International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

HYBRID - A Mathematical Programming Package

Makowski, M. and Sosnowski, J.S.

IIASA Collaborative Paper
April 1984

HYBRID — A MATHEMATICAL PROGRAMMING PACKAGE

Marek Makowski
Janusz S. Sosnowski
Systems Research Institute, Warsaw, Poland

April 1984
CP-84-9

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

PREFACE

This paper describes a new mathematical programming package developed by the authors and implemented both on the VAX at IIASA and at the Computing Center of the State Planning Commission in Warsaw.

The new package, called HYBRID, is designed for the solution of linear programming problems, making use of a particular implementation of the Lagrange multiplier method. The version described here is limited to static LP problems (including multiobjective problems that may be reformulated as LP problems), but it will eventually be extended to deal with dynamic problems.

HYBRID is intended for use with real-world problems that require scenario analysis, and is therefore oriented towards an interactive mode of operation in which a sequence of problems is to be solved under different conditions (e.g., with different objective functions, reference points, constraints or bounds).

This report provides all the information necessary to use the HYBRID package at IIASA and also discusses the methodological issues associated with the chosen solution technique.

ANDRZEJ WIERZBICKI
Chairman
System and Decision Sciences

CONTENTS

HYBRID: A MATHEMATICAL PROGRAMMING PACKAGE

Marek Makowski and Janusz S. Sosnowski

Systems Research Institute of the Polish Academy of Sciences,
Warsaw, Poland

1. INTRODUCTION

1.1 Scope of the report

The purpose of this report is to:

- provide the information necessary to use the HYBRID package and to understand its general structure and capabilities
- discuss the methodological issues associated with the chosen solution technique

We assume that the reader is either familiar with the standard formulation of LP problems or at least has an appropriate manual to refer to.

1.2 Purpose of the HYBRID package

HYBRID is a mathematical programming package which includes all the functions necessary for the solution of linear programming problems. The current version of HYBRID is restricted to static LP problems. (This also includes multiobjective optimization problems that may be reformulated as LP problems – see [1–3]).

HYBRID has been designed more for real-world problems that require scenario analysis than for artificial (e.g., randomly generated) problems. Thus HYBRID is oriented towards an interactive mode of operation in which a sequence of problems is to be solved under different conditions (e.g., different objective functions, reference points, values of constraints or bounds).

The solution technique may also be used to solve quadratic problems with virtually no changes in the algorithm. However, a routine to input and handle the relevant data and a corresponding standard for data input have yet to be designed and implemented.

HYBRID will also be extended to deal with dynamic LP problems, using state equations and the reduction of gradients to control subspaces.

1.3 Outline of the solution technique

HYBRID uses a particular implementation of the Lagrange multiplier method for solving linear programming problems. Complex linear constraints are included within the augmented Lagrangian. The LP problem is solved by minimizing a sequence of quadratic functions subject to simple constraints (lower and upper bounds). This minimization is achieved by the use of a method which combines the conjugate gradient method and an active constraints strategy. The method exploits the sparseness of the matrix structure. The simple constraints are not violated during optimization and the resulting sequence of multipliers is feasible for the dual problem. The complex constraints may be violated, however, and therefore the algorithm may be started from any point that satisfies the simple constraints.

2. STRUCTURE AND GENERAL FEATURES OF THE PACKAGE

2.1 General description

The package is constructed in modules to provide a reasonably high level of flexibility and efficiency. This is crucial for planned extensions of the package (see Section 1.2) and possible modification of the algorithm (see Section 6). The source code for HYBRID consists of approximately 5000 lines of FORTRAN 77 and has been implemented on two computers – the VAX 11/780 at IIASA and the UNIVAC 1100 at the Computing Center of the State Planning Commission in Warsaw. (Please consult Dr. Z. Fortuna of the IIASA Computer Services group for details of the IIASA implementation.)

The chosen method of allocating storage takes maximum advantage of the available word sizes and of the features of typical real-world problems. In general, the matrix of constraints is large and sparse with the number of unique elements being much smaller than the number of non-zero elements. A super-sparse-matrix technique is therefore applied to the data that define the problem to be solved. This involves the construction of a table of these unique elements, one four-byte (36 bits on UNIVAC) word being used for four indices (2 logical and 2 integer). All data is packed in blank common to minimize the storage area used.

