An efficient algorithm for generalized SS/TDMA scheduling with satellite cluster

Sehun Kim
and
Soo-Hyun Kim

Department of Management Science
Korea Advanced Institute of Science and Technology
373-1, Kusong-Dong, Yusong-Gu
Teajon 305-701, Korea
E-mail: shkim@mgt.kaist.ac.kr

Summary

We consider the satellite cluster scheduling problem which is one of the most interesting problems in satellite communication scheduling area. This problem is known to be NP-complete and a couple of heuristic algorithms had been developed. In this paper, we suggest another algorithm for this problem which has the same computational complexity as the best existing one and provides much better solution quality. Extensive computational simulation results are reported.
Key words : SS/TDMA, Satellite Cluster Scheduling Problem, ISL.

I. INTRODUCTION

As the demands of satellite communication rapidly grow, the natural resources they use, the RF spectrum and the geosynchronous orbit, are becoming highly crowded. The utilization of the above natural resources is optimized by employing multibeam antennae and satellite-switched time-division multiple-access *(SS/TDMA)* techniques[1-8]. In an SS/TDMA system, a satellite has a number of spot beam antennas, each providing coverage for a limited geographical zone. The solid-state RF switch on-board the satellite provides connections between the various uplink and downlink beams according to the TDMA frame. The frame is divided into time slots. Each time slot represents a particular switching matrix configuration, which allows to transmit a certain number of packets between the connected uplink and downlink beams without conflict. Single satellite SS/TDMA systems were extensively studied under various optimization criterions[1-8].

In many practical situations, ground stations exchanging traffic are not always visible by the same satellite. Current practices involve either the use of ground communication lines or rerouting the traffic via an intermediate ground station which is in the line of sight of two satellites, one visible by the transmitter and the other visible by the receiver. In both cases, extra earth resources are used, thus reducing the total efficiency of the system. A more efficient solution for the above problem is the interconnection of satellites by ***intersatellite links (ISL)***, creating a satellite communication network[9-11]. The ISL system characteristics that can be utilized for supporting the existing systems are : coverage extension, improved orbit/spectrum utilization, reduced number of earth station antennas, improved transmission quality, and continued service cost reduction[11]. Therefore SS/TDMA satellites with no on-board buffering, interconnected through intersatellite links, are very promising.

However, in this case we need to solve the satellite cluster scheduling problem which has to account for additional constraints in addition to the single satellite SS/TDMA scheduling constraints. Specifically, the continuous assignment from the source earth zone to the destination earth zone through the ISL should account for 1) transmission *conflicts* on the ISL's and 2) the traffic arriving from the ISL should be immediately switched to the appropriate downlink.

It has been proven that the time slot assignment problem for satellite clusters with an arbitrary number of satellites is NP-complete, even for quite restricted intersatellite link patterns and simplified models[9]. Heuristic algorithms were presented in Reference 9 for two satellites, each covering similar number of disjoint ground stations, and one ISL. In Reference 11, presented was a heuristic algorithm for an arbitrary number of satellites, each covering an arbitrary number of disjoint zones and an *arbitrary* configuration of interconnection through an *arbitrary* number of intersatellite links. The algorithm is based on the algorithm for openshop scheduling problem, and has $O(rM^2)$ computational complexity, where r is the number of non-zero elements in the traffic matrix, M is the number of geographical zones. However, the quality of its solutions is not quite satisfactory. The simulation result in Reference 11 shows that the solution could be deviated from its optimal solution by 9.03%.

In this paper, we suggest a new simple algorithm for the SS/TDMA slot assignment problem for a satellite cluster. This algorithm is based on the observation that the traffic demand from one satellite to another with the smallest number of intersatellite communication channels may cause a bottleneck in the scheduling and, hence, should be scheduled first. The algorithm has $O(rM^2)$ computational complexity like the one in Reference 11. However, computational test shows that our algorithm generates much better solutions than the one in Reference 11 in most cases. Furthermore, we generalize the SS/TDMA slot assignment problem in one more respect. In practice, each satellite may have multiple on-board transponders. Actually, many researchers have investigated the system with an arbitrary number of transponders in the case of single satellite system[1−4]. In this paper, we consider a more generalized model which has not only an ***arbitrary*** number of satellites, each covering an ***arbitrary*** number of disjoint zones, and an ***arbitrary*** configuration of interconnection through an ***arbitrary*** number of intersatellite links, but also an ***arbitrary*** number of on-board transponders.

