonincreasing order of the evaluation value $Eval_2(k)$ until a BS is selected.

For BS_k,

If *Tabu_Time(k) < Current_Iteration*, select the BS_k.

Else if *T_Asp(k) = 1*, select the BS_k.

Otherwise repeat for the next BS in the order.

(c) Move the selected BS_k from Candidate_List to Active_List.

Set *Tabu_Time(k) := Current_Iteration +* T_1, *Move_Freq(k) := Move_Freq(k) + 1*.

(d) Go to Step 7.

Step 7. Reassignment after Add Move

(a) For a new BS which is moved into Active_List

If *P(i,new)* is the largest among *P(i,k)* for each TDA_i and BS_k in Active_List and $RC(new) \geq d_i$ then

set $y_{i,new} := 1$ (TDA_i is covered by new BS), $RC(new) := RC(new) - d_i$,

$y_{i,old} := 0$ (TDA_i is not covered by old BS), and $RC(old) := RC(old) + d_i$.

(b) Go to Step 4.

Step 8. Updating Solution

(a) Compute the *New_Solution_Value* = $\sum_{k=1}^{K_1+K_2} c_k z_k$.

(b) Set *Active_Freq(k) := Active_Freq(k) + 1* for BSs in Active_List.

(c) If *New_Solution_Value < Best_Solution_Value*,

set *Best_Solution_Value := New_Solution_Value*, *NoImprove := 0* and go to Step 3.

Otherwise *NoImprove := NoImprove + 1*;

(d) If *NoImprove < Nmax*, then go to Step 3.

(e) If *NoImprove = Nmax* and *Best_Solution_Value < Record_Solution_Value*,

set *Record_Solution_Value := Best_Solution_Value* and *Best_Solution_Value :=* ∞.

(f) If *Diversification < Dmax*, go to Step 9.

Otherwise **STOP**.

Step 9. Diversification

(a) Two diversification strategies are performed.

(i) Capacity Diversification

ΔM := unit capacity variation in a BS, Δc := unit cost variation in a BS;

v_k := the number of units increased or decreased ($0 \leq v_k \leq V$);

Consider *RC(k)* of BSs in the Best_Solution of the previous path.

If $RC(k) > \Delta M$ and $v_k > 0$, $M_k := M_k - \Delta M$ and $c_k = c_k - \Delta c$.

If $RC(k) = 0$ and $v_k < V$, $M_k := M_k + \Delta M$ and $c_k = c_k + \Delta c$.

(ii) Coverage Diversification

Sort all new BSs in nondecreasing order of *Move_Freq(k) + Active_Freq(k)* and store the BSs in Diverse_List.

Set *Active_List* $:= \phi$.

While the solution infeasible, move BSs one by one from Diverse_List to Active_List.

(b) Set *NoImprove* $:= 0$, *Move_Freq(k)* $:= 0$, *Active_Freq(k)* $:=0$ for $k = 1, 2, ..., K_1+K_2$.

(c) Set *Diversification* := *Diversification* + *1*.

(d) Go to Step 3.

In the above procedure, notice that the main computational burden occurs in Step 4 when TDAs are reassigned after a Drop move. For each TDA the procedure sorts K_2 base stations in nonincreasing order of received power. This procedure requires $O(K_2 \log K_2)$ steps. Since the ordering is performed for TDAs previously covered by the dropped base station, the overall complexity reduces to $O(NK_2 \log K_2)$ in the worst case.

