LEWIS GRANT
IN-18-CR
84703
P-113

THE OHIO STATE UNIVERSITY

MATHEMATICAL PROGRAMMING FORMULATIONS FOR SATELLITE SYNTHESIS

Puneet Bhasin
Charles H. Reilly

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 718688-4
Grant NAG3-159
July 1987

National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

(NASA-CR-181151) MATHEMATICAL PROGRAMMING FORMULATIONS FOR SATELLITE SYNTHESIS (Ohio State Univ.) 113 p Avail: NTIS HC A06/MF A01 CSCL 22B

N87-25419

Unclas
G3/18 0084703

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO.	2.	3. Recipient's Accession No.

4. Title and Subtitle MATHEMATICAL PROGRAMMING FORMULATIONS FOR SATELLITE SYNTHESIS	5. Report Date July 1987
	6.
7. Author(s) Puneet Bhasin and Charles H. Reilly	8. Performing Organization Rept. No. 718688-4
9. Performing Organization Name and Address The Ohio State University ElectroScience Laboratory 1320 Kinnear Road Columbus, Ohio 43212	10. Project/Task/Work Unit No.
	11. Contract(C) or Grant(G) No (C) (G) NAG 3-159
12. Sponsoring Organization Name and Address NASA - Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	13. Type of Report & Period Covered Technical
	14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

The problem of satellite synthesis can be described as optimally allotting locations and sometimes frequencies and polarizations, to communication satellites so that interference from unwanted satellite signals does not exceed a specified threshold. In this report, mathematical programming models and optimization methods are used to solve satellite synthesis problems. A nonlinear programming formulation which is solved using Zoutendijk's method and a gradient search method is described. Nine mixed integer programming models are considered. Results of computer runs with these nine models and five geographically compatible scenarios are presented and evaluated. A heuristic solution procedure is also used to solve two of the models studied. Heuristic solutions to three large synthesis problems are presented. The results of our analysis show that the heuristic performs very well, both in terms of solution quality and solution time, on the two models it was applied. We conclude that the heuristic procedure is the best of the methods considered for solving satellite synthesis problems.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited.	Unclassified	113
	20. Security Class (This Page) Unclassified	22. Price

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

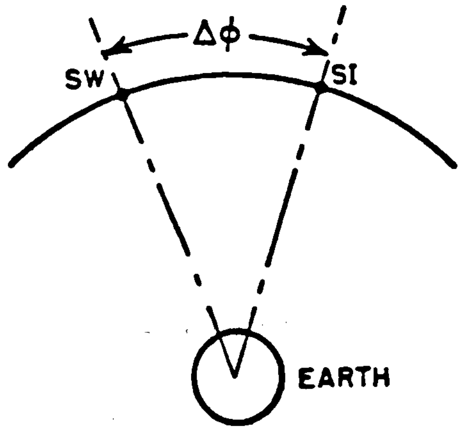
CHAPTER I

INTRODUCTION

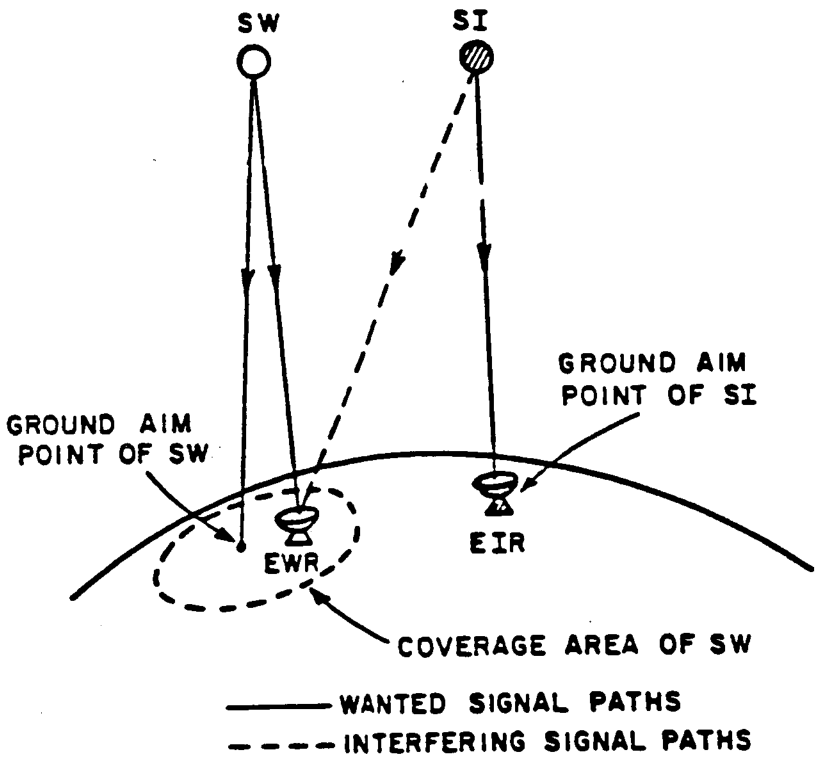
The problem of Fixed Satellite Service (FSS) system synthesis can be described as optimally allotting locations, frequencies and polarizations to communication satellites. These satellites are deployed in geostationary orbit (5.6 earth radii above the equator) for transmitting signals to earth receivers. An orbital arc of feasible locations is determined for each satellite, based on the location and geometry of its service area. These arcs are comprised of locations from which every point in the intended service area is visible. A set of test points on the boundary of each sevice area is defined to facilitate the evaluation of a synthesis solution on the basis of interference. A feasible frequency band at which signals may be transmitted is specified, an accompanying polarization scheme is included as well.

The primary objective in satellite synthesis is to allot orbital positions, frequencies and polarizations to satellites so that interference from unwanted satellite signals does not exceed a specified threshold at any (test) point in any service area. The interference geometry, based on single-entry interference between two down-link communication circuits is shown in Fig. 1 [11]. The following notation is used: S - satellite, W - wanted network, I - interfering network, E - earth station, R - receiving antenna. In this figure, beams transmitted from the two satellites are aimed at their respective service areas. However, some signals transmitted from sateliite SI are received at the service area of satellite SW. These signals interfere with the signals transmitted from satellite SW and thus cause a deterioration in the quality of the signal of satellite SW received in its service area. The synthesis problem is to specify locations, frequencies and polarizations for satellites SW and SI so that interference between these two satellites does not exceed a specified threshold anywhere in the service areas of satellites SW and SI.

In real synthesis problems, there are typically many more than two satellites and two service areas. Aggregate carrier-to-interference (C/I) ratios calculated at every test point of each service area are used to measure the relative strength of wanted and interfering signals. Any solution in which all aggregate C/I ratios exceed a certain threshold is considered acceptable. With this primary criterion in mind, other objectives for the FSS satellite



(a) Overall geometry



(b) Detail with Earth radius exaggerated for clarity

Figure 1: INTERFERENCE GEOMETRY

system synthesis problem can be defined. As an example of this, we may wish to minimize the sum of the absolute deviations of allotted satellite locations from certain predetermined "desired" locations.

The practical importance and political ramifications of a problem of this nature are obvious when one considers the proliferation of space technology to even the smallest and poorest countries. The need for orbital space and frequency and polarization allotments for present and future satellites illustrates the importance of prudent use of the geostationary orbit and the frequency spectrum. Since complex resource allocation problems are often attacked by formulating them as mathematical programming (optimization) problems, mathematical programming models and optimization methods described here should provide tools for the complex orbit/spectrum allocation decisions to be made by those responsible for the governance of international communications, such as delegates to the World Administrative Radio Conference's (WARC) and the International Telecommunication Union.

In the present work, alternate mathematical programming formulations for FSS satellite synthesis are considered. Most of the models are similar but have different objective functions. Our aims are (1) to study different solution techniques for satellite synthesis, and (2) to suggest an appropriate mathematical programming model, and an accompanying solution strategy, that can be reliably solved for synthesis solutions in a reasonable amount of time.

The remainder of this manuscript is organized as follows. Chapter II contains a review of literature on the subject of satellite synthesis. A nonlinear programming formulation of a satellite synthesis problem and three applicable solution methods are described in Chapter III. Nine mixed integer programming (MIP) formulations are described in the next chapter and shown in Appendix A. Results of computer runs with these nine models and five small-to-medium size geographically compatible scenarios are also presented and evaluated. In Chapter V, we present solutions to two of the models studied that were found with a heuristic solution procedure. Results for this approximate method are compared with those obtained with branch-and-bound, an exact method. Solutions to a few large synthesis problems which have also been found with this heuristic are presented. Finally, in the last chapter, conclusions and recommendations about strategies for solving satellite synthesis problems are discussed.

CHAPTER II

LITERATURE REVIEW

MODELING APPROACHES FOR SATELLITE SYNTHESIS

Satellite synthesis problems have been the subject of much research. Much of the work pertains to satellites in the Broadcasting Satellite Service (BSS). Solution strategies suggested for these problems range from exact algorithms to heuristic techniques. Some of these approaches can be modified to suit the FSS also.

There have been approaches that consider only frequency allotments. Cameron [3] has formulated an integer programming frequency allotment model that minimizes the number of channels allotted to satellites, subject to co-channel interference restrictions. Levis et al. [9] have suggested another integer programming formulation for this same problem. They have also formulated an integer programming model that considers the allotment of multiple channels to each service area and takes into account adjacent-channel interference. Baybars [1] has formulated an integer programming model that minimizes the number of channels used while considering both co-channel and adjacent-channel interference.

Ito et al. [8] have formulated a satellite synthesis model that considers the allotment of satellite locations only. Their objective is to minimize the distance between the easternmost and westernmost allotted satellite locations. The model is a nonlinear program that they suggest be solved using the Sequential Unconstrained Minimization Technique. In solving the model, satellites are positioned one at a time according to a predetermined launch sequence. Every satellite that is "launched" is positioned in the best possible manner relative to the other satellites that have already been launched. Such a method does not guarantee global optimality.

Levis et al. [10] have formulated a nonlinear programming model for allotting locations and frequencies to BSS satellites. An alternating polarization scheme is assumed. Their model attempts to maximize the minimum of the aggregate C/I ratios computed for all test points in every service area; locations and frequencies are restricted to lie within specified bounds. Levis et al. [10] and Martin et al. [14] have suggested that this model be solved using an extended gradient search procedure. Reilly et al. [18] have recommended a cyclic coordinate search procedure for the same problem. An extensive computer experiment to evaluate the quality of the synthesis solutions found by these two search techniques was conducted by Reilly et al. [19]. They conclude that the cyclic coordinate search procedure finds better solutions, but it does so at greater computing expense. This experiment and the two search methods are described in more detail in the next chapter.

Heuristic methods for location, frequency and polarization allotments have been suggested by Nedzela and Sidney [16], Chouinard and Vachon [4], and Christensen [5].

REQUIRED SATELLITE SEPARATION CONCEPTS

Wang [22] has developed a procedure to calculate the required orbital separation between two satellites with a known mean longitude that guarantees single-entry co-channel C/I ratios at test points along the boundaries of the intended service areas are at least equal to some acceptable threshold. Aggregate C/I ratio requirements can be satisfied by specifying a higher threshold for the single-entry C/I requirements [22].

We define $\Delta\phi_{ijk}$ to be the required separation between satellites i and j when their mean longitude is k degrees. Typically $\Delta\phi_{ijk}$ values vary according to a bath-tub curve over the set of feasible locations [22]. We refer to the maximum over all feasible k of the $\Delta\phi_{ijk}$ as Δs_{ij}. The significance of Δs_{ij} is that it is a worst-case required separation. Any pair of satellites i and j, when separated by Δs_{ij}, would satisfy our single-entry interference criteria irrespective of their locations in the feasible arc.

In all the synthesis examples to be presented, we use Δs_{ij} values calculated to ensure that aggregate C/I requirements of 25dB are met. The $\Delta\phi_{ijk}$ values are calculated based on a single entry co-channel C/I ratio requirement of 30dB. The notion of requiring an extra 5dB over the single-entry interference criterion to meet an aggregate interference criterion was used at WARC '77 [6] and has proven to be valid for scenarios considered by Levis et al. [11]. The $\Delta\phi_{ijk}$ values and the Δs_{ij} values used in the experiments described in the forthcoming chapters have been calculated using Wang's [22] method.

Detailed descriptions of the separation concept for satellites with elliptical-beam antennas can be found in Wang [22], Levis et al. [11], and Reilly et al. [18]. Yamamura and Levis [21] have done similar work for satellites with circular-beam antennas, which is also described in Levis et al. [11].

Some of the models described in the literature have used similar separation concepts. A model developed by Ito et al. [8] uses a separation matrix to ensure that single-entry interference requirements will be met. Required satellite spacing calculations are based on the current locations of the satellites already positioned ("launched"). Christensen [5] calculates separation values under the assumption that the required separation is approximately constant and independent of the satellite orbital longitudes.

