韓國經營科學會誌
第30권 第1號
2005年 3月

정지궤도 위성의 궤도 선정을 위한 알고리즘

김수현* · 김세헌**

Algorithms for Determining the Geostationary Satellite Orbital Positions

Soo-Hyun Kim* · Sehun Kim**

Abstract

We consider the optimization problem of the geostationary satellite orbital positions, which is very fundamental and important in setting up the new satellite launching plan. We convert the problem into a discrete optimization problem. However, the converted problem is too complex to find an optimal solution. Therefore, we develope the solution procedures using simulated annealing technique. The results of applying our method to some examples are reported.

Keyword : Satellite, Orbital Position, Geostationary Orbit, Simulated Annealing

1. Introduction

Geostationary orbit is considered as a kind of natural resource which must be shared with all nations. However, due to the rapid increase of the number of geostationary satellites, the use of the geostationary satellite orbit by satellites sharing the same frequency band leads to an interference environment. It is called *intersatellite interference*, more precisely, interference between geostationary satellite networks. It should be properly analyzed to allow for an adequate and efficient use of the available orbit and spectrum resources. An effective means is highly required to reduce this interference so as to accommodate as many

논문접수일 : 2004년 2월 23일 논문게재확정일 : 2005년 2월 6일
* 배재대학교 경영학과
** 한국과학기술원 산업공학과

satellites as possible in the geostationary orbit while keeping the interference level within the predetermined criteria.

Several studies concerning the efficient use of orbital and spectral resources have been made [4, 6, 10, 11, 13]. With increasing numbers of satellites in orbit, the use of more sophisticated techniques to optimize satellite orbital positions has become necessary, particularly in congested segments of the geostationary orbit.

In applying optimization methods to identify orbital positions, one would, in general, use an objective function which is usually related to one of two basic criteria. The first criterion would be to minimize the total occupied orbital arc, implying interference levels which are close to the maximum allowed level [6]. The second criterion would be to minimize the interference levels, disregarding the orbital arc effectively used [4]. However, the first criterion seems to be inadequate for application in practical situations where the orbital positions of satellites must usually satisfy arc constraints imposed by either technical or political reasons.

In this paper, we propose the optimization methods which use an objective function related to the second criterion. The methods are based on the idea of minimizing the aggregate interference levels. This criterion has been applied by Fortes, *et al.* [4], but it is not like ours. We think that our criterion seems to be adequate in practical situations because some assumptions, which are not necessary here, were made in [4]. In particular, the assumption that the ordering of satellites is prefixed is not practical.

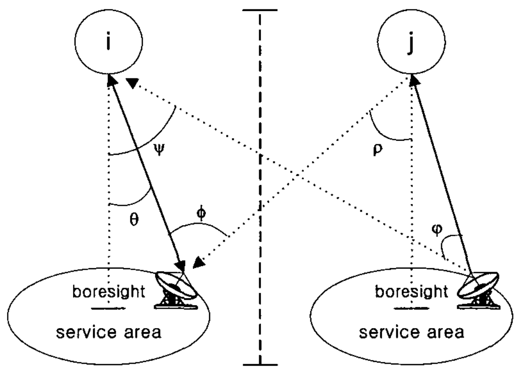
Our optimization criterion function is nonlinear and does not have a special mathematical property (*i.e.* convexity) on which nonlinear optimization technique works well. Therefore, we use the approach which divides the orbit into a number of equal-sized segments. That is, the original problem is converted into a discrete optimization problem. However, it is still difficult to find an optimal solution for the converted problem. In this paper, we apply the *Simulated Annealing* method to this converted discrete optimization problem.

This paper is organized in the following way. In Section 2, the problem is defined and formulated. The solution procedure using simulated annealing is described in Section 3. Section 4 shows the demonstration of the solution quality of our method. Section 5 concludes the paper.

