韓國經營科學會誌
第31권 第3號
2006年 9月


# 멀티캐스트 CDMA 네트워크에서의 Soft-combine을 지원할 기지국의 선정*

김재훈** · †명영수***

# Optimal Soft-combine Zone Configuration in a Multicast CDMA Network*

Jae-Hoon Kim** · Young-Soo Myung***

■ Abstract ■

In this paper we deal with a cell planning issue arisen in a CDMA based multicast network. In a CDMA based wireless network, a terminal can significantly reduce the bit error rate via the cohesion of data streams from multiple base stations. In this case, multiple base stations have to be operated according to a common time line. The cells whose base stations are operated as such are called soft-combined cells. Therefore, a terminal can take advantage of error rate reduction, if the terminal is in a *soft-combined cell* and at least one neighboring cell is also soft-combined. However, as soft-combining operation gives heavy burden to the network controller, the limited number of cells can be soft-combined. Our problem is to find a limited number of soft-combined cells such that the benefit of the soft-combining operation is maximized.

Keyword : Multicast CDMA Network, Cell Selection, Combinatorial Optimization

논문접수일 : 2005년 12월 3일　　논문게재확정일 : 2006년 6월 19일
* This work was supported by Korea Research Foundation Grant(KRF-2004-041-B00141).
** Network R&D Center, SK Telecom
*** 단국대학교 경영학부
† 교신저자

# 1. Introduction

Mobile multicast has recently emerged and become a topic of significant discussion on both of technical advances and practical usefulness. Mobile multicast makes various types of on-demanded services available in the mobile environment. From a technical point of view, the area of mobile multicast radio technologies has become very diverse and manifested through the emergence of many international standardization activities along with regional proprietary solutions. Especially, in 3G mobile radio standardization, multicast services, named as Multicast Service (BCMCS) in cdma2000 1xEV-DO [5, 6] and Multimedia Broadcast/Multicast Service (MBMS) in UMTS WCDMA [2], are now being deployed in CDMA-based 3G networks. The former one, which specifies mobile multicast support for the 3rd generation cdma2000 cellular, satisfies the perceived market demand while minimizing resource usage in the radio access network [14]. The latter one, for the next version of UMTS, is looking at defining point to multipoint bearers to send multimedia services to several users in a cell over one radio bearer [4, 13].

The nature of multicast in CDMA networks is not quite different from that of wired line networks. Needless to say, a single data stream per link is the essential feature of multicast and there is no difference between wired and wireless networks on this point. However, in last-mile access lines, *interfaces-thru-air* of wireless network induces significant difference from wired line networks. In a wired line network, the data stream should be distributed via separated access line to each user in last-mile access line, but in a wireless network, only a single data stream with specific discriminator, such as multicast code or frequency, is dispersed into the air. The terminals extract the data stream and decode it to receive a specific multicast service.

In a CDMA based wireless network, a terminal can hold multiple connections from near base stations and receives an identical multicast content via multiple connections, that is similar to soft handover procedure. In this case, the terminal can significantly reduce the bit error rate via the cohesion of data stream from multiple base stations. The cohesion of data streams from two different base stations is possible only when two base stations have to emit data stream containing the same content at the same time epoch. In other words, the two base stations should be operated according to a common time line. The cells whose base stations are operated as such are called *soft-combined cells*. For the details of the soft-combining operation, refer to [1]. We will say that two cells are neighbor if a terminal in each cell can receive data from the base station in the other center. Therefore, if a terminal locates in a soft-combined cell and at least one neighboring cell is also soft-combined, the terminal can take advantage of error rate reduction.

The operations, onto the common time line, require strict cell management by a network controller, such as Base Station Controller (BSC) or Radio Network Controller (RNC). In order to transmit contents to a group of base stations through a common time line, the network controller has to keep the path information to the corresponding base stations and analyze the delay effect of specific path to each base station. The network controller should give time stamps to every multicast packet to specify the time epoch for packet emitting. As those operations for keeping common time line yield additional heavy burden to the network controller, a network controller can handle only a limited number of soft-combined cells.

In this paper, we deal with a problem of selecting the cells to be soft-combined in order to maximize the benefit of the soft-combining operation. As we mentioned, a terminal in a soft-combined cell can take advantage of error rate reduction, if at least one neighboring cell is also soft-combined. Our objective is to maximize the multicast traffic to be benefited by the soft-combining operation. Although it is hard to measure the volume of multicast traffic, it is shown that the proportion of multicast traffic to the total traffic is uniform on the average. So, our objective is to maximize the total traffic demand of the cells that are benefited by the soft-combining operation. Instead of maximizing the volume of multicast traffic benefited, one may also think about more specific criteria related with the grade of service, such as the average error rate of the multicast traffic. It would be better if new criteria reflect the difference of the benefit among cells having different number of the soft-combined neighboring cells. However, it seems to be difficult to come up with a good method to solve a model with such complex objective function.