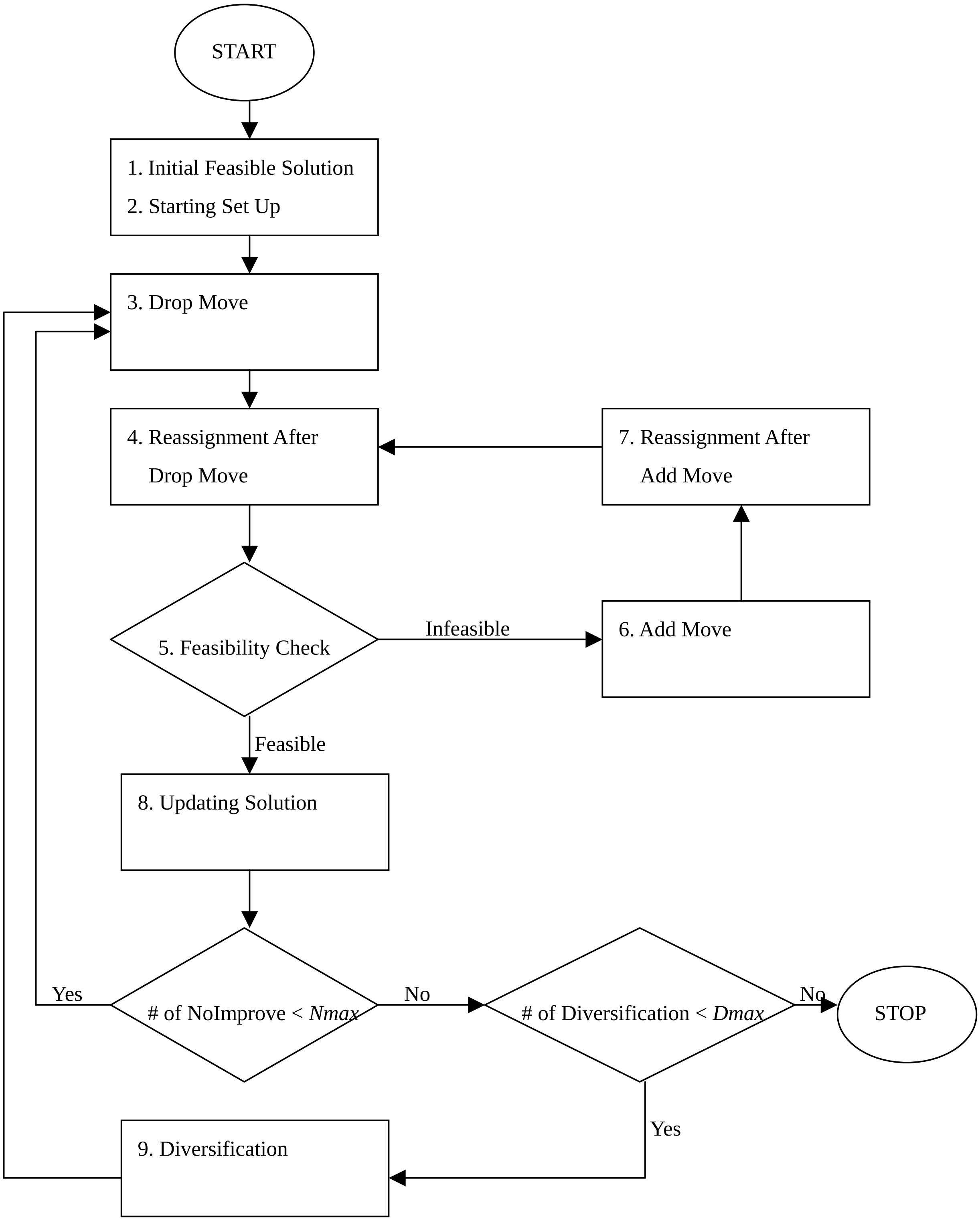

Figure 1. Proposed Tabu Search procedure for the cell planning

4. Computational Results

In this section, we test the efficiency of the proposed algorithm for the cell planning with capacity expansion. The algorithm described in the previous section was implemented in Visual C++ (Version 5.02), and run on a 200 MHz Intel Pentium based personal computer with 64 Mbyte of memory under Windows 95. We assume that all TDAs are square as shown in Figure 2 and 6. The traffic demand at each TDA has uniformly distributed with integer values (in Erlang) over [1, 6] in AMPS and [1, 9] in CDMA. The locations of candidate base stations are randomly generated.

To compare the performance of the proposed tabu search, Parallel Genetic Algorithm (PGA) with island model is considered. 500 chromosomes are divided into five islands and evolved until the best fitness at each island is equal to the average fitness of the island. At the end of each generation, the best solution is copied to the next island. To represent the chromosome Grouping GA [17] is employed which is known to be superior in grouping problems including set covering. Each string has two parts: TDA part and BS part. In TDA part a base station is assigned to each TDA. The base stations employed in the TDA part are then represented in the BS part. The superiority of Grouping GA over standard GA is mainly due to group-oriented operators. Group-oriented crossover and mutation [17] are applied to the chromosome. Tournament selection is performed to generate a population for the following generation.

4.1 Test on AMPS System

The capacity of each new base station is assumed 46 Erlangs as in Section 2.1. The size of a TDA is assumed 600 m$\times$600 m with traffic demand distributed uniformly over 1, 2, ..., 6 Erlangs.

The received power *P(i,k)* at BS_k from TDA_i is simplified as follows: Using the propagation model in Section 2.2 the coverage radius by a base station is computed which satisfies the receiver sensitivity at the base station. For example, with receiver sensitivity –112 dBm, the transmission power of a mobile is 28 dBm (= 600 mW), G_c = 6 dB, G_m = 3 dB, h_c = 25 m, h_m = 1.5 m, and L_m = -45 dB, the radius is computed as r = 2.4 km from the propagation model.

In Figure 2, the received power at the base station located in the center is represented with 10 different levels depending on the location of each TDA. The 10 levels are discretized depending on the received power such that level 1 corresponds to [-112 dBm

~ -110 dBm] and level 10 to [-88 dBm ~ ∞].

0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
0	**0**	**2**	**3**	**4**	**3**	**2**	**0**	**0**
0	**2**	**5**	**6**	**7**	**6**	**5**	**2**	**0**
0	**3**	**6**	**8**	**9**	**8**	**6**	**3**	**2**
1	**4**	**7**	**9**	**10**	**9**	**7**	**4**	**1**
0	**3**	**6**	**8**	**9**	**8**	**6**	**3**	**0**
0	**2**	**5**	**6**	**7**	**6**	**5**	**2**	**0**
0	**0**	**2**	**3**	**4**	**3**	**2**	**0**	**0**
0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

Figure 2. Received Power Levels in AMPS

Now, we first test two initial solution strategies: All Candidate Base Stations and Random Feasible Base Stations. Problems of 20×20 TDAs are tested for three different cases of coverage factor, i.e., α=0.90, 0.95 and 0.99. For each case 10 instances are experimented and the average is shown in Figure 3. As shown in the figure, the method of Random Feasible Base Stations is superior to that of All Candidate Base Stations in view of computational time. The solution quality seems to be almost same in all three cases. Tabu parameters used in the test are chosen from the following experiments.

Before solving the cell planning problem we test the performance of tabu parameters: Tabu-Time size, *Nmax*, and *Dmax*. Tabu-Time size represents the number of iterations during which a base station is not allowed to be added or dropped. *Nmax* represents the

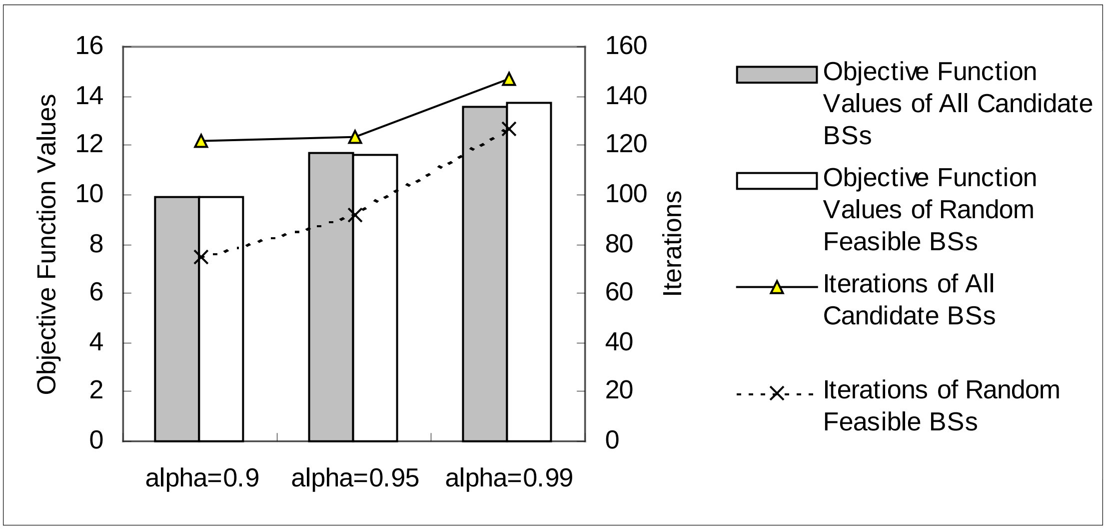

Figure 3. Test of Initial Feasible Solutions

number of consecutive iterations allowed for the search to continue without cost improvement. A diversification procedure is followed if no cost improvement is obtained during the *Nmax* iterations. *Dmax* is the stopping criterion which represents the number of diversifications in the search process.