2. Problem Formulation

2.1 Interference Evaluation

To analyze the interference into or from an adjacent satellite system, let us consider the satellite link and interference paths (dotted arrows) between two satellite networks i and j in [Figure 1] [1].



[Figure 1] Intersatellite interference

Let j be the existing satellite network and i be the proposed new satellite network. Then the satellite network i is affected by two interference sources : the uplink interference sigals from earth stations in the existing satellite network j and the downlink interference signals coming from satellite j. The total carrier-to-interference ratio due to these two interference sources represents the interference generated by the existing satellite network j into the proposed satellite network i. The total carrier-to-interference ratio in satellite network i caused by the adjacent satellite network j is

$$(C/I)_{ij} = -10\log\left[10^{-\frac{(C/I)_u}{10}} + 10^{-\frac{(C/I)_d}{10}}\right] [\mathrm{dB}].$$

where $(C/I)_u$ is the uplink carrier-to-interference ratio and $(C/I)_d$ is the downlink carrier-to-interference ratio.

Let us consider the general situation in which n satellites are placed side by side in the geostationary satellite orbit, sharing a common band of the frequency spectrum. Then, we have to aggregate the effect of multiple interfering satellites. Hence, the aggregate carrier-to-interference ratio in satellite network i caused by the adjacent satellite networks is

$$(C/I)_i = -10\log\left[\sum_{j\neq i} 10^{-\frac{(C/I)_{ij}}{10}}\right] [\mathrm{dB}].$$

To guarantee the service quality, satellite systems define the minimum carrier-to-interference ratio which is called *required carrier-to-interference ratio*. Then, the difference between the actual carrier-to-interference ratio and the required carrier-to-interference ratio means the degree of quality. The difference is called *margin*, that is, the margin of satellite i (denoted by MG_i) is evaluated by subtracting the required $(C/I)_i$ from the actual $(C/I)_i$.

2.2 Formulation

Let us assume that the total number of satellite networks is n, which includes the proposed ones. Let us also assume that all the satellites of proposed networks have the orbital upper and lower boundaries, and all the satellites of existing networks, of course, are fixed on their orbital positions. It is quite common that the preferable satellite position is preassigned by a satellite owner and the satellite location is fixed after it is launched. These kinds of constraints are very important in practical applications. In this paper, we consider three cases as follows.

A. (Case 1)

First, the case where the number of proposed networks is only one. This case is applicable to determine the orbital position of a new satellite which is ready to give the communications services. In this case, n-1 satellites' location are prefixed and the location for new proposed satellite i has to be determined.

(C1) $\max\{f(x_i) \mid x_i \le \theta_i^{\max}, x_i \ge \theta_i^{\min}\}$

where $\theta_i^{\max}$ and $\theta_i^{\min}$ denote, respectively, the upper and lower limits of the possible satellite position for network i, x_i is the orbital position of satellite i, and $f(x_i)$ is MG_i.

B. (Case 2)

Second, the case where all networks are the proposed ones. When new orbit is assigned to each of nations, the results of this case can be utilized. In this case, any network is not pre-fixed and we have to identify orbital positions for them.

(C2) $\max\{f(x)\,|\,x_i \le \theta_i^{max}, x_i \ge \theta_i^{min},\ i=1,\cdots,n\}$

where x is a vector on orbital longitudes of the n networks, $f(x)$ is min $\{f_1(x),\cdots,f_n(x)\}$, and $f_i(x)$ is MG_i .

C. (Case 3)

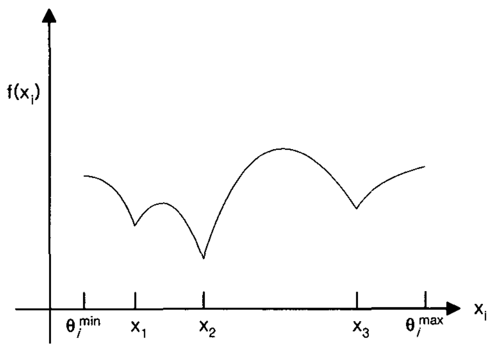
Third, the case where the number of proposed networks is between two and n-1. This case is very useful for the adjustment of orbital positions among the proposed networks. In this case, $p(1<p<n)$ networks are not pre-fixed and we have to identify orbital positions for them.