SYNTHESIS MODELS USING WANG'S SEPARATION CONCEPT

The satellite separation concept has opened new avenues for the solution of satellite synthesis problems, as will be discussed in Chapters IV and V. The repeated, complex interference calculations seen in nonlinear programming models for satellite synthesis can be avoided. Such models have been used by Levis et al. [10], Martin et. al. [14], Reilly et. al. [19] and Ito et al [8]. Models that utilize the separation concept are a potentially valuable alternative to the nonlinear programming synthesis models suggested to date.

Reilly et. al. [20] have incorporated the satellite separation concept in a mixed integer programming (MIP) model and an almost linear programming model for allotting satellite locations. The logical relationships between satellite locations that are required to enforce the satellite separation constraints prohibit formulating the model as a pure linear program. Instead the simplex method with restricted basis entry, which does not ensure optimality, is used to find solutions to the almost linear programming model [20]. The MIP model can be solved to optimality with a branch-and-bound algorithm, but solution times for large problems may be prohibitive. The objective in both of these models is to minimize the sum of the deviations between prescribed (optimal) locations and the corresponding "desired" satellite locations. Additional MIP formulations that use the satellite separation concept are described in Chapter IV.

Gonsalvez [7] has developed a switching heuristic, or permutation algorithm, that is applicable to satellite synthesis problems as well as to some scheduling problems. He uses an MIP synthesis model suggested by Mount-Campbell et al. [15] which incorporates the minimum required satellite separation concept. An initial ordering of the satellites is selected. With this initial ordering the synthesis problem reduces to a linear program. The linear program is then solved to find optimal satellite locations for the given ordering. The algorithm proceeds by considering permutations of small groups of adjacent satellites and reoptimizing the linear programming solution as needed.

Finite termination is guaranteed, but optimality is not. Feasible solutions to large synthesis problems (up to 81 satellites) have been obtained with this heuristic in very reasonable solution times. Results obtained with this heuristic for some large problems are described in Chapter V. We also compare results for the switching heuristic with results for branch-and-bound.

CHAPTER III

AN EVALUATION OF THREE SEARCH METHODS FOR A NONLINEAR PROGRAMMING FORMULATION FOR SATELLITE SYNTHESIS

NONLINEAR PROGRAMMING FORMULATION

As mentioned in the previous chapter, a nonlinear programming formulation of satellite synthesis problems has been suggested by Levis et al. [10]. The model as presented in Reilly et al. [19] is:

$$\text{Minimize} \quad f(x) = \sum_{i} \sum_{j \in J_i} \sum_{k \in K_i} \exp(\alpha - (C/I)_{ijk}) \qquad (3.1)$$

subject to,

$$e_i \leq x_{l_i} \leq w_i \qquad \forall\, i \qquad (3.2)$$

$$l_i \leq x_{f_i} \leq h_i \qquad \forall\, i \qquad (3.3)$$

where x_{l_i} is the location of satellite i, x_{f_i} is the first (lowest) frequency in the family of channels assigned to satellite i, e_i (w_i) is the easternmost (westernmost) feasible location for satellite i, l_i (h_i) is the lowest (highest) frequency that can be assigned to satellite i, α

is a scaling constant, J_i is the set of test points in the service area served by satellite i, and K_i is the set of frequencies at which signals are to be transmitted from satellite i [19].

The objective function (3.1) is intended to maximize the worst aggregate C/I ratio at any test point. By raising e to the $(-C/I)_{ijk}$ power and summing over i,j, and k, the worst C/I ratios contribute the most to the objective function value. The constraints (3.2) and (3.3) are simple bound constraints and ensure that the location and frequency decision variables remain feasible.

This synthesis formulation is a difficult optimization problem to solve. The objective function in this model is not convex. In fact, it is characterized by numerous peaks and valleys of varying heights and depths [19]. The computation of the C/I ratios is much more complicated than it might appear from the formulation. The C/I ratios are actually a function of topocentric and satellite-centered angles, frequency discrimination, antenna gains and discrimination patterns, elliptical beam parameters, and transmitted power [19]. For brevity the entire expression for C/I has not been presented. In this model C/I is treated as a function of locations and frequencies only.

COMPUTER EXPERIMENTS WITH THE GRADIENT AND CYCLIC COORDINATE SEARCH PROCEDURES

Reilly et al. [19] have used the gradient search technique and the cyclic coordinate search technique to solve the model given in the previous section.

The gradient search technique is a method that can be used to solve unconstrained nonlinear optimization problems. It can be modified to solve problems with constraints such as the ones present in the model we consider here. The method works as follows. We compute the gradient of the objective function at a feasible point. A line search is conducted in the negative gradient direction to find an improved solution. The method proceeds iteratively by restarting at the improved solution and computing the gradient again. The method terminates if a line search does not yield an improved solution, if a global optima (for convex programming problems) or a local optima (for non-convex programming problems) is found, or if the procedure goes through a predetermined number of iterations. Only solutions which are feasible are examined during the line search.

The cyclic coordinate search procedure begins with the selection of a feasible point. Line searches are conducted, in turn, in both the positive and negative coordinate directions for each of the decision variables. The method terminates when a line search in the coordinate directions for each of the decision variables yields no improvement in the objective function value. The disadvantage of this method is that the objective function is evaluated many times. However, complex gradient computations are never done.

Reilly et al. [19] designed an extensive computer experiment to assess the performance of the gradient and cyclic coordinate search algorithms as synthesis tools. A synthesis problem that consisted of seven South American countries (Argentina, Bolivia, Brazil Chile, Paraguay, Peru and Uruguay) was used. Each country is to be served by one satellite. Each of the satellites is assumed to be capable of transmitting signals over a family of three channels with alternating polarization. Decision variables need only be defined for the lowest frequency assigned to each satellite. The remaining two frequencies are then the next two higher frequencies. All seven satellites are assumed to have the same feasible arc and feasible spectrum.

Since the performance of the gradient and cyclic coordinate search techniques can be affected by various factors, a list of factors that could possibly affect performance was compiled. The factors and factor levels used are listed in Table 1. A 1/4-fractional factorial design, which included 64 runs, was chosen by Reilly et al. [19]. The important conclusions from the experiment are:

(1) Thirty-two of the 34 best results were obtained by using the cyclic coordinate procedure. The average worst C/I for the gradient runs was 19.73dB, and for the cyclic coordinate runs the average was 41.10dB. Only three of the gradient search runs resulted in a worst C/I value greater than the acceptability threshold of 30dB. On the other hand, all cyclic coordinate runs satisfied this criterion of acceptability.

(2) A cyclic coordinate iteration requires about 5.6 times as much CPU time as a gradient search iteration requires. (This conclusion is valid for the methods as they were implemented by Reilly et. al. [19]).

(3) Better results are found when a longer feasible arc segment (80° - 110° W) is used.

(4) When the starting locations are centered in the feasible arc, the initial locations have little effect on the average of the observed worst C/I ratios. However, when the satellites are located near the western boundary of the feasible arc, a better solution is found when the satellites are collocated than when they are separated by 1°. This is especially true for the

Table 1

FACTORS AND FACTOR LEVELS

Factor	Low (-1)	High (+1)
	Factor Levels	
A - Algorithm	Gradient Search	Cyclic Coordinate Search (or) Zoutendijk's Method.
B - Location Spacing	0^{o}	1^{o}
C - Frequency Spacing	0 Mhz	5 Mhz
D - Starting Locations	Centered in Feasible Arc	Spaced From Western-most Feasible Loc.
E - Starting Frequency	Centered in Available Spectrum	Spaced From Highest Available Spectrum
F - Arc Length	90^{o} - 110^{o} W	80^{o} - 110^{o} W
G - Frequency Spectrum	12233-12300 Mhz	12200 - 12300 Mhz
H - Run Time	5 CPU minutes or 10 iterations or cycles	10 CPU minutes or 30 iterations or cycles

Note. For the work carried out previously [19] a level of +1 for factor A represented the cyclic coordinate search procedure. For the present work a level of +1 represents Zoutendijk's method.

gradient runs.

(5) The poor performance of the gradient search algorithm can be attributed to a "jamming" effect that is often observed close to a boundary of the feasible region. This can be explained as follows. Some components of the gradient may indicate that moves near a boundary look promising. The result will be that the line segment searched in the negative gradient direction may be quite short, thereby prohibiting substantial repositioning of the satellites. In effect, the algorithm becomes "jammed" against a boundary.

COMPUTER EXPERIMENTS WITH ZOUTENDIJK'S METHOD

The discussion above indicates that although better solutions are available, the "jamming" effect close to a boundary can result in poor performance of the gradient search algorithm. If we could avoid being trapped along a boundary, we could possibly find better solutions than those found with the gradient search with much shorter solution times than those obtained by the more computationally demanding cyclic coordinate search procedure. An alternate method to the gradient and cyclic coordinate searches, in terms of "jamming" and solution time, respectively, is Zoutendijk's method.

Zoutendijk's method, or the method of feasible directions, is an algorithm which can prevent "jamming" [24]. The method involves solving a linear program to determine the most favorable direction in which to conduct a line search. The directions generated are a compromise between lining up with the negative gradient, as the gradient search does, and avoiding the boundary of the feasible region. For convex programming problems, this method will converge to a global optimum. However, as mentioned earlier in this chapter, the objective function in our problem is not convex and we would most likely find a local, rather than a global, optimum. At every iteration of Zoutendijk's algorithm a linear program has to be solved. Due to the special nature of our problem, this linear program is very sparse (Table 2) and reasonable solution times to the linear program are usually observed.

Given a feasible starting solution x_o to the nonlinear optimization problem,

$$\text{Minimize} \qquad z = f(x) \tag{3.4}$$

subject to,

$$g(x) \leq 0 \tag{3.5}$$

the direction-finding linear program to be solved is,

Table 2

CONSTRAINT COEFFICIENT MATRIX OF SUBPROBLEM IN ZOUTENDIJK'S METHOD

$-\nabla_{x_{1_1}} f$	$+\nabla_{x_{1_1}} f$	$-\nabla_{x_{1_2}} f$	$+\nabla_{x_{1_2}} f$	....				$-\nabla_{x_{f_7}} f$	$+\nabla_{x_{f_7}} f$	+ 1
-1	+1	0	0	0	0	0	....	0	0	+1
+1	-1	0	0	0	0	0	....	0	0	+1
0	0	-1	+1	0	0	0	....	0	0	+1
0	0	+1	-1	0	0	0	....	0	0	+1
0	0	0	0	-1	+1	0	....	0	0	+1
0	0	0	0	+1	-1	0	....	0	0	+1
0	0	0	0	0	0	-1	+1	0	0	+1
0	0	0	0	0	0	-1	+1	0	0	+1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
0	0	0	0	0	......	-1	+1	0	0	+1
0	0	0	0	0	......	+1	-1	0	0	+1
0	0	0	0	0	......	0	0	-1	+1	+1
0	0	0	0	0	......	0	0	+1	-1	+1
+1	+1	+1	+1	+1	......	+1	+1	+1	+1	0

Minimize ρ (3.6)

subject to,

$$\nabla f(x_o)y - \rho \leq 0 \tag{3.7}$$

$$g(x_o) + \nabla g(x_o)y - \rho u \leq 0 \tag{3.8}$$

$$\sum_i |y_i| \leq 1 \tag{3.9}$$

where ρ is a scalar, $\nabla f(x_o)$ is the gradient of the objective function at x_o, $\nabla g(x_o)$ is the gradient of the constraints at x_o, $g(x_0)$ is the value of the constraint functions (3.5) at x_o, $u=(1,1,1....,1)$, and $y=(y_1,y_2,.....,y_n)$ is the direction vector. We replace $|y_i|$ with $(y_i^+ + y_i^-)$, where y_i^+ and y_i^- are restricted to non-negative values, to make constraint (3.9) linear before the linear program is solved.

The algorithm for Zoutendijk's method is initiated at a feasible point. The linear program, given by equations (3.6) - (3.9), is solved to give us an optimal direction y in which to conduct a line search. Every time an improved solution is found, we start from the improved solution and solve another linear program. We have implemented the algorithm so that it terminates either when the solution to the linear program is zero, when 13 searches of 10 points each in the y direction yield no improved solution, or when m^* iterations are completed. The maximum number of iterations, m^*, was set at ten for the experiments we present here.

Formally the entire algorithm can be stated as follows:

Step 1: Choose a feasible starting solution $x_o = (x_{1_o}, x_{f_o})$. Let $m = 0$.

Step 2: If $m \geq m^*$, go to Step 6. Otherwise, compute $f(x_{1_o}, x_{f_o})$.