HYBRID makes it easier to verify a model or modify a problem; it also facilitates scenario analysis and reduces the problems caused by inappropriate scaling.

The data format for input and output files follows that adopted by most commercial mathematical programming systems.

2.2 Options available

Only the main options are listed below. The full array of options is given in the listings illustrating the diagnostics (see Appendices 1 and 2).

HYBRID offers the following features:

- A main storage area called the *communication region* which contains all the information corresponding to a run. The communication region is stored on disk in certain situations to allow recovery from failed (or interrupted) runs or to run a modified problem using previously obtained information without necessitating the reading and processing of the MPS input file.
- Modification of a problem at any stage of its solution (i.e., by changing the matrix coefficients, introducing or altering right-hand sides, ranges or bounds).
- Problem scaling (as described by the authors in [4] and briefly discussed in Section 3.3).
- More comprehensive diagnostics, including the checking of parallel rows, the detection of columns and rows which are empty or contain only one entry, the splitting of columns, the recognition of inconsistencies in right-hand sides, ranges and bounds, and various other features that are useful in debugging a problem formulation.
- The display of a matrix by rows (printing the nonzero elements and names of the corresponding columns, right-hand sides and ranges).
- The display of a matrix by columns (analogous to displaying by rows).
- A check of the feasibility of a problem prior to its optimization.
- Regularization of the problem (see Section 5.6).

– The provision of more detailed information for an infeasible or unbounded problem.

2.3 Control statements

The sequence of operations executed by HYBRID is controlled by the user through statements provided in a specification file (see Section 3.2). Some of these control statements are listed below. It is recommended that only the statements mentioned here should be redefined by the user: as yet there is insufficient information on the effects of changing the values of other parameters or options. The authors of the package hope to formulate guidelines governing the modification of these additional parameters/options in due course.

A control statement activates or deactivates a certain option, defines or redefines the logical number of an input/output unit, or sets the value of a parameter. Each statement has a default value which is initialized prior to starting the run. These default values are also given below.

The control statements are divided into six groups:

1. Statements without parameters
2. Statements that define the logical number of an input/output unit
3. Statements with character string parameters
4. Statements with integer parameters
5. Statements with real parameters
6. Statements with a single parameter which may be either a character string or a real number

Each group of statements is discussed separately below.

2.3.1 Statements without parameters

Each statement of this type activates or deactivates a certain action, and therefore they are listed in pairs. The only exception is the RECOVERY statement, which is dealt with in Section 2.3.2 because it may also define an input/output unit. The letter D indicates the default setting.

MAXIMIZE	D	Defines type of optimization of the objective function
MINIMIZE		

SCALE	D	Activates the scaling routine
NOSCALE		
MODIFY		Activates the routine for problem modification
NOMODIFY	D	
BYROWS		Displays the matrix by rows
NOBROWS	D	
BYCOLS		Displays the matrix by columns
NOBCOLS	D	
ACCEPT		Accepts minor errors in the MPS file (such as
NOACCEPT	D	zero elements, duplicated elements, etc.). Such errors are reported but do not cause termination of the run prior to optimization if the ACCEPT option is set
REGZERO		Regularizes the problem (see Section 5.8)
NOREGUL	D	
GETFEAS		Checks feasibility prior to optimization.
NOGETFEAS	D	This action should be avoided for problems likely to have a feasible solution because its use increases the total number of iterations
PARALLEL		Checks for parallel rows. This option is time-
NOPARALLEL	D	consuming but helps to identify dominating rows and pairs of constraints that may be replaced by a single constraint with appropriate range
ZERMULT		Sets the initial value of the Lagrange
COMMULT	D	multipliers to zero. The alternative is to compute the initial multipliers before the first iteration

2.3.2 Definition of input/output units

RECOVERY	3	Inputs the contents of the communication region. This statement causes the recovery data to be read. The absence of this statement suppresses recovery action
FINPUT	3	Inputs MPS data file
FMODIF	4	Inputs data for modification of the problem
FMULTI	8	Inputs data for multiobjective optimization
FDIAG	6	Outputs the diagnostics of the problem
FSOLUTION	7	Outputs the results
FSECURE	11	Two files used interchangeably to secure the
FBSECURE	12	contents of the communication region
FBYROWS	14	Displays a matrix by rows
FBYCOLS	15	Displays a matrix by columns
FINFO	6	Gives information issued during optimization