Our test shows that the Tabu-Time size is dependent on the tabu list and problem size. In the test, since the size of Candidate_List is larger than that of Active_List, it is clear that Tabu-Time size of a base station in the Candidate_List needs to be larger than that in the Active_List.

Larger Tabu-Time size showed better performance as the problem size (number of TDAs to cover) increases. Test result shows that the following pair of Tabu-Time sizes in Active_List and Candidate_List are appropriate: (1,3) for problems with 10×10 TDAs, (2,5) for 20×20 TDAs and (3,7) for 30×30 TDAs.

The test result of *Nmax* is presented in Figure 4. By assuming that an appropriate value of *Nmax* is proportional to the number of candidate base stations K_2, test is performed for 5 different values. Figure 4 shows that $Nmax = 1.2 \times K_2$ is appropriate in view of iterations and objective function values.

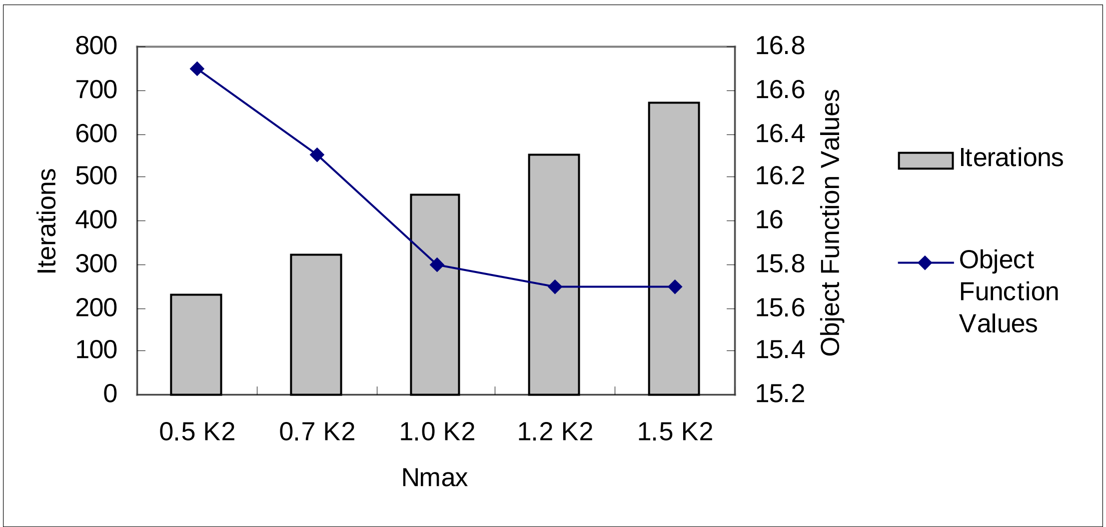

Figure 4. Test of *Nmax* in Problems with 20×20 TDAs

The number of diversifications in tabu search is deeply related to the solution quality. The test on *Dmax* is performed for three different values of coverage factor α. For each case, fifty problems are experimented to determine the value of *Dmax*. In each case of α, the portion among fifty problems which gives no further improvement for increased value of *Dmax* is plotted. Figure 5 shows that the number of required diversifications increases with the increase of coverage factor α. From the figure it seems to be

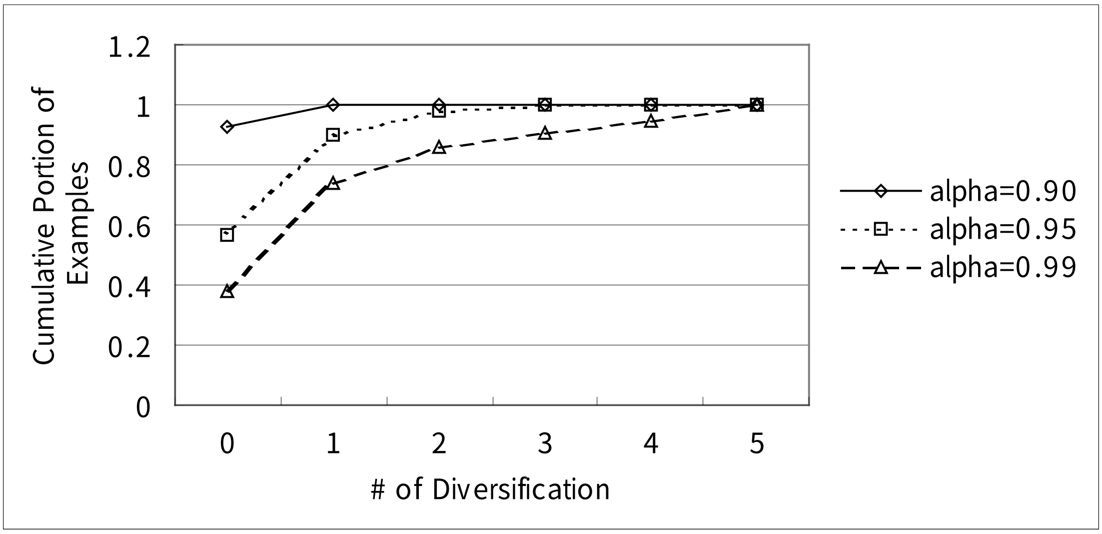

Figure 5. Test of *Dmax* on Problems with 20×20 TDAs

reasonable to perform 1, 3 and 5 diversifications for the coverage factor α=0.90, 0.95 and 0.99 respectively.

Table 1, 2, and 3 show the performance of the proposed tabu search with operators and parameters obtained in the preliminary tests. CPLEX [16] is used to obtain optimal solutions. The parallel GA is also experimented. Since our attention is focused on the solution quality, the PGA is run until the best solution at each island becomes equal to the average fitness of the island. The number of new base stations and computational times are presented in the tables. The performance of the proposed tabu search is outstanding for problems with coverage factors α = 0.9 and 0.95. The gap from the optimal solution is approximately 1 ~ 2 % even in problems with 900 TDAs. The gap is slightly increased to 5 ~ 6 % in problems with 900 TDAs with coverage factor α = 0.99. The parallel GA gives near optimal solution in problems with 100 TDAs. However, the performance is degraded as the problem size increases. The gap from the optimal solution approaches 20% in problems with 900 TDAs.