Step 3: Solve the direction-finding linear program. If the solution to this linear program is zero, go to Step 6. Otherwise, let $n = 0$ and $m = m+1$.

Step 4: Let $n = n + 1$. Determine the length d of the line segment from (x_{1_o}, x_{f_o}) to the nearest boundary in the y direction. Evaluate the objective function (3.1) at ten equally spaced feasible points in the y direction,

$$(x_1, x_f) = (x_{1_o}, x_{f_o}) + \frac{kd \cdot y}{(0.2)^n}, \qquad k = 1,2.....10.$$

Select the most favorable of these ten points, (x_1^*, x_f^*).

If $f(x_1^*, x_f^*) < f(x_{1_o}, x_{f_o})$, set $(x_{1_o}, x_{f_o}) = (x_1^*, x_f^*)$ and go to Step 2. Otherwise go to Step 5.

Step 5: If $n > 12$, go to Step 6. Otherwise go to Step 4.

Step 6: Stop.

The computer code used for the gradient runs in the experiments conducted by Reilly et al. [19] has been modified. FSS antenna patterns have been used instead of the BSS patterns used in [19]. All channels are assumed to have the same polarization. To solve the direction-finding linear program at each iteration, a linear programming package called ZX3LP, available through the IMSL Library, was linked to this computer code. For an n-satellite problem, the linear program to be solved has 2n+1 variables and 4n+2 constraints - 4n constraints resulting from equation (3.8) above and one each for equations (3.7) and (3.9). The first constraint, resulting from (3.7) has coefficients equal to the gradient of the objective function with respect to each of the decision variables. The values of these gradient components range from 10^{-4} to 10^{-9}. The rest of the constraint coefficients are 0 or 1. As one can see from Table 2, the constraint coefficient matrix of the linear program to be solved is very sparse. Due to the ill-conditioned constraint coefficient matrix, the gradient component values are scaled to avoid accuracy errors.

A total of fourteen runs, seven each with the gradient search algorithm and Zoutendijk's algorithm, have been made. The same factors and factor levels used in [19] have been used here also. They are shown in Table 1. For the present work, a level of +1 for factor A (algorithm) indicates Zoutendijk's method, and a level of -1 indicates the gradient search method. Factor H (run time/iteration count) was set at the low level of -1 for all the runs. The results of this experiment are summarized in Table 3. A stop code of 1,2, or 3 indicates that the maximum iteration count was reached, that the allotted CPU time had expired, or that all attempts to find a better solution than the final reported solution failed. Finally, a stop code of 4 indicates that the current solution is at a Kuhn-Tucker point, a local optima. (Note, however, that this did not occur).

All the runs summarized in Table 3 were made on an IBM 3081-D at The Ohio State University.

CONCLUSIONS ON SEARCH METHODS FOR NONLINEAR PROGRAMMING FORMULATION

It is seen from Table 3 that in each pair of runs, either Zoutendijk's method performed better than the gradient search or it yielded an acceptable solution, where acceptable means any solution with a worst C/I value of at least 30dB. Only two of the gradient runs and three of the Zoutendijk runs meet our criterion of acceptability. In addition to these, two other Zoutendijk runs (1b and 5b) came very close to the 30dB acceptability threshold. Both of

Table 3

SUMMARY OF COMPUTER RESULTS

Run	A	B	C	D	E	F	G	H	Worst C/I	Iter./ Cycles	CPU Time (sec)	Stop Type
1a	-1	+1	+1	+1	-1	+1	-1	-1	5.59	6	264	3
1b	+1	+1	+1	+1	-1	+1	-1	-1	29.03	10	77	1
2a	-1	+1	+1	-1	+1	+1	-1	-1	28.38	5	62	3
2b	+1	+1	+1	-1	+1	+1	-1	-1	36.05	10	64	1
3a	-1	+1	+1	+1	+1	-1	+1	-1	5.59	3	122	3
3b	+1	+1	+1	+1	+1	-1	+1	-1	24.39	7	85	3
4a	-1	+1	-1	-1	+1	-1	-1	-1	8.62	2	76	3
4b	+1	+1	-1	-1	+1	-1	-1	-1	17.21	3	88	3
5a	-1	+1	-1	+1	+1	+1	+1	-1	5.43	8	300	2
5b	+1	+1	-1	+1	+1	+1	+1	-1	29.41	10	61	1
6a	-1	-1	+1	-1	-1	+1	+1	-1	41.21	5	34	3
6b	+1	-1	+1	-1	-1	+1	+1	-1	32.62	10	54	1
7a	-1	-1	+1	+1	-1	-1	+1	-1	47.97	10	32	1
7b	+1	-1	+1	+1	-1	-1	+1	-1	35.81	10	47	1

these runs terminated because the maximum iteration count was exceeded. It is possible that another iteration would have increased the worst C/I value to an acceptable level. It is also observed that whenever the gradient search performed poorly, Zoutendijk's method found much improved solutions. On the other hand, for gradient runs which produced acceptable solutions, Zoutendijk's method produced less attractive, but acceptable, solutions. The average C/I value for the gradient runs is 20.39dB while for the Zoutendijk runs it is 29.22dB.

Runs made with Zoutendijk's algorithm always resulted in more iterations. Five of the seven Zoutendijk runs terminated because the maximum iteration count was exceeded. This implies that more improvement may have been possible in these runs. Only one of the gradient runs terminated due to the iteration count. No Zoutendijk run terminated at a Kuhn-Tucker point, again implying that more improvement was possible.

Another significant observation that is made from the results presented here concerns solution times. The average run time per iteration for the gradient runs is 26.08 seconds while the corresponding value for the Zoutendijk runs is 10.25 seconds. The total average time for the gradient runs is 127 seconds, while for the Zoutendijk runs it is 68 seconds. Thus, on the whole, Zoutendijk's method appears to work faster. It can be seen from Run 5 that while the gradient search run had two iterations less than Zoutendijk's algorithm, it took nearly five times as much time. Such a result can be explained by the fact that the gradient search can become jammed against a boundary and perform fruitless line searches over very short line segments, thereby evaluating the objective function many times for naught.

It is also observed that if the satellites are separated by 1^{o} and are located at the western boundary, then the gradient search performs poorly. This is presumably because the easternmost satellites cause a blocking effect on the other satellites. For the same scenario, Zoutendijk's method, because of the nature of the algorithm, does very well. Another observation, that is not evident from the results presented here, is that satellite locations have a much greater effect than frequencies in determining the magnitude of the worst C/I values [2].

To summarize the findings from the experiments presented here it can be concluded that Zoutendijk's method looks very promising. It has performed better than the gradient search, both in terms of the worst C/I values and in terms of solution times. However, neither method guarantees an acceptable solution. To do full justice to the comparison of the gradient search and Zoutendijk algorithms, an extensive experiment, with several test problems, would have to be conducted. Alternate solution techniques, some of which are described in the next two chapters, are available. These methods, unlike the nonlinear programming solution techniques, guarantee that any solution found is acceptable.

CHAPTER IV

MIXED INTEGER PROGRAMMING FORMULATIONS FOR SATELLITE SYNTHESIS PROBLEMS

INTRODUCTION

In this chapter we consider mixed integer programming (MIP) formulations of satellite synthesis problems. It is evident from the discussion of the previous chapters that the major criterion to be satisfied in the satellite synthesis problem is that C/I ratios must not fall below a certain threshold value. The minimum satellite separation concept [22] is a valuable tool in ensuring that we do not exceed predetermined acceptable interference limits, and has been used in the formulations discussed below. This minimum satellite separation concept can be used in many synthesis models with a variety of objective functions.

In the present work nine MIP models with different objective functions have been considered. The results of computer experiments with these models and five geographically compatible scenarios is presented. Both point and arc allotments have been considered. The purpose of this work is to evaluate the results from our different models in terms of robustness of objective function and solution quality. We try to identify objectives which are "universal" in the sense that while meeting a particular criterion they do well against other objectives also. We look at the number of feasible solutions found with each model during a certain period of execution. Percentage improvements in objective function values are also considered. The initial ordering, the final ordering, and the change in the ordering of satellites from the first feasible solution to the final solution are studied.

OBJECTIVES FOR THE MIP FORMULATIONS

The nine different formulations (objectives) that have been considered are:

1. Maximizing the minimum of the extra separation beyond Δs_{ij} for any pair of satellites i and j. This objective would tend to maximize the minimum C/I ratio and is thus similar to the objective used for the nonlinear programming formulation discussed in the previous chapter.

2. Maximizing the minimum gap between adjacent satellites. The use of this objective would leave large gaps between satellites, facilitating introduction of new satellites between existing satellites at a future time.

3. Minimizing the total deviation of allotted locations from given "desired" locations, subject to feasible arc restrictions. The use of this objective can enable administrations to place satellites close to some pre-determined desired locations. In the experiments presented here, the desired location used for each satellite is the center of its visible arc.

4. Minimizing the maximum deviation of an allotted location from its given "desired" location, subject to feasible arc restrictions. This model attempts to do the same as model 3 except that we look at the maximum deviation rather than the total deviation of allotted locations from desired locations.

5. Minimizing the total weighted deviation of allotted locations from given "desired" locations. This model fulfills the same objective as models 3 and 4 above. The weight used for each satellite is inversely proportional to the length of the satellite's feasible arc. This weighting scheme, by giving greater weight to administrations with smaller feasible arcs, would attempt to ensure a feasible solution.

6. Minimizing the maximum deviation of an allotted location from the closest boundary of the associated satellite's feasible arc. The solution to a synthesis problem with

this objective would leave a large gap in the center of the available arc enabling easy insertion of future satellites.

7. Minimizing the length of the arc between the easternmost and westernmost allotted locations. This objective attempts to place satellites as close together as possible and thereby tends to conserve the geostationary orbit.

8. Maximizing the minimum length arc segment allotted to any satellite. This objective differs from objectives 1 through 7 in that it allots an arc segment to each satellite rather than a point. This formulation gives an administration the flexibilty to place their satellite anywhere in the allotted arc segment. Arcs are permitted to overlap if the minimum required satellite separation is zero. This model allots each administration the same length arc segment.

9. Maximizing the minimum length weighted arc segment allotted to any satellite. This objective might be more practical than that in model 8 above because we could allot arcs to administrations (countries) such that the length of each allotted arc segment is proportional to some criterion and also give administrations the flexibility to place satellites anywhere in their

allotted arc segments. The weighting criterion to be used could be a country's population, telephone traffic, land area, population density, etc. The weights we use here are proportional to the population of each country.

For the sake of brevity, only the formulation for objective 1 is given below. Formulations (2) through (9) and the notation used in them is given in Appendix A. The notation that we use for formulation (model) 1 is given below:

Parameters:

$E_j(W_j)$ = easternmost (westernmost) feasible location for satellite j, in degrees west longitude.

E = minimum $[E_j]$ over all j (assumed to be zero).

W = maximum $[W_j]$ over all j.

Δs_{ij} = worst-case minimum required satellite separation between satellites i and j, in degrees longitude.

Decision variables:

x_j = orbital location of satellite j (degrees west of E), in degrees west longitude.

$$x_{ij} = \begin{cases} 1 & \text{if satellite } i \text{ is located west of satellite } j, \\ 0 & \text{otherwise} \end{cases}$$

a = minimimum of the extra separation beyond Δs_{ij} for all pairs of satellites i and j.

Model 1: Maximize the minimum of the extra separation beyond a required minimum satellite separation.

$$\text{Maximize} \quad z = a \tag{4.1}$$

subject to,

$$(x_i - x_j) + [2(E-W) - \Delta s_{ij}]x_{ij} - a \geq 2(E-W) \quad (4.2) \ \forall i,j \ni i<j$$

$$(-x_i + x_j) - [2(E-W) - \Delta s_{ij}]x_{ij} - a \geq \Delta s_{ij} \quad (4.3) \ \forall i,j \ni i<j$$

$$E_j \leq x_j \leq W_j \quad (4.4) \ \forall j$$

$$x_{ij} = 0 \text{ or } 1 \quad (4.5) \ \forall i,j \ni i<j$$

$$x_j \geq 0 \quad (4.6) \ \forall j$$

$$a \geq 0 \tag{4.7}$$

The objective function (4.1) maximizes the minimum of the extra separation between two satellites beyond their required minimum separation. For each pair of satellites i and j, exactly one of the constraints (4.2) and (4.3) will be redundant. The binary variable x_{ij} which takes value 1 or 0, depending on whether satellite i is located to the west of satellite j, determines the redundant constraint. If $x_{ij}=1$ then constraint (4.3) is redundant. If $x_{ij}=0$ then constraint (4.2) is redundant. The non-redundant constraint then enforces the required separation between satellites i and j. Constraints (4.2) and (4.3) also measure the minimum separation beyond a required satellite separation, which is denoted by a. Constraints (4.2) and (4.3) are an example of dichotomous constraints commonly used in mixed integer programming formulations to enforce "either-or" conditions.

