T-1509

SEARCHING FOR THE MULTIPLIERS IN LINEAR BINARY-INTEGER PROBLEMS WITH LINEAR PROGRAMMING

By

Robert W. Wiley

T-1509

A Thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematics).

Signed: Robert W Wiley

Golden, Colorado

Date: Dec, 8, 1972

Approved: Hunter Swanson
Thesis Advisor

Joseph R Lee
Head of Department

Golden, Colorado

Date: Dec. 8, 1972

ABSTRACT

Although the linear binary-integer problem, a problem where the variables are either 0 or 1, is frequently encountered, no simple method for solving it exists. However there are several rigorous and many heuristic methods by which the binary-integer problem can be solved. One of the methods of solution involves forming two problems related to the original problem. One of the problems is a Lagrangian and the other is a linear programming problem. Here the shadow prices from the linear programming problem are the multipliers in the Lagrangian problem. The technique iterates on the two problems to converge on the optimum solution.

CONTENTS

FIGURES AND TABLES

INTRODUCTION

Frequently managers are forced to decide from a list of possible projects which projects to attempt and which to discard or delay. One method of doing this is to create a mathematical model of the problem and solve the model. The model will consist of maximizing the expected benefit (profit, utility, usefulness, etc.) of all the projects that are to be done while making sure that for the projects that will be done the total man-power needed doesn't exceed the man-power available, the total cost doesn't exceed the budget, and the material needed doesn't exceed the material available.

Problems of this type might be typified by the back-packer who was planning to spend several days in the country and there are many items that he would like to take. However his pack has limited size, he can only carry so much weight, and many items must be purchased. Given a list of all items, a measure of the benefit or usefulness of each, the size, the weight, and the cost of each item, the maximum allowable weight, the maximum size, and the budget limitations, he wishes to maximize total utility without violating any of the limitations. Mathematically, this can be formulated as

$$\text{Maximize} \qquad H(x) = \sum_{i=1}^{n} H_i X_i$$

$$\text{Subject to} \quad C_j(x) = \sum_{i=1}^{n} C_{j,i} X_i \leq c_j \qquad j=1,2,3$$

$$X_i = 0 \text{ or } 1 \qquad H_i \geq 0 \qquad C_{j,i} \geq 0 \qquad \text{for all } i \text{ and } j$$

where H_i is the benefit associated with the ith item, $C_{1,i}$ is the cost of the ith item, $C_{2,i}$ is the weight of the ith item, $C_{3,i}$ is the size of the ith item, c_1 is his budget allowance, c_2 is the maximum total weight, c_3 is the maximum size, and $X_i=0$ if the item is not included, and $X_i=1$ if the item is included in the pack.

This problem type is also encountered in loading submarines where size and weight are critical; planning space projects where size, weight, and moment of inertia must be considered; and in almost any problem based on a go, no-go decision.

It is the purpose of this thesis to present one method for solving problems of this type, called linear binary-integer problems. The basic idea of this technique, presented by Brooks[1], used linear programming in an iterative manner to solve the binary-integer problem.

THE LINEAR BINARY-INTEGER PROBLEM

In general, the binary-integer problem may be modeled as:

$$\text{Maximize} \qquad H(x) = \sum_{i=1}^{n} H_i X_i$$

$$\text{Subject to} \qquad C_j(x) = \sum_{i=1}^{n} C_{j,i} X_i \leq c_j \qquad j=1,\ldots,m$$

where H_i = the return or profit from doing project i,

$C_{j,i}$ = the amount of resource j required to do project i,

c_j = the total amount of resource j available,

m = the number of resources considered,

n = the number of projects proposed,

and the variable X_i is either 0 or 1

X_i = 1 if project i is to be done and 0 otherwise.

Note that when the problem has been solved, because X_i is either 0 or 1, H(x) then becomes the sum of the returns of the projects that are to be done and $C_j(x)$ becomes the amount of resource j needed for the projects selected.

Another problem frequently encountered is one in which the variables Y_i assume non negative integer values. However this problem can be reformulated into a binary-integer problem by replacing each such variable Y_i by $2^0 X_{1,i} + 2^1 X_{2,i} + \ldots + 2^{k-1} X_{k,i}$, where k is sufficiently large such that $2^k \geq Y_i^\circ$. Here Y_i° is the least upper bound on the

variable Y_i found by setting all other variables to 0. Also for each variable Y_i the constraint

$$2^0X_{1,i} + 2^1X_{2,i}+\ldots+2^{k-1}X_{k,i} \leq Y_i^0$$

must be added. Here Y_i^0 is an integer and $X_{j,i}$ is either 0 or 1. For example the problem

$$\text{Max.} \quad Z=3\,Y_1 +4\,Y_2$$

$$Y_1 +2\,Y_2 \leq 6$$

$$3\,Y_1 + Y_2 \leq 10$$

$$Y_1, Y_2 \text{ are non negative integers}$$

could be reformulated as

$$\text{Max.} \quad Z=3\,X_{1,1}+6\,X_{2,1}+4\,X_{1,2}+8\,X_{2,2}$$

$$\text{S.T.} \quad X_{1,1}+2\,X_{2,1}+2\,X_{1,2}+4\,X_{2,2} \leq 6$$

$$3\,X_{1,1}+6\,X_{2,1}+ X_{1,2+2}\,X_{2,2} \leq 10$$

$$X_{1,1}+2\,X_{2,1} \leq 3$$

$$X_{1,2}+2\,X_{2,2} \leq 3$$

$$X_{i,j} = 0 \text{ or } 1$$

since from $Y_1+2Y_2\leq 6$, $Y_2\leq 3$; and from $3Y_1+ Y_2\leq 10$, $Y_1\leq 3$. Some difficulties in doing this are that the number of variables can become quite large, the coefficients increase rapidly in magnitude, the number of constraints increase, and the linear binary-integer problem is frequently hard to solve.

Although the problem looks simple, solving problems with relatively few variables can become herculean. This is

because the variables are either 0 or 1 and thus the problem doesn't yield itself to linear programming. However there are several rigorous methods such as explicit enumeration, implicit enumeration done by Balas[2], Geoffrion[3], and Plane[4] and branch and bound done by Davis[5], Lawler[6], and Tuan[7] that yield optimal solutions. These rigorous techniques involve searching over all 2^n potential solutions where n is the number of variables. Explicit enumeration involves examining all possible solutions directly, eliminating those that are infeasible and finding the optimum of those that are left. Implicit enumeration involves examining a few solutions explicitly, and based upon these solutions eliminating all solutions that are obviously not feasible or better than the present optimum. Another solution is then examined explicitly and the procedure is repeated on those solutions that haven't been examined or eliminated. Branch and bound sets up a tree, as in figure 1, and proceeds down a branch until the end of the branch is reached, the solution becomes infeasible, or no solution down the branch will be better than the present optimum. At this point the routine then bounds to another branch. This procedure is continued until all possible branches have been investigated. There are also many heuristics, one of which is a vector search technique by Senju[8]. This technique, although it doesn't guarantee an optimal solution, does provide a near optimum solution very quickly. It does this by setting all the

