SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

152011

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

CONTRIBUTED PROGRAM LIBRARY SUBM
(for IBM S/360, 1130 and 1800)

SHARE Program Library Agency
Triangle Universities Computation Center
P. O. Box 12076
Research Triangle Park, N. C. 27709

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

(1) **Program Order Number** *(to be filled in by PID)* 360D-15.2.011

(2) **System Type** *(machine)* S/360

(3) **Search Key** . . . . . ZERO-ONE, INTEGER LINEAR /PROGRAMMING /WITH/ HEURISTICS

(4) **Programming Language** . . . . . B.D. Holcomb

(5) **Author's Name and Address** . . . . Oak Ridge National Laboratory
Bldg. 4500-N, H-28
P.O. Box X
Oak Ridge, TN 37830

(6) **Direct Inquiries to Name and Address**
(if different than Author)

(7) **Title of Program** . . . . ZERO-ONE INTEGER PROGRAMMING WITH HEURISTICS

(8) **Submitter's User Group Affiliation Code and Installation Code** . . . . . S OR

(9) **Submitter's Own Program Identification and Suffix** *(optional)* . . . . .

(10) **Primary Subject Code** 15.2

(11) **Secondary Subject Codes** . . . . .

(12) **Operating or Monitor System Required** . . . . . SEE ABSTRACT

(13) **New or Revision Code** *(if revision, show prior Program Order Number in item 1)* . . . . . N

(14) **Year Completed** 68

(15) **Date of Submittal** 12 06 68

(16) **Documentation** *(number of original pages submitted)* 12

(17) **Abstract** *(should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.*

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

a. Purpose
b. Programming Language used
c. Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
d. Field of application
e. Type of routine (main program, subroutine, etc.)
f. Specific description of machine requirements
g. Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT

The ZERO-ONE INTEGER PROGRAMMING WITH HEURISTICS program is designed to solve linear programming problems whose variables are restricted to values of zero or one. The program utilizes the well known additive algorithm of Egon Balas combined with a group of user selected heuristic test options designed to speed solution time by taking advantage of individual problem characteristics.

The program deck consists of a main program and four subroutines written in FORTRAN plus a three card object deck of a clock reading function. The program has been tested on the IBM 360 Model 50 using OS360 Release 16. However, the use of any IBM 360 Model 40 or larger with OS360 should not cause difficulties.

(Please attach additional pages if necessary) . . . . . **Total pages attached** _____

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

(18) **Signature of Submitter and Date** Brad D. Holcomb 12-06-68

(19) **Signature of Installation Addressee** C. P. Allen

T4SF

CONTENTS

Purpose

To solve a linear programming problem with zero-one variables of the form

$$\text{Minimize} \quad \sum_{j=1}^{n} c_j x_j$$

Subject to m constraints

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1,2,3,\ldots,m$$

$$\text{and } x_j = 0 \text{ or } 1 \text{ for all } j.$$

The program required that the constraints be inequalities of the form $\sum ax \leq b$.

Any problem not in this form must be brought to this form by:

(a) Replacing all equations by two inequalities.

(b) Multiplying by -1 all inequalities of the form $\sum ax \geq b$.

Algorithm

The algorithm is a variant of the 0-1 programming algorithm published by Egon Balas [1]. Being an implicit enumeration scheme, the efficiency of the algorithm depends solely on the effectiveness of the tests used to eliminate from consideration large portions of the 2^n possible choices. Several heuristic tests suggested in the literature have been incorporated in the code. These tests which augment or replace existing tests in the algorithm allow the formulator to employ ad hoc information in selecting which of several variants he wishes to be used in solving the problem. The algorithm is additive as well as all-integer. This avoids the problem of round-off errors while taking advantage of the added speed of fixed point operations. A more detailed discussion of the algorithm may be found in reference [2].

Limitations

This code is written in FORTRAN IV for the IBM System/360 with an internal clock*. The code presently provides for running problems with a maximum dimension of 100 x 200. If larger problems are to be solved, the dimensions of the variables in the labeled common sections must be changed. The variable NRA, appearing in the main program and representing the number of rows of the A matrix, must also reflect this change of dimension.

All essential variables are fixed point and therefore, limited to a maximum magnitude of 2,147,483,647. Cost or constraint coefficients not in integer form can usually be brought to an acceptable integer form with appropriate scaling.

*If such is not available, remove references to ICLOCK.