Constraints of type (4.4) ensure that all allotted satellite locations are feasible. In the MIP code used to solve this synthesis problem, constraint (4.4) is enforced as a simple bound on the variable x_j. The integrality of the binary variables is enforced by (4.5) and non-negativity requirements of continuous variables is ensured by (4.6) and (4.7).

For a problem with m satellites, this formulation entails m(m-1) structural constraints (4.2) and (4.3), m+1 continuous variables and m(m-1)/2 binary variables.

The formulations given in Appendix A are very similar to the one shown above. Each of these formulations uses the minimum required satellite separation, Δs_{ij}. However, in several of the formulations, additional variables are introduced.

COMPUTER EXPERIMENTS WITH MIP MODELS

Five scenarios were used to evaluate the nine MIP models. These scenarios are listed below. The countries and satellites which make up each scenario are given in Table 4.

(1) E. Europe (12 satellites/12 service areas).

(2) W. Europe (12 satellites/12 service areas).

(3) S. America (13 satellites/12 service areas).

(4) N. Africa (10 satellites/10 service areas).

(5) S.E. Asia (10 satellites/10 service areas).

The results of computer experiments with these five scenarios and our nine models are given in Appendix B. An IBM linear programming/mixed integer programming package called MIP/370 has been used for all the runs. The runs for these experiments were made on an IBM 4341 at The Ohio State University. The maximum CPU time specified was 20 minutes and maximum node storage was set at 5000 nodes. Computer runs were automatically terminated if either of these limits was exceeded.

To evaluate the universality, or robustness, of each of the models, the final solution from formulations 1 through

Table 4

GEOGRAPHIC SCENARIOS USED IN EXPERIMENTS

	EAST EUROPE	WEST EUROPE	SOUTH AMERICA	NORTH AFRICA	S.E. ASIA
1.	Finland	Italy	Bolivia	Libya	Taiwan
2.	Bulgaria	Norway	Brazil*	Nigeria	Cambodia
3.	Romania	Denmark	Chile	Mali	Malaysia
4.	Greece	Belgium	Colombia	Morocco	Vietnam
5.	Albania	France	Guyana	Sudan	China
6.	Poland	Switzerland	Peru	Egypt	Burma
7.	Hungary	Netherland	Paraguay	Chad	Thailand
8.	Sweden	Spain	Equador	Tunisia	Laos
9.	Austria	Ireland	Venezuala	Algeria	Indonesia
10.	E. Germany	W. Germany	Argentina	Mauritius	Philipp-ines
11.	Yugoslavia	Portugal	Uruguay		
12.	Czechoz-lavakia	United Kingdom	Surinam/ F. Guyana		

* - 2 Satellites.

7 was evaluated in the objective function of each of the other six models for each scenario. The results of this analysis can be found in Appendix C. The initial ordering, the final ordering, and the change in the ordering of satellites from the first feasible solution to the final solution was also investigated. Our findings are given in Appendix D.

Table 5 gives the average number of feasible solutions found with each of the models for the five scenarios. The average percentage improvement and the average percentage improvement with respect to time for each of the models is also given in Table 5.

From the results given in Appendices B,C and D, and in Table 5, we make the observations discussed in the next section.

OBSERVATIONS OF COMPUTER RUNS ON THE MIP MODELS

As mentioned earlier in this chapter, we have considered both point allotment models (models 1 through 7) and arc allotment models (models 8 and 9). We will analyze the results of computer runs of the point allotment models separately from the results for the arc allotment models.

Point Allotment Models

On an examination of the results given in the appendices and from the discussion below it will become clear that models 3,4, and 5 are robust. Furthermore, solution times for these models are less than those for the other point allotment models. Hence, in addition to overall results, we will also give results pertaining to these three models.

The tables in Appendix B give us the times to the first feasible solution and the final feasible solution for each of the branch-and-bound runs. We see that for all our models and for each of the scenarios considered, the first feasible solution was found very fast. For all the point allotment models, the average time to the first feasible solution was 11.9 seconds. On the other hand, the time to the first feasible solution for models 3,4, and 5 was only 5.5 seconds.

As mentioned earlier, the maximum CPU time specified for all the runs was 1200 seconds. However, we find little or no improvement in the objective function values during the last few minutes of execution, even though optimal solutions were found for but a few of the runs. The average time to the best solution for all point allotment models was 385 seconds. On computing the same quantity for models 3,4, and 5 we get a value of 165 seconds. An optimal solution was found and proved for models 3,4, and 5 in three of the five scenarios. A proven optimal solution was found for model 6 in one scenario. All other point allotment runs terminated at the maximum allotted CPU time.

Table 5 gives us the average number of feasible solutions found for each of our models. It is seen that model 2 found an unusually large number of feasible solutions. This can be explained as follows. Recall that model 2 maximizes the minimum gap between adjacent satellites. Once the entire arc is occupied, the gap between each pair of adjacent satellites is the same. In each of our runs, this gap was larger than the largest required satellite separation value. Hence, alternate solutions to the model are found by switching satellites from one position to another. The relative locations and hence the objective function values remain the same for such solutions. Each time a satellite is switched we find a different feasible solution. Since such switches occur many times we find many different feasible solutions.

We also see from Table 5 that models 3,4, and 5 find a relatively small number of feasible solutions. This is because some of the runs terminated when an optimal solution was found, before the maximum CPU time expired. The numbers of feasible solutions found in 20 minutes of CPU time for models 1, 6 and 7, are nearly the same.

Table 5

AVERAGE NUMBER OF FEASIBLE SOLUTIONS AND PERCENTAGE IMPROVEMENTS IN OBJECTIVE FUNCTION VALUE

	AVG. NO. OF FEASIBLE SOLUTIONS	% IMPROVEMENT	% IMPROVEMENT PER CPU MINUTE
Model 1	13.0	64.16	13.69
Model 2	68.0	31.48	15.84
Model 3	6.4	20.20	18.70
Model 4	3.6	11.35	29.69
Model 5	6.4	25.41	53.59
Model 6	10.4	82.38	35.16
Model 7	11.8	21.32	1.93
Model 8	10.0	38.64	10.88
Model 9	8.0	22.80	9.41

NOTE. All the above runs were given a maximum CPU time of 20 minutes. Some of the runs reached optimality before the maximum time expired.

Table 5 also gives the percentage improvement in solution value from the first feasible solution to the best solution, and the percentage improvement with respect to the time to find the best solution. A significant point emerging from this analysis concerns models 3 and 5. Recall that model 3 minimizes the total deviation from desired locations. In model 5 also we minimize the sum of deviations but with a weighted objective function and without enforcing the visible arc restrictions. From Table 5 we see that model 5 converges to its final solution faster than model 3. Appendix C, in which we compare the solution for each model against the objective function values of the other models, shows us that the solution for both of these models is similar. We thus conclude that it might be advantageous to use model 5 instead of model 3. However, we must note that the solution from model 5 might not be feasible, in which case we would have to use model 3.

We see from Table 5 that models 1 and 6 show a significant improvement in objective function value from the first feasible solution to the best solution found. Model 4 shows the least percentage improvement from the first feasible solution to the best solution. We also note from Table 5 that model 5 has the fastest convergence to the final solution, while model 7 converges very slowly to its final solution.

The solutions from models 1 through 7 were evaluated in the other six models by calculating their "would-be" objective function values. These solution values are reported in Appendix C. The following observations have been made:

(1) The best solution to model 1 gave an objective function value close to that of the best solution to model 2, and vice versa.

(2) Also, the best solution to model 3 gave an objective function value close to that of the best solution to model 5, and vice versa.

(3) The solutions to models 3,4, and 5 gave good values for the objective function of model 7, but the reverse was not true. Recall that in our case the desired locations are assumed to be the center of the feasible arcs. By minimizing deviations of prescribed satellite locations from desired locations, we also conserve the geostationary orbit, but not vice versa.

Arc Allotment Models

In models 8 and 9 we allot arc segments to satellites. In model 8 the length of the arc segment allotted to every satellite (administration) is the same. In model 9 the the length of the arc segment allotted to every administration is proportional to the population of the administration.

From Appendix B we see that the overall average time to the first feasible solution for model 8 is 11.0 seconds while for model 9 it is 7.5 seconds. The overall average time to the best solution found is 557 seconds for model 8 and 333 seconds for model 9.

A proven optimal solution was found for model 8 in one scenario and for model 9 in two scenarios. From Table 5 we see that the average number of feasible solutions found for models 8 and 9 is ten and eight, respectively.

From Table 5 we also note that model 8 shows a greater percentage improvement in objective function value from the first feasible solution to the final solution than model 9. However, the rate of convergence for models 8 and 9 is nearly the same.

Analysis of Satellite Orderings

Appendix D gives the ordering of satellites for each model and each scenario. In this Appendix, (F) refers to the ordering at the first feasible solution and (B) is the ordering of satellites at the best solution found. For the arc allotment models we use the center of the allotted arc to determine the order of satellite allotments. For all the point and arc allotment models considered we see a significant change in the orderings from the first feasible solution to the best solution. It is also seen that, for the final solution found, there is some similarity in the ordering at the boundaries. For example, from Table 21 we see that satellites 4, 5, and 11 are found close to or at the boundaries of the arc occupied by the eastern European satellites. From Table 24 we see that satellites 5,6, and 7 are located close to the boundary of the arc occupied by the African satellites. Similar results are found for other scenarios. Satellites which show this kind of similarity at the boundaries usually have feasible arcs whose eastern and/or western edges extend beyond those for other satellites. Such observations may be useful in developing heuristic techniques which reduce the number of binary variables. Satellites with feasible arcs extending beyond the eastern or western edges of the arcs of other satellites could be fixed in position. As an example, for a 13-satellite problem, if we could fix four satellites in positions at either end of the ordering, we would reduce the number of binary variables from 78 to 36. This significant reduction in the number of binary variables reduces the complexity of the problem considerably.

CHAPTER V

APPLYING A HEURISTIC ALGORITHM FOR SATELLITE SYNTHESIS

The satellite synthesis problem can be considered to be comprised of essentially two problems - a satellite ordering problem and a satellite location problem (Mount-Campbell et al. [15]). For a given ordering of satellites, the synthesis problem reduces to a linear program. This linear program can be solved to find the optimal locations of satellites subject to required minimum satellite separation and feasibile arc constraints. The determination of an optimal, or even good, ordering is still difficult.

Gonsalvez [7] has developed a heuristic algorithm that uses this notion to find solutions to satellite synthesis problems. In the current implementation of this algorithm, we have the option to minimize either the sum of deviations of allotted locations from desired locations or the maximum deviation of an allotted location from the corresponding desired location [7].

The heuristic algorithm is initiated with the specification of an initial ordering of satellites. Given this ordering we solve a linear program to find optimal satellite locations. Next, we generate another ordering by permuting ("switching") the order of satellites within a group of k adjacent satellites. For every promising switch, the linear programming solution is reoptimized using the principles of sensitivity analysis. If a proposed switch does not yield an improved solution, the switch is reversed. If an improved solution is found, that solution becomes the incumbent and we look for other promising switches starting from this improved solution. This procedure is repeated till all orderings that can be obtained by permuting k adjacent satellites at a time have been considered. When no promising permutations are identified, k is incremented by 1 (to a maximum of 5), or the algorithm is terminated.

To guarantee optimality of the solution found with the "switching" heuristic, one would have to examine all n! orderings, where n is the number of satellites. This is not practical for large problems. Since the objective is to minimize the sum of deviations from desired locations or the maximum deviation from a desired location, we begin with the initial ordering based on the order of desired locations and permute small groups of adjacent satellites to generate different orderings.

Formally the algorithm for the switching heuristic can be stated as follows:

Step 1: Generate an ordering of the satellites based on their desired locations.

Step 2: Given the ordering from Step 1, and a particular objective, solve a linear program to find the optimal locations for the satellites.

Step 3: Select a particular adjacent-satellite group size, $(2 \leq k \leq 5)$.

Permute the order of satellites within this group (i.e., switch satellites in position 1 to k with the satellites in positions k+1 to n being fixed in position (not location)).

Reoptimize the linear programming solution as needed.

Repeat till all orderings within a group have been considered.

Step 4: Repeat Step 3, switching satellites 2 to k+1, 3 to k+2,.....,n-k+1 to n.

Step 5: (This step is optional).

Increase group size by 1. Start from Step 3 with the best solution found so far.

Step 6: If no improved solution is found, stop.