THE BRANCH AND BOUND TREE FOR THREE VARIABLES

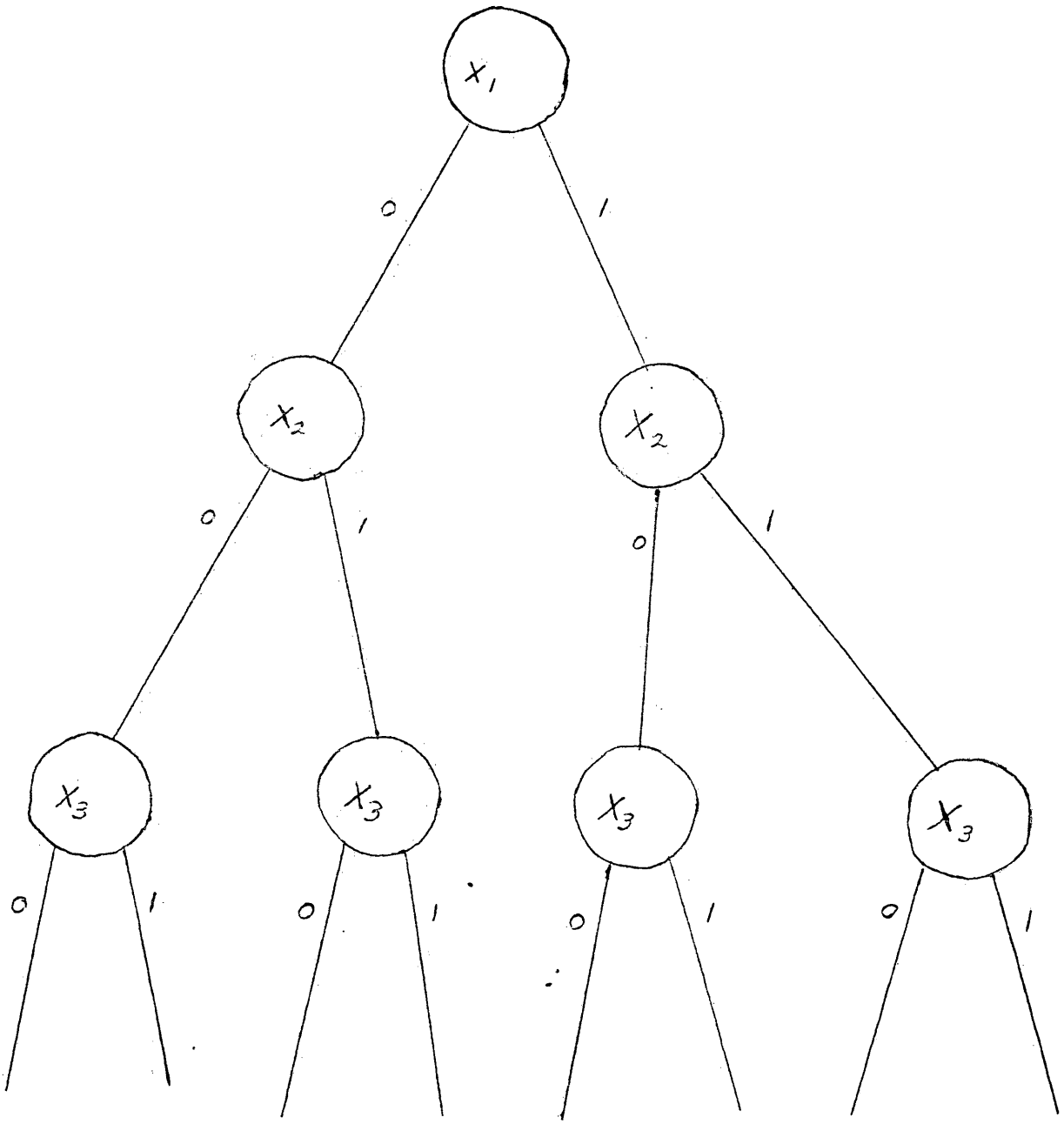

Figure 1

variables equal to 1, and ordering the variables in terms of benefit/use of resource. Then it sets the variables equal to 0 starting at the low order until a feasible solution is reached.

Everett's technique of Generalized Lagrange Multipliers[9] can also be used to solve binary-integer problems. Everett's technique involves reducing the original problem to an unconstrained optimization problem through the use of lagrange multipliers. The main problem in this technique is that Everett doesn't present any method by which the multipliers may be found. However Gulley[10] showed that in problems with one constraint the multiplier is determined by the ratio of the coefficients of the objective function to the coefficients of the constraint. It was also shown that in this case Everett's technique reduced to the Slippery Algorithm. Terry[11] presented a method of finding the multipliers by a random search technique in the general problem.

SOLVING THE LINEAR BINARY-INTEGER PROBLEM WITH G L M

The G L M Technique

Everett states that one way to solve

Maximize $U(x)$

Subject to $R_j(x) \leq r_j \qquad j=1,\ldots,m, \qquad (2)$

where $U(x)$ and $R_j(x)$ are any function of the vector x, is to reformulate it as

Maximize $U(x)-\sum_j \lambda_j R_j(x) \qquad (3)$

Then for any set of non negative λ's, $\lambda_1,\ldots,\lambda_m$, the set of X's (x*) that maximizes (3) also maximizes (2) over every set $x \epsilon S$ (where S is the set of all possible solutions) such that $R_j(x) \leq R_j(x^*)$ for all j. That is the solution to (3) is also the solution to (2) where the r_j are determined by $R_j(x^*)$. Therefore by determing several different λ's, and the corresponding r_j's it is possible to perform sensitivity analysis by comparing the different r_j and the corresponding $U(x^*)$. Although having x* determine r_j is beneficial in sensitivity analysis, it also introduces the added problem that the r_j determined by x* may not be the r_j that were desired. Therefore if there is a solution set, $x^* \epsilon S$, that maximizes (3), it may not provide the optimum solution to (2) if the r_j in (2) are fixed. Therefore to solve (2) the following must be done:

1. Generate a set of lambdas (λ^k) based upon past λ^i's and the corresponding solution sets x^i's.

2. Determine the set x^k corresponding to λ^k.
3. If x^k is not acceptable then set k to k+1 and return to step 1, otherwise stop.

This requires that a method for determining λ^k and x^k is known and a means of knowing when x^k is acceptable. While in general it may be difficult to find the maximum of the unconstrained problem, with the linear binary-integer problem it is very simple. All that must be done is computed

$$A_i=H_i-\sum_{j=1}^{m} \lambda_i c_{j,i}$$

for each i. Then if A_i is negative, set $X_i=0$ and if A_i is positive, set $X_i=1$. The case where $A_i=0$ is a special case as X_i can be either 0 or 1, without affecting the maximization of the Lagrangian function. Therefore when $A_i=0$ the resource requirements for both cases should be considered to determine which answer is best. The other requirements, namely determining λ^k and knowing when x^k is acceptable, will be discussed in the next section.

However, there is another problem and it is that there are usually lattice points in the constraint space that cannot be generated by any set of lambdas. That is there are combinations of the X_i's for which no λ's exist or there are c_j's for which no combination of X_i's exist, such as

$$3X_1+5X_2\leq 7.$$

These lattice points, called gaps[9], are influenced by the number of constraints and the number of variables. Since

the optimization is done over the n+1 dimensional space of payoff versus resource expenditure (PR-space)[9], increasing the number of constraints aids in reducing the number of gaps, or modifying the location of the gaps so that they become less critical to the problem. However, adding constraints to a problem doesn't seem to help remove the gaps. This is because the added constraints usually have the same gaps that all the other constraints have and only increases computational time. Although optimality cannot be verified where a gap exists, bounds on the solution may be obtained and near-optimal solution can be computed. By looking at the solutions that require more resources than at the gap an upper bound can be obtained and by searching over the solutions that require less resources a lower bound can be found. And as a set of lambdas define the right-hand sides of the constraint, it is possible to determine what can be done with a little extra resource.

Using Linear Programming to Determine the Multipliers

As was mentioned before, a means for determining λ^k is needed. To do this Brooks[1] recommends the use of linear programming. A linear programming problem is a problem where the objective function and the constraints are all linear and the variables are all non-negative real numbers. Associated with every linear programming problem, called the primal, is a related problem called the dual. The

primal problem is formulated as

$$\text{Max.} \quad Z=\sum_i d_i X_i$$

$$\text{S.T.} \quad \sum_i a_{j,i} X_i \leq b_j \qquad j=1,\ldots,m$$

$$X_i \geq 0.$$

The associated dual is formulated as

$$\text{Min.} \quad V=\sum_j b_j Y_j$$

$$\text{S.T.} \quad \sum_j a_{j,i} Y_j \leq d_i \qquad i=1,\ldots,n$$

$$Y_i \leq 0.$$

Here $a_{j,i}$, b_j , and d_i are the same in the two problems. It is obvious that Y_j is related to the jth primal constraint, and in fact Y_j^* is referred to as the shadow price of the jth constraint. These shadow prices are used by Brooks' as Everett's multipliers. The primal problem is usually solved by the simplex method[12] which yields the solution to both the primal and the dual by solving only one.