(C3) $\max\{f(x)\,|\,x_i \le \theta_i^{max}, x_i \ge \theta_i^{min},\ i=1,\cdots,p\}$

where x is a vector on orbital longitudes of the n networks, $f(x)$ is min $\{f_1(x),\cdots,f_n(x)\}$, and $f_i(x)$ is MG_i.

3. Algorithms

3.1 Case 1



[Figure 2] An example of $f(x_i)$

[Figure 2] shows that the transition of objective value as x_i , *i.e.*, the orbital position of satellite i, increases from θ_i^{min} to θ_i^{max}. We observe that the objective value is local minimum when satellite i has the orbital position at x_1, x_2 or x_3, which are the orbital positions of the existing satellites. The interval between θ_i^{min} and θ_i^{max} is divided into four sub-intervals such as [θ_i^{min}, x_1], [x_1, x_2], [x_2, x_3], and [x_3, θ_i^{max}]. Note that an unique local maximum exists in each sub-interval. That is, the objective function is strictly unimodal in this sub-interval.

Therefore, we can find a maximum over this sub-interval fast by applying an efficient search algorithm such as Fibonacci method [2]. Assume that the number of this sub-intervals between θ_i^{min} and θ_i^{max} is n_s+1, where n_s is the number of existing satellites between θ_i^{min} and θ_i^{max}. Then, we come to find n_s+1 local maximums over these sub-intervals. Among these, the largest one is an approximated solution to the problem (C1).

3.2 Case 2

Problem (C2) has a nonlinear objective function and does not have a special mathematical property. Therefore, to handle this case, the orbit is uniformly divided into a number of discrete segments. The unit width of a segment can be arbitrarily determined. The unit width is denoted by δ. Then the discrete placement of satellites is made by placing the satellites at the integral multiple of unit segment. Owing to the segmentation, this problem is converted into a discrete optimization problem. Of course, this converted problem is an approximate one of the original problem. However, this converted problem is still difficult to solve.

This part will be devoted to finding a very good solution via an efficient and robust optimization algorithm-*simulated annealing* [5]. Simulated annealing is a general method for the approximate solution of difficult combinatorial optimization problems. In order to apply simulated annealing to problem (C2), we have to define the corresponding discrete solution space, the profit function, and the neighborhood structure.

- Solution space

Since we have *n* proposed satellites, the current state of orbital positions is represented by a n-dimensional vector x as below.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}$$

A component x_i of vector x denotes the orbital position of satellite i where $1 \le i \le n$, and $\theta_i^{\min} \le x_i \le \theta_i^{\max}$.

- Profit function

The problem introduced here is aimed at maximizing the margin of aggregate *C/I*. Therefore the profit function is $f(x) = \min\{f_1(x), \cdots, f_i(x), \cdots, f_n(x)\}$, which is the objective function of problem (C2).

- Neighborhood structure

Here, given an $x = (x_1, x_2, \cdots, x_n)^T$, the neighborhood $N_e(x)$ is defined as follows :

$$N_e(x) = \{\bar{x} \| \bar{x} = (x_1, x_2, \cdots, \bar{x}_i, \cdots, \bar{x}_j, \cdots, x_n)^T\},$$

where $i \neq j$, $1 \le i, j \le n$, and $\theta_i^{\min} \le \bar{x}_i \le \theta_i^{\max}$ ($\theta_j^{\min} \le \bar{x}_j \le \theta_j^{\max}$).