Based on the above observation, our problem can be described as the following combinatorial optimization problem, referred as the soft-combined cell selection problem (SCSP). We are given the set of cells with traffic demand and the maximum allowable number of soft-combined cells. We are also given information on the neighboring relation among cells. Then our problem is to find the prescribed number of cells such that each selected cell has at least one selected neighbor. The objective of the problem is to maximize the total traffic demand of the selected cells. Although this problem is very simple and has an interesting feature, it has not been addressed in the literature. The most similar problem to the SCSP is the $k$-MST problem defined as follows. Given a graph $G=(V,E)$ with non-negative cost for each edge, the $k$-MST problem is that of finding the minimum-cost tree in G that spans at least $k$ nodes. The $k$-MST problem is known to be NP-hard [9] and a couple of researchers [7, 11] have found approximation algorithms for the $k$-MST problem.

The remaining part of the paper is organized as follows. In Section 2, we formulate the SCSP as a 0-1 integer programming model, and show that the problem is *NP-hard*. In Section 3, we present algorithms that provide lower and upper bounds of the problem. Computational results for evaluating the performance of the proposed algorithm are presented in Section 4.

## 2. Problem Statement and Formulation

For a service area consisting of $n$ cells, we define the following notation. Let $N=\{1,2,\cdots,n\}$ be the set of cells, and $d(i)$ denotes the traffic demand of cell $i$ for each $i \in N$. For each cell $i \in N$, $N(i)$ denotes the set of the neighboring cells to cell $i$. Let $k$ be the maximum allowable number of soft-combined cells. Then the SCSP is to find a set of $k$ cells, such that each cell in the set has at least one neighbor in the set. The objective of the problem is to maximize the total traffic demand of the selected cells.

To formulate the SCSP, we define 0-1 variables $x(i)$ for each cell $i \in N$ such that $x(i)=1$ if cell $i$ is selected for soft-combining, and 0, otherwise. Then the SCSP can be formulated as the following 0-1 integer programming problem.

$$
\begin{aligned}
\text{(P)} \quad \text{Max} \quad & \sum_{i \in N} d(i)x(i) && && (1)\\
\text{s.t.} \quad & \sum_{i \in N} x(i) \le k, && && (2)\\
& x(i) - \sum_{j \in N(i)} x(j) \le 0, && i \in N && (3)\\
& x(i) \in \{0,1\}, && i \in N && (4)
\end{aligned}
$$

The constraint (2) implies that the selected cells are not more than the maximum allowable numbers. The constraint (3) ensures that if a cell is selected as a soft-combined cell, at least one cell neighboring to it has to be also soft-combined.

Now we show that the SCSP is an *NP-hard* problem. To prove this, we will show that the node-cover problem, denoted by NODE-COVER, can be transformed in polynomial time to the decision version of the SCSP, denoted by D-SCSP. Since NODE-COVER is *NP-complete* [10], the above statement implies that D-SCSP is also *NP-complete*. NODE-COVER and D-SCSP are defined as follows.

**NODE-COVER** : Given a graph $G=(V,E)$ and an integer $t<|V|$, the node-cover problem is defined to determine whether a graph has a set $C$ of at most $t$ nodes such that all edges of $G$ are adjacent to at least one node of $C$. We call such a set $C$ as a node-cover of $G$.

**D-SCSP** : Given an instance of SCSP, the decision version of SCSP is to determine whether there exists a feasible solution of the SCSP with the objective value not less than a given value.

**Theorem 1.** *D-SCSP is NP-complete.*

Proof. We give a polynomial time reduction from NODE-COVER to D-SCSP. Consider an instance of NODE-COVER for a given $G=(V,E)$ and $t$. Our reduction maps the given instance of NODE-COVER to an instance of D-SCSP as follows. Each cell in $N$ corresponds to either a node or an edge in $G$, i.e., $N=V \cup E$. Then we set the traffic demand of each cell such that $d(i)=0$ if cell $i$ corresponds to a node of $V$, and $d(i)=1$ if cell $i$ corresponds to an edge of $E$. Now we define neighborhood relation between two cells of $N$. Every cell corresponding a node can be a neighbor only to a cell corresponding an edge and vice versa. Between such pair of cells, each cell can be a neighbor to each other, only when if the corresponding node is one of the end nodes of the corresponding edge. If we let $k=|E|+t$, there is a node cover $C$ with $|C| \le t$ if and only if there exists a feasible solution of SCSP with the objective value not less than $|E|$.

Note that even if a given graph is planar, the corresponding NODE-COVER is *NP*-complete [10]. Therefore, when each cell represents a part of a region, the corresponding SCSP is still *NP*-hard.