Brooks' recommendation is based upon the observation that if (2) is a linear programming problem then x° and λ° satisfy the condition that x° maximizes (3), where the λ's are the elements of λ°, if x° solves (2) and λ° solves the dual of (2). Therefore by solving

$$\text{Maximize} \quad Y(u)=\sum_{i=1}^{n} u_i H(x^i)$$

$$\text{Subject to} \quad W(u)=\sum_{i=1}^{n} u_i C_j(x^i) \leq c_j \qquad j=1,\ldots,m$$

$$\sum_j u_i = 1$$

the shadow prices become the multipliers. Here u is an arbitrary variable, and the x^i is a given solution set. Thus, to solve (1) by the method recommended by Brooks, m+2 different elements of the set S are required in order to provide the coefficients of the linear programming problem. The right-hand side c_j is the same as in the linear binary-integer problem (see example 1). Once the shadow prices of the linear programming problem are known they are assigned as values of the lambdas (λ^*). These lambdas are used to compute a new x*. This new x* is used to compute new coefficients for one of the variables of the linear programming problem. The coefficients that will be replaced are the coefficients of one of the variables that remained non basic. This process is continued until

$$H(x^*)-\sum_i \lambda_i^* C_i(x^*) \leq \lambda_o^*$$

where λ_o^* is the shadow price associated with the constraint $\sum_{i=1}^{n} u_i=1$.

For example, if the original problem is

$$\text{Max.} \quad Z=4X_1+3X_2+2X_3+X_4$$

$$\text{S.T.} \quad 3X_1+4X_2+2X_3+X_4 \leq 6$$

$$X_1+2X_2+3X_3+4X_4 \leq 6$$

$$X_i = 0 \text{ or } 1$$

and the following vectors are chosen from among possible solutions:

(0,0,1,1)

(0,1,0,1)

(1,0,0,1)

(1,0,1,0)

Then the linear programming problem is

$$\text{Max.} \quad Y=3u_1+4u_2+5u_3+6u_4$$

$$\text{S.T.} \quad 3u_1+5u_2+4u_3+5u_4 \leq 6$$

$$7u_1+6u_2+5u_3+4u_4 \leq 6$$

$$u_1+u_2+u_3+u_4=1$$

Where the coefficients of u_1, u_2, u_3, and u_4 are determined by (0,0,1,1); (0,1,0,1); (1,0,0,1); and (1,0,1,0) respectively.

It is obvious that the initial solution vectors used to start the problem can greatly affect the number of iterations. Also, since there are not limitations on the elements of the set S used, the initial solution vectors used can be either feasible or infeasible. Therefore, if one knew which solution vectors will produce the optimum set of lambdas, the problem could be run very quickly. However, there is no way of knowing beforehand which solution vectors will produce the desired result. For this reason the vectors used in the program were the rows of an m+2 and m+2 identity matrix. However, after the final answer was obtained many of the vectors used to obtain the linear programming problem were infeasible. This would

indicate that possibly a better starting solution vector would be one that is infeasible or possibly a set of vectors where some are feasible and some are infeasible.

As was mentioned before, if there are gaps in the region of the optimal solution, then optimality cannot be verified. In this case the technique will do one of two things. First the technique may find a nearby solution. This nearby solution may be either feasible and suboptimal or it may be infeasible. The infeasible solution is suspected to be caused by the linear programming problem linearizing over the gap. In this manner infeasible solutions may be introduced into the G L M technique through the multipliers. If the technique fails to find a nearby solution then the technique will start looping. That is, it will obtain the same solution repeatedly, choose the same column to replace, and in this way it will fail to obtain an optimal solution.

The Algorithm

From the previous discussion the following algorithm for the solution of the linear binary-integer problem has been developed.

Step 1: Arbitrarly choose $x^k \in S \quad k=1,\ldots,m+2$

Step 2: Compute $H(x^k)$ and $C_j(x^k)$ for all i and j

Step 3: Set up the linear programming problem

$$\text{Max.} \quad z=\sum_k u_k H(x^k)$$

$$\text{S.T.} \quad \sum_k u_k C_j(x^k) \leq c_j \quad j=1,\ldots,m$$

$$\sum_k u_k = 1$$

Step 4: Solve the linear programming problem and set

λ_j = the shadow price of the jth constraint = Y_j^* $\quad j=1,\ldots,m$

λ_o = the shadow price of $\sum_k u_k = 1$

Step 5: Solve Max. $H(x^*) - \sum_{j=1}^{m} \lambda_j^* C_j(x^*)$

$x^* \in S$

Step 6: If $H(x^*) - \sum_{j=1}^{m} \lambda_j^* C_j(x^*) = \lambda_o^*$ stop.

Otherwise find k such that $u_k = 0$.

Replace $C_j(x^k)$ with $C_j(x^*)$ $\quad j=1,\ldots,m$

and $H(x^k)$ with $H(x^*)$.

Go to Step 3.

COMPUTATIONAL RESULTS

To determine the efficiency of the technique of Brooks and Geoffrion, the author has solved several problems using a FORTRAN IV program on a Digital Equipment Corporation PDP-10 owned by the Colorado School of Mines. The method of Senju and Toyoda was also programmed to provide computational times and answers to aid in determining the efficiency of the technique proposed by Brooks and Geoffrion. The optimal answers were found by either a Balas code or, in cases where the number of variables was less than 5 or there was only one constraint, the answers were obtained by hand. As was mentioned before, there are some cases where $A_i=0$. In these cases the corresponding X_i was set to 0. At the end of the program if A_i was 0 then X_i was set to 0 and flagged with a -1. It should be noted that for the problems run if all the X_i's that were flagged with a -1 are set to 1 the solution is still feasible. This would seem to indicate that if $A_i=0$ then X_i should be set to 0 unless optimality has been reached then X_i should be set to 1. Table 1 shows the results of the program and the last column shows the value of the objective function if all variables where $A_i=0$ are set to 1.

Note that in all cases where G L M obtained a feasible solution, the method of Senju and Toyoda was as good as if not better than the solution obtained by G L M. Also, as

expected, there was a problem where G L M failed to find a solution. It should also be pointed out that in all cases where an answer was found the objective function was better than 75% of the optimum. However there were a few cases where the solution was infeasible, and depending upon the problem this may be unacceptable. Also the degree of error in the suboptimal solution might be unacceptable.

The problems used to check out the program and to determine the efficiency of the technique of Brooks and Geoffricn came from three sources. Seven of the problems were from an article by Peterson[13], five were from an article by Trauth and Woolsey[14], and three were developed by the author to check against a hand solution.

TABLE I

Problem Source	Size *	Z	SENJU AND TOYODA			GLM			
			Solution Time **	% of Optimum	Z	Solution Time **	$A_i=0$ % Optimum	$X_i=0$ Z	$A_i=0$; $X_i=1$ Z
Peterson[13]	(6,10)	3800	0	100.	3800	600	84.21	3200	3700
Peterson	(10,10)	8706	33	95.76	8336	550	78.52	6836	7436
Peterson	(15,10)	4015	50	93.65	3760	1166	77.33	3105	3105
Peterson	(20,10)	6120	33	98.20	6010	1683	106.21	6500	6500
Peterson	(28,10)	12400	67	100.	12400	--	--	--	--
Peterson	(39,5)	10618	50	95.99	10192	1383	107.39	11403	11403
Peterson	(50,5)	16537	117	95.74	15832	2567	108.54	17949	17949
Trauth[14]	(10,1)	52	17	100.	52	33	100.	52	52
Trauth	(10,1)	58	17	89.66	52	33	89.66	52	52
Trauth	(10,1)	62	0	83.87	52	33	83.87	52	52
Trauth	(10,1)	67	17	100.	67	33	77.61	52	67
Trauth	(10,1)	68	0	98.53	67	17	98.53	67	67
Author	(4,2)	7	0	100.	7	33	100.	7	7
Author	(3,2)	6	0	83.33	5	17	50.	3	5
Author	(4,3)	8	0	100.	8	66	75.	6	6

* (N,M) N Number of Variables, M Number of Constraints

** In Milliseconds

CONCLUSION

For the problems run, the use of linear programming to find the multipliers required more time than the method of Senju and Toyoda. Also, although the method of Senju and Toyoda cannot guarantee an optimal solution, it was in each case equal to if not better than the answer from G L M.