To solve the linear program, the algorithm uses the formulation of model 3 suggested by Mount-Campbell et al. [15]. It has been found that increasing the group size k from 2 to 5, in steps of 1, is most effective [7]. For further details on the switching heuristic the reader is referred to [7].

The switching algorithm can use either the worst-case required satellite separations (Δs) or the location-dependent satellite separations ($\Delta\phi$) to guarantee interference requirements are met. If the $\Delta\phi$ values are to be used, then $\Delta\phi$ matrices, calculated for selected satellite longitudes, must be input. This option cannot be implemented conveniently in a branch-and-bound algorithm. It represents a distinct advantage of the switching algorithm, because the smaller location-dependent satellite separations may make it easier to find feasible solutions to tightly-constrained problems.

We have used the switching heuristic to find solutions to some large and medium-size problems, in addition to the five scenarios introduced in Chapter IV. In Table 6 results for this algorithm on a 36-satellite scenario are given. Results obtained from a larger 59-satellite scenario are given in Table 7. Results from a 81-satellite scenario are given in Table 8.

In Table 9 we present results obtained with the switching heuristic for the five scenarios used to evaluate the MIP formulations in the previous chapter. The objective function is to minimize the sum of deviations from desired locations. We compare these results with the results obtained for the MIP runs. The CPU time for the switching runs is the total run time. For the MIP runs, it is the time to the best solution if an optimal solution was not found or the total run time if optimality was proved. We must note that the CPU times are not exactly comparable since two different computer systems have been used. We do know that the IBM 3081-D is faster than the IBM 4341.

Table 10 gives results similar to those in Table 9, but the objective function is to minimize the maximum deviation from given desired locations.

From Tables 6,7, and 8 we see that the switching heuristic finds feasible solutions very fast. A 36-satellite scenario, comprised of European and African satellites, was modelled using formulation 3, the minimization of the total deviation from desired locations, given in Appendix A. We ran this model on the IBM 4341 using MIP/370 and found no feasible solution in 120 minutes of CPU time. On the other hand, the switching heuristic found a feasible solution to this same scenario in just 0.4 seconds. The times obtained

Table 6

SWITCHING HEURISTIC WITH 36 SATELLITES

OBJECTIVE: MINIMIZE THE TOTAL DEVIATION FROM GIVEN DESIRED LOCATIONS.

CODE USED: SWITCHING HEURISTIC WITH INCREASING 'k'.

SCENARIO : EUROPE + AFRICA (36 SATELLITES).

DELTA 'S'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	642.49	38.17	100.45	0.40
FINAL SOLN.	169.29	25.32	72.56	436

DELTA 'Φ'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	642.19	38.17	100.45	1.60
FINAL SOLN.	121.79	14.08	72.00	541

Table 7

SWITCHING HEURISTIC WITH 59 SATELLITES

OBJECTIVE: MINIMIZE THE TOTAL DEVIATION FROM GIVEN DESIRED LOCATIONS.

CODE USED: SWITCHING HEURISTIC WITH INCREASING 'k'.

SCENARIO : OASTS2G1 (59 SATELLITES).

DELTA 'S'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	-	-	-	1200*

DELTA 'Φ'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	475.55	26.95	94.15	153.8
FINAL SOLN.	443.98	24.54	92.35	1200*

* - TERMINATING DUE TO TIME LIMIT

Table 8

SWITCHING HEURISTIC WITH 81 SATELLITES

OBJECTIVE: MINIMIZE THE TOTAL DEVIATION FROM GIVEN DESIRED LOCATIONS.

CODE USED: SWITCHING HEURISTIC WITH INCREASING 'k'.

SCENARIO : OASTS2G1 + CARRIBEAN (81 SATELLITES).

DELTA 'S'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	-	-	-	600*

DELTA 'Φ'

	TOT DEV.	MAX DEV.	ORBITAL ARC	CPU TIME (Sec)
FIRST FEASIBLE SOLN.	1033.29	53.53	133.07	354.7
FINAL SOLN.	832.46	47.01	126.55	600*

* - TERMINATING DUE TO TIME LIMIT

for the 59-satellite and 81-satellite scenarios are also reasonable.

From Tables 9 and 10 we see that the switching heuristic does better than the MIP formulations in terms of solution time in nearly all cases. The objective function values are also comparable. In six of the ten runs given in these two tables, we find that the switching heuristic came up with the same solution as the MIP solution. In four of these cases the switching heuristic found an optimal solution. In the other four computer runs, where we did not find the same solution with both approaches, the switching heuristic solution's value is nearly the same as that for the best solution found by the branch-and-bound algorithm. The results presented here indicate that the switching heuristic performs exceedingly well, both in terms of solution time and solution quality. The advantages of this algorithm can be summarized as follows:

(1) The switching heuristic finds feasible solutions very fast.

(2) The final solution values are nearly as good as those obtained by truncated runs with a branch-and-bound algorithm. In some cases, the switching heuristic converges to the optimum solution.

Table 9

SWITCHING HEURISTIC VS. MIP MODEL 3

OBJECTIVE: MINIMIZE THE TOTAL DEVIATION FROM GIVEN DESIRED LOCATIONS.

CODE USED: SWITCHING HEURISTIC WITH INCREASING 'k'.

OPTION : DELTA 'S'

SCENARIO	SWITCHING HEUR.		MIP FORMULATION	
	Obj. Fn. Value	CPU Time (Sec)	Obj. Fn. Value	CPU Time (Sec)
E. EUROPE (12)	52.74	7.34	49.87	613.0
W. EUROPE (12)	32.29	11.34	29.67*	41.4
S. AMERICA (13)	30.44	11.35	30.44	86.4
N. AFRICA (10)	8.65	3.68	8.65*	3.6
S.E. ASIA (10)	23.05	5.31	23.05*	96.0

NOTE: SW runs have been made on the IBM 3081-D.
MIP runs have been made on the IBM 4341.
For SW runs, CPU time = total run time
For MIP runs CPU time = total run time for optimal solutions, (or)
= time to best solution if soln. is not optimal
Figures in parentheses indicate number of satellites.

* - PROVEN OPTIMUM

Table 10

SWITCHING HEURISTIC VS. MIP MODEL 4

OBJECTIVE: MINIMIZE THE MAXIMUM DEVIATION FROM GIVEN DESIRED LOCATIONS.

CODE USED: SWITCHING HEURISTIC WITH INCREASING 'k'.

OPTION : DELTA 'S'

SCENARIO	SWITCHING HEUR.		MIP FORMULATION	
	Obj. Fn. Value	CPU Time (Sec)	Obj. Fn. Value	CPU Time (Sec)
E. EUROPE (12)	9.41	8.68	8.48	1137.0
W. EUROPE (12)	6.62	8.35	6.62*	155.4
S. AMERICA (13)	6.96	10.23	6.60	12.6
N. AFRICA (10)	1.64	3.29	1.64*	0.4
S.E. ASIA (10)	5.19	5.10	5.19*	79.8

NOTE: SW runs have been made on the IBM 3081-D.
MIP runs have been made on the IBM 4341.
For SW Runs CPU time = total run time
For MIP runs CPU time = total run time for optimal solutions
= time to best solution if soln. is not optimal
Figures in parentheses indicate number of satellites

* - PROVEN OPTIMUM

(3) The switching heuristic is independent of the objective function. The current implementation includes the option of choosing between two objective functions; the algorithm can be modified to solve other models also, including arc allotments [7].

(4) The switching heuristic can make use of location-dependent satellite separations ($\Delta\phi$) values. This means that it is more likely to find a feasible solutions when one exists, especially for tightly-constrained problems.

CHAPTER VI

RECOMMENDATIONS

Our aims were (1) to analyze different solution techniques for solving satellite synthesis problems and (2) to suggest an appropriate mathematical programming model, and an accompanying solution strategy, that can reliably solve synthesis problems at reasonable cost. In the previous chapters we have discussed three distinct solution strategies, namely, nonlinear programming search methods, branch-and-bound, and a switching heuristic. We have also experimented with different satellite synthesis models. We have evaluated these strategies and the models.

In Chapter III we see that though it does not guarantee that C/I needs will be met, Zoutendijk's method does better than the gradient search in terms of solution time and solution quality. The "jamming" effect, which is a potential problem with the gradient search method, is not observed when Zoutendijk's algorithm is used. Results presented by Reilly et al. [19] indicate that the cyclic coordinate search method is likely to give worst C/I values that are better than those obtained with Zoutendijk's method. However, solution times are likely to be much longer. It must be pointed out that even the cyclic coordinate search method, like the gradient search and Zoutendijk's method, does not ensure an acceptable solution. Approaches that guarantee that acceptable solutions will be found, provided such solutions exist, are more useful.

In Chapter IV we see that by introducing the minimum satellite separation concept into an MIP model we guarantee that any solution found will meet our interference requirements and hence be acceptable. We have also seen that for the problems we have considered here (10-13 satellites) the first feasible solution is found very fast. However, any real-life problem would be larger than this. Solution times for integer programming problems typically increase exponentially with problem size. An example of this was mentioned in Chapter V where we said that no solution was found to a 36-satellite MIP model in two hours of CPU time.

The nonlinear programming methods discussed in Chapter III, unlike the branch-and-bound algorithm, do not ensure that globally optimum solutions will be found. We have seen that in spite of extensive line searches, none of the nonlinear programming algorithms converged to even a local optimum. On the other hand, a few of the MIP models have homed in on globally optimum solutions. For the smaller

(7-satellite) problems solved by the nonlinear programming methods, the time to find the first feasible solution, if one is found, is greater than the time to find the first feasible solution for the larger (10 to 13-satellite) MIP models [2]. Also, the MIP models guarantee an acceptable solution if a feasible solution is found. Clearly, the mixed integer programming models are preferred over the nonlinear programming model of Chapter II.

The switching heuristic finds feasible solutions relatively quickly. Solution quality is also nearly as good as that obtained by truncated branch-and-bound runs. In addition to this, the switching heuristic makes use of the smaller, location-dependent satellite separations ($\Delta\Phi$). It is independent of the objective function and can be modified to solve models other than the deviation models we considered in Chapter V.

From the experiments presented in the previous two chapters, it is clear that for small problems the solutions to models solved by the truncated branch-and-bound method are as good or slightly better than the solutions to similar models solved by the switching algorithm. If computer time is limited, then we recommend that the switching heuristic be used. For larger problems, the MIP models would require prohibitive amounts of computing resources. Such problems are better solved by the switching heuristic.

Since the nonlinear programming methods and branch-and-bound have known computational shortcomings, we conclude that the heuristic procedure is the best of the methods considered for solving satellite synthesis problems.

Our other important conclusions come from Chapter IV in which we compare mixed integer programming models with different objective functions. In that chapter we show that the objective of minimizing deviations from desired locations (models 3,4, and 5) have an edge over the other objectives because:

(1) Optimal solutions are found with models 3,4, and 5 a greater number of times than with any other model.

(2) The time to the first feasible solution and the time to the best solution for models 3,4, and 5 are less than half of the overall average times for the other point allotment models.

(3) The solutions to models 3,4, and 5 give good values in the objective function of model 7 (minimizing the total arc occupied by the satellites to be allotted locations).

Summarizing, we recommend that for point allotment problems model 3,4, or 5 be used. Model 5 might not find a feasible solution for some problems and one must keep this in mind while deciding between models 3 and 5.

The two arc allotment models we consider are models 8 and 9. We have mentioned in Chapter IV that model 8 finds solutions in which the length of the arc segment allotted to every administration is the same. On the other hand, model 9 finds solutions in which the length of the arc segment allotted to every administration is proportional to some criterion. We also note that the time to the first feasible solution and the time to the best solution is less for model 9 than for model 8. Hence, for arc allotment problems, model 9 seems more practical.

Appendix A

MIXED INTEGER PROGRAMMING MODELS

NOTATION

We define the following notation for the formulations discussed in this Appendix:

Parameters.

$E_j(W_j)$ = Easternmost (westernmost) feasible location for satellite j, in degrees west longitude.

E = Minimum $[E_j]$ over all j (assumed to be zero).

W = Maximum $[W_j]$ over all j.

Δs_{ij} = Worst-case minimum required separation between satellites i and j, in degrees longitude.

M = Large positive constant

$$\delta_{ij} = \begin{cases} -M & \text{if } \Delta s_{ij}=0 \\ \Delta s_{ij} & \text{if } \Delta s_{ij} > 0 \end{cases}$$

D_j = Desired location of satellite j, in degrees west longitude
= $(W_j - E_j)/2$ (default)

L_j = Constant inversely proportional to the length of the feasible arc of satellite j (used in model 5).
= $1/(W_j - E_j)$.

P_j = Population of country with satellite j.