# 3. Solution Method

Now, we develop a practical algorithm to solve the SCSP. As the problem is *NP-hard*, we are focusing on producing a feasible solution of good quality that provides a lower bound for the problem. We also show how to obtain an upper bound that can be used to evaluate the quality of a generated feasible solution. These two procedures can also be used when obtaining an exact solution through a branch and bound method.

## 3.1 Calculating upper bounds

In this section, we describe a procedure for computing upper bounds of the problem. Consider the LP (Linear Programming) relaxation of (P) where integrality condition (4) is replaced by the following constraints.

$$x(i) \le 1, \quad i \in N \tag{5}$$

$$x(i) \ge 0, \quad i \in N \tag{6}$$

Let the LP relaxation of (P) be (LP). The optimal objective value of (LP) is an upper bound for the optimal value of (P).

To obtain an upper bound, we generate a dual feasible solution instead of exactly solving (LP). We come up with a dual method to exploit the good structure of the dual of (LP). As (LP) also has rather compact structure, a commercial LP solver such as CPLEX may be a good alternative. However, our method enriches the list of solution tools and our computational experiments will show that the dual method performs well. The dual of (LP), denoted by (DLP), is as follows :

(DLP)

$$\text{Min} \quad \lambda k + \sum_{i \in N} u(i) \tag{7}$$

$$\text{s.t.} \quad \lambda + w(i) - \sum_{j \in N(i)} w(j) + u(i) \ge d(i), \quad i \in N \tag{8}$$

$$\lambda, w(i), u(i) \ge 0, \quad i \in N \tag{9}$$

In this formulation, $\lambda$, $w(i)$, and $u(i)$ correspond to (2), (3), and (5), respectively.

Our algorithm obtains a feasible solution of (DLP) whose objective value is an upper bound of both (LP) and (P). Our algorithm may be viewed as a dual ascent heuristic, which successfully applied to several combinatorial optimization problems [3, 8, 15]. Let's define $s(i)$, $I^-$ and, $I^0$ as follows :

$$s(i) = \lambda + w(i) - \sum_{j \in N(i)} w(j) + u(i) - d(i),$$

$$I^- = \{i \in N \mid s(i) < 0\}, \text{ and } I^0 = \{i \in N \mid s(i) = 0\}.$$

We initially set all the dual variables $\lambda$, $w(i)$'s, and $u(i)$'s equal to 0 and increase them to satisfy the feasibility condition, that $I^- = \varnothing$. Our strategy is to increase dual variables while keeping the objective value as low as possible. Since $w(i)$'s do not appear in the objective function, we try to update $w(i)$'s first to increase $s(i)$ for some $i \in I^-$ without violating the feasibility condition for $i \in N \setminus I^-$. While the increase of $u(i)$ directly increases both $s(i)$ and the objective value by the same amount, the increase of $\lambda$ increases all $s(i)$'s by the same amount and the objective value by $k$ times the amount. So, our algorithm increases $\lambda$ only when it can increase $s(i)$'s for at least $k+1$ elements in $I^-$. Our algorithm is formally described as follows :

**Algorithm UB**

1. Initially set $w(i) = u(i) = 0$, $\forall\, i \in N$ and $\lambda = 0$.
2. For each $i \in I^-$, if $s(j) > 0$ for all $j \in N(i)$, then set $w(i) \leftarrow w(i) + \min\{|s(i)|, \min_{j \in N(i)} s(j)\}$ and update $s(j)$ for $j \in N(i) \cup \{i\}$ and $I^-$.
3. If $|I^-| > k$, set $\lambda \leftarrow \lambda + \min\{|s(i)| \mid i \in I^-\}$ and update $s(i)$ for $i \in N$ and $I^-$.
4. If any update in steps 2 or 3, go to step 2.
5. For each $i \in I^-$, set $u(i) \leftarrow |s(i)|$ and $s(i) \leftarrow 0$.

## 3.2 Calculating lower bounds

Now we consider how to obtain a lower bound

(LB) for (P). Our algorithm, called LB, is an add-type heuristic that iteratively selects a subset of cells not more than $k$ such that each cell in the subset is selected together with at least one neighboring cell. In each iteration, we permanently include a cell $i \in N$ into the list of the selected cells, say SELECT, when either cell $i$ is neighboring to any cell in SELECT, or cell $i$ can be included together with a neighboring cell in the candidate set, called CANDID. CANDID is an initially null set and includes cells that failed to be included into SELECT in the previous iterations.