A strategy for picking a solution $\bar{x}$ in $N_e(x)$ is as follows: We choose two numbers i and j such that $i = \operatorname{Argmax}\{f_i(x): i = 1, \cdots, n\}$, $j = \operatorname{Argmin}\{f_j(x): j = 1, \cdots, n\}$. Set $\bar{x} = x$. Update $\bar{x}_i$ and $\bar{x}_j$ as follows : $\bar{x}_i := \theta_i^{\min} + \delta \times \text{random}(m_i)$, $\bar{x}_j := \theta_j^{\min} + \delta \times \text{random}(m_j)$, where m_i is the greatest integer number less than or equal to $\frac{\theta_i^{\max} - \theta_i^{\min}}{\delta}$, and random(m_i) is a function which generates a random integer number between 0 and m_i.

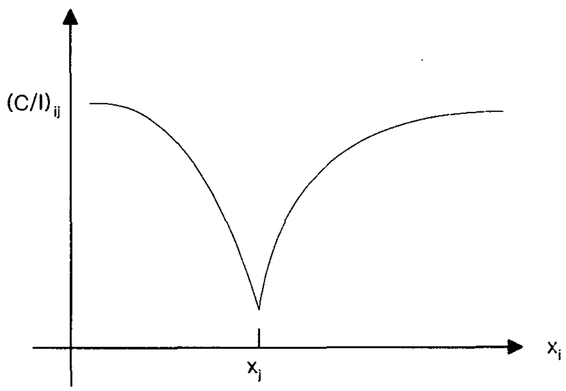
3.3 Case 3

We handle this case like (Case 2). In this case, however, $n-p$ existing satellites have their fixed orbital positions. Therefore, we do not need to change the positions of these $n-p$ satellites. For the convenience, the proposed satellites are denoted by $\{1, \cdots, p\}$, and the existing satellites are denoted by $\{p+1, \cdots, n\}$. Then, we have a n-dimensional vector x where $\theta_i^{\min} \le x_i \le \theta_i^{\max}$ for $i \in \{1, \cdots, p\}$, and x_i is fixed for $i \in \{p+1, \cdots, n\}$.

The profit function is the objective function of problem (C3). The neighborhood is defined as $N_e(x) = \{\bar{x} | \bar{x} = (x_1, x_2, \cdots, \bar{x}_i, \cdots, \bar{x}_j, \cdots, x_n)^T\}$, where $1 \le i, j \le p$, and $\theta_i^{\min} \le \bar{x}_i \le \theta_i^{\max}$ ($\theta_j^{\min} \le \bar{x}_j \le \theta_j^{\max}$). A strategy for picking a solution $\bar{x}$ in $N_e(x)$ is very similar to (Case 2). But, if the chosen i is greater than p then find the currently nearest, from i, proposed satellite i^* and replace i with i^*. If the chosen j is greater than p then do the same way as above. If i is equal to j, update $\overline{x_i}$ only.

4. Implementation

First, we investigated the total carrier-to-interference ratio. The total carrier-to-interference ratio in network i due to single-entry interference from network j, $(C/I)_{ij}$, is shown in [Figure 3]. The larger the orbital distance between two satellites i and j, the bigger the value of $(C/I)_{ij}$. And the speed of decreasing of $(C/I)_{ij}$ is very fast when the distance is small.



[Figure 3] An example of $(C/I)_{ij}$

For the implementation, we used the new function $\alpha_{ij}\log(|x_i - x_j|^2 + 1)$ for $(C/I)_{ij}$, which has the very same shape as the one in [Figure 3]. The relation coefficient α_{ij}, $(0 \leq \alpha_{ij} \leq 1)$, is to deal with the similarity between two networks i and j. If two networks have a lot of similarity, which means they make a lot of interference to each other, we assign a small value to α_{ij}. The α_{ij} also makes it possible to consider other interference-making factors which we can not deal with easily. We assumed that all the networks have the same required aggregate C/I ratio.

The entire scheme of simulated annealing is summarized as follows (for details, see [5]).

Step 1 : Determine an initial T.