2.3.3 Statements with character string parameters

GOAL	Name of the neutral row taken as the objective function. If absent, the neutral row encountered first is assumed to be the objective function
RHS	Name of the set of right-hand sides and ranges. If absent, the first such name encountered is taken
BOUNDS	Name of the bounds set. If absent, the first such name encountered is taken
NAME	Name of the problem. If absent, the name found in the MPS file is assumed

2.3.4 Statements with integer parameters

MROWS	100	Maximum number of rows
MCOLS	5*MROWS	Maximum number of columns
MELEM	5*MCOLS	Maximum number of nonzero matrix elements
MDIFF	MELEM	Maximum number of unique quantities defining the problem
MTIME	10	Number of CPU minutes allocated for the run
MITER	0	Maximum number of iterations

MERRORS	50	Maximum number of errors allowed on the MPS file (before processing is terminated)
ITSCAL	30	Maximum number of iterations during scaling
ISECURE	500	Number of iterations after which the communication region is stored (in addition to secure action after each update of multipliers and termination of the run)
INORM	1	L_∞ norm is assumed in the stopping criterion. For L_2 norm, 0 should be specified

2.3.5 Statements with real parameters

BIGN	1.e+30	Any number greater then BIGN is treated as infinite
TZERO	1.e-30	Any number of absolute value less then TZERO is replaced by 0
SMALL	1.e-6	Any number in the result file with absolute value less then SMALL is replaced by 0
FEAS	1.e-4	Feasibility tolerance
SETA	.5	Parameters for scaling (see [4])
SBETA	.5	
SEPS	.985	
SEP1	.01	
RO	1.	Penalty parameters for Lagrangian (see Section 5)
ROST	2.	
ROS2	4.	
ROMX	512.	
RETA	0.	Regularization parameters (see Section 5.6)
RMET	0.	are redefined (see Section 5.7)
RSET	0.	
EPS	0.	Stopping-criteria parameters are redefined
EPSD	0.	(see Section 5.7)
EPSS	0.	

2.3.6 Statements with mixed parameters

Both of these statements take either a real-valued parameter or the word NONE. The statements are used to define the lower and upper bounds for variables. This may be changed for selected variables through appropriate definitions in the BOUNDS section of the MPS file. The default values are the following:

LBOUND 0.

UBOUND NONE

3. USER-SUPPLIED INFORMATION

3.1 General remarks

The user can supply information of four types:

- the problem specification
- the formulation of the LP problem
- modifications to the problem being solved
- the formulation of a multiobjective optimization problem

Problem specification is optional. If the specification file is empty all of the control statements take their default values. Problem specification is discussed in more detail in Section 3.2.

The formulation of the LP problem is necessary for the initial run (cold start) but not for subsequent or modification runs. Problem formulation is discussed in more detail in Section 3.3.

The problem may be modified on either an initial run or a recovery run (after finding an optimal solution, in the case of an infeasible/unbounded problem or following an interrupted run). The way in which the problem may be modified is discussed in more detail in Section 3.4.

The formulation of multiobjective optimization problems using HYBRID has not yet been fully debugged, and so the approach described in [4] should be used to generate the relevant MPS file. Software for this purpose is available on both the VAX at IIASA and the UNIVAC 1100 in Warsaw.

3.2 Problem specification

The user specifies the problem and may control some of the operations performed by HYBRID with the help of the specification file containing the control statements. The definitions and default values of these statements are given in Section 2.3.

Statements may be given in any order. The only exception is RECOVERY which – if it appears – must be the first statement in the specification file. Note that each new value for a given control statement will overwrite the previous one (either the default value or the value restored from a recovery file or previously defined in the same specification file) without any specific warning.

A statement in the specification file is recognized by the first four characters of the keyword and – if required – by a parameter following the keyword. The keywords are given in full in Section 2.3 and may be used in this form for the sake of clarity. Each statement should be specified in free format on a separate line. Only the first 30 characters are processed. Blank(s) are used to separate the keyword and its parameter, and therefore blanks cannot be embedded in either the keyword or the parameters. The last column (i.e., the 30th) must contain a blank.