As different sequences of scanning cells for selection produce different solutions, we have tried two different strategies for sequencing. The first strategy uses information on dual solutions obtained through the upper bounding process. We classify the nodes into two groups, $I^0$ and $N-I^0$ and select a cell in $I^0$ before any cell in $N-I^0$. Among the cells in the same group, a cell with larger traffic demand, $d(i)$, is scanned earlier. The second strategy simply considers a cell with the largest traffic demand first.

**Algorithm LB**

1. Initially set SELECT=$\varnothing$ and CANDID=$\varnothing$.
2. while |SELECT|<$k$ do the following steps.
2.1 Select a cell $i \in N$-(SELECT$\cup$CANDID) according to a predetermined sequence. If any cell neighboring to cell $i$ already exists in SELECT, then include cell $i$ into SELECT. Otherwise, go to the next step.
2.2 If |SELECT|$\geq k-1$, include cell $i$ into CANDID and return to step 2.1. Otherwise, go to the next step.
2.3 If a cell neighboring to cell $i$ exists in CANDID, include cell $i$ into SELECT. Additionally include the cells of CANDID neighboring to cell $i$ into SELECT until |SELECT|=$k$. If CANDID has no cell neighboring to cell $i$, include cell $i$ into CANDID and return to step 2.1.

## 4. Computational Results

The proposed algorithm for calculating lower and upper bounds of the problem was coded in the language C and test runs were performed on a PC with 1.4GHz Pentium M processor. We performed computational experiments using two classes of problems. In the first set of test problems, all cells are assumed as omni-type with hexagonal form and in the second set of test problems, all cells are assumed to have non hexagonal form. The traffic demand of each cell is given as downlink load factor (see ref. [4]).

We have tested 1000 data instances with different numbers of cells, different values of $k$, and different neighboring relation. In the first set of test problems, the networks have cells from 129 to 529. The second set includes the problems with rather large networks having cells from 500 to 2000. We test the different levels of the maximum number of soft-combined cells by setting $k$ equal to 10%, 20%, 30%, 40%, 50% of the total number of cells. In a network of hexagonal cells, each cell has six neighboring cells. In a network of non-hexagonal cells, we set neighboring relation arbitrary and make three different cases according to the average number of neighbors per cell. In each case, the average numbers of neighbors per cell are 10%, 20%, and 30% of the total cells, respectively.

<Table 1> shows the results for the first set of test problems and <Table 2> for the second

set of test problems. We classified each group by the size of networks, the average number of neighbors, and the levels of $k$. We have tested 10 instances per group and present the average value of 10 instances for each group in the tables. Upper and lower bounds, denoted by UB and LB, respectively, were obtained using the procedures presented in Section 3.1 and 3.2. In order to evaluate our dual heuristic of solving the LP relaxation, we compare it with the optimal LP value. We also compare the computing time of the dual heuristic with the time spent to solve (LP) by CPLEX 9.0 callable library. We have compared the two different strategies of sequencing cells in Algorithm LB.

<Table 1> and <Table 2> show that our dual heuristic found an optimal or near optimal solution quickly and all the ratios of (UB-LB)/LB for the tested instances are within a reasonable range. We also have found that no big difference between the two strategies of sequencing cells in Algorithm LB.