Program Input

All data for this code is read in subroutine INPUT. Following is the description of the data deck needed to solve a given problem:

Identification Card

Name	Format	Columns	Description
ID(I)	18A4,1A2	7-80	Alphameric Identification (Columns 1-6 ignored)

Dimension Card

Name	Format	Columns	Description
M	I5	1-5	No. of rows in the A matrix
N	I5	6-10	No. of columns in the A matrix
ZS	I10	11-20	A nonzero value for ZS signals the program that a starting solution will be furnished with the data
LTIME	I5	21-25	Running time limit in minutes

Option Card - (The desired options are given a nonzero value)

Choice among the following options should be considered in 3 groupings:

Providing a starting solution: Either NS1 or NR2 or neither

Reducing the set of candidate variables: NF1 and (NR1 or NF2), any one singly, or none

Simplifying selection of the candidate variable: NH1, or none.

If no options are specified, the standard Balas algorithm is utilized.

Option	Format	Columns	Description
NR1	I5	1-5	A ranking of the c_j/a_{ij} for all i,j is used to determine program strategy. Requires variations in the magnitude of the ratio c_j/a_{ij} to be effective.
NR2	I5	6-10	Provides the program with a starting solution which satisfies the "worst" constraint.
NF1	I5	11-15	Program strategy, effective only when some constraint coefficients a_{ij} are such that $c_j * a_{ij} > 0$.
NF2	I5	16-20	Provides for more than one variable to enter the basis in an iteration. Requires variations in the magnitude of the ratio c_j/a_{ij} to be effective.
NS1	I5	21-25	Provides the program with a starting solution satisfying a surrogate constraint which is the sum of the m constraints.
NH1	I5	26-30	Avoids the more time-consuming test for incoming variables and uses the first eligible variable encountered.

Format Card

Name	Format	Columns	Description
FORMAT	1A4,1A2	1-6	The word "FORMAT"
(FMT(I))	18A4,1A2	7-80	The format specification for the data that follows (enclosed in parentheses), e.g. FORMAT (8I10).

Cost Coefficients

Name	Format	Columns	Description
C(J), J=1,N	FMT(I)	As specified by FMT(I)	Integer coefficients of the objective function.

Constraint Cards

Name	Format	Columns	Description
A(I,J), J=1,N	FMT(I)	As specified by FMT(I)	Integer coefficients of the constraint matrix A. These are read one row at a time with each row beginning on a new card.

B Vector

Name	Format	Columns	Description
B(I), I=1,M	FMT(I)	As specified by FMT(I)	Integer coefficients of the right hand side vector.

The above cards form the data deck for a standard problem. Should the user wish to supply a starting solution (any n-dimensional vector consisting of 0's and 1's) to the code the following data is needed and is read only when the variables ZS on the dimension card is nonzero.

Solution

Name	Format	Columns	Description
JS(J), J=1,N	16I5	1-80	The values of the variables for for the starting solution.*

*Assuming X(J) is the desired starting solution, JS(J) is defined as:

$$JS(J) = \begin{cases} X(J) & , \text{ if } C(J) \geq 0. \\ 1-X(J) & , \text{ if } C(J) < 0. \end{cases}$$

Bookkeeping Array

Name	Format	Columns	Description
INB(J), J=1,N	16I5	1-80	Indicates the order in which the individual elements of the solution entered the basis.**

When a problem is terminated after exceeding the time limit, the JS and INB arrays are printed. The condition of the system prior to termination may be restored by supplying this information as a starting solution.

Multiple problems may be run by stacking data decks one after another.

**A suggested set of values is the following:

$$INB(J) = \begin{cases} 0 & , \text{ if } JS(J) = 0 \\ \sum_{K=1}^{J} JS(K) & , \text{ if } JS(J) = 1 \end{cases}$$

Sample Problem (from reference [1])

Maximize

$$10x_1 - 7x_2 + x_3 - 12x_4 + 2x_5 + 8x_6 - 3x_7 - x_8 + 5x_9 + 3x_{10}$$

Sugject to:

$$3x_1 + 12x_2 - 8x_3 - x_4 - 7x_9 + 2x_{10} \geq -8,$$

$$x_2 + 10x_3 + 5x_5 - x_6 + 7x_7 + x_8 \leq 13,$$

$$-5x_1 - 3x_2 + x_3 - 2x_8 - x_{10} = -6,$$

$$- 4x_3 + 2x_4 - 5x_6 - x_7 + 9x_8 - 2x_9 \geq -8,$$

$$- 9x_2 + 12x_4 - 7x_5 + 6x_6 - 2x_8 - 15x_9 - 3x_{10} \geq -12,$$

$$8x_1 + 5x_2 - 2x_3 - 7x_4 + x_5 - 5x_7 + 10x_9 \leq 16,$$

$$x_j = 0 \text{ or } 1. \qquad (j = 1,\ldots,10)$$

References

1. Balas, Egon, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Operations Research, Volume 13, No. 4, (July-August, 1965), pp. 517-546.

2. Holcomb, B. D., "An Implementation of a Zero-One Programming Algorithm," Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, Report No. CTC-3, September 6, 1968.