The specification file is read from the unit with logical number 2 until a star (*) is encountered in the first column or EOF (end of file) is reached. The user may control printing by placing the PRINT or NOPRINT statement in the specification file. Each statement is checked for validity and error messages are printed if a statement is incorrect. If the number of errors occurring during the processing of the specification file reaches 30 the run is terminated. Any error detected during processing causes termination of the run after the specification file has been processed. The diagnostics are printed on unit number 6. If no error occurs and the PRINT directive is not in effect, no information is issued. In any case, the current values of all control statements are listed in the diagnostics file (see the examples given in Appendices 1 and 2).

A line that contains the character "c" in the first column followed by a blank in the second column is treated as a comment and ignored. There is no restriction on the contents of the remaining columns.

A control parameter is therefore defined by a default value, by a value restored from a recovery file, or by a statement in the specification file. The values of the parameters can also be overwritten in this sequence, i.e., a default

value is overwritten by a value from a recovery file, which may itself be overwritten by a statement in the specification file.

3.3 Formulation of the LP problem

At present the problem to be solved has to be presented in standard MPS format; this may be done using a commercial problem generator or a generator tailored specifically to the problem. We shall therefore make only a few general suggestions and comments on this part of the system.

Scaling the problem is very important for numerical reliability. It is generally agreed that the data and the variables should be as close to 1.0 as possible. However, since the scaling is performed by HYBRID there are not very strict restrictions on the user. Our main comment is that care should be taken in the formulation of the problem to ensure that only significant variables are included in the constraints. This requirement is easily fulfilled for real-world problems. Since HYBRID scales the values the user need not worry about differences in the magnitude of the coefficients.

We recommend that lower and upper bounds should be specified for the variables and in the constraints whenever sensible values are known. This is useful in defining the admissible region over which optimization is to be performed and usually results in a decrease in computation time.

The names of rows and columns should start with a letter or a number to avoid possible confusion with names generated by the package for multiobjective optimization problems. Names should not include a blank because of the syntax rules used in the modification routine.

Any line in the MPS file may contain a star (*) in the first column. Any such line is treated as a comment and there are no restrictions on the contents of the remaining columns.

3.4 Modification of the problem

A user may modify the problem being solved by activating the modification routine. This is done by inserting the keyword MODIFY in the specification file.

The modification lines should follow the MPS standard with the following exceptions:

1. Data are read in free format, and therefore there is no need to worry about placing data in the fields prescribed by the MPS standard.
2. Sections may occur in any order, and may also be subdivided.
3. Only 37 columns are processed.
4. Due to the problem of repacking the data (which has not yet been completely overcome), reclassifying a row or introducing new non-zero elements in the matrix is not allowed.
5. To remove a range the names of the rows affected should be specified with value 0. in the ranges section. Negative values are however illegal.

The data which are to be modified are read from the unit specified by the user (see Section 3.2) until a star (*) is found in the first column or EOF (end of file) is reached. The user may specify PRINT and NOPRINT commands in a way analogous to that described in Section 3.2.

Any user who does not want to follow the format restrictions imposed by the MPS standard should instead observe the following syntax rules:

1. Section names should follow MPS format.
2. Lines (with the exception of section names, comments, PRINT and NOPRINT commands and the star character that serves as an optional EOF mark) should have a blank in the first and 37th columns.
3. Fields are separated by at least one blank.
4. The number and contents of all fields must correspond to the information required by the modified section.
5. A line is treated as a comment if it contains the character "c" in the first column followed by a blank in the second column.

Since it is assumed that the sets of bounds and right-hand sides have been chosen, no associated name is needed. Thus, in the modification lines the corresponding fields should contain blanks (if prepared according to MPS format) or be absent (if in free format).

In addition to possible error diagnostics, other information is printed during modification. An example of modification diagnostics is presented in Appendix 2.

Processing may be terminated if the number of errors detected during modification exceeds MERRORS (see Section 2.3).

4. HYBRID-SUPPLIED INFORMATION

4.1 Initial information and problem diagnostics

Examples of the diagnostics are given in Appendices 1 and 2, and therefore only a brief description will be given here.

The information supplied may be divided into the following classes:

1. If the recovery option is activated, information about the recovery file (name of problem, date and time of creation, status of solution, size, etc.) is printed.
2. A summary of the current values of all control statements (see Section 2.3 and