〈Table 1〉 Computational results for hexagonal cell networks

| N | k | LP (CPLEX) | | UB | | LB(strategy 1) | | LB(strategy 2) | | Gap* |
|---|---|---|---|---|---|---|---|---|---|---|
| | | value | CPU (sec) | value | CPU (sec) | value | CPU (sec) | value | CPU (sec) | |
| 129 | 13 | 11.84 | 0.0060 | 11.91 | 0.0000 | 11.74 | 0.0000 | 11.70 | 0.0000 | 0.01 |
| | 26 | 22.53 | 0.0080 | 22.64 | 0.0000 | 22.23 | 0.0000 | 22.05 | 0.0000 | 0.02 |
| | 39 | 32.11 | 0.0080 | 32.17 | 0.0000 | 31.36 | 0.0000 | 31.25 | 0.0000 | 0.02 |
| | 52 | 40.63 | 0.0080 | 40.70 | 0.0000 | 39.44 | 0.0000 | 39.27 | 0.0000 | 0.03 |
| | 65 | 47.95 | 0.0050 | 47.98 | 0.0010 | 45.99 | 0.0000 | 45.83 | 0.0000 | 0.04 |
| 228 | 23 | 21.20 | 0.0100 | 21.25 | 0.0010 | 21.07 | 0.0000 | 21.06 | 0.0000 | 0.01 |
| | 46 | 40.25 | 0.0100 | 40.34 | 0.0000 | 39.71 | 0.0010 | 39.65 | 0.0000 | 0.02 |
| | 68 | 56.93 | 0.0090 | 57.06 | 0.0010 | 55.70 | 0.0000 | 55.55 | 0.0020 | 0.02 |
| | 91 | 72.32 | 0.0090 | 72.37 | 0.0010 | 70.15 | 0.0010 | 70.03 | 0.0000 | 0.03 |
| | 114 | 85.28 | 0.0090 | 85.31 | 0.0000 | 82.03 | 0.0010 | 82.11 | 0.0000 | 0.04 |
| 336 | 34 | 31.28 | 0.0150 | 31.36 | 0.0000 | 31.09 | 0.0010 | 30.99 | 0.0000 | 0.01 |
| | 67 | 58.87 | 0.0140 | 59.03 | 0.0010 | 58.13 | 0.0010 | 57.92 | 0.0010 | 0.02 |
| | 101 | 84.44 | 0.0110 | 84.57 | 0.0020 | 82.70 | 0.0010 | 82.50 | 0.0000 | 0.02 |
| | 134 | 106.79 | 0.0100 | 106.87 | 0.0040 | 103.93 | 0.0020 | 103.75 | 0.0000 | 0.03 |
| | 168 | 125.97 | 0.0100 | 126.03 | 0.0040 | 121.54 | 0.0000 | 121.50 | 0.0000 | 0.04 |
| 428 | 43 | 39.83 | 0.0130 | 39.93 | 0.0060 | 39.55 | 0.0030 | 39.42 | 0.0010 | 0.01 |
| | 86 | 76.05 | 0.0140 | 76.23 | 0.0060 | 75.05 | 0.0010 | 74.66 | 0.0000 | 0.02 |
| | 128 | 107.21 | 0.0130 | 107.40 | 0.0060 | 104.62 | 0.0020 | 104.57 | 0.0000 | 0.03 |
| | 171 | 135.74 | 0.0120 | 135.82 | 0.0050 | 131.44 | 0.0030 | 131.56 | 0.0000 | 0.03 |
| | 214 | 159.88 | 0.0160 | 159.91 | 0.0020 | 154.01 | 0.0010 | 153.88 | 0.0000 | 0.04 |
| 529 | 53 | 48.96 | 0.0160 | 49.10 | 0.0080 | 48.68 | 0.0040 | 48.56 | 0.0010 | 0.01 |
| | 106 | 93.47 | 0.0140 | 93.71 | 0.0100 | 92.25 | 0.0050 | 91.79 | 0.0000 | 0.02 |
| | 159 | 133.46 | 0.0160 | 133.64 | 0.0040 | 130.36 | 0.0080 | 130.17 | 0.0010 | 0.03 |
| | 212 | 168.61 | 0.0140 | 168.74 | 0.0060 | 163.57 | 0.0040 | 163.09 | 0.0010 | 0.03 |
| | 265 | 197.65 | 0.0130 | 197.68 | 0.0060 | 190.80 | 0.0030 | 190.63 | 0.0010 | 0.04 |