4.2 Test on CDMA System

The system capacity of CDMA is usually known to be 6 8 times of AMPS [1]. It means that CDMA is appropriate for the increased dense traffic area with microcells. We thus assume that the size of a TDA is 300 m×300 m with traffic demand distributed uniformly over 1, 2, ..., 9 Erlangs.

The received power at a base station can be computed as in the AMPS of section 4.1. Due to the reduced cell size, the transmission power of a mobile is reduced to 200 300 mW as in PCS phones.

In Figure 6, the received power at the base station located in the center is represented with 15 different levels. The 15 levels are discretized depending on the received power such that level 1 corresponds to [-112 dBm ~ -111 dBm] and level 15 to [-76 dBm ~ ∞].

For the capacity of base stations, we assume each existing base station uses 4 FAs which is equal to 345 Erlangs. The capacity of each new base station is assumed to have either 2, 3, or 4 FAs, which correspond to 165, 255, and 345 Erlangs respectively. Since the cost of a base station consists of fixed and variable parts, we assume the cost of base station to be 6, 8, and 10 respectively for the 2, 3, and 4 FAs.

As in AMPS, the strategy of Random Feasible Base Stations is employed to have initial feasible solution. Tabu-Time sizes in Active_List and Candidate_List are selected as (2,3), (3,5), and (5,8) for problems with 20×20, 30×30, and 50×50 TDAs respectively.

0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	2	2	3	4	3	2	2	0	0	0	0	0
0	0	0	1	3	4	5	6	6	6	5	4	3	1	0	0	0
0	0	1	3	5	7	8	8	9	8	8	7	5	3	1	0	0
0	0	3	5	7	9	9	10	11	10	9	9	7	5	3	0	0
0	2	4	7	9	10	11	12	12	12	11	10	9	7	4	2	0
0	2	5	8	9	11	13	13	14	13	13	11	9	8	5	2	0
0	3	6	8	10	12	13	14	15	14	13	12	10	8	6	3	0
1	4	6	9	11	12	14	15	15	15	14	12	11	9	6	4	1
0	3	6	8	10	12	13	14	15	14	13	12	10	8	6	3	0
0	2	5	8	9	11	13	13	14	13	13	11	9	8	5	2	0
0	2	4	7	9	10	11	12	12	12	11	10	9	7	4	2	0
0	0	3	5	7	9	9	10	11	10	9	9	7	5	3	0	0
0	0	1	3	5	7	8	8	9	8	8	7	5	3	1	0	0
0	0	0	1	3	4	5	6	6	6	5	4	3	1	0	0	0
0	0	0	0	0	2	2	3	4	3	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 6. Received Power Levels in CDMA

In addition to the above fixed tabu time, we test the performance of dynamic tabu time. A sequence of Tabu-Time (1,2), (2,3), and (3,5) are repeatedly employed in problems with 20×20 TDAs. (2,3), (3,5), and (5,8) for 30×30 and (3,5), (5,8), and

(8,11) for 50×50 are tested. Figure 7 shows the dynamic tabu tenure is slightly better than the fixed one with reasonable increase of tabu iterations. We thus employ dynamic tabu tenure in the tests to follow.

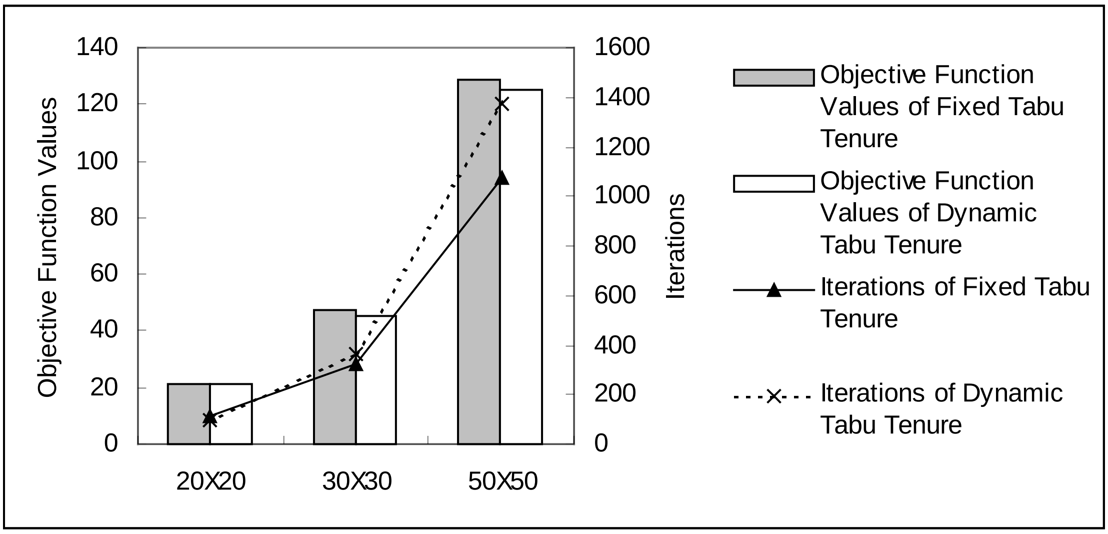

Figure 7. Test of Fixed/Dynamic Tabu Tenure Strategies

The test results on *Nmax* and *Dmax* are shown in Figure 8 and 9 respectively. Figure 8 shows that diversification with capacity change gives best performance. In other words, the performance of tabu search is increased with the diversification which increases or decreases the capacity of a new base station depending on the current usage of the capacity. When the capacity of a base station is fully employed, the capacity is