It is the author's opinion that for problems with few variables some other technique might be used, as in general these techniques require less time on small problems than the method of using linear programming to find Everett's multipliers. However, as this technique is rather insensitive, in terms of time, to increases in the number of variables and constraints, when compared to most other methods, it might prove to be efficient on large problems. The major drawback is that the technique cannot always guarantee an optimal or feasible solution. However, in problems where the amount of the resources available is not absolutely fixed, the solution is the best for the resources used. Also, if the solution is feasible it can be fed into a branch and bound algorithm as a lower bound to obtain an optimum, or if the solution is infeasible it can be adjusted to where it is feasible and then fed into a branch and bound algorithm.

APPENDIX

Contained in the following pages is the listing of the FORTRAN IV program used to solve the linear binary-integer problems reported in this paper. Also contained is the output from the linear binary-integer problems run.

```
      COMMON/SENJU1/A(50,50),B(50),C(50)
      DIMENSION LABEL(20),LFORM(20),IX(70),IY(70)
      INTEGER TTY,CDR,BLANK
      DATA TTY,CDR,LPT/4,5,6/
1     FORMAT (2I10)
2     FORMAT (20A5)
3     FORMAT (/'       THE OPTIMUM VALUE IS',I10)
4     FORMAT (//6X,'THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS
     1 FOLLOWS'//26X,'SENJU',5X,'BROOKS')
5     FORMAT (18X,'X(',I3,')=',2X,I2,9X,I2)
6     FORMAT (/6X,'THE TIME REQUIRED TO RUN BROOKS WAS ',I6,' MILLISECON
     1DS',//6X,'THE VALUE OF THE OBJECTIVE FUNCTION WAS ',I10,/6X,' WHIC
     2H IS ',F7.2,' PERCENT OF THE OPTIMUM')
7     FORMAT (//6X,'THE TIME REQUIRED TO RUN SENJU WAS ',I6,' MILLISECON
     1DS',//6X,'THE VALUE OF THE OBJECTIVE FUNCTION WAS ',I10,/6X,' WHIC
     2H IS ',F7.2,' PERCENT OF THE OPTIMUM')
8     FORMAT (6X,'BROOKS FAILED TO OBTAIN A SOLUTION TO THE PROBLEM'//6X
     1,'THE VALUES FOR X OBTAINED BY SENJU ARE'//)
9     FORMAT (10F8.1)
60    FORMAT ('       FOR THE PROBLEM'//)
61      FORMAT (/)
      READ (TTY,2) INLIF,IOFIL
      CALL IFILE(CDR,INLIF)
      CALL OFILE (LPT,IOFIL)
      READ (CDR,2) (LABEL(I),I=1,20)
      READ (CDR,1) M,N
      READ (CDR,2) (LFORM(I),I=1,20)
C
C          INPUT THE DATA
C
      DO 10 I=1,M
      READ (CDR,LFORM) (A(I,J),J=1,N),B(I)
10    CONTINUE
      READ (CDR,LFORM) (C(J),J=1,N)
      WRITE (LPT,2)(LABEL(I),I=1,20)
      WRITE (LPT,61)
      WRITE (LPT,60)
      WRITE (LPT,9) (C(J),J=1,N)
      WRITE (LPT,61)
      DO 62 I=1,M
      WRITE (LPT,9) (A(I,J),J=1,N),B(I)
62     WRITE (LPT,61)
      CALL RTIME (IT1)
C
C          SOLVE THE PROBLEM USING THE METHOD OF SENJU AND TOYODA
C
      CALL SENJU (M,N,IX)
      CALL RTIME (IT2)
C
C        SOLVE THE PROBLEM USING G L M
C
      CALL BROOKS (M,N,IY,NK)
      CALL RTIME (IT3)
      ITS=IT2-IT1
      ITB=IT3-IT2
      SOPT=0.
      BOPT=0.
      READ (CDR,1) IOPT
C
C          OUTPUT THE RESULTS
```

```
C
      WRITE (LPT,3) IOPT
      IF(NK) 16,11,11
11    WRITE (LPT,4)
      DO 15 I=1,N
      WRITE(LPT,5) I,IX(I),IY(I)
      IF(IX(I)) 13,13,12
12    SOPT=SOPT+C(I)
13    IF(IY(I))15,15,14
14    BOPT=BOPT+C(I)
15    CONTINUE
      PCTB=100.*BOPT/FLOAT(IOPT)
      PCTS=100.*SOPT/FLOAT(IOPT)
      IBOPT=BOPT
      WRITE (LPT,6) ITB,IBOPT,PCTB
      ISOPT=SOPT
      WRITE (LPT,7) ITS,ISOPT,PCTS
      STOP
16    WRITE (LPT,8)
      DO 18 I=1,N
      WRITE (LPT,5) I,IX(I)
      IF(IX(I)) 18,18,17
17    SOPT=SOPT+C(I)
18    CONTINUE
      ISOPT=SOPT
      PCTS=100.*SOPT/FLOAT(IOPT)
      WRITE (LPT,7) ITS,ISOPT,PCTS
      STOP
      END
      SUBROUTINE SENJU (M,N,IX)
C
C          THIS ROUTINE SOLVES BINARY INTEGER PROBLEMS USING THE
C          TECHNIQUE OF SENJU AND TOYODA
C
      DIMENSION IX(70),R(50),S(50),G(50),AI(50)
      COMMON/SENJU1/ A(50,50),B(50),C(50)
      ERR=0.000001
      DO 4 J=1,N
4     IX(J)=1
C
C          SET UP THE VECTORS
C
      SM=0.
      DO 8 I=1,M
      RR=0.
      DO 5 J=1,N
5     RR=RR+A(I,J)
      R(I)=RR
      SS=RR-B(I)
      IF(SS-ERR) 6,6,7
6     SS=0.
      GO TO 8
7     SM=SM+SS*SS
8     S(I)=SS
      IF(SM-ERR) 24,24,9
9     SM=1./SQRT(SM)
      DO 10 I=1,M
10    S(I)=S(I)*SM
      DO 12 J=1,N
      U=0.
```

```
      DO 11 I=1,M
11    U=U+A(I,J)*S(I)
      G(J)=C(J)/U
12    AI(J)=J
C
C        ORDER THE VECTORS
C
      CALL SORT2 (G,AI,N)
      II=1
C
C         DETERMINE WHICH VARIABLES ARE TO BE SET TO ZERO
C
17    JJ=AI(II)
      IND=1
      DO 14 I=1,M
      R(I)=R(I)-A(I,JJ)
      IF(R(I)-B(I)) 14,14,13
13    IND=2
14    CONTINUE
      IX(JJ)=0
      GO TO (15,16) IND
16    II=II+1
      GO TO 17
15    DO 18 I=1,M
18    S(I)=R(I)-B(I)
      JI=II-1
      DO 22 K=1,JI
      II=II-1
      JJ=AI(II)
      DO 20 I=1,M
      IF(S(I)+A(I,JJ)) 20,20,22
20    CONTINUE
      IX(JJ)=1
      DO 21 I=1,M
21    S(I)=S(I)+A(I,JJ)
22    CONTINUE
24    RETURN
23    FORMAT (3012)
      END
      SUBROUTINE SORT2 (A,B,N)
C
C         THIS ROUTINE ORDERS ANY VECTOR A WITH N ELEMENTS AND PUTS
C         THE ELEMENTS OF B IN THE CORRESPONDING ORDER.
C
      DIMENSION A(50),B(50)
      NN=N-1
      DO 4 I=1,NN
      AMIN=A(I)
      IND=1
      II=I+1
      DO 2 J=II,N
      IF(AMIN-A(J)) 2,2,1
1     AMIN=A(J)
      IHOLD=J
      IND=2
2     CONTINUE
      GO TO ( 4,3) IND
3     BMIN=B(IHOLD)
      A(IHOLD)=A(I)
      B(IHOLD)=B(I)
```

```
      A(I)=AMIN
      B(I)=BMIN
4     CONTINUE
      RETURN
      END
      SUBROUTINE BROOKS (M,N,IY,NK)
C
C          THIS ROUTINE USES THE METHOD PROPOSED BY BROOKS AND
C          GEOFFRION TO FIND EVERETT'S MULTIPLIERS FOR THE SOLUTION
C          OF THE BINARY-INTEGER PROBLEM
C
      COMMON/SENJU1/A(50,50),B(50),C(50)
      COMMON /GLM1/ D(50,50),M1,N2
      COMMON /GLM2/ II(50),JI(50)
      DIMENSION U(50,50),IY(70),ALAM(50)
      ERR=0.000001
      NK=0
      NN=N+1
      MM1=M+1
      M1=M+2
      N2=M+3
C
C          SET UP THE LINEAR PROGRAMMING PROBLEM
C
      DO 4 J=1,M1
      U(1,J)=C(J)
3     IL=1
      DO 4 I=1,M
      IL=IL+1
      U(IL,J)=A(I,J)
4     CONTINUE
      DO 5 I=1,M
      IL=I+1
5     U(IL,N2)=B(I)
      DO 2 J=1,N2
2     U(M1,J)=1.
6     DO 7 I=1,M1
      DO 7 J=1,N2
7     D(I,J)=U(I,J)
      DO 8 I=1,M1
      JI(I)=I+M1
8     II(I)=I
C
C          SOLVE THE LINEAR PROGRAMMING PROBLEM
C
      CALL LP
C
C          FIND THE MULTIPLIERS
C
      DO 25 I=1,M1
25    ALAM(I)=0.
      CMAX=0.
      DO 11 J=1,M1
      IJ=II(J)
       IF(IJ-M1) 9,9,10
9     IF(D(1,J)-CMAX) 11,24,24
24    JJ=IJ
      CMAX=D(1,J)
      GO TO 11
10     ALAM(IJ-M1)=D(1,J)
```

```
11    CONTINUE
C
C          SOLVE THE BINARY-INTEGER PROBLEM USING EVERETT'S TECHNIQUE
C          WITH THE MULTIPLIERS JUST DETERMINED
C
      FT=0.
      DO 16 J=1,N
      F=C(J)
      IL=1
      DO 12 I=1,M
      IL=IL+1
12    F=F-ALAM(IL)*A(I,J)
      IF(F-ERR) 26,15,14
26    IF(F+ERR) 13,15,15
13    IY(J)=0
      GO TO 16
14    IY(J)=1
      FT=FT+F
      GO TO 16
15    IY(J)=-1
16    CONTINUE
44    FORMAT (I5)
      IF(JJ-LJJ)91,63,91
63    IF(NK-M2) 64,65,65
64    NK=NK+1
      GO TO 92
C
C          IF THE SOLUTION JUST DETERMINED IS NOT OPTIMAL UPDATE
C          THE LINEAR PROGRAMMING PROBLEM AND ITERATE AGAIN.
C          OTHERWISE STOP.
C
91    NK=0
      LJJ=JJ
92    IF(FT-ALAM(M1)-ERR) 19,19,17
17    DO 251 I=1,MM1
251   U(I,JJ)=0.
      DO 18 J=1,N
      IF(IY(J)) 18,13,23
23    U(1,JJ)=U(1,JJ)+C(J)
      IL=1
      DO 20 I=1,M
      IL=IL+1
20    U(IL,JJ)=A(I,J)+U(IL,JJ)
18    CONTINUE
      GO TO 6
19    DO 41 I=1,M
      BB=B(I)
      DO 37 J=1,N
      IF(IY(J)) 37,37,36
36    BB=BB-A(I,J)
37    CONTINUE
      DO 41 J=1,N
      IF(IY(J)) 38,41,41
38    IF(BB-A(I,J)) 39,41,41
39    IY(J)=0
41    CONTINUE
      RETURN
65    NK=-1
      RETURN
      END
```

```
      SUBROUTINE GAUSS (K,L)
C
C          THIS ROUTINE PERFORMS GAUSS JORDAN ELIMINATION
C
      COMMON /GLM1/ A(50,50),M,N
      AP=1./A(K,L)
      A(K,L)=AP
      DO 3 I=1,M
      IF(I-K) 2,3,2
2     AC=-A(I,L)*AP
      A(I,L)=AC
      DO 6 J=1,N
      IF(J-L) 4,6,4
4     A(I,J)=A(I,J)+A(K,J)*AC
6     CONTINUE
3     CONTINUE
      DO 1 J=1,N
      IF(J-L) 5,1,5
5     A(K,J)=A(K,J)*AP
1     CONTINUE
      RETURN
      END
      SUBROUTINE LP
C
C          THIS POUTINE USES THE SIMPLEX METHOD TO SOLVE THE PRIMAL
C          FEASIBLE LINEAR PROGRAMMING PROBLEM
C
      COMMON /GLM1/ A(50,50),M,N
      COMMON /GLM2/ II(50),JI(50)
      DO 10 J=1,N
10    A(1,J)=-A(1,J)
1     JCOL=0
      CMAX=0,
      DO 3 J=1,N
      IF(A(1,J)-CMAX)2,3,3
2     CMAX=A(1,J)
      JCOL=J
3      CONTINUE
      IF(JCOL) 9,9,4
4     IROW=0
      RMIN=10000000.
      DO 7 I=2,M
      IF(A(I,JCOL)-0.000001) 7,7,5
5     R=A(I,N)/A(I,JCOL)
      IF(R-RMIN) 6,7,7
6     IROW=I
      RMIN=R
7     CONTINUE
      IF(IROW) 9,9,8
8     CALL GAUSS (IROW,JCOL)
      IS=II(JCOL)
      II(JCOL)=JI(IROW)
      JI(IROW)=IS
      GO TO 1
9     RETURN
      END
```