주) * GAP=(UB-best LB)/best LB

〈Table 2〉 Computational results for non-hexagonal cell networks

| N | Density | k | LP(CPLEX) value | LP(CPLEX) CPU (sec) | UB value | UB CPU (sec) | LB(strategy 1) value | LB(strategy 1) CPU (sec) | LB(strategy 2) value | LB(strategy 2) CPU (sec) | Gap* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 0.1 | 10 | 9.33 | 0.0070 | 9.37 | 0.0000 | 9.21 | 0.0000 | 9.20 | 0.0000 | 0.01 |
| | | 20 | 17.61 | 0.0050 | 17.63 | 0.0000 | 17.08 | 0.0000 | 17.08 | 0.0000 | 0.03 |
| | | 30 | 25.46 | 0.0080 | 25.48 | 0.0000 | 24.57 | 0.0000 | 24.60 | 0.0000 | 0.04 |
| | | 40 | 31.14 | 0.0070 | 31.14 | 0.0010 | 29.45 | 0.0000 | 29.47 | 0.0000 | 0.06 |
| | | 50 | 37.32 | 0.0080 | 37.33 | 0.0000 | 36.04 | 0.0000 | 36.05 | 0.0010 | 0.04 |
| | 0.2 | 10 | 9.48 | 0.0050 | 9.49 | 0.0000 | 9.35 | 0.0000 | 9.36 | 0.0000 | 0.01 |
| | | 20 | 17.73 | 0.0060 | 17.73 | 0.0000 | 17.35 | 0.0000 | 17.35 | 0.0000 | 0.02 |
| | | 30 | 25.36 | 0.0061 | 25.36 | 0.0000 | 24.87 | 0.0010 | 24.87 | 0.0000 | 0.02 |
| | | 40 | 31.82 | 0.0060 | 31.82 | 0.0000 | 31.16 | 0.0000 | 31.16 | 0.0000 | 0.02 |
| | | 50 | 37.41 | 0.0050 | 37.41 | 0.0010 | 36.50 | 0.0000 | 36.50 | 0.0000 | 0.03 |
| | 0.3 | 10 | 9.48 | 0.0060 | 9.48 | 0.0010 | 9.39 | 0.0000 | 9.39 | 0.0000 | 0.01 |
| | | 20 | 18.12 | 0.0060 | 18.12 | 0.0010 | 17.83 | 0.0000 | 17.83 | 0.0000 | 0.02 |
| | | 30 | 25.19 | 0.0070 | 25.19 | 0.0010 | 24.86 | 0.0000 | 24.86 | 0.0000 | 0.01 |
| | | 40 | 31.58 | 0.0070 | 31.58 | 0.0000 | 31.17 | 0.0010 | 31.17 | 0.0000 | 0.01 |
| | | 50 | 37.23 | 0.0060 | 37.23 | 0.0000 | 36.50 | 0.0000 | 36.50 | 0.0000 | 0.02 |
| 500 | 0.1 | 50 | 47.52 | 0.0292 | 47.52 | 0.0110 | 47.09 | 0.0030 | 47.09 | 0.0010 | 0.01 |
| | | 100 | 89.65 | 0.0290 | 89.65 | 0.0100 | 88.92 | 0.0020 | 88.92 | 0.0010 | 0.01 |
| | | 150 | 126.78 | 0.0271 | 126.78 | 0.0100 | 125.45 | 0.0050 | 125.45 | 0.0010 | 0.01 |
| | | 200 | 160.67 | 0.0300 | 160.67 | 0.0090 | 158.89 | 0.0020 | 158.89 | 0.0000 | 0.01 |
| | | 250 | 186.31 | 0.0300 | 186.31 | 0.0091 | 183.68 | 0.0010 | 183.68 | 0.0000 | 0.01 |
| | 0.2 | 50 | 47.33 | 0.0571 | 47.33 | 0.0150 | 47.15 | 0.0030 | 47.15 | 0.0000 | 0.00 |
| | | 100 | 90.29 | 0.0531 | 90.29 | 0.0140 | 90.06 | 0.0040 | 90.06 | 0.0010 | 0.00 |
| | | 150 | 126.64 | 0.0531 | 126.64 | 0.0130 | 126.00 | 0.0040 | 126.00 | 0.0020 | 0.01 |
| | | 200 | 161.63 | 0.0540 | 161.63 | 0.0150 | 160.86 | 0.0010 | 160.86 | 0.0011 | 0.00 |
| | | 250 | 189.29 | 0.0550 | 189.29 | 0.0100 | 188.08 | 0.0030 | 188.08 | 0.0000 | 0.01 |
| | 0.3 | 50 | 47.53 | 0.0812 | 47.53 | 0.0200 | 47.36 | 0.0000 | 47.36 | 0.0000 | 0.00 |
| | | 100 | 90.02 | 0.0810 | 90.02 | 0.0171 | 89.82 | 0.0030 | 89.82 | 0.0030 | 0.00 |
| | | 150 | 127.25 | 0.0791 | 127.25 | 0.0181 | 126.70 | 0.0020 | 126.70 | 0.0020 | 0.00 |
| | | 200 | 160.52 | 0.0822 | 160.52 | 0.0130 | 159.85 | 0.0020 | 159.85 | 0.0010 | 0.00 |
| | | 250 | 186.87 | 0.0811 | 186.87 | 0.0140 | 186.49 | 0.0030 | 186.49 | 0.0020 | 0.00 |
| 1000 | 0.1 | 100 | 95.06 | 0.1232 | 95.06 | 0.0531 | 94.62 | 0.0130 | 94.62 | 0.0040 | 0.00 |
| | | 200 | 179.26 | 0.1281 | 179.26 | 0.0521 | 178.25 | 0.0110 | 178.25 | 0.0050 | 0.01 |
| | | 300 | 255.36 | 0.1241 | 255.36 | 0.0431 | 253.97 | 0.0140 | 253.97 | 0.0050 | 0.01 |
| | | 400 | 321.84 | 0.1251 | 321.84 | 0.0470 | 320.20 | 0.0121 | 320.20 | 0.0040 | 0.01 |
| | | 500 | 376.02 | 0.1262 | 376.02 | 0.0351 | 373.68 | 0.0120 | 373.68 | 0.0030 | 0.01 |
| | 0.2 | 100 | 94.93 | 0.2212 | 94.93 | 0.0662 | 94.77 | 0.0110 | 94.77 | 0.0040 | 0.00 |
| | | 200 | 179.55 | 0.2323 | 179.55 | 0.0622 | 179.16 | 0.0120 | 179.16 | 0.0030 | 0.00 |
| | | 300 | 256.73 | 0.2294 | 256.73 | 0.0550 | 256.01 | 0.0170 | 256.01 | 0.0030 | 0.00 |
| | | 400 | 322.49 | 0.2212 | 322.49 | 0.0532 | 321.45 | 0.0140 | 321.45 | 0.0050 | 0.00 |
| | | 500 | 376.75 | 0.2274 | 376.75 | 0.0502 | 375.57 | 0.0130 | 375.57 | 0.0030 | 0.00 |
| | 0.3 | 100 | 95.00 | 0.3193 | 95.00 | 0.0821 | 94.88 | 0.0140 | 94.88 | 0.0050 | 0.00 |
| | | 200 | 179.81 | 0.3163 | 179.81 | 0.0791 | 179.60 | 0.0130 | 179.60 | 0.0040 | 0.00 |
| | | 300 | 254.92 | 0.3143 | 254.92 | 0.0732 | 254.35 | 0.0130 | 254.35 | 0.0040 | 0.00 |
| | | 400 | 321.81 | 0.3157 | 321.81 | 0.0690 | 321.30 | 0.0120 | 321.30 | 0.0040 | 0.00 |
| | | 500 | 376.26 | 0.3157 | 376.26 | 0.0670 | 375.37 | 0.0110 | 375.37 | 0.0050 | 0.00 |