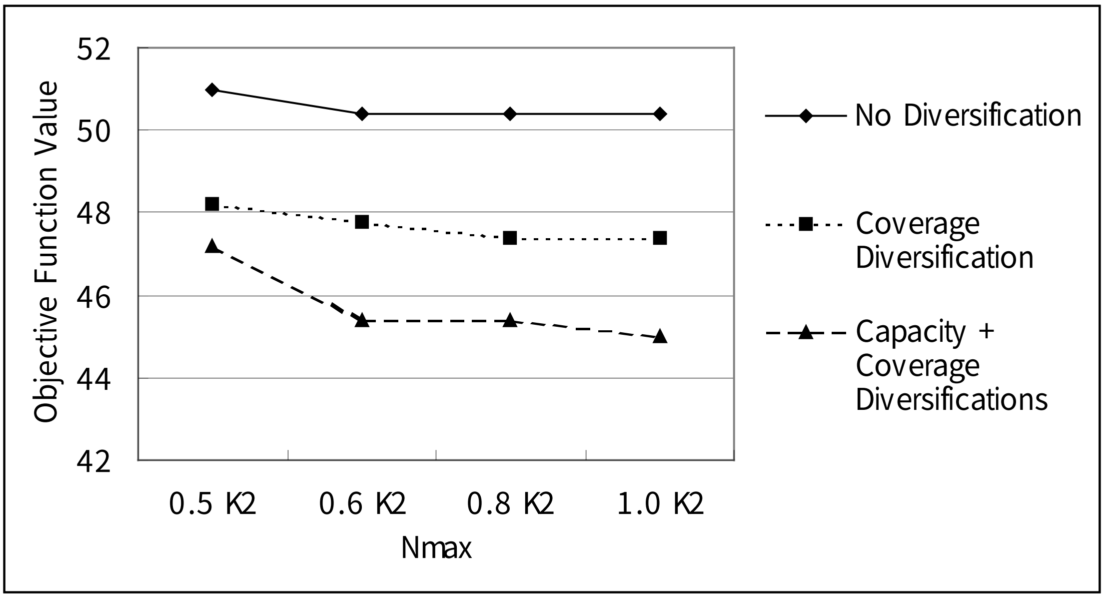

Figure 8. Test of *Nmax* in Problems with 30×30 TDAs

increased by 1 FA, and the residual capacity is decreased by 1 FA. The effectiveness of the restarting strategy is also illustrated in the figure. Better solutions are obtained by starting base stations with least frequencies in the old solutions and moves. Diversification is performed when no functional improvement is obtained for consecutive $Nmax = 0.6 \times K_2$ iterations.

The number of diversifications required to stop tabu search is largely dependent on the problem complexity. The difficulty of a problem is increased with the increase of traffic coverage factor. From Figure 9 the reasonable number of diversifications seams to be 9, 12, and 15 respectively for α =0.90, 0.95, and 0.99.

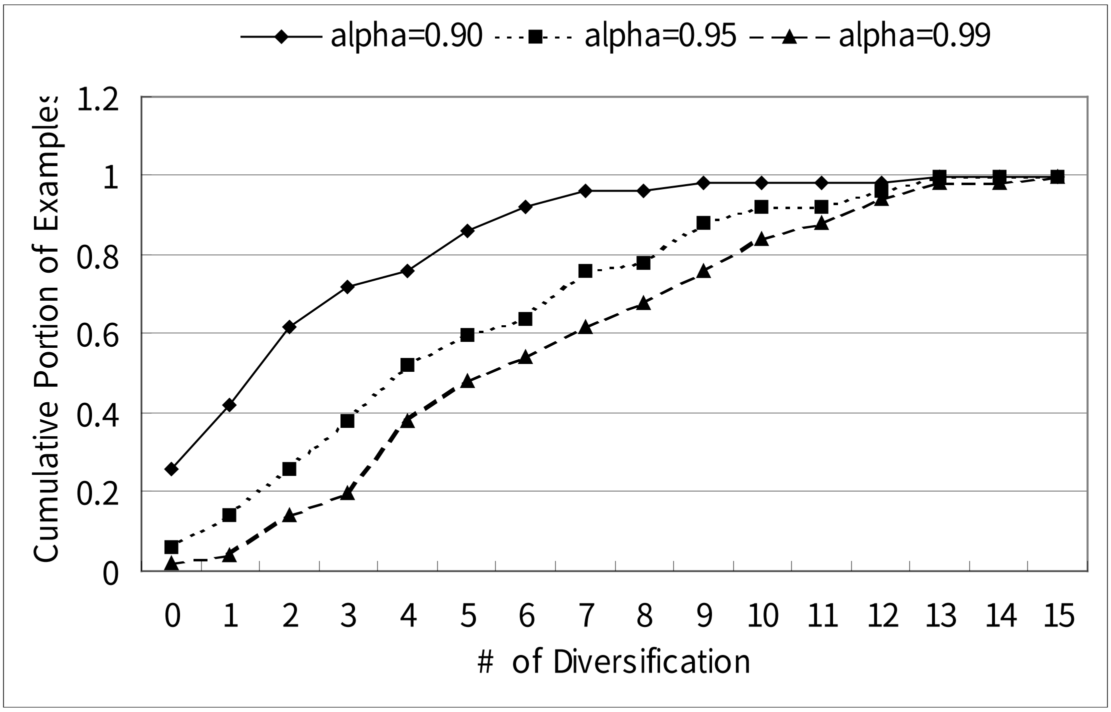

Figure 9. Test of *Dmax* in Problems with 30×30 TDAs

Table 4, 5, and 6 show the computational results of tabu search for problems under CDMA system. For problems with 400 TDAs tabu search provides solutions which are within 5% from the optimal even in case of α= 0.99. However, in most problems with 900 and 2,500 TDAs, we failed to obtain optimal solutions with CPLEX within the CPU time limit of 10,000 seconds. Fortunately in problems with 900 TDAs, the proposed tabu search provides optimal solutions in almost all cases of α=0.90 and 0.95. Solutions by the tabu search match with the lower bounds by CPLEX. For α=0.99 the gap from the lower bound is approximately 15% in 900 TDAs. In problems with 2,500 TDAs, the gaps from the lower bounds are 1%, 5%, and 15% respectively for α= 0.90, 0.95 and 0.99. Note that the gaps remain same in problems with 900 and 2,500 TDAs.

The performance of the parallel GA is not promising compared to the proposed tabu search. In many problems the PGA fails to meet the coverage factor α, which is due to the penalty method used in GA to handle the constraints in the problem. Moreover, the gaps from the lower bounds are 28-30 % for problems with 2,500 TDAs regardless of the coverage factor α.

5. Conclusion

Cell planning problem with capacity expansion is examined in wireless communications. The problem decides the location and capacity of each new base station to cover expanded and increased traffic demand. The objective is to minimize the cost of new base stations. The coverage by the new and existing base stations is constrained to satisfy a proper portion of traffic demands. The received signal power at the base station also has to meet the receiver sensitivity. The cell planning is formulated as an integer linear programming problem and solved by a Tabu Search algorithm.

In the tabu search intensification by add and drop move is implemented by short-term memory embodied by two tabu lists: Candidate_List and Active_List. Different tabu times are applied to the base stations to add among those in the Candidate_List and to drop in the Active_List. Two diversification strategies are employed to explore new solution space. The capacity diversification is designed to investigate proper capacities of new base stations. The capacity of each new base station is increased or decreased depending on the usage of the capacity. The coverage diversification restarts the tabu search from new assignment of bast station locations. It is implemented by examining the base stations with least frequencies in the old solutions.