This paper is organized in the following way. The problem fomulation and a theoretical lower bound on the switching duration are given in Section II. Section III presents our heuristic algorithm. In Section IV, two examples are presented for the demonstration of our algorithm and extensive computational test results of our algorithm are reported and compared with the one in Reference 11. Section V concludes the paper.

II. PROBLEM FORMULATION

We consider a cluster consisting of S satellites, $C = \{1, 2, \ldots, S\}$, and a set of M disjoint geographical zones, $Z = \{1, \ldots, M\}$. Satellite p in the cluster covers a subset, Z_p, of Z. We assume that no zone is covered by more than one satellite. Hence, $Z_p \cap Z_q = \emptyset$ for two different satellites p, q. All uplinks and downlinks are assumed to have equal bandwidth. For two different satellites p, q, there are l_{pq} intersatellite links from satellite p to satellite q. In addition, there are l_{pp} transponders in satellite p.

The traffic demand is chracterized by an $M \times M$ matrix D with entry d_{ij} representing the amount of traffic from uplink beam (source zone) i to downlink beam (destination zone) j, measured in time slot units. If zone i is visible by satellite p, zone j by satellite q ($q \neq p$), and the two satellites are not interconnected by ISL's, then d_{ij} and d_{ji} are equal to 0. We denote by $D(p,q)$ the $|Z_p| \times |Z_q|$ submatrix of D representing the traffic between zones visible by satellite p and zones visible by satelite q. Of course $D(p,p)$ is the $|Z_p| \times |Z_p|$ submatrix of D representing the traffic between zones visible by satellite p alone. We refer to $D(p,q)$ as the ***intersatellite submatrix***. Note that the transmission of the traffic in the intersatellite submatrix $D(p,q)$ requires both a transponder and an ISL simultaneously. $D(p,\cdot)$ is $|Z_p| \times |Z|$ submatrix of D representing the traffic originating from zones in Z_p. Also $D(\cdot,q)$ is $|Z| \times |Z_q|$ submatrix of D representing the traffic arriving to zones in Z_q. We shall use ***line*** to refer either to a row or to a column of a matrix.

The scheduling algorithm has to decompose the given traffic matrix D into distinct ***switching matrices***, $D = D_1 + D_2 + \ldots + D_n$ where n denotes the number of ***switching configurations***. Such a matrix characterizes a particular switching configuration and its corresponding traffic load being switched without ***conflict***. To obtain a conflict free assignment, a switching matrix D_i must be an $M \times M$ matrix with at most one positive entry in each line, at most l_{pq} positive entries in each submatrix corresponding to $D(p,q)$ where $p \neq q$, and at most $l_{pp}(l_{qq})$ positive entries in each submatrix corresponding to $D(p,\cdot)(D(\cdot,q))$, respectively. The largest entry in a switching matrix D_i dictates the ***switching duration*** of D_i denoted by L_i. The ***total duration*** needed to schedule the complete traffic matrix D is given by $L = L_1 + L_2 + \ldots + L_n$. A schedule for D is ***optimal*** if its total duration is minimal.

Bertossi, *et al.* considered a special case when there are two satellites, each has $|Z_p|$ transponders and one ISL[9]. Ganz, *et al.* considered the case where there are an arbitrary number of satellites and each satellite p in the cluster has $|Z_p|$ transponders and an arbitrary number of ISL's[11]. In our paper, we consider the case when there are an arbitrary number of satellites and each satellite p has l_{pp} ($1 \leq l_{pp} \leq |Z_p|$) transponders and an arbitrary number of ISL's.

We first genenalize the theoretical ***lower bound*** on the minimal duration given in Reference 11 to our system with an arbitrary number of transponders. We denote by r_i the sum of entries in the ith row of the traffic matrix D and by c_j the sum of entries in the jth column of D. Let $T(p,q)$ denote the amount of traffic in the submatrix $D(p,q)$, that is, the sum of all entries in $D(p,q)$. Let $T(p,\cdot)$ ($T(\cdot,q)$) denote the amount of traffic in the submatrix $D(p,\cdot)$ ($D(\cdot,q)$), respectively.