C_j = Constant proportional to the weight assigned to each satellite j (used in model 9).
$= (p_j) / \sum_j (p_j)$

<u>Decision Variables:</u>

x_j = Orbital location of satellite j (degrees west of E), in degrees west longitude.

$x_j^+(x_j^-)$ = Degrees west (east) of D_j that satellite j is located, in degrees west longitude.

x_0 = Dummy satellite location corresponding to the easternmost satellite location, in degrees west longitude.

x_{n+1} = Dummy satellite location corresponding to the westernmost satellite location, in degrees west longitude.

$w_j(e_j)$ = Western (eastern) edge of the arc segment allotted satellite j, in degrees west longitude.

$$x_{ij} = \begin{cases} 1 & \text{if satellite i is located to the west of satellite j.} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{oj} = \begin{cases} 1 & \text{if satellite j is located closer to its western boundary than to its eastern boundary.} \\ 0 & \text{otherwise} \end{cases}$$

$$x'_{ij} = \begin{cases} 1 & \text{if the western-edge of satellite } i \text{ is located to the west of western-edge of satellite } j. \\ 0 & \text{otherwise} \end{cases}$$

n = Minimum gap between adjacent satellites, over all pairs of adjacent satellites, in degrees longitude.

p = Maximum deviation of allotted location from given desired satellite location, over all satellites, in degrees longitude.

q = Maximum deviation of allotted satellite location from the closest boundary of satellite's feasible arc, over all satellites, in degrees longitude.

r = Length of the minimum arc segment allotted to a satellite, over all satellites, in degrees longitude.

t = Length of the minimum weighted arc segment allotted to a satellite, over all satellites, in degrees longitude.

Model 2: Maximize the minimum gap between adjacent satellites.

$$\text{Maximize} \qquad z = n \qquad \text{(A.1)}$$

subject to,

$$(x_i - x_j) + [(E-W) - \Delta s_{ij}]x_{ij} \geq (E-W) \qquad \text{(A.2)} \ \forall i,j \ni i<j$$

$$(-x_i + x_j) - [(E-W) - \Delta s_{ij}]x_{ij} \geq \Delta s_{ij} \qquad \text{(A.3)} \ \forall i,j \ni i<j$$

$$(x_i - x_j) + [2(E-W)]x_{ij} - n \geq 2(E-W) \qquad \text{(A.4)} \ \forall i,j \ni i<j$$

$$(-x_i + x_j) - [2(E-W)]x_{ij} - n \geq 0 \qquad \text{(A.5)} \ \forall i,j \ni i<j$$

$$E_j \leq x_j \leq W_j \qquad \text{(A.6)} \ \forall j$$

$$x_{ij} = 0 \text{ or } 1 \qquad \text{(A.7)} \ \forall i,j \ni i<j$$

$$x_j \geq 0 \qquad \text{(A.8)} \ \forall j$$

$$n \geq 0 \qquad \text{(A.9)}$$

The objective function (A.1) maximizes the minimum gap between a pair of adjacent satellites. Constraints (A.2) and (A.3) are dichotomous constraints which ensure that minimum satellite separation requirements are enforced. Constraints (A.4) and (A.5) are also dichotomous constraints and they measure the minimum gap between adjacent satellites for all pairs of satellites i and j. This minimum gap is denoted by n. Constraint (A.6) ensures that the allotted location for satellites is feasible. The integrality of the binary variables is enforced by (A.7). The non-negativity of the continuous variables is enforced by (A.8) and (A.9).

For a problem with m satellites, this formulation entails 2m(m-1) structural constraints (A.2), (A.3), (A.4) and (A.5), m+1 continuous variables and m(m-1)/2 binary variables.

Model 3: Minimize the total deviation of allotted locations from given desired locations, subject to feasible arc restrictions [11].

$$\text{Minimize} \qquad z = \sum_{i} (x_j^{+} + x_j^{-}) \qquad (A.10)$$

subject to,

$$(x_i - x_j) + [(E-W) - \Delta s_{ij}] x_{ij} \geq (E-W) \qquad (A.11)\ \forall i,j \ni i<j$$

$$(-x_i + x_j) - [(E-W) - \Delta s_{ij}] x_{ij} \geq \Delta s_{ij} \qquad (A.12)\ \forall i,j \ni i<j$$

$$x_j - x_j^{+} + x_j^{-} = D_j \qquad (A.13)\ \forall j$$

$$E_j \leq x_j \leq W_j \qquad (A.14)\ \forall j$$

$$x_{ij} = 0 \text{ or } 1 \qquad (A.15)\ \forall i,j \ni i<j$$

$$x_j, x_j^{+}, x_j^{-} \geq 0 \qquad (A.16)\ \forall j$$

The objective function (A.10) minimizes the total deviation of allotted locations from desired locations. Constraints (A.11) and (A.12) are dichotomous constraints which ensure that minimum satellite separation requirements are enforced. The deviation of the allotted location from the desired location for each satellite is measured in constraint (A.13). Constraint (A.14) ensures that the allotted location for each satellite is feasible. The integrality of the binary variables is enforced by (A.15). The non-negativity of the continuous variables is enforced by (A.16).

For a problem with m satellites, this formulation entails m^2 structural constraints (A.11), (A.12) and (A.13), 3m continuous variables and m(m-1)/2 binary variables.

Model 4: Minimize the maximum deviation of allotted location from given desired location, subject to feasible arc restrictions [20].

$$\text{Minimize} \quad z = p \qquad (A.17)$$

subject to,

$$(x_i - x_j) + [(E-W) - \Delta s_{ij}]x_{ij} \geq (E-W) \qquad (A.18)\ \forall i,j \ni i<j$$

$$(-x_i + x_j) - [(E-W) - \Delta s_{ij}]x_{ij} \geq \Delta s_{ij} \qquad (A.19)\ \forall i,j \ni i<j$$

$$x_j - x_j^+ + x_j^- = D_j \qquad (A.20)\ \forall j$$

$$x_j^+ + x_j^- - p \leq 0 \qquad (A.21)\ \forall j$$

$$E_j \leq x_j \leq W_j \qquad (A.22)\ \forall j$$

$$x_{ij} = 0 \text{ or } 1 \qquad (A.23)\ \forall i,j \ni i<j$$

$$x_j, x_j^+, x_j^- \geq 0 \qquad (A.24)\ \forall j$$

$$p \geq 0 \qquad (A.25)$$

The objective function (A.17) minimizes p, the maximum deviation of an allotted location from the corresponding desired location. Constraints (A.18) and (A.19) are dichotomous constraints which ensure that minimum satellite separation requirements are enforced. The deviation of the allotted location from the corresponding desired location for each satellites is measured in constraint (A.20). The maximum deviation of an allotted location from its corresponding desired location is measured by constraint (A.21). Constraint (A.22) ensures that the allotted location for all satellites is feasible. The integrality of the binary variables is enforced by (A.23). The non-negativity of the continuous variables is enforced by (A.24) and (A.25).

For a problem with m satellites, this formulation entails m(m+1) structural constraints (A.18), (A.19), (A.20) and (A.21), 3m+1 continuous variables and m(m-1)/2 binary variables.

Model 5: Minimize the total weighted deviation of allotted locations from given desired locations [20].

Minimize $$z = \sum_j L_j(x_j^+ + x_j^-) \qquad (A.26)$$

subject to,

$$(x_i-x_j) + [(E-W) - \Delta s_{ij}]x_{ij} \geq (E-W) \qquad (A.27)\ \forall i,j \ni i<j$$

$$(-x_i+x_j) - [(E-W) - \Delta s_{ij}]x_{ij} \geq \Delta s_{ij} \qquad (A.28)\ \forall i,j \ni i<j$$

$$x_j - x_j^+ + x_j^- = D_j \qquad (A.29)\ \forall j$$

$$x_{ij} = 0 \text{ or } 1 \qquad (A.30)\ \forall i,j \ni i<j$$

$$x_j, x_j^+, x_j^- \geq 0 \qquad (A.31)\ \forall j$$

The objective function (A.26) minimizes the total weighted deviation of allotted locations from given desired locations. The weight, L_j, assigned to each satellite j is the reciprocal of the length of the feasible arc of satellite j. By weighing the objective function in this way we give more weight to administrations with smaller arcs and thereby attempt to ensure that the solution will be feasible. Constraints (A.27) and (A.28) are dichotomous constraints which ensure that minimum satellite separation requirements are enforced. The deviation of the allotted location from the corresponding desired location for each satellite is measured in constraint (A.29). The integrality of the binary variables is enforced by (A.30). The non-negativity of the continuous variables is enforced by (A.31).

For a problem with m satellites, this formulation entails m^2 structural constraints (A.27), (A.28) and (A.29), 3m continuous variables and m(m-1)/2 binary variables.

Model 6: Minimize the maximum deviation of allotted location from the closest boundary of feasible arc.

Minimize $\quad z = q \qquad$ (A.32)

subject to,

$(x_i - x_j) + [(E-W) - \Delta s_{ij}]x_{ij} \geq (E-W) \qquad$ (A.33) $\forall i,j \ni i<j$

$(-x_i + x_j) - [(E-W) - \Delta s_{ij}]x_{ij} \geq \Delta s_{ij} \qquad$ (A.34) $\forall i,j \ni i<j$

$(W_j - x_j) + (W-E)x_{oj} - q \leq (W-E) \qquad$ (A.35) $\forall j$

$(x_j - E_j) - (W-E)x_{oj} - q \leq 0 \qquad$ (A.36) $\forall j$

$E_j \leq x_j \leq W_j \qquad$ (A.37) $\forall j$

$x_{ij} = 0$ or $1 \qquad$ (A.38) $\forall i,j \ni i<j$

$x_{oj} = 0$ or $1 \qquad$ (A.39) $\forall j$

$x_j \geq 0 \qquad$ (A.40) $\forall j$

$q \geq 0 \qquad$ (A.41)

The objective function (A.32) minimizes the maximum deviation of a satellite's allotted location from the closest boundary of that satellite's feasible arc. Constraints (A.33) and (A.34) are dichotomous constraints which ensure that minimum satellite separation requirements are enforced. Constraints (A.35) and (A.36) are also dichotomous or "either-or" constraints. Exactly one of these constraints will be redundant. The binary variable x_{oj} takes values of 1 or 0, depending on whether satellite j is located closer to its western boundary or eastern boundary. If satellite j is located closer to its western boundary the constraint (A.36) is redundant and (A.35) measures the deviation from the western boundary. If $x_{oj}=0$ then constraint (A.36) measures the deviation of satellite j from the eastern boundary. The maximum deviation from the closest boundary is denoted by q. Constraint (A.37) ensures that the allotted location for satellites is feasible. The integrality of the binary variables is enforced by (A.38) and (A.39). The non-negativity of the continuous variables is enforced by (A.40) and (A.41).

For a problem with m satellites, this formulation entails m(m+1) structural constraints (A 33), (A.34), (A.35) and (A.36), m+1 continuous variables and m(m+1)/2 binary variables.

Model 7: Minimize the length of the total arc occupied by the satellites to be allotted locations [20].

Minimize $\quad z = x_{n+1} - x_0 \qquad$ (A.42)

subject to,

$(x_i - x_j) + [(E-W) - \Delta s_{ij}]x_{ij} \geq (E-W) \qquad$ (A.43) $\forall i,j \ni i<j$

$(-x_i + x_j) - [(E-W) - \Delta s_{ij}]x_{ij} \geq \Delta s_{ij} \qquad$ (A.44) $\forall i,j \ni i<j$

$x_{n+1} - x_j \geq 0 \qquad$ (A.45) $\forall j$

$x_0 - x_j \leq 0 \qquad$ (A.46) $\forall j$

$E_j \leq x_j \leq W_j \qquad$ (A.47) $\forall j$

$x_{ij} = 0$ or $1 \qquad$ (A.48) $\forall i,j \ni i<j$

$x_j \geq 0 \qquad$ (A.49) $\forall j$

In this model two dummy variables, x_0 and x_{n+1}, are introduced. These correspond to the easternmost and westernmost allotted satellite locations, respectively. The objective function (A.42) minimizes the total arc occupied by the allotted satellites, which is equal to the distance between the easternmost and westernmost satellite locations.

Constraints (A.43) and (A.44) are dichotomous constraints which enforce the minimum satellite separation requirements. Constraints (A.45) and (A.46) ensure that the dummy satellites occupy locations that are separated by at least the greatest distance between any two satellites. Constraint (A.47) ensures that the allotted location for each satellite is feasible. The integrality of the binary variables is enforced by (A.48). The non-negativity of the continuous variables is enforced by (A.49).

For a problem with m satellites, this formulation entails $m(m+1)$ structural constraints (A.43), (A.44), (A.45) and (A.46), $m+2$ continuous variables and $m(m-1)/2$ binary variables.