Computational experiments of the proposed tabu search are performed for the cell planning problems in two different environments: AMPS and CDMA system. The proposed procedure illustrates outstanding performance in AMPS. The gap from the optimal solution is 1~2% for coverage factor α=0.9 and 0.95. The gap is slightly increased to 5~6% in problems with 900 TDAs with α=0.99. Compared to the PGA 10~30 % cost reduction is obtained by the tabu search in problems with 900 TDAs.

Test on CDMA shows that two diversification strategies are highly effective. Compared to the no diversification 6% cost reduction is obtained by the coverage diversification and 10% by both the coverage and capacity strategies. In problems with 400 TDAs tabu search provides solutions which are within 5% from the optimal even in case of α=0.99. The proposed procedure presents optimal solutions in almost all instances of 900 TDAs with α=0.90 and 0.95. The gap from the lower bound is approximately 15% in problems with α=0.99. However, the gap remains same in problems of 2,500 TDAs with α=0.99, which shows the robustness of the proposed tabu search. The proposed tabu search also outperforms the PGA. Approximately 10~20 % cost reduction is obtained in problems with 2,500 TDAs.

References

[1] T. S. Rappaport, *Wireless Communications*, Prentice Hall PTR, 1996.

[2] W. C. Y. Lee, *Mobile Cellular Telecommunications Systems*, McGraw-Hill Book Company, 1989.

[3] F. Glover, "Tabu Search," *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206, Summer 1989.

[4] E. Rolland, H. Pirkul, and F. Glover, "Tabu Search for Graph Partitioning," *Annals of Operations Research*, Vol. 63, pp. 209-232, 1996.

[5] H. D. Sherali, C. M. Pendyala, and T. S. Rappaport, "Optimal Location of Transmitters for Micro-Cellular Radio Communication System Design," *IEEE Journal on Selected areas in communications*, Vol. 14, No. 4, pp. 662-673, 1996.

[6] B. Chamaret, S. Josselin, P. Kuonen, M. Pizarroso, B. Salas-Manzanedo, S. Ubeda, and D. Wagner, "Radio Network Optimization with Maximum Independent Set Search," *IEEE VTC*, pp. 770-774, 1997.

[7] P. Calegari, F. Guidec, P. Kuonen, and D. Wagner, "Genetic Approach to Radio Network Optimization for Mobile Systems," *IEEE VTC*, pp. 755-759, 1997.

[8] P. Calegari, F. Guidec, P. Kuonen, and D. Kobler, “Parallel Island-based Genetic Algorithm for Radio Network Design,” *Journal of Parallel and Distributed Computing*, Vol. 47, pp. 86-90, 1997.

[9] K. Tutschku, “Demand-based Radio Network Planning of Cellular Communication Systems,” *Proceedings of the IEEE Infocom 98*, pp. 1054-1061, 1998.

[10] C. Yu, S. Subramanian, and N. Jain, “CDMA Cell Site Optimization Using a Set Covering Algorithm,” *in Proceedings of the 8-th International Network Planning Symposium*, pp. 75-78, 1998.

[11] M.R. Garey, D.S. Johnson, and L. Stockmeyer, “Some simplified NP-complete graph problems,” *Theoretical Computer Science*, Vol. 1, pp. 237-267, 1976.

[12] K. Tutschku, “Interference minimization using automatic design of cellular communication networks,” *IEEE VTC*, pp. 634-638, 1998.

[13] M. H. Wright, “Optimization Methods for Base Station Placement in Wireless Application,” *IEEE VTC*, pp. 387-391, 1998.
[14] *The CDMA Network Engineering Handbook*, Qualcomm, Vol.1, March 1. 1993.
[15] '94 CDMA Forum, 1994.2.28. - 3.1., San Diego
[16] CPLEX 4.0, CPLEX Optimization Inc., 1998.
[17] E. Falkenauer, “A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems,” *Evolutionary Computation*, Vol. 2, No. 2, pp. 123-144, 1994.

Table 1. Computational Results with 10×10 TDAs

Problem Number	Total Traffic Demand	α= 0.90			α= 0.95			α= 0.99		
		Tabu Search	Genetic Algorithm	Optimal Solution	Tabu Search	Genetic Algorithm	Optimal Solution	Tabu Search	Genetic Algorithm	Optimal Solution
1	347	3 (0.05)	3* (101.61)	3 (2.53)	4 (0.05)	4 (72.17)	4 (2.37)	4 (0.17)	4* (110.90)	4 (2.25)
2	350	3 (0.05)	5 (121.55)	3 (2.47)	4 (0.11)	5 (3.68)	4 (2.43)	4 (0.11)	4 (114.91)	4 (2.42)
3	341	3 (0.06)	3* (107.43)	3 (2.10)	4 (0.06)	4 (5.27)	4 (6.48)	4 (0.11)	4* (5.50)	4 (2.58)
4	335	3 (0.05)	3 (102.00)	3 (1.92)	3 (0.05)	4 (6.86)	3 (2.91)	4 (0.11)	4 (5.54)	4 (1.93)
5	365	4 (0.06)	3* (108.15)	4 (2.31)	4 (0.06)	4* (39.1)	4 (1.98)	5 (0.11)	4 (9.01)	4 (2.25)
6	345	3 (0.06)	3* (103.10)	3 (2.20)	4 (0.10)	4* (113.48)	4 (2.08)	5 (0.11)	5 (113.03)	4 (2.08)
7	347	3 (0.05)	3* (103.31)	3 (2.30)	4 (0.06)	4 (111.12)	4 (1.92)	4 (0.11)	4* (19.55)	4 (2.25)
8	341	3 (0.06)	3 (37.85)	3 (2.09)	4 (0.11)	3* (107.00)	4 3.73)	4 (0.17)	4 (4.67)	4 (2.08)
9	363	4 (0.05)	3* (104.91)	4 (2.36)	4 (0.05)	4 (39.49)	4 (2.47)	5 (0.11)	4* (8.35)	4 (2.25)
10	343	3 (0.06)	3 (101.72)	3 (1.93)	4 (0.06)	3* (40.76)	4 (5.61)	4 (0.11)	4 (9.66)	4 (2.03)
Average		3.2 (0.05)	3.2 (99.16)	3.2 (2.22)	3.9 (0.07)	3.9 (53.89)	3.9 (3.20)	4.3 (0.12)	4.1 (40.11)	4.0 (2.21)