Theorem 1 : Any schedule for D has length not smaller than

$$LB = max \left\{ \begin{array}{l} \max_{1\leq i\leq M}\{r_i\};\ \max_{1\leq i\leq M}\{c_i\}; \\ \max_{1\leq p,q\leq S, p\neq q}\left\{\left\lceil \frac{T(p,q)}{l_{pq}} \right\rceil\right\}; \\ \max_{1\leq p\leq S}\left\{\left\lceil \frac{T(p,\cdot)}{l_{pp}} \right\rceil\right\};\ \max_{1\leq q\leq S}\left\{\left\lceil \frac{T(\cdot,q)}{l_{qq}} \right\rceil\right\}; \end{array} \right\}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x.

Proof : All entries in the same line must be transmitted sequentially to avoid conflicts. And entries in the intersatellite submatrix can be transmitted at most l_{pq} units in a time unit. Also, entries in the submatrix $D(p,\cdot)$ ($D(\cdot,q)$) can be transmitted at most l_{pp} (l_{qq}) units in a time unit, respectively. Hence a lower bound is given as above. ■

III. ALGORITHM

The optimal time slot assignment problem has been proved to be *NP-complete*[9]. In this paper we present a fast heuristic algorithm called *SCS* (Satellite Cluster Scheduling). Let the *degree* of row i (column j) denoted by d_i^r (d_j^c) be the number of non-zero elements in row i (column j), respectively.

There are l_{pq} communication channels from satellite p to satellite q. The traffic demand from one satellite to another with the smallest number of communication channels l_{pq} may cause a bottleneck in the scheduling. Hence this demand should be scheduled first to avoid unnecessary delay. Futhermore, the degree of a line in D implies the number of alternatives in selecting a cell in the line. The traffic demand between uplink and downlink zones in $D(p,q)$ with the smallest row and column degrees in D may also cause a bottleneck in the scheduling. Hence this demand should be scheduled first.

Based on the above arguments, we first choose a submatrix $D(p,q)$ which has the smallest value of l_{pq}, $p,q \in C$. If we do not have available transponders in satellite p and satellite q, then all the remaining lines in the submatrix are removed and the degree of each line is updated. Then we choose the next pair of satellites with the next smallest l_{pq}. Otherwise, we find a row i with minimum degree and find a column j with minimum degree and non-zero d_{ij}, where d_{ij} is the (i,j) element in D. If we find more than one column with the minimum degree, choose the column j with maximum remaining traffic d_{ij} since it is desirable to transport the maximum traffic first through the switching matrix. The entry in the selected cell is stored in the switching matrix. Since a switching matrix can have only one nonzero entry in each line, remove the row i and column j from the traffic matrix and update the degree of lines. And then choose the next cell in $D(p,q)$ similarly and repeat this procedure until either the number of selected cells is equal to l_{pq}, or no available transponders exist, or all lines in the submatrix are removed. In the first and second case, all remaining lines in the submatrix are also removed and the degree of each line is updated.

Now, we have obtained a temporary switching matrix from zones in Z_p to zones in Z_q. Then we choose the next pair of satellites with the next smallest l_{pq} and construct a temporary switching matrix between this pair of satellites in a similar way. This process continues until no lines are left in the traffic matrix. Then we come to obtain a complete temporary switching matrix. Since the duration time of this switching matrix is equal to the value of its largest non-zero entry, it is economical to truncate all its value to the value of its smallest non-zero entry. Now we have constructed the first complete switching matrix D_1. The resultant switching matrix is subtracted from the original traffic matrix D. The whole process explained above is repeated until no traffic is left in D.

This algorithm is described below in detail. Let us denote the number of non-zero elements in row i (column j) of $D(p,q)$ by $d_i^r(q)$ ($d_j^c(p)$) respectively.

Algorithm *SCS (Satellite Cluster Scheduling)* :
STEP 0 : *(Initialization)*

Let D be a given traffic matrix. Set $k \leftarrow 1$.
Each ordered pair (p, q) is indexed from 1 to S^2 according to the ascending order of l_{pq}.