PETERSON PROBLEM #1

FOR THE PROBLEM

100,0	600,0	1200,0	2400,0	500,0	2000,0	
8,0	12,0	13,0	64,0	22,0	41,0	80,0
8,0	12,0	13,0	75,0	22,0	41,0	96,0
3,0	6,0	4,0	18,0	6,0	4,0	20,0
5,0	10,0	8,0	32,0	6,0	12,0	36,0
5,0	13,0	8,0	42,0	6,0	20,0	44,0
5,0	13,0	8,0	48,0	6,0	20,0	48,0
0,0	0,0	0,0	0,0	8,0	0,0	10,0
3,0	0,0	4,0	0,0	8,0	0,0	18,0
3,0	2,0	4,0	0,0	8,0	4,0	22,0
3,0	2,0	4,0	8,0	8,0	4,0	24,0

THE OPTIMUM VALUE IS 3800

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	1	0
X(3)=	1	1
X(4)=	0	0
X(5)=	0	-1
X(6)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 600 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 3200
WHICH IS 84,21 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 3800
WHICH IS 100,00 PERCENT OF THE OPTIMUM

PETERSON PROBLEM #2

FOR THE PROBLEM

600,1	310,5	1800,0	3850,0	18,6	198,7	882,0	4200,0	402,5	327,0
20,0	5,0	100,0	200,0	2,0	4,0	60,0	150,0	80,0	40,0
450,0									
20,0	7,0	130,0	280,0	2,0	8,0	110,0	210,0	100,0	40,0
540,0									
60,0	3,0	50,0	100,0	4,0	2,0	20,0	40,0	6,0	12,0
200,0									
60,0	8,0	70,0	200,0	4,0	6,0	40,0	70,0	16,0	20,0
360,0									
60,0	13,0	70,0	150,0	4,0	10,0	60,0	90,0	20,0	24,0
440,0									
60,0	13,0	70,0	280,0	4,0	10,0	70,0	105,0	22,0	28,0
480,0									
5,0	2,0	20,0	100,0	2,0	5,0	10,0	60,0	0,0	0,0
200,0									
45,0	14,0	80,0	180,0	6,0	10,0	40,0	160,0	20,0	0,0
360,0									
55,0	14,0	80,0	200,0	6,0	10,0	50,0	140,0	30,0	40,0
440,0									
65,0	14,0	80,0	220,0	6,0	10,0	50,0	180,0	30,0	50,0
480,0									

THE OPTIMUM VALUE IS.. 8706

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	=1
X(2)=	1	1
X(3)=	1	1
X(4)=	0	0

```
X(  5)=   1          0
X(  6)=   1          1
X(  7)=   1          0
X(  8)=   1          1
X(  9)=   0          0
X( 10)=   1          1

THE TIME REQUIRED TO RUN BROOKS WAS    550 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS        6836
 WHICH IS   78.52 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS     33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS        8336
 WHICH IS   95.76 PERCENT OF THE OPTIMUM
```