* represents the solution which does not satisfy the coverage factor (α) in GA

The numbers in the parenthesis represent the CPU seconds

Table 2. Computational Results with 20×20 TDAs

Problem Number	Total Traffic Demand	α= 0.90			α= 0.95			α= 0.99		
		Tabu Search	Genetic Algorithm	Optimal Solution	Tabu Search	Genetic Algorithm	Optimal Solution	Tabu Search	Genetic Algorithm	Optimal Solution
1	3192	12 (0.92)	13* (443.80)	12 (455.28)	13 (4.29)	16 (175.26)	13 (103.43)	16 (7.14)	16 (143.63)	14 (205.48)
2	3113	11 (0.81)	12* (157.86)	11 (138.09)	13 (3.62)	16 (175.43)	13 (86.41)	15 (6.86)	14* (140.23)	14 (112.49)
3	3223	11 (0.87)	14* (445.89)	11 (127.76)	13 (3.57)	16 (180.05)	13 (89.09)	15 (6.76)	16 (118.36)	14 (124.69)
4	3081	11 (0.98)	12* (463.79)	11 (128.75)	13 (3.79)	15* (504.33)	13 (91.51)	15 (7.14)	15 (220.03)	14 (88.06)
5	3130	12 (0.98)	12* (474.23)	12 (85.70)	14 (3.85)	16 (210.86)	14 (86.23)	16 (7.19)	17 (118.42)	15 (112.65)
6	3082	13 (0.86)	14* (485.10)	13 (210.46)	15 (3.68)	15* (497.24)	14 (136.17)	16 (6.87)	19 (107.70)	16 (121.11)
7	3172	12 (0.82)	14 (492.52)	12 (91.56)	14 (3.62)	14* (451.65)	13 (95.46)	15 (7.47)	16 (156.04)	15 (280.08)
8	3171	12 (0.92)	12* (493.61)	12 (89.36)	14 (4.01)	17 (472.79)	14 (91.77)	16 (6.65)	17* (164.00)	15 (95.02)
9	3090	12 (0.76)	13* (569.24)	12 (90.41)	14 (3.79)	17 (138.63)	14 (242.12)	16 (6.31)	18 (125.12)	15 (91.26)
10	3046	11 (1.08)	12* (455.88)	11 (160.78)	13 (3.62)	15* (452.64)	13 (103.39)	15 (6.98)	17 (129.07)	14 (81.68)
Average		11.7 (0.90)	12.8 (478.19)	11.7 (157.82)	13.6 (3.78)	15.7 (325.89)	13.4 (112.56)	15.5 (6.94)	16.5 (142.27)	14.6 (131.25)

* represents the solution which does not satisfy the coverage factor (α) in GA

The numbers in the parenthesis represent the CPU seconds

Table 3. Computational Results with 30×30 TDAs

Problem Number	Total Traffic Demand	α= 0.90 Tabu Search	α= 0.90 Genetic Algorithm	α= 0.90 Optimal Solution	α= 0.95 Tabu Search	α= 0.95 Genetic Algorithm	α= 0.95 Optimal Solution	α= 0.99 Tabu Search	α= 0.99 Genetic Algorithm	α= 0.99 Optimal Solution
1	3192	27 (22.14)	34 (2263.20)	27 (2389.18)	31 (44.16)	39 (1121.74)	30 (3067.57)	35 (82.55)	39 (670.91)	33 (2934.48)
2	3113	26 (20.93)	31 (2218.44)	25 (2920.26)	29 (46.46)	36* (2254.41)	29 (2780.75)	33 (79.53)	38 (935.49)	31 (3064.95)
3	3223	28 (20.70)	34* (2329.83)	28 (3287.02)	31 (45.31)	42 (874.63)	31 (3053.85)	35 (82.66)	40 (774.40)	34 (2953.94)
4	3081	25 (22.57)	28* (2306.98)	25 (2949.75)	29 (47.46)	38 (828.22)	28 (3280.11)	32 (92.71)	36 (951.37)	31 (3708.02)
5	3130	26 (21.75)	32* (2418.59)	26 (2706.27)	30 (43.94)	39 (927.54)	29 (2769.61)	33 (80.69)	40 (537.33)	32 (3043.78)
6	3082	25 (21.04)	31* (2307.36)	25 (2593.80)	28 (45.70)	37 (1620.25)	28 (2911.99)	33 (85.57)	38* (641.14)	31 (2870.97)
7	3172	27 (20.43)	31* (2311.43)	27 (2688.28)	30 (46.08)	39 (995.75)	30 (3178.09)	35 (78.88)	39 (1026.17)	33 (2912.60)
8	3171	27 (20.87)	28* (2380.85)	27 (3490.68)	30 (44.33)	39 (1338.92)	30 (3347.42)	35 (76.95)	39 (725.51)	33 (3204.36)
9	3090	25 (20.16)	31 (2354.99)	25 (3355.73)	29 (41.30)	38* (2238.54)	28 (2813.95)	34 (73.71)	37 (708.59)	31 (3732.58)
10	3046	24 (20.48)	29* (2340.98)	24 (3060.44)	28 (42.24)	35 (1013.48)	27 (3177.22)	32 (82.33)	35* (869.63)	30 (3498.60)
Average		26.0 (21.11)	30.9 (2323.27)	25.9 (2989.14)	29.5 (44.70)	38.2 (1321.35)	29.0 (3038.06)	33.7 (81.56)	38.1 (697.92)	31.9 (3192.43)