STEP 1 : *(Obtaining a temporary switching matrix)*

(1.0) $D_k \leftarrow 0; ID \leftarrow 1$.
$\bar{D} \leftarrow D; T^r(p) \leftarrow l_{pp}, T^c(p) \leftarrow l_{pp}$ for all $p \in C$.
Find $d_i^r(q)$ for all $i \in Z, q \in C$.
Set $d_i^r \leftarrow \sum_{q=1}^{s} d_i^r(q)$ for all $i \in Z$.
Find $d_j^c(p)$ for all $j \in Z, p \in C$.
Set $d_j^c \leftarrow \sum_{p=1}^{s} d_j^c(p)$ for all $j \in Z$.

(1.1) Select the ordered pair $(\bar{p}, \bar{q})$ with index ID.

(1.2) *(Obtaining a temporary switching matrix for $(\bar{p}, \bar{q})$)*

(1.2.0) If $(T^r(\bar{p})) > 0)$ and $(T^c(\bar{q})) > 0)$ then goto Step (1.2.1). Otherwise, goto Step (1.3).

(1.2.1) *(Select a cell)*
$i^* \leftarrow \{i \mid min \ \{d_i^r \mid i \in Z_{\bar{p}}, d_i^r(\bar{q}) \neq 0\}\}$.
If i^* is not found, goto Step (1.4). Otherwise,
$j^* \leftarrow \{j \mid min \ \{d_j^c \mid j \in Z_{\bar{q}}, \bar{d}_{i^*j} \neq 0\}\}$. If we find two or more j^*, then choose j^* with the largest $\bar{d}_{i^*j}$.
Store (i^*, j^*) entry in the switching matrix D_k.
$T^r(\bar{p}) \leftarrow T^r(\bar{p}) - 1; T^c(\bar{q}) \leftarrow T^c(\bar{q}) - 1$.

(1.2.2) *(Update the degree of lines)*
For each non-zero $\bar{d}_{i^*j} (j \in Z)$, $d_j^c \leftarrow d_j^c - 1; \bar{d}_{i^*j} \leftarrow 0$;
if $j \in Z_p, p \in C$ then $d_j^c(p) \leftarrow d_j^c(p) - 1$.
For each non-zero $\bar{d}_{ij^*} (i \in Z)$, $d_i^r \leftarrow d_i^r - 1; \bar{d}_{ij^*} \leftarrow 0$;
if $i \in Z_p, p \in C$ then $d_i^r(p) \leftarrow d_i^r(p) - 1$.
$d_{i^*}^r(p) \leftarrow 0, d_{j^*}^c(p) \leftarrow 0$ for all $p \in C; d_{i^*}^r \leftarrow 0; d_{j^*}^c \leftarrow 0$.

(1.2.3) If (the number of selected cells is less than $l_{\bar{p}\bar{q}}$) then goto Step (1.2.0). Otherwise, goto Step (1.3).

(1.3) For each non-zero $d_i^r(\bar{q}), d_j^c(\bar{p})$,
$d_i^r \leftarrow d_i^r - d_i^r(\bar{q}); d_j^c \leftarrow d_j^c - d_j^c(\bar{p}); d_i^r(\bar{q}) \leftarrow 0; d_j^c(\bar{p}) \leftarrow 0$.
Goto Step (1.4).

(1.4) If non-zero $d_i^r, d_j^c (i, j \in Z)$ exist, $ID \leftarrow ID + 1$; goto Step (1.1).
Otherwise, goto Step 2.

STEP 2 : *(Obtaining a switching matrix)*
Find the smallest non-zero entry, let d^*, in D_k.
Form a switching matrix D_k by truncating all its non-zero entries to the value of d^*.

STEP 3 : *(Adjusting D)*
Set $D \leftarrow D - D_k$. If D contains no non-zero entry, then STOP.
Otherwise, set $k \leftarrow k + 1$; goto STEP 1. ∎

Theorem 2 : The overall time complexity of the algorithm *SCS* is $O(rM^2)$, where r is the number of non-zero elements in D, M is the number of zones.