PETERSON PROBLEM #3

FOR THE PROBLEM

100.0	220.0	90.0	400.0	300.0	400.0	205.0	120.0	160.0	580.0
400.0	140.0	100.0	1300.0	650.0					
8.0	4.0	13.0	80.0	70.0	80.0	45.0	15.0	28.0	90.0
130.0	32.0	20.0	120.0	40.0	550.0				
8.0	44.0	13.0	100.0	100.0	90.0	75.0	25.0	28.0	120.0
130.0	32.0	40.0	160.0	40.0	700.0				
3.0	6.0	4.0	20.0	20.0	30.0	8.0	3.0	12.0	14.0
40.0	6.0	3.0	20.0	5.0	130.0				
5.0	9.0	6.0	40.0	30.0	40.0	16.0	5.0	18.0	24.0
60.0	16.0	11.0	30.0	25.0	240.0				
5.0	11.0	7.0	50.0	40.0	40.0	19.0	7.0	18.0	29.0
70.0	21.0	17.0	30.0	25.0	280.0				
5.0	11.0	7.0	55.0	40.0	40.0	21.0	9.0	18.0	29.0
70.0	21.0	17.0	35.0	25.0	310.0				
0.0	0.0	1.0	10.0	4.0	10.0	0.0	6.0	0.0	6.0
32.0	3.0	0.0	70.0	10.0	110.0				
3.0	4.0	5.0	20.0	14.0	20.0	6.0	12.0	10.0	18.0
42.0	9.0	12.0	100.0	20.0	205.0				
3.0	6.0	9.0	30.0	29.0	20.0	12.0	12.0	10.0	30.0
42.0	18.0	15.0	110.0	20.0	260.0				
3.0	8.0	9.0	35.0	29.0	20.0	16.0	15.0	10.0	30.0
42.0	20.0	18.0	120.0	20.0	275.0				

THE OPTIMUM VALUE IS 4015

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	1
X(2)=	1	1
X(3)=	1	1

X(4)=	0	1
X(5)=	0	1
X(6)=	1	1
X(7)=	0	1
X(8)=	1	0
X(9)=	1	1
X(10)=	1	1
X(11)=	0	0
X(12)=	1	0
X(13)=	0	0
X(14)=	1	0
X(15)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 1166 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 3105
WHICH IS 77.33 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 50 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 3760
WHICH IS 93.65 PERCENT OF THE OPTIMUM

PETERSON PROBLEM #4

FOR THE PROBLEM

100.0	220.0	90.0	410.0	300.0	400.0	205.0	120.0	160.0	580.0
400.0	140.0	100.0	1300.0	650.0	320.0	480.0	80.0	60.0	2550.0
8.0	4.0	13.0	60.0	70.0	80.0	45.0	15.0	28.0	90.0
130.0	32.0	20.0	120.0	40.0	30.0	20.0	6.0	3.0	180.0
550.0									
8.0	44.0	13.0	100.0	100.0	90.0	75.0	25.0	28.0	120.0
130.0	32.0	40.0	160.0	40.0	60.0	55.0	10.0	6.0	240.0
700.0									
3.0	6.0	4.0	20.0	20.0	30.0	8.0	3.0	12.0	14.0
40.0	6.0	3.0	20.0	5.0	0.0	5.0	3.0	0.0	20.0
130.0									
5.0	9.0	6.0	40.0	30.0	40.0	16.0	5.0	18.0	24.0
60.0	16.0	11.0	30.0	25.0	10.0	13.0	5.0	1.0	80.0
240.0									
5.0	11.0	7.0	50.0	40.0	40.0	19.0	7.0	18.0	29.0
70.0	21.0	17.0	30.0	25.0	15.0	25.0	5.0	1.0	100.0
280.0									
5.0	11.0	7.0	55.0	40.0	40.0	21.0	9.0	18.0	29.0
70.0	21.0	17.0	35.0	25.0	20.0	25.0	5.0	2.0	110.0
310.0									
0.0	0.0	1.0	10.0	4.0	10.0	0.0	6.0	0.0	6.0
32.0	3.0	0.0	70.0	10.0	0.0	0.0	0.0	0.0	0.0
110.0									
3.0	4.0	5.0	20.0	14.0	20.0	6.0	12.0	10.0	18.0
42.0	9.0	12.0	100.0	20.0	5.0	6.0	4.0	1.0	20.0
225.0									
3.0	6.0	9.0	30.0	29.0	20.0	12.0	12.0	10.0	30.0
42.0	18.0	18.0	110.0	20.0	15.0	18.0	7.0	2.0	40.0
260.0									
3.0	8.0	9.0	35.0	29.0	20.0	16.0	15.0	10.0	30.0
42.0	20.0	18.0	120.0	20.0	28.0	22.0	7.0	3.0	50.0
275.0									

THE OPTIMUM VALUE IS 6120

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	1
X(2)=	1	1
X(3)=	1	0
X(4)=	0	0
X(5)=	0	0
X(6)=	0	0
X(7)=	0	0
X(8)=	0	0
X(9)=	1	1
X(10)=	0	1
X(11)=	0	0
X(12)=	0	0
X(13)=	0	0
X(14)=	1	1
X(15)=	1	1
X(16)=	1	1
X(17)=	1	1
X(18)=	1	1
X(19)=	1	1
X(20)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 1683 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 6500
WHICH IS 106.21 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 6010
WHICH IS 98.20 PERCENT OF THE OPTIMUM

PETERSON PROBLEM #5

10x28

FOR THE PROBLEM

100.0	220.0	90.0	400.0	300.0	400.0	205.0	120.0	160.0	580.0
420.0	140.0	100.0	1300.0	650.0	320.0	480.0	80.0	60.0	2550.0
3180.0	1190.0	950.0	450.0	300.0	220.0	200.0	520.0		
8.0	4.0	13.0	80.0	70.0	80.0	45.0	15.0	28.0	90.0
130.0	32.0	20.0	120.0	40.0	30.0	20.0	6.0	3.0	180.0
220.0	50.0	30.0	50.0	12.0	5.0	8.0	18.0	930.0	
8.0	44.0	13.0	100.0	100.0	90.0	75.0	25.0	28.0	120.0
130.0	32.0	40.0	160.0	42.0	60.0	55.0	10.0	6.0	240.0
290.0	50.0	90.0	70.0	27.0	17.0	8.0	28.0	1210.0	
3.0	6.0	4.0	20.0	20.0	30.0	8.0	3.0	12.0	14.0
42.0	6.0	3.0	20.0	5.0	0.0	5.0	3.0	0.0	20.0
30.0	40.0	10.0	0.0	5.0	0.0	0.0	10.0	272.0	
5.0	9.0	6.0	40.0	30.0	40.0	16.0	5.0	18.0	24.0
60.0	16.0	11.0	30.0	25.0	10.0	13.0	5.0	1.0	80.0
60.0	50.0	20.0	30.0	12.0	5.0	3.0	20.0	426.0	
5.0	11.0	7.0	50.0	40.0	40.0	19.0	7.0	18.0	29.0
70.0	21.0	17.0	30.0	25.0	15.0	25.0	5.0	1.0	100.0
70.0	55.0	22.0	50.0	15.0	15.0	6.0	20.0	532.0	
5.0	11.0	7.0	55.0	40.0	40.0	21.0	9.0	18.0	29.0
70.0	21.0	17.0	35.0	25.0	20.0	25.0	5.0	2.0	110.0
70.0	55.0	20.0	50.0	20.0	15.0	6.0	20.0	572.0	
0.0	0.0	1.0	10.0	4.0	10.0	0.0	6.0	0.0	6.0
32.0	3.0	0.0	70.0	10.0	0.0	0.0	0.0	0.0	0.0
30.0	10.0	0.0	10.0	10.0	5.0	0.0	10.0	240.0	
3.0	4.0	5.0	20.0	14.0	20.0	6.0	12.0	10.0	18.0
42.0	9.0	12.0	100.0	20.0	5.0	6.0	4.0	1.0	20.0
50.0	30.0	5.0	20.0	20.0	10.0	10.0	20.0	400.0	
3.0	6.0	9.0	30.0	29.0	20.0	12.0	12.0	10.0	30.0
42.0	18.0	18.0	110.0	20.0	15.0	18.0	7.0	2.0	40.0
60.0	50.0	25.0	25.0	25.0	15.0	10.0	28.0	470.0	
3.0	8.0	9.0	35.0	29.0	20.0	16.0	15.0	10.0	30.0
42.0	20.0	18.0	120.0	20.0	20.0	22.0	7.0	3.0	50.0
60.0	55.0	25.0	30.0	25.0	15.0	10.0	28.0	490.0	