〈Table 2〉 계속

| N | Density | k | LP(CPLEX) | | UB | | LB(strategy 1) | | LB(strategy 2) | | Gap* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | value | CPU (sec) | value | CPU (sec) | value | CPU (sec) | value | CPU (sec) | |
| 1500 | 0.1 | 150 | 142.28 | 0.2704 | 142.28 | 0.1281 | 141.86 | 0.0280 | 141.86 | 0.0150 | 0.00 |
| | | 300 | 269.40 | 0.2674 | 269.40 | 0.1151 | 268.39 | 0.0271 | 268.39 | 0.0090 | 0.00 |
| | | 450 | 382.09 | 0.2725 | 382.09 | 0.1061 | 380.76 | 0.0270 | 380.76 | 0.0090 | 0.00 |
| | | 600 | 481.29 | 0.2614 | 481.29 | 0.0931 | 479.15 | 0.0290 | 479.15 | 0.0070 | 0.00 |
| | | 750 | 564.11 | 0.2574 | 564.11 | 0.0871 | 561.27 | 0.0280 | 561.27 | 0.0090 | 0.01 |
| | 0.2 | 150 | 142.48 | 0.5117 | 142.48 | 0.1705 | 142.31 | 0.0280 | 142.31 | 0.0291 | 0.00 |
| | | 300 | 269.78 | 0.5107 | 269.78 | 0.1573 | 269.47 | 0.0271 | 269.47 | 0.0232 | 0.00 |
| | | 450 | 382.31 | 0.5069 | 382.31 | 0.1451 | 381.78 | 0.0270 | 381.78 | 0.0160 | 0.00 |
| | | 600 | 480.82 | 0.5048 | 480.82 | 0.1301 | 479.59 | 0.0280 | 479.59 | 0.0141 | 0.00 |
| | | 750 | 561.64 | 0.5019 | 561.64 | 0.1141 | 560.49 | 0.0300 | 560.49 | 0.0200 | 0.00 |
| | 0.3 | 150 | 142.13 | 0.7841 | 142.13 | 0.2041 | 141.97 | 0.0271 | 141.97 | 0.0180 | 0.00 |
| | | 300 | 268.77 | 0.7692 | 268.77 | 0.2022 | 268.58 | 0.0270 | 268.58 | 0.0130 | 0.00 |
| | | 450 | 382.09 | 0.7621 | 382.09 | 0.1811 | 381.65 | 0.0380 | 381.65 | 0.0200 | 0.00 |
| | | 600 | 481.67 | 0.7740 | 481.67 | 0.1853 | 481.04 | 0.0290 | 481.04 | 0.0200 | 0.00 |
| | | 750 | 563.04 | 0.7692 | 563.04 | 0.1643 | 562.20 | 0.0301 | 562.20 | 0.0160 | 0.00 |
| 2000 | 0.1 | 200 | 189.73 | 0.5087 | 189.73 | 0.2224 | 189.17 | 0.0530 | 189.17 | 0.0201 | 0.00 |
| | | 400 | 358.45 | 0.5057 | 358.45 | 0.2134 | 357.31 | 0.0540 | 357.31 | 0.0220 | 0.00 |
| | | 600 | 512.18 | 0.5017 | 512.18 | 0.1933 | 510.88 | 0.0560 | 510.88 | 0.0220 | 0.00 |
| | | 800 | 640.90 | 0.4939 | 640.90 | 0.1721 | 639.36 | 0.0560 | 639.36 | 0.0241 | 0.00 |
| | | 1000 | 751.47 | 0.4916 | 751.47 | 0.1583 | 748.94 | 0.0571 | 748.94 | 0.0220 | 0.00 |
| | 0.2 | 200 | 190.04 | 0.9432 | 190.04 | 0.2887 | 189.87 | 0.0530 | 189.87 | 0.0220 | 0.00 |
| | | 400 | 360.01 | 0.9521 | 360.01 | 0.2675 | 359.58 | 0.0651 | 359.58 | 0.0332 | 0.00 |
| | | 600 | 509.93 | 0.9295 | 509.93 | 0.2544 | 509.36 | 0.0570 | 509.36 | 0.0200 | 0.00 |
| | | 800 | 640.95 | 0.9405 | 640.95 | 0.2274 | 639.95 | 0.0600 | 639.95 | 0.0290 | 0.00 |
| | | 1000 | 749.33 | 0.9413 | 749.33 | 0.2143 | 748.45 | 0.0542 | 748.45 | 0.0200 | 0.00 |
| | 0.3 | 200 | 189.90 | 4.2982 | 189.90 | 0.3719 | 189.74 | 0.0520 | 189.74 | 0.0180 | 0.00 |
| | | 400 | 360.21 | 4.6416 | 360.21 | 0.3525 | 359.97 | 0.0511 | 359.97 | 0.0221 | 0.00 |
| | | 600 | 512.66 | 3.9858 | 512.66 | 0.3102 | 512.28 | 0.0522 | 512.28 | 0.0200 | 0.00 |
| | | 800 | 640.49 | 4.0749 | 640.49 | 0.3174 | 639.70 | 0.0681 | 639.70 | 0.0310 | 0.00 |
| | | 1000 | 746.76 | 3.6753 | 746.76 | 0.3005 | 746.30 | 0.0551 | 746.30 | 0.0170 | 0.00 |