Model 8: Maximize the length of the minimum arc segment allotted to any satellite [17].

Maximize $\quad z = r$ (A.50)

subject to,

$(e_i - w_j) + [(E-W) - \delta_{ij}]x'_{ij} \geq (E-W)$ (A.51) $\forall i,j \ni i<j$

$(e_j - w_i) + [(E-W) - \delta_{ij}]x'_{ij} \geq \delta_{ij}$ (A.52) $\forall i,j \ni i<j$

$w_j - e_j - r \geq 0$ (A.53) $\forall j$

$e_j \geq E_j$ (A.54) $\forall j$

$w_j \leq W_j$ (A.55) $\forall j$

$x'_{ij} = 0$ or 1 (A.56) $\forall i,j \ni i<j$

$e_j, w_j \geq 0$ (A.57) $\forall j$

$r \geq 0$ (A.58)

The objective function (A.50) maximizes the length of the shortest arc segment allotted to any satellite. Constraints (A.51) and (A.52) are dichotomous constraints which ensure minimum satellite separation requirements by enforcing the required satellite separation between the western edge of an allotted arc and the eastern edge of the arc of adjacent satellites. By replacing Δs_{ij} by δ_{ij} we permit arcs to overlap if the minimum required satellite separation is zero. The length of the minimum arc segment allotted to any satellite is measured in constraint (A.53), and is denoted by r. Constraints (A.54) and (A.55) ensure that the arc segment allotted to each satellite lies within its feasible arc. The integrality of the binary variables is enforced by (A.56). The non-negativity of the continuous variables is enforced by (A.57) and (A.58).

For a problem with m satellites, this formulation entails m^2 structural constraints (A.51), (A.52) and (A.53), 2m+1 continuous variables and m(m-1)/2 binary variables.

Model 9: Maximize the length of the minimum weighted arc segment allotted to any satellite [17].

$$\text{Maximize} \quad z = t \tag{A.59}$$

subject to,

$$(e_i - w_j) + [(E-W) - \delta_{ij}]x'_{ij} \geq (E-W) \qquad \text{(A.60)} \ \forall i,j \ni i<j$$

$$(e_j - w_i) + [(E-W) - \delta_{ij}]x'_{ij} \geq \delta_{ij} \qquad \text{(A.61)} \ \forall i,j \ni i<j$$

$$w_j - e_j - c_j t \geq 0 \qquad \text{(A.62)} \ \forall j$$

$$e_j \geq E_j \qquad \text{(A.63)} \ \forall j$$

$$w_j \leq W_j \qquad \text{(A.64)} \ \forall j$$

$$x'_{ij} = 0 \text{ or } 1 \qquad \text{(A.65)} \ \forall i,j \ni i<j$$

$$e_j, w_j \geq 0 \qquad \text{(A.66)} \ \forall j$$

$$t \geq 0 \tag{A.67}$$

The objective function (A.59) maximizes the length of the shortest weighted arc segment allotted to any satellite. Constraints (A.60) and (A.61) are dichotomous constraints which ensure minimum satellite separation requirements by enforcing the required separation between the western edge of an allotted arc and the eastern edge of the arc of adjacent satellites. By replacing Δs_{ij} by δ_{ij} we permit arcs to overlap if the minimum required satellite separation is zero. The length of the minimum weighted arc segment allotted to any satellite is measured in constraint (A.62), and is denoted by t. Constraints (A.63) and (A.64) ensure that the arc segment allotted to each satellite lies within its feasible arc. The integrality of the binary variables is enforced by (A.65). The non-negativity of the continuous variables is enforced by (A.66) and (A.67).

For a problem with m satellites, this formulation entails m^2 structural constraints (A.60), (A.61) and (A.62), 2m+1 continuous variables and m(m-1)/2 binary variables.

Appendix B

COMPUTER RESULTS OF MIP FORMULATIONS

Table 11

COMPUTER RESULTS OF MIP FORMULATIONS (E. EUROPE)

MODEL	FIRST FEASIBLE SOLUTION		BEST SOLUTION		FINAL SOLN.
	Obj Fn. Value	Time Min	Obj Fn. Value	Time Min.	Time Min.
1	4.54	.31	9.33	3.45	20.07[2]
2	7.45	.71	11.27	1.66	20.20[2]
3	77.06	.15	49.87	10.21	20.12[2]
4	12.32	.19	8.48	18.95	20.06[2]
5	69.75	.13	58.21	3.53	20.03[2]
6	14.22	.11	2.71	3.58	20.09[2]
7	32.81	.23	25.48	7.86	20.21[2]
8	5.51	.19	8.75	14.49	20.22[2]
9	6.43	.17	10.06	1.46	20.19[2]

NOTE: 1 - Proven Optimum solution

2 - Terminating at the maximum allotted CPU time.

Table 12

COMPUTER RESULTS OF MIP FORMULATIONS (W. EUROPE)

MODEL	FIRST FEASIBLE SOLUTION		BEST SOLUTION		FINAL SOLN.
	Obj Fn. Value	Time Min.	Obj Fn. Value	Time Min.	Time Min.
1	4.77	.29	8.38	10.38	20.11[2]
2	8.22	.55	10.36	1.68	20.18[2]
3	61.92	.09	29.67	0.69	5.22[1]
4	7.25	.17	6.62	0.56	2.59[1]
5	46.12	.10	26.69	3.45	7.15[1]
6	12.33	.17	7.51	18.67	20.11[2]
7	33.13	.26	23.71	19.35	20.02[2]
8	5.61	.10	6.00	0.21	0.28[1]
9	7.79	.21	10.14	19.73	20.06[2]

NOTE: 1 - Proven Optimum solution

2 - Terminating at the maximum allotted CPU time.

Table 13

COMPUTER RESULTS OF MIP FORMULATIONS (S. AMERICA)

MODEL	FIRST FEASIBLE SOLUTION		BEST SOLUTION		FINAL SOLN.
	Obj Fn. Value	Time Min.	Obj Fn. Value	Time Min.	Time Min.
1	7.77	.41	10.55	11.83	20.22[2]
2	11.27	1.05	13.20	15.25	20.15[2]
3	33.55	.13	30.44	1.44	20.02[2]
4	6.96	.19	6.60	0.21	20.02[2]
5	38.02	.10	27.70	1.30	20.04[2]
6	41.64	.09	0.86	17.07	20.05[2]
7	28.66	.35	22.50	6.27	20.08[2]
8	7.68	.37	9.85	16.41	20.24[2]
9	9.67	.15	11.99	2.30	20.01[2]

NOTE: 1 - Proven Optimum solution

2 - Terminating at the maximum allotted CPU time.

Table 14

COMPUTER RESULTS OF MIP FORMULATIONS (N. AFRICA)

MODEL	FIRST FEASIBLE SOLUTION		BEST SOLUTION		FINAL SOLN.
	Obj Fn. Value	Time Min	Obj Fn. Value	Time Min.	Time Min.
1	11.10	.15	16.90	2.11	20.13 [2]
2	12.67	.30	18.44	5.36	20.20 [2]
3	8.65	.01	8.65	.01	0.06 [1]
4	1.64	.01	1.64	.01	0.07 [1]
5	6.85	.01	6.85	.01	0.05 [1]
6	2.99	.01	0.00	.77	0.81 [1]
7	17.48	.10	16.27	15.47	20.04 [2]
8	11.35	.15	14.85	8.36	20.19 [2]
9	12.37	.07	14.42	4.27	20.10 [2]

NOTE: 1 - Proven Optimum solution

2 - Terminating at the maximum allotted CPU time.

Table 15

COMPUTER RESULTS OF MIP FORMULATIONS (S.E. ASIA)

MODEL	FIRST FEASIBLE SOLUTION		BEST SOLUTION		FINAL SOLN.
	Obj Fn. Value	Time Min	Obj Fn. Value	Time Min.	Time Min.
1	9.69	.13	14.69	17.14	20.09[2]
2	14.00	.26	16.44	.75	20.20[2]
3	24.11	.02	23.05	.54	1.60[1]
4	5.88	.05	5.19	.11	1.33[1]
5	26.59	.03	18.31	.14	0.91[1]
6	12.25	.05	0.74	6.05	20.04[2]
7	26.16	.08	18.99	18.88	20.04[2]
8	9.06	.11	13.18	6.93	20.11[2]
9	8.67	.03	8.67	.03	0.09[1]

NOTE: 1 - Proven Optimum solution

2 - Terminating at the maximum allotted CPU time.

Appendix C

COMPARISON OF SOLUTION QUALITY OF MIP MODELS

Table 16

COMPARISON OF SOLUTION QUALITY (E. EUROPE)

BEST SOLN. TO MODEL	OBJECTIVE FUNCTION VALUE IN MODEL						
	1	2	3	4	5	6	7
1	9.3	9.3	376.8	62.0	357.7	55.7	124.0
2	6.7	11.3	412.8	62.0	404.9	50.4	124.0
3	0.0	0.6	49.9	10.7	51.2	61.0	28.4
4	0.0	0.5	71.6	8.5	70.2	54.3	27.3
5	0.0	0.3	60.9	14.5	58.2	59.5	32.0
6	0.0	1.3	593.0	61.0	563.1	2.7	124.0
7	0.0	0.4	270.6	38.4	282.3	47.9	25.5

c.2

Table 17

COMPARISON OF SOLUTION QUALITY (W. EUROPE)

BEST SOLN. TO MODEL	OBJECTIVE FUNCTION VALUE IN MODEL						
	1	2	3	4	5	6	7
1	8.4	9.3	349.5	59.0	365.4	55.5	114.0
2	8.0	10.4	366.6	59.0	358.3	50.5	114.0
3	0.0	1.3	29.7	11.8	26.6	61.4	34.8
4	0.0	1.8	53.4	6.6	96.7	61.4	35.3
5	0.0	1.3	29.7	11.8	26.6	61.4	34.8
6	0.0	1.3	575.9	62.0	568.0	7.5	120.0
7	0.0	0.9	230.6	35.3	256.1	53.1	23.7

Table 18

COMPARISON OF SOLUTION QUALITY (S. AMERICA)

BEST SOLN. TO MODEL	OBJECTIVE FUNCTION VALUE IN MODEL						
	1	2	3	4	5	6	7
1	10.6	10.6	496.0	68.0	407.3	61.4	152.0
2	8.1	13.2	530.4	68.0	436.5	55.6	160.0
3	0.0	0.6	30.4	11.4	28.4	68.0	35.0
4	0.0	0.2	55.3	6.6	46.3	66.8	22.5
5	0.0	0.3	30.4	10.0	27.7	67.3	33.8
6	0.0	0.5	774.7	68.0	647.2	0.9	160.0
7	0.0	0.0	556.4	57.6	468.7	29.0	22.5

Table 19

COMPARISON OF SOLUTION QUALITY (N. AFRICA)

BEST SOLN. TO MODEL	OBJECTIVE FUNCTION VALUE IN MODEL						
	1	2	3	4	5	6	7
1	16.9	17.4	400.7	64.0	323.5	57.8	166.0
2	18.4	13.4	431.1	63.0	350.8	45.8	166.0
3	0.0	2.0	8.7	3.3	7.1	64.3	43.0
4	0.0	0.7	13.0	1.6	10.4	63.4	44.3
5	0.0	2.0	8.7	3.3	6.9	64.0	46.3
6	2.9	6 0	616.0	65.0	500.0	0.0	166.0
7	0.0	0.9	363.4	60.0	295.3	48.8	16.3

Table 20

COMPARISON OF SOLUTION QUALITY (S.E. ASIA)

BEST SOLN. TO MODEL	OBJECTIVE FUNCTION VALUE IN MODEL						
	1	2	3	4	5	6	7
1	14.7	15.3	386.9	64.0	316.3	47.1	148.0
2	12.2	16.4	391.1	64.0	314.5	49.8	148.0
3	0.0	1.4	23.0	7.1	18.4	66.7	27.6
4	0.0	0.4	40.0	5.2	34.6	62.8	32.2
5	0.0	1.4	23.1	9.8	18.3	66.8	27.8
6	0.4	1.4	588.7	67.0	498.9	0.7	144.0
7	0.0	0.0	319.9	47.8	279.8	37.9	19.0

Appendix D

ORDERING OF SATELLITE ALLOTMENTS IN MIP MODELS

Table 21

CHANGE IN THE ORDERING OF SATELLITES (E. EUROPE)