Proof : $O(M^2)$ time is needed to calculate the degree of lines in step (1.0). Selecting a cell in step (1.2.1) requires $O(M)$ time, because $|Z_{\bar{p}}|$ is less than M and $d_{i^*}^r(\bar{q})$ is less than M. Step (1.2.2) requires $O(M)$ time, since the number of non-zero elements in a line is at most M. At most $O(M)$ iterations of step (1.2.1) and (1.2.2) are needed to find a switching matrix. It is because a switching matrix has at most M non-zero elements and the total number of transponders is less than or equal to M. And step (1.3) requires $O(SM)$ time to find a switching matrix, where S is the number of satellites, since the number of non-zero $d_i^r(q), i \in Z, q \in C$ is at most SM. In the worst case, the number of switching matrices generated is $O(r)$, since at least one non-zero entry is entirely scheduled in each switching matrix. Thus the worst case overall time complexity of this algorithm is $O(rM^2 + rM^2 + rSM) = O(rM^2)$ since $S \leq M$. ∎

IV. EXAMPLES AND SIMULATION RESULTS

IV-1. Examples

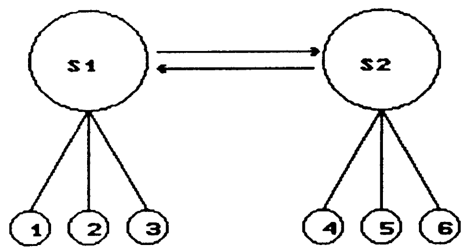
In this section we demonstrate our algorithm for two example systems in Reference 11. For each example we present the scheduling provided by SCS and the scheduling provided in Reference 11 for the purpose of comparison. The examples are as follows :

Example 1 : There are a cluster of two satellites, each satellite covering four zones. There are two intersatellite links from satellite 1 to satellite 2, and one intersatellite link from satellite 2 to satellite 1. The zones covered by satellite 1 and 2 are $Z_1 = \{1,2,3,4\}$, $Z_2 = \{5,6,7,8\}$, respectively. An 8×8 traffic matrix is shown in Fig. 1. The lower bound from Theorem 1 is 6. The total duration of this system obtained by SCS is 9, which is the same as that provided by Reference 11. In this case, two methods generate the same quality solutions.

$$D = \begin{pmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 1. 8×8 traffic matrix D.

Example 2 : There are a cluster of two satellites, each satellite covering three zones. There is one intersatellite link from each satellite to another. The zones covered by satellites 1 and 2 are $Z_1 = \{1,2,3\}$, $Z_2 = \{4,5,6\}$, respectively. A 6×6 traffic matrix is shown in Fig. 2. The lower bound from Theorem 1 is 3.



$$D = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Fig. 2. A cluster of two satellites, and the ISL traffic matrix D.

A. The scheduling provided by SCS

We obtain a decomposition of D into switching matrices with duration time 3. This duration is the same as the lower bound LB in Theorem 1. Hence our solution is an optimal solution. We will represent the switching matrix as a set of *(row,column)* having non-zero element in the switching matrix.

$D_1 = \{(1,6),(2,3),(3,1),(4,2),(5,4),(6,5)\}$;
$D_2 = \{(1,2),(2,5),(3,3),(4,4),(5,1),(6,6)\}$;
$D_3 = \{(1,1),(2,2),(3,4),(4,6),(5,5),(6,3)\}$.

B. The scheduling provided by Reference 11

In Reference 11 a decomposition of D into switching matrices with duration time 5 is obtained as follows

$D_1 = \{(1,1),(2,2),(3,3),(4,4),(5,5),(6,6)\}$;
$D_2 = \{(1,2),(2,3),(3,1),(4,6),(5,4),(6,5)\}$;
$D_3 = \{(1,6),(4,2)\}$;
$D_4 = \{(2,5),(5,1)\}$.
$D_5 = \{(3,4),(6,3)\}$.
In this case, our method generated a better quality solution than Reference 11.

IV-2. Simulation Results

SCS is implemented in Pascal and simulation test has been performed. Test examples have $S = 2,3,4, M = 6,8,12$. For each case we have applied the SCS to 100 matrices containing integers randomly generated from a uniform distribution between 0 and k, $k = 5, 10, 20, 50$. This random generation format is exactly the same as that in Reference 11. The lower bound LB, the average duration $\bar{L}$, the average surplus percentage from the lower bound are reported.

Table 1 shows the simulation results when each satellite has M/S transponders and each ordered pair of satellites has an ISL. This case can also be solved by the algorithm presented in Reference 11. Hence the results of both methods are reported in this Table 1. The table shows that our algorithm generates much better solutions in all cases than Reference 11. Furthermore, for problems 6 and 8-11, our algorithm generates exact optimal solutions.