Step 2 : Given an x, pick randomly a solution $\bar{x}$ in $N_e(x)$, and let Δ be the change in profit value of $\bar{x}$ from that of x.

Step 3 : If $\Delta \geq 0$, then $x := \bar{x}$. Otherwise, let $x := \bar{x}$ with probability $e^{\frac{\Delta}{T}}$.

Step 4 : If it is concluded that a sufficient number of trials have been made with the current T (i.e., in *equilibrium*), then go to Step 5. Otherwise return to Step 2 with the current x.

Step 5 : If the current T is concluded to be sufficiently small (i.e., *frozen*), then go to Step 6. Otherwise reduce the T (e.g., $T := rT$ with a constant r satisfying $0 < r < 1$) and return to Step 2.

Step 6 : Halt after outputting the best solution obtained so far as the computed approximate solution x.

In fact, it is necessary to enumerate all possible combinations to get the optimal solution. However, it is impossible to get the optimal solution because of the astronomical computation time. Let us compare the solution quality of our methods in the case of the very small example problems which are summarized in <Table 1> and <Table 2>.

〈Table 1〉 Computational results for (Case 2)

Prob.		a_{ij}			θ_i^{min}	θ_i^{max}	our method		optimal	
	i,j	1	2	3			x_i	$f_i(x)$	x_i	$f_i(x)$
1-1	1	-	0.9	0.7	20	25	24.36	-1.6468	24.36	-1.6465
	2	0.9	-	0.2	20	30	30.00	-2.1157	30.00	-2.1156
	3	0.7	0.2	-	15	25	15.01	-2.1165	15.00	-2.1162
1-2	1	-	0.9	0.7	10	20	10.01	-0.7928	10.00	-0.7897
	2	0.9	-	0.2	20	40	39.96	-1.6248	40.00	-1.6254
	3	0.7	0.2	-	10	30	29.86	-1.9575	30.00	-1.9567
1-3	1	-	0.5	0.3	50	60	50.00	-2.3843	50.00	-2.3854
	2	0.5	-	0.1	55	60	55.03	-2.6165	55.00	-2.6172
	3	0.3	0.1	-	53	58	57.99	-2.6947	58.00	-2.6940

〈Table 2〉 Computational results for (Case 3)

Prob.		a_{ij}			θ_i^{min}	θ_i^{max}	our method		optimal	
	i,j	1	2	3			x_i	$f_i(x)$	x_i	$f_i(x)$
2-1	1	-	0.9	0.7	20	25	24.30	-1.6477	24.36	-1.6465
	2	0.9	-	0.2	20	30	29.95	-2.1153	30.00	-2.1156
	3	0.7	0.2		15		15.00	-2.1182	15.00	-2.1162
2-2	1	-	0.9	0.7	10	20	10.00	-0.7909	10.00	-0.7897
	2	0.9	-	0.2	20	40	39.90	-1.6275	40.00	-1.6254
	3	0.7	0.2	-	30		30.00	-1.9577	30.00	-1.9567
2-3	1	-	0.5	0.3	50	60	50.00	-2.3854	50.00	-2.3854
	2	0.5	-	0.1	55	60	55.00	-2.6172	55.00	-2.6172
	3	0.3	0.1	-	58		58.00	-2.6940	58.00	-2.6940

In order to apply simulated annealing, we set the initial T to 1 and set r to 0.7. The simple approach of permitting 5,000 state transitions at each T level has been used. When T is less than 10^{-6}, we conclude it is frozen. The unit width, δ, of a segment is set to 0.01.

<Table 1> shows the result for (Case 2). The number of satellites is three. We have solved three problems and for each of them the initial solution is set to the mid-point between θ_i^{min} and θ_i^{max}. In <Table 1>, we can compare the optimal solutions with the ones we have found. <Table 2> shows the difference between the optimal solutions and the ones we have found for (Case 3). In this case, the satellite 3 of each problem is fixed. The optimal solutions can be obtained by enumerating all possible combinations since the problems above are very small. From <Table 1> and <Table 2>, we can say that our algorithms provide very good performance.