THE OPTIMUM VALUE IS 12400
BROOKS FAILED TO OBTAIN A SOLUTION TO THE PROBLEM

THE VALUES FOR X OBTAINED BY SENJU ARE

X(1)=	1
X(2)=	1
X(3)=	1
X(4)=	0
X(5)=	0
X(6)=	0
X(7)=	0
X(8)=	0
X(9)=	1
X(10)=	0
X(11)=	0
X(12)=	0
X(13)=	0
X(14)=	1
X(15)=	1
X(16)=	1
X(17)=	1
X(18)=	1
X(19)=	1
X(20)=	1
X(21)=	1
X(22)=	1
X(23)=	1
X(24)=	0
X(25)=	1
X(26)=	1
X(27)=	1
X(28)=	1

THE TIME REQUIRED TO RUN SENJU WAS 67 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 12400
WHICH IS 100.00 PERCENT OF THE OPTIMUM

PETERSON PROBLEM #6

(206/206)

FOR THE PROBLEM

560.0	1125.0	300.0	620.0	2100.0	431.0	68.0	328.0	47.0	122.0
322.0	196.0	41.0	25.0	425.0	4260.0	416.0	115.0	82.0	22.0
631.0	132.0	420.0	86.0	42.0	103.0	215.0	81.0	91.0	26.0
49.0	420.0	316.0	72.0	71.0	49.0	108.0	116.0	90.0	
40.0	91.0	10.0	30.0	160.0	20.0	3.0	12.0	3.0	18.0
9.0	25.0	1.0	1.0	10.0	280.0	10.0	8.0	1.0	1.0
49.0	8.0	21.0	6.0	1.0	5.0	10.0	8.0	2.0	1.0
0.0	10.0	42.0	6.0	4.0	8.0	0.0	10.0	1.0	600.0
16.0	92.0	41.0	16.0	150.0	23.0	4.0	18.0	6.0	0.0
12.0	8.0	2.0	1.0	0.0	200.0	20.0	6.0	2.0	1.0
70.0	9.0	22.0	4.0	1.0	5.0	10.0	6.0	4.0	0.0
4.0	12.0	8.0	4.0	3.0	0.0	10.0	0.0	6.0	500.0
38.0	39.0	32.0	71.0	80.0	26.0	5.0	40.0	8.0	12.0
30.0	15.0	0.0	1.0	23.0	100.0	0.0	20.0	3.0	0.0
40.0	6.0	8.0	0.0	6.0	4.0	22.0	4.0	6.0	1.0
5.0	14.0	8.0	2.0	8.0	0.0	20.0	0.0	0.0	500.0
8.0	71.0	30.0	60.0	200.0	18.0	6.0	30.0	4.0	8.0
31.0	6.0	3.0	0.0	18.0	60.0	21.0	4.0	0.0	2.0
32.0	15.0	31.0	2.0	2.0	7.0	8.0	2.0	8.0	0.0
2.0	8.0	6.0	7.0	1.0	0.0	0.0	20.0	8.0	500.0
38.0	52.0	30.0	42.0	170.0	9.0	7.0	20.0	0.0	3.0
21.0	4.0	1.0	2.0	14.0	310.0	8.0	4.0	6.0	1.0
18.0	15.0	38.0	10.0	4.0	8.0	6.0	0.0	0.0	3.0
0.0	10.0	6.0	1.0	3.0	0.0	3.0	5.0	4.0	600.0

THE OPTIMUM VALUE IS 10618

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	1
X(2)=	0	1
X(3)=	0	0
X(4)=	1	1
X(5)=	0	0
X(6)=	1	1
X(7)=	1	0
X(8)=	0	1
X(9)=	0	1
X(10)=	1	0
X(11)=	1	1

X(12)=	1	0
X(13)=	1	1
X(14)=	1	1
X(15)=	1	1
X(16)=	1	1
X(17)=	1	1
X(18)=	1	1
X(19)=	1	1
X(20)=	1	1
X(21)=	0	1
X(22)=	1	0
X(23)=	1	1
X(24)=	1	0
X(25)=	1	1
X(26)=	1	1
X(27)=	1	1
X(28)=	1	1
X(29)=	1	1
X(30)=	1	1
X(31)=	1	1
X(32)=	1	1
X(33)=	1	0
X(34)=	1	1
X(35)=	1	1
X(36)=	1	1
X(37)=	1	1
X(38)=	1	1
X(39)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 1383 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 11403
WHICH IS 107,39 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 50 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 10192
WHICH IS 95,99 PERCENT OF THE OPTIMUM

PETERSON PROBLEM #7

(256/266)

FOR THE PROBLEM

560.0	1125.0	300.0	620.0	2100.0	431.0	68.0	328.0	47.0	122.0
322.0	196.0	41.0	25.0	425.0	4260.0	416.0	115.0	82.0	22.0
631.0	132.0	420.0	86.0	42.0	103.0	215.0	61.0	91.0	26.0
49.0	420.0	316.0	72.0	71.0	49.0	108.0	116.0	90.0	738.0
1811.0	430.0	3060.0	215.0	58.0	296.0	620.0	418.0	47.0	61.0
40.0	91.0	10.0	30.0	160.0	20.0	3.0	12.0	3.0	18.0
9.0	25.0	1.0	1.0	10.0	260.0	12.0	8.0	1.0	1.0
49.0	8.0	21.0	6.0	1.0	5.0	10.0	8.0	2.0	1.0
0.0	10.0	42.0	6.0	4.0	8.0	0.0	10.0	1.0	40.0
86.0	11.0	120.0	8.0	3.0	32.0	28.0	13.0	2.0	4.0
800.0									
16.0	92.0	41.0	16.0	150.0	23.0	4.0	18.0	6.0	0.0
12.0	8.0	2.0	1.0	0.0	200.0	20.0	6.0	2.0	1.0
70.0	9.0	22.0	4.0	1.0	5.0	10.0	6.0	4.0	0.0
4.0	12.0	8.0	4.0	3.0	0.0	10.0	0.0	6.0	28.0
93.0	9.0	30.0	22.0	0.0	36.0	45.0	13.0	2.0	2.0
650.0									
38.0	39.0	32.0	71.0	80.0	26.0	5.0	40.0	8.0	12.0
30.0	15.0	0.0	1.0	23.0	100.0	0.0	20.0	3.0	0.0
40.0	6.0	8.0	0.0	6.0	4.0	22.0	4.0	6.0	1.0
5.0	14.0	8.0	2.0	8.0	0.0	20.0	0.0	0.0	6.0
12.0	6.0	80.0	13.0	6.0	22.0	14.0	0.0	1.0	2.0
550.0									
8.0	71.0	30.0	60.0	200.0	18.0	6.0	30.0	4.0	8.0
31.0	6.0	3.0	0.0	18.0	60.0	21.0	4.0	0.0	2.0
32.0	15.0	31.0	2.0	2.0	7.0	8.0	2.0	8.0	0.0
2.0	8.0	6.0	7.0	1.0	0.0	0.0	20.0	8.0	14.0
20.0	2.0	40.0	6.0	1.0	14.0	20.0	12.0	0.0	1.0
550.0									
38.0	52.0	30.0	42.0	170.0	9.0	7.0	20.0	0.0	3.0
21.0	4.0	1.0	2.0	14.0	310.0	8.0	4.0	6.0	1.0
18.0	15.0	38.0	10.0	4.0	8.0	6.0	0.0	0.0	3.0
0.0	10.0	6.0	1.0	3.0	0.0	3.0	5.0	4.0	0.0
30.0	12.0	16.0	18.0	3.0	16.0	22.0	30.0	4.0	0.0
650.0									