In Table 1, we also report the average number of switching configurations $\bar{n}$ since this colud be another important factor in the SS/TDMA scheduling. Since both algorithms were not developed to minimize the number of switching configurations, both algorithms provide similar results for this numbers. It could be an interesting future research to adjust our algorithm so that it can also reduce the number of switching configurations.

Table 2-4 show the results, obtained by SCS, for more general systems with an *arbitrary* number of intesatellite links and on-board transponders. Even though the algorithm in Reference 11 was not developed to handle this case, we can easily modify this algorithm to solve this case also. The modified algorithm is denoted by [11]* and compared with ours. Table 2 is the results when there are two satellites, and $\{l_{pq}\}$ is given as

$$\{l_{pq}\} = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}.$$

Our algorithm generates schedules which have surplus from LB about 7 %.

Table 3 is the case where there are 3 satellites and $\{l_{pq}\}$ is given as

$$\{l_{pq}\} = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 4 \end{pmatrix}.$$

In this case, the surplus from LB is around 4 %.

Table 4 is the case where there are 4 satellites and

$$\{l_{pq}\} = \begin{pmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}.$$

The surplus from LB is around 6 %. Table 2-4 also show our algorithm generates much better solutions than [11]*.

Table 1. Computational Results

Prob.	S	M	k	LB	$\bar{L}$		Surplus from LB(%)		$\bar{n}$	
					SCS	[11]	SCS	[11]	SCS	[11]
1	2	6	5	26.20	26.47	27.23	0.69	3.23	19.17	18.82
2	2	6	10	51.01	51.33	53.77	0.38	4.83	25.70	29.94
3	2	6	20	97.59	98.22	101.66	0.58	4.22	29.94	29.57
4	2	6	50	255.17	256.92	264.78	0.64	4.31	33.19	33.21
5	2	8	5	43.35	43.37	43.74	0.05	0.68	30.75	30.36
6	2	8	10	88.24	88.24	89.17	0.00	1.26	43.01	42.60
7	2	8	20	171.27	171.28	171.82	0.01	0.75	50.72	51.13
8	2	8	50	433.96	433.96	439.25	0.00	1.06	58.67	58.26
9	2	12	5	95.90	95.90	96.15	0.00	0.28	63.67	63.35
10	2	12	10	187.94	187.94	188.54	0.00	0.44	87.17	86.28
11	2	12	20	376.17	376.17	377.89	0.00	0.44	107.25	108.70
12	3	12	5	48.65	50.94	52.55	4.23	7.39	45.83	45.83
13	3	12	10	95.23	99.20	103.34	4.39	8.91	72.53	72.96
14	3	12	20	189.58	198.39	204.46	4.08	7.88	98.52	98.89
15	4	12	5	40.69	41.32	48.05	1.60	19.86	39.11	42.47
16	4	12	10	81.26	82.65	94.72	1.98	15.91	65.34	68.30
17	4	12	20	161.13	162.92	190.16	1.09	16.32	93.01	96.18

Table 2. Computational Results

Prob.	S	M	k	LB	$\bar{L}$		Surplus from LB(%)		$\bar{n}$	
					SCS	[11]*	SCS	[11]*	SCS	[11]*
1	2	6	5	26.01	27.78	28.51	6.29	9.19	19.63	19.42
2	2	6	10	51.95	56.28	57.37	7.60	8.87	26.19	25.93
3	2	6	20	104.00	112.56	114.38	7.28	9.12	30.37	30.11

Table 3. Computational Results

Prob.	S	M	k	LB	$\bar{L}$		Surplus from LB(%)		$\bar{n}$	
					SCS	[11]*	SCS	[11]*	SCS	[11]*
1	3	12	5	63.31	66.10	68.85	4.59	8.97	55.31	56.01
2	3	12	10	124.99	130.64	135.82	3.99	8.40	82.96	83.80
3	3	12	20	247.85	259.50	269.84	4.92	9.83	107.46	107.33

Table 4. Computational Results

Prob.	S	M	k	LB	$\bar{L}$		Surplus from LB(%)		$\bar{n}$	
					SCS	[11]*	SCS	[11]*	SCS	[11]*
1	4	12	5	47.95	51.11	58.11	6.02	20.76	44.93	47.10
2	4	12	10	96.48	102.30	116.63	6.28	20.31	71.92	74.11
3	4	12	20	188.87	200.84	230.62	6.65	21.32	97.57	100.11

V. CONCLUSIONS

In this paper, we considered the satellite cluster scheduling problem which is one of the most interesting problems in satellite communication scheduling area. Our presented algorithm has low computational complexity and provides a solution very close to the optimal schedule.