주) * GAP=(UB-best LB)/best LB

## 5. Conclusion

In this paper, we have dealt with the problem of finding the optimal set of cells to be soft-combined such that for each selected cell, at least one cell neighboring to it also has to be selected. Our objective is to maximize the total traffic demand of the selected cells. This problem has an application in cell planning for a CDMA based multicast network. We have shown that the problem is *NP-hard* and introduced a 0-1 integer programming model for the problem. We also presented an algorithm that provides lower and upper bounds of the problem. Computational results have shown that the proposed algorithm is a practical one.

## References

[1] 3GPP TR 25.803 S-CCPCH performance for MBMS, Sep. 2005.

[2] 3GPP TS 22.146 Multimedia Broadcast/Multicast Service : Stage1, 3GPP, Mar. 2003.

[3] Balakrishnan, A., T.L. Magnanti and R.T. Wong, "A dual-ascent procedure for large-scale uncapacitated network design," *Operations Research*, Vol.37(1989), pp.716-740.

[4] Boni, A. et al., "Multimedia Broadcast Multicast Service - Technology Overview and Service Aspects," Fifth IEE International Conference on 3G Mobile Communication Technologies, 2004.

[5] Broadcast-Multicast Service (BCMCS) Framework Draft Document, 3GPP2, 2003.

[6] CDMA2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification, 3GPP2 TSG-C SWG2.5, Aug. 2003.

[7] Chudak, F., T. Roughgarden, and D.P. Williamson, "Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation," *Mathematical Programming*, Vol.100(2004), pp.411-421.

[8] Erlenkotter, D., "A dual based procedure for uncapacitated facility location," *Operations Research*, Vol.26(1978), pp.992-1009.

[9] Fischetti, M., H. Hamacher, K. Jørnsten, and F. Maffioli, "Weighted k-Cardinality Trees : Complexity and Polyhedral Structure," *Networks*, Vol.24(1994), pp.11-21.

[10] Garey, M.R. and D.S. Johnson, *Computers and intractability : a Guide to the theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.

[11] Garg, N., "A 3-Approximation for the Minimum Tree Spanning k Vertices," Proceedings of the 37th Annual Symposium on Foundations of Computer Science, (1996), pp.302-309.

[12] Holma, H. and A. Toskala, *WCDMA for UMTS*, Wiley, 2000.

[13] Tatesh, S. and S.K. Palat, "Future Evolution of 3G Networks," International Conference on Information and Communication Technologies : From Theory to Applications, 2004.

[14] Wang, J., R. Sinnarajah, T. Chen, Y. Wei, and E. Tiedemann, "Broadcast and Multicast Services in cdma 2000," *IEEE communications magazine*, 2004.

[15] Wong, R.T., "A dual ascent approach for Steiner tree problems on a directed graph," *Mathematical Programming*, Vol.28(1984), pp.271-287.