* represents the solution which does not satisfy the coverage factor (α) in GA

The numbers in the parenthesis represent the CPU seconds

Table 4. Computational Results with 20×20 TDAs

Problem Number	Total Traffic Demand	α= 0.90			α= 0.95			α= 0.99		
		Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX
1	2040	16 (0.55)	16 (245.46)	16 (2776.92)	18 (1.16)	18* (270.61)	18 (171.64)	22 (1.37)	22 (40.92)	20 (1927.00)
2	1973	14 (0.44)	14* (47.07)	14 (97.83)	16 (1.10)	18 (54.87)	16 (92.49)	18 (1.16)	18* (62.23)	18 (101.12)
3	1941	14 (0.50)	14* (232.77)	14 (282.70)	16 (0.82)	16* (280.45)	16 (2735.84)	20 (1.37)	18* (68.76)	18 (3550.61)
4	2053	16 (0.49)	16 (238.15)	16 (109.69)	18 (0.88)	18 (70.36)	18 (113.04)	22 (1.21)	24 (60.42)	20 (108.31)
5	1955	14 (0.55)	14 (43.83)	14 (104.25)	16 (0.82)	16 (41.80)	16 (110.24)	18 (1.32)	18* (61.41)	18 (107.65)
6	1994	14 (0.49)	14 (43.56)	14 (96.73)	18 (0.94)	18 (81.78)	18** (10000)	22 (1.59)	24 (50.09)	18 (15.21)
7	2019	16 (0.39)	18 (51.19)	16 (3211.33)	18 (0.73)	18 (289.51)	18** (10000)	20 (1.21)	18* (75.03)	20** (10000)
8	1957	14 (0.49)	16 (60.20)	14 (117.38)	16 (0.99)	18 (62.39)	16 (110.40)	22 (1.04)	22 (54.43)	18 (103.69)
9	1966	14 (0.50)	14* (49.49)	14 (108.37)	16 (0.82)	16* (62.56)	16 (102.70)	20 (1.82)	18 (63.76)	18 (104.74)
10	1988	14 (0.38)	14* (218.38)	14 (194.48	18 (0.77)	18 (252.65)	16 (182.68)	20 (1.10)	20 (74.59)	18 (14.56)
Average		14.6 (0.48)	15.0 (123.01)	14.6 (709.97)	17.0 (0.90)	17.4 (146.70)	16.8	20.4 (1.40)	20.2 (61.16)	18.6

* represents the solution which does not satisfy the coverage factor(α) in GA.

** represents the lower bound of the solution.

The numbers in the parenthesis represent the CPU seconds.

Table 5. Computational Results with 30×30 TDAs

Problem Number	Total Traffic Demand	α= 0.90			α= 0.95			α= 0.99		
		Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX
1	4490	28 (6.26)	28* (259.41)	28 (159.73)	36 (9.18)	44 (1040.68)	36** (10000)	46 (15.05)	44* (335.15)	40** (10000)
2	4444	28 (5.93)	28* (1058.91)	28** (10000)	36 (9.50)	36* (1001.63)	36** (10000)	48 (20.44)	44 (394.91)	44** (10000)
3	4434	28 (4.39)	32 (197.24)	28** (10000)	34 (7.52)	38 (262.16)	34** (10000)	42 (15.54)	48 (268.20)	38 (167.37)
4	4517	30 (5.82)	30 (339.55)	30** (10000)	36 (9.29)	36* (446.38)	36** (10000)	44 (21.09)	50 (1183.04)	40 (394.31)
5	4426	28 (6.65)	28* (950.26)	28** (10000)	34 (10.93)	38 (279.46)	34** (10000)	44 (14.83)	42* (273.91)	38 (188.44)
6	4522	32 (5.55)	32 (925.44)	30** (10000)	36 (8.45)	36* (303.52)	36** (10000)	48 (14.61)	52 (1087.47)	40 (294.02)
7	4437	28 (5.44)	28* (248.15)	28** (10000)	36 (7.69)	36* (449.84)	36** (10000)	44 (14.56)	44 (316.59)	38 (157.96)
8	4420	28 (7.30)	30* (314.78)	28** (10000)	34 (10.22)	34* (295.89)	34** (10000)	44 (14.50)	44* (315.11)	38** (10000)
9	4452	30 (5.38)	30 (989.82)	28** (10000)	36 (9.17)	42 (357.84)	36** (10000)	50 (16.14)	52 (421.61)	42** (10000)
10	4561	32 (4.89)	32 (328.40)	30** (10000)	38 (8.95)	42 (1130.64)	38** (10000)	50 (17.64	44* (266.99)	44** (10000)
Average		29.2 (5.76)	29.8 (561.20)	28.6	35.6 (9.09)	38.2 (556.80)	35.6	46.0 (16.44)	46.4 (486.30)	40.2

* represents the solution which does not satisfy the coverage factor (α) in GA.

** represents the lower bound of the solution.

The numbers in the parenthesis represent the CPU seconds.

Table 6. Computational Results with 50×50 TDAs

Problem Number	Total Traffic Demand	α= 0.90			α= 0.95			α= 0.99		
		Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX	Tabu Search	Genetic Algorithm	Solution by CPLEX
1	12461	80 (122.54)	106 (3708.72)	78** (10000)	104 (187.52)	126 (3626.26)	96** (10000)	130 (321.75)	140* (3501.61)	118** (10000)
2	12540	84 (113.26)	104 (3706.59)	78 (6534.87)	102 (201.41)	130 (3388.25)	100** (10000)	128 (303.68)	142* (3513.26)	110** (10000)
3	12686	84 (129.18)	110 (3429.21)	84** (10000)	106 (180.40)	126 (3296.46)	100 (6094.52)	132 (337.41)	148 (3976.06)	116** (10000)
4	12407	80 (116.83)	98* (3167.93)	76** (10000)	98 (178.35)	100* (3211.00)	94** (10000)	126 (139.39)	140* (4577.27)	108** (10000)
5	12387	78 (124.08)	98* (3236.54)	76** (10000)	100 (211.51)	124* (3374.35)	94** (10000)	126 (301.65)	142* (4257.49)	108** (10000)
6	12481	82 (119.08)	102 (3312.23)	78** (10000)	102 (180.65)	126 (3323.71)	96** (10000)	134 (362.31)	148 (2841.07)	114** (10000)
7	12507	82 (106.17)	108 (3344.53)	80** (10000)	102 (181.86)	128 (3357.65)	96** (10000)	130 (352.13)	148 (2838.88)	110** (10000)
8	12461	80 (115.73)	104 (3336.67)	80** (10000)	100 (211.30)	130 (3414.06)	96** (10000)	130 (280.34)	148 (4255.52)	114** (10000)
9	12401	78 (106.28)	90* (3307.72)	76** (10000)	100 (172.69)	124* (3389.17)	94** (10000)	126 (330.48)	158 (3236.43)	112** (10000)
10	12569	82 (116.71)	108* (3300.09)	80** (10000)	100 (199.27)	132 (3511.45)	100** (10000)	130 (313.27)	148 (3652.88)	118** (10000)
Average		81.0 (116.99)	102.8 (3385.02)	80.0	101.4 (190.50)	124.6 (3389.24)	96.8	129.2 (304.24)	146.2 (3665.05)	112.8

* represents the solution which does not satisfy the coverage factor (α) in GA.

** represents the lower bound of the solution.

The numbers in the parenthesis represent the CPU seconds.