OBJ 1(F)	1	8	10	6	4	12	9	11	7	5	3	2
OBJ 1(B)	11	7	12	3	10	8	1	2	6	4	9	5
OBJ 2(F)	1	10	11	12	6	9	8	7	4	3	5	2
OBJ 2(B)	8	3	11	6	10	2	9	1	12	7	4	5
OBJ 3(F)	2	3	1	4	7	10	9	11	12	8	6	5
OBJ 3(B)	11	12	8	9	1	5	7	10	4	6	2	3
OBJ 4(F)	2	3	1	4	9	11	12	8	10	7	5	6
OBJ 4(B)	12	8	6	5	11	10	2	9	1	7	4	3
OBJ 5(F)	2	3	1	5	7	10	9	8	6	4	11	12
OBJ 5(B)	9	12	11	6	4	1	7	5	10	3	8	2
OBJ 6(F)	2	5	4	3	11	7	9	6	1	10	12	8
OBJ 6(B)	12	5	7	2	6	1	10	11	9	8	3	4
OBJ 7(F)	3	9	10	12	4	7	11	8	6	5	1	2
OBJ 7(B)	1	8	9	4	6	2	11	10	7	5	12	3
OBJ 8(F)	2	5	1	4	6	11	10	8	9	12	7	3
OBJ 8(B)	2	4	11	7	1	5	9	10	3	6	8	12
OBJ 9(F)	2	5	4	1	10	3	6	7	12	9	11	8
OBJ 9(B)	3	7	4	9	2	1	12	5	10	8	6	11

Table 22

CHANGE IN THE ORDERING OF SATELLITES (W. EUROPE)

OBJ 1(F)	3	2	8	9	10	11	12	7	4	5	6	1
OBJ 1(B)	1	9	12	4	11	6	8	2	7	10	3	5
OBJ 2(F)	5	3	2	9	11	12	8	10	6	7	4	1
OBJ 2(B)	1	9	7	4	11	12	8	2	10	6	3	5
OBJ 3(F)	2	3	1	9	4	6	7	5	10	11	12	8
OBJ 3(B)	11	12	7	10	9	8	6	5	3	1	4	2
OBJ 4(F)	2	3	1	4	5	6	10	7	8	12	9	11
OBJ 4(B)	11	9	12	8	7	10	4	6	5	3	1	2
OBJ 5(F)	2	1	3	5	6	7	8	4	10	11	12	9
OBJ 5(B)	11	12	7	10	9	8	6	5	3	1	4	2
OBJ 6(F)	5	1	4	7	6	11	12	9	2	3	8	10
OBJ 6(B)	7	1	4	3	2	9	11	10	8	12	6	5
OBJ 7(F)	3	5	6	10	12	7	8	9	11	4	2	1
OBJ 7(B)	4	8	12	9	1	3	11	5	2	6	10	7
OBJ 8(F)	4	1	6	7	2	5	12	3	10	8	11	9
OBJ 8(B)	7	4	11	1	2	9	6	3	5	12	8	10
OBJ 9(F)	4	5	3	7	2	8	12	11	9	10	6	1
OBJ 9(B)	5	4	11	6	2	7	12	3	1	8	9	10

Table 23

CHANGE IN THE ORDERING OF SATELLITES (S. AMERICA)

OBJ 1(F)	2	13	9	5	4	12	7	3	10	11	1	8	6
OBJ 1(B)	9	7	4	13	1	5	11	12	8	6	3	2	10
OBJ 2(F)	2	3	13	4	12	5	11	8	1	9	7	10	6
OBJ 2(B)	6	9	1	8	4	7	13	2	3	5	12	11	10
OBJ 3(F)	3	13	10	11	8	7	2	12	1	5	4	9	6
OBJ 3(B)	6	9	1	5	4	12	2	7	8	11	10	13	3
OBJ 4(F)	3	13	10	8	7	11	2	12	1	5	4	9	6
OBJ 4(B)	6	9	4	5	1	12	2	7	11	13	8	10	3
OBJ 5(F)	3	11	13	10	8	7	2	1	5	4	12	9	6
OBJ 5(B)	6	9	1	12	4	5	2	7	8	11	10	13	3
OBJ 6(F)	10	11	2	8	3	13	9	6	12	5	4	7	1
OBJ 6(B)	6	2	5	7	1	13	12	4	9	3	8	11	10
OBJ 7(F)	3	12	11	13	10	9	8	7	4	6	2	5	1
OBJ 7(B)	13	1	9	12	8	6	11	7	2	5	10	4	3
OBJ 8(F)	2	5	4	6	11	7	9	10	12	13	3	8	1
OBJ 8(B)	10	2	11	12	8	6	13	3	1	5	4	7	9
OBJ 9(F)	7	11	10	8	3	6	1	5	13	12	2	4	9
OBJ 9(B)	7	8	10	11	3	13	4	12	1	5	2	9	6

Table 24

CHANGE IN THE ORDERING OF SATELLITES (N. AFRICA)

OBJ 1(F)	10	5	1	4	6	3	8	7	9	2
OBJ 1(B)	5	2	4	1	3	7	10	8	9	6
OBJ 2(F)	10	5	4	1	6	7	8	9	3	2
OBJ 2(B)	5	3	1	9	8	4	10	2	7	6
OBJ 3(F)	6	7	8	1	9	2	10	3	4	5
OBJ 3(B)	5	4	3	10	2	9	1	8	7	6
OBJ 4(F)	6	7	8	1	9	2	10	3	4	5
OBJ 4(B)	5	4	3	10	9	2	1	8	7	6
OBJ 5(F)	6	7	8	1	9	2	10	3	4	5
OBJ 5(B)	5	4	3	10	2	9	1	8	7	6
OBJ 6(F)	6	7	8	1	9	2	10	3	5	4
OBJ 6(B)	5	3	9	8	7	4	10	2	1	6
OBJ 7(F)	2	4	7	8	9	5	10	6	3	1
OBJ 7(B)	1	5	7	10	8	3	9	6	4	2
OBJ 8(F)	1	3	5	6	10	7	4	8	9	2
OBJ 8(B)	7	9	8	10	6	3	1	4	2	5
OBJ 9(F)	7	2	8	10	6	9	4	1	5	3
OBJ 9(B)	7	2	6	10	1	3	8	9	4	5

Table 25

CHANGE IN THE ORDERING OF SATELLITES (S.E. ASIA)

OBJ 1(F)	1	9	6	4	7	8	10	5	3	2
OBJ 1(B)	10	4	7	6	3	8	5	1	9	2
OBJ 2(F)	1	6	4	8	10	5	3	9	7	2
OBJ 2(B)	10	5	2	6	7	8	3	9	4	1
OBJ 3(F)	3	1	2	4	7	8	5	9	6	10
OBJ 3(B)	10	6	9	5	8	7	4	1	2	3
OBJ 4(F)	3	1	7	2	4	5	8	6	9	10
OBJ 4(B)	10	9	4	6	5	8	7	3	2	1
OBJ 5(F)	1	3	2	4	7	8	5	10	9	6
OBJ 5(B)	9	10	6	5	8	7	4	1	2	3
OBJ 6(F)	2	5	4	7	6	9	3	10	8	1
OBJ 6(B)	9	4	7	1	8	10	3	6	5	2
OBJ 7(F)	6	1	4	9	8	10	7	5	2	3
OBJ 7(B)	9	6	3	8	5	2	7	10	1	4
OBJ 8(F)	7	9	1	4	10	8	5	3	6	2
OBJ 8(B)	2	3	6	7	8	5	9	1	4	10
OBJ 9(F)	1	2	5	3	4	9	8	6	7	10
OBJ 9(B)	1	2	5	3	4	8	9	6	7	10

BIBLIOGRAPHY

1.) Baybars, I.,"Optimal Assignment of Broadcasting Frequencies", EUROPEAN JOURNAL OF OPERATIONS RESEARCH, Vol. 9, No. 3, pp. 257-263, March, 1982.

2.) Bhasin, P., "Application of Zoutendijk's Method to Satellite Synthesis" Project ISE 832, Department of Industrial and Systems Engineering, The Ohio State University, Sp. 1986.

3.) Cameron, S., "The Solution of the Graph-Coloring Problem as a Set Covering Problem", IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY, Vol. EMC-19, No. 3, August, 1977.

4.) Chouinard, G., and M. Vachon, " A Synthesis of a Plan by Computer Using an Assignment Method and Inductive Selection", ITU/CITEL/DOC-Canada Preparatory Seminar for 1983 RARC, Ottawa, Canada, May, 1981.

5.) Christensen, J., "BSS CAPS, A System Description", ITU/CITEL/ DOC-Canada Preparatory Seminar For 1983 RARC, Ottawa, Canada, May, 1981.

6.) Final Acts of the World Administrative Radio Conference for the Planning of the Broadcasting-Satellite-Service in Frequency Bands 11.7-12.2 Ghz (in Regions 2 and 3) and 11.7-12.5 Ghz (in Region 1)' Annex 9, "Criteria for Sharing Between Services", Geneva, Switzerland, p. 104, 1977.

7.) Gonsalvez, D., "On Orbital Allotment for Geostationary Satellites", Doctoral Dissertation, The Ohio State University, Dec., 1986.

8.) Ito, Y., T. Mizuno, and T. Muratani, " Effective Utilization of Geostationary Orbit Through Optimization", IEEE TRANSACTIONS ON COMMUNICATIONS, Vol. COM-27, No. 10, pp 1551-1558 October, 1979.

9.) Levis, C., C. Martin, D. Gonsalvez, C. Wang, "Engineering Calculations for Communication Satellite Systems Planning", The Ohio State University, ElectroScience Laboratory, First Summary Report 713533-2 for Grant NAG 3-159, March 1983.

10.) Levis, C., C. Martin, D. Gonsalvez, C. Wang, "Engineering Calculations for Communication Satellite Systems Planning", The Ohio State University, ElectroScience Laboratory, Second Interim Report 713533-3 for Grant NAG 3-159, June 1983.

11.) Levis, C., C. Wang, Y. Yamamura, C.. Reilly, and D. Gonsalvez "The Role of Service Areas in the Optimization of FSS Orbital and Frequency Assignments", A COLLECTION OF TECHNICAL PAPERS, AIAA 11th Communications Satellite Systems Conference, pp. 190-198, 1986.

12.) Luenberger, D., "Introduction to Linear and Nonlinear Programming", Addison- Wesely Publishing Company, 1973.

13.) Martin, C., D. Gonsalvez, C. Levis, C. Wang, "Engineering Calculations for Communication Satellite Systems Planning", The Ohio University, ElectroScience Laboratory, December 1983.

14.) Martin, C., D. Gonsalvez, C. Levis, C. Wang, "Engineering Calculations for Communication Satellite Systems Planning", The Ohio University, ElectroScience Laboratory, Interim Report 713533-4 for Grant NAG 3-159, December 1985.

15.) Mount-Campbell, C., C. Reilly and D. Gonsalvez, "A Mixed Integer Linear Programming Formulation of the FSS Synthesis Problem Using Minimum Required Pair-Wise Separations", Working Paper Series Number 1986-006, The Ohio State University, 1986.

16.) Nedzela, M. and J. Sidney, "A Synthesis Algorithm for Broadcasting-Satellite-Service", ITU/CITEL/DOC-Canada Preparatory Seminar for 1983 RARC, Ottawa, Canada, May, 1981.

17.) Reilly, C., "A Satellite Synthesis Model for Orbital Arc Allotment Optimization", Working Paper Series Number 1986-016, The Ohio State University, July, 1986.

18.) Reilly, C., C. Levis, C. Mount-Campbell, D. Gonsalvez, C. Wang and Y. Yamamura, "Engineering Calculations for Communication Satellite Systems Planning", The Ohio State University, ElectroScience Laboratory, Interim Report 716548-2 for Grant NAG 3-159.

19.) Reilly, C., C. Mount-Campbell, D. Gonsalvez, C. Martin, C. Levis, and C. Wang, "Broadcasting Satellite Synthesis Using Gradient and Cyclic Coordinate Search Procedures", A COLLECTION OF TECHNICAL PAPERS, AIAA 11th. Communication Satellite Systems Conference, pp 237-245, 1986.

20.) Reilly, C., C. Mount-Campbell, D. Gonsalvez, C. Levis, "Alternate Mathematical Programming Formulations for FSS Synthesis", Working Paper Series Number 1986-005, The Ohio State University, July, 1986.

21.) Yamamura, Y. and C. Levis, "Calculation of Allowable Orbital Spacings for the Fixed Satellite Service", Technical Report 716548-1 for Grant NAG 3-159, July, 1985.

22.) Wang, C., "Optimization of Orbital Assignment and Specification of Service Areas in Satellite Communications", Doctoral Dissertation, The Ohio State University, June, 1986.

23.) Zangwill, W. I. , "Nonlinear Programming - A Unified Approach", Prentice Hall Inc., NJ, 1969.

24.) Zoutendijk, G., "Mathematical Programming Methods", North Holland Publishing Company, 1976.