THE OPTIMUM VALUE IS 16537

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

SENJU BROOKS

X(1)=	1	0
X(2)=	0	1
X(3)=	0	0
X(4)=	0	1
X(5)=	0	0
X(6)=	1	1
X(7)=	1	0
X(8)=	0	1
X(9)=	0	1
X(10)=	0	0
X(11)=	0	1
X(12)=	0	0
X(13)=	1	1
X(14)=	1	1
X(15)=	1	1
X(16)=	1	1
X(17)=	1	1
X(18)=	0	0
X(19)=	1	1
X(20)=	1	1
X(21)=	0	1
X(22)=	0	0
X(23)=	1	1
X(24)=	1	0
X(25)=	1	0
X(26)=	1	1
X(27)=	1	1
X(28)=	1	1
X(29)=	1	1
X(30)=	1	1
X(31)=	1	1
X(32)=	1	1
X(33)=	1	0
X(34)=	1	1
X(35)=	1	1
X(36)=	1	1
X(37)=	1	1
X(38)=	0	1
X(39)=	1	1
X(40)=	1	1
X(41)=	1	1
X(42)=	1	1
X(43)=	1	1
X(44)=	0	0
X(45)=	1	1
X(46)=	0	0
X(47)=	1	1
X(48)=	1	1
X(49)=	1	1
X(50)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 2567 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 17949
WHICH IS 108.54 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 117 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 15832

WHICH IS 95.74 PERCENT OF THE OPTIMUM

TRAUTH AND WOOLSEY PROBLEM #1

FOR THE PROBLEM

20.0 18.0 17.0 15.0 15.0 10.0 5.0 3.0 1.0 1.0

30.0 25.0 20.0 18.0 17.0 11.0 5.0 2.0 1.0 1.0
60.0

THE OPTIMUM VALUE IS 52

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	0	0
X(5)=	1	1
X(6)=	1	1
X(7)=	1	1
X(8)=	1	1
X(9)=	1	1
X(10)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 100.00 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 17 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 100.00 PERCENT OF THE OPTIMUM

TRAUTH AND WOOLSEY PROBLEM #2

FOR THE PROBLEM

20.0	18.0	17.0	15.0	15.0	10.0	5.0	3.0	1.0	1.0
30.0	25.0	20.0	18.0	17.0	11.0	5.0	2.0	1.0	1.0

67.0

THE OPTIMUM VALUE IS 58.

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	0	0
X(5)=	1	1
X(6)=	1	1
X(7)=	1	1
X(8)=	1	1
X(9)=	1	1
X(10)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 89.66 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 17 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 89.66 PERCENT OF THE OPTIMUM

TRAUTH AND WOOLSEY PROBLEM #3

FOR THE PROBLEM

20.0 18.0 17.0 15.0 15.0 10.0 5.0 3.0 1.0 1.0

30.0 25.0 20.0 18.0 17.0 11.0 5.0 2.0 1.0 1.0
70.0

THE OPTIMUM VALUE IS 62

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	0	0
X(5)=	1	1
X(6)=	1	1
X(7)=	1	1
X(8)=	1	1
X(9)=	1	1
X(10)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 83.87 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 83.87 PERCENT OF THE OPTIMUM

TRAUTH AND WOOLSEY PROBLEM #4

FOR THE PROBLEM

20.0 18.0 17.0 15.0 15.0 10.0 5.0 3.0 1.0 1.0

30.0 25.0 20.0 18.0 17.0 11.0 5.0 2.0 1.0 1.0
75.0

THE OPTIMUM VALUE IS 67

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	1	-1
X(5)=	1	1
X(6)=	1	1
X(7)=	1	1
X(8)=	1	1
X(9)=	1	1
X(10)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 52
WHICH IS 77.61 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 17 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 67
WHICH IS 100.00 PERCENT OF THE OPTIMUM

TRAUTH AND WOOLSEY PROBLEM #5

FOR THE PROBLEM

20.0 18.0 17.0 15.0 15.0 10.0 5.0 3.0 1.0 1.0

30.0 25.0 20.0 18.0 17.0 11.0 5.0 2.0 1.0 1.0
80.0

THE OPTIMUM VALUE IS 68

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	1	1
X(5)=	1	1
X(6)=	1	1
X(7)=	1	1
X(8)=	1	1
X(9)=	1	1
X(10)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 17 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 67
WHICH IS 96.53 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 67
WHICH IS 96.53 PERCENT OF THE OPTIMUM

TEST PROBLEM #1

FOR THE PROBLEM

1.0 2.0 3.0 4.0

4.0 3.0 2.0 1.0 4.0

1.0 4.0 2.0 3.0 6.0

THE OPTIMUM VALUE IS 7

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	0	0
X(2)=	0	0
X(3)=	1	1
X(4)=	1	1

THE TIME REQUIRED TO RUN BROOKS WAS 33 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 7
WHICH IS 100.00 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 7
WHICH IS 100.00 PERCENT OF THE OPTIMUM

TEST PROBLEM #2

FOR THE PROBLEM

3.0 2.0 4.0

1.0 1.0 1.0 2.0

2.0 1.0 4.0 5.0

THE OPTIMUM VALUE IS 6

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	1
X(2)=	1	-1
X(3)=	0	0

THE TIME REQUIRED TO RUN BROOKS WAS 17 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 3
WHICH IS 52.00 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 5
WHICH IS 83.33 PERCENT OF THE OPTIMUM

TEST PROBLEM #3

FOR THE PROBLEM

2.0	3.0	4.0	6.0	
1.0	4.0	3.0	1.0	6.0
2.0	6.0	7.0	12.0	15.0
7.0	4.0	3.0	2.0	11.0

THE OPTIMUM VALUE IS 8

THE VALUES FOR X FOR THE DIFFERENT TECHNIQUES ARE AS FOLLOWS

	SENJU	BROOKS
X(1)=	1	1
X(2)=	0	0
X(3)=	0	1
X(4)=	1	0

THE TIME REQUIRED TO RUN BROOKS WAS 66 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 6
WHICH IS 75.00 PERCENT OF THE OPTIMUM

THE TIME REQUIRED TO RUN SENJU WAS 0 MILLISECONDS

THE VALUE OF THE OBJECTIVE FUNCTION WAS 8
WHICH IS 100.00 PERCENT OF THE OPTIMUM

LITERATURE CITED

1. Brooks, R., and Geoffrion, A., Finding Everett's Lagrange Multipliers by Linear Programming: Operations Research, V. 14, No. 6, pp. 1149-1153 (1966)

2. Balas, E., An Additive Algorithm for Solving Linear Programs With Zero-One Variables: Operations Research, V. 13, No. 3, pp. 517-546 (1965)

3. Geoffrion, A.M., An Improved Implicit Enumeration Approach for Integer Programming: Operations Research, V. 17, No. 3, pp. 437-454 (1969)

4. Plane, D.R. and McMillan, C., Jr., Discrete Optimization, Prentice Hall, Englewood Cliffs, New Jersey (1971)

5. Davis, R.E., Kendrick, D.A., and Weitzman, M., A Branch-and-Bound Algorithm for Zero-One Mixed Integer Programming Problems: Operations Research, V. 19, No. 4, pp. 1036-1044 (1971)

6. Lawler, E.L., and Wood, D.E., Branch-and-Bound Methods: A Survey: Operations Research, V. 14, No. 4, pp. 699-719 (1966)

7. Tuan, N.P., A Flexible Tree-Search Method for Integer Programming Problems: Operations Research, V. 19, No. 1, pp. 115-119 (1971)

8. Senju, S., and Toyoda, Y., An Approach to Linear Programming With 0-1 Variables: Management Science, V. 15, No. 4, p. B-196 - B-107 (1968)

9. Everett, H., III, General Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources: Operations Research, V. 11, No. 3, pp. 399-417 (1963)

10. Gulley, D.A., On Not Searching for the Multiplier in Knapsack Problems: Masters Thesis, Colorado School of Mines, Golden, Colorado (1972)

11. Terry, J.R., An Application of Generalized Lagrange Multipliers to Binary-Integer Programming: Masters Thesis, Colorado School of Mines, Golden, Colorado (1972)

12. Gass, S.I., Linear Programming: McGraw Hill, New York, (1969)

13. Peterson, C.C., Computational Experience With Variants of the Balas Algorithm Applied to the Selection of R&D Projects: Management Science, V. 13, pp. 736-750 (1967)

14. Trauth, C.A., Jr., and Woolsey, R.E., Integer Linear Programming, A Study in Computational Efficiency: Management Science, V. 15, p. 481 (1968)