This type of scheduling will be more important and interesting problems in the future when the low earth orbit satellite communication systems become more commonly used. In this case, the scheduling problem becomes dynamic. Extending our algorithm to cover this case would be an interesting future research work.

REFERENCES

1. G. Bongiovanni, D. Coppersmith, and C. K. Wong, "An optimal time slot assignment algorithm for an SS/TDMA system with variable number of transponders," *IEEE Trans. Com.*, **COM-29** (5), 721-726 (1981).
2. I. S. Gopal, G. Bongiovanni, M. A. Bonuccelli, D. T. Tang, and C. K. Wong, "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Trans. Com.*, **COM-30** (11), 2475-2481 (1982).
3. C. A. Pomalaza-Raez, "A note on efficient SS/TDMA assignment algorithms," *IEEE Trans. Com.*, **COM-36** (9), 1078-1082 (1988).
4. A. Ganz and Y. Gao, "Efficient algorithms for an SS/TDMA scheduling," *IEEE Trans. Com.*, **COM-40** (8), 1367-1374 (1992).
5. Y. Ito, Y. Urano, T. Muratani, and M. Yamaguchi, "Analysis of a switch matrix for an SS/TDMA system," *Proc. IEEE*, **65** (3), 411-419 (1977).
6. T. Inukai, "Comments on 'Analysis of a switch matrix for an SS/TDMA system,'" *Proc. IEEE*, **66** (12), 1669-1670 (1978).
7. T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Com.*, **COM-27** (10), 1449-1455 (1979).
8. I. S. Gopal and C. K. Wong, "Minimizing the number of switchings in an SS/TDMA system," *IEEE Trans. Com.*, **COM-33** (6), 497-501 (1985).
9. A. A. Bertossi, G. Bongiovanni, and A. Bonuccelli, "Time slot assignment in SS/TDMA systems with intersatellite links," *IEEE Trans. Com.*, **COM-35** (6), 602-608 (1987).
10. F. Takahata, "An optimum traffic loading to intersatellite links," *IEEE J. Select. Areas Com.*, **SAC-5** (5), 662-673 (1987).
11. A. Ganz and Y. Gao, "SS/TDMA scheduling for satellite clusters," *IEEE Trans. Com.*, **COM-40** (3), 597-603 (1992).
12. S. Kim and S.-H. Kim, "Time slot assignment in a heterogeneous environment of SS/TDMA system," Working Paper TOSS 93-5, *Telecommunication Optimization Lab.*, Dept. of Management Science, KAIST, 1993.

BIOGRAPHIES OF THE AUTHORS

Sehun Kim received the B.S. degree in physics from Seoul National University, Seoul, Korea, in 1972, and the M.S. and Ph.D. degrees in operations research from Stanford University in 1978 and 1981, respectively.

From 1981 to 1982 he worked at Systems Control, Inc. In 1982, he joined the faculty of the Korea Advanced Institute of Science and Technology (KAIST), where he is currently a Professor of management science. He was a visiting professor at Arizona State University from 1986 to 1987. His research has been in the areas of combinatorial and nonlinear optimization. Recently, he is working on the application of optimization techniques to the design and analysis of computer and communication systems. He has published a number of papers in *Mathematical Programming*, *Operations Research Letters*, *Journal of Optimization Theory and Applications*, *Journal of Applied Physics*, *IEEE Trans. on Vehicular Technology*.

He served as Editor of the *Journal of Korean Operations Research Society* from 1989 to 1991.

Soo-Hyun Kim was born in Kwangju, Korea, on October 1, 1968. He received the B.S. degree in statistics from Korea University and the M.S. degree in management science from the Korea Advanced Institute of Science and Technology (KAIST), in 1991 and 1993, respectively.

Currently, he is a doctoral student in management science at KAIST. His research interests are in the areas of combinatorial and nonlinear optimization, and optimal design and analysis of communication systems, especially, satellite communication systems and cellular mobile systems.