In addition, our algorithms have a very low computational burden. Our algorithm gives a good solution after $5000(k^*+1)$ iterations, while k^* is the smallest integer number k satisfying $Tr^k < 10^{-6}$. Since $T=1$ and $r=0.7$, therefore $k^*=39$ and we need 200,000 iterations to get the final solution. However, the total number of iterations to get an optimal solution is very big. For instance, we can obtain the optimal solution to problem 1-1 by computing 500,000,000 iterations since the unit width of a segment is 0.01. <Table 3> compares the computational burden of the problems in <Table 1> and <Table 2>. For problem 2-1, 2-2, and 2-3, each exhaustive search has a relatively low iterations because the satellite 3 of each problem is fixed.

〈Table 3〉 Computational burden

Prob.	our method(a)	exhaustive search(b)	difference(b/a)
1-1	200,000 iterations	500,000,000 iterations	2,500
1-2	200,000 iterations	4,000,000,000 iterations	20,000
1-3	200,000 iterations	250,000,000 iterations	1,250
2-1	200,000 iterations	500,000 iterations	2.5
2-2	200,000 iterations	2,000,000 iterations	10
2-3	200,000 iterations	500,000 iterations	2.5

5. Conclusions

Optimal methods to identify satellite orbital positions were suggested. We considered three cases which are possible in real situation. The first case was worked out by applying a search algorithm such as Fibonacci method, and we found a near optimum with ease. Then, we applied the methods based on simulated annealing procedure to the other cases. By using the methods we obtained very good solutions.

The methods introduced in this paper can be applied to decide the orbital positions in satellite cluster systems, which is comprised of tens of satellites interconnected each other. Note that, since our simulated annealing procedure is not dependent on the type of objective function, any kind of objective function can be used.

References

[1] Ha, T.T., *Digital satellite communications*, McGraw-Hill, 1990.

[2] Avriel, M., *Nonlinear programming : analysis and methods*, Prentice-Hall, 1976.

[3] Fiacco and McCormick, *Nonlinear programming : sequential unconstrained minimization techniques*, John Wiley & Sons, 1968.

[4] Fortes, J.M. and E.C. Ribeirete, "Optimization of satellite orbital positions using an objective function related to aggregate interference levels," *International Journal of Satellite Communications*, Vol.8(1990), pp. 103-109.

[5] Ibaraki, T., *Enumerative approaches to combinatorial optimization*, J.C. Baltzer AG, 1987.

[6] Ito, Y., T. Mizuno and T. Muratani, "Effective Utilization of Geostationary Orbit Through Optimization," *IEEE Trans. Commun.*, Vol.27(1979), pp.1551-1558.

[7] Jahn, A., "Resource management model and performance evaluation for satellite communications," *International Journal of Satellite Communications*, Vol.19(2001), pp. 169-203.

[8] Jeruchim, M., "A survey of interference

problems and applications to geostationary satellite networks," *Proc. IEEE*, Vol.65 (1977), pp.317-331.

[9] Luenberger, D.G., *Introduction to linear and nonlinear programming*, Addison-Wesley, 1973.

[10] Mizuike, T. and Y. Ito, "Optimization of frequency assignment," *IEEE Trans. Commun.*, Vol.37(1989), pp.1031-1041.

[11] Okinaka, H. *et al.*, "Intermodulation interference-minimum frequency assignment for satellite SCPC systems," *IEEE Trans. Commun.*, Vol.32(1984), pp.462-448.

[12] Thuve, H., "Frequency planning as a set partitioning problem," *European J. Oper. Res.*, Vol.6(1981), pp.29-37.

[13] Weiss, H.J., "Relating to the efficiency of utilization of geostationary orbit/spectrum in the fixed satellite service," *Proc. IEEE*, Vol.68(1980), pp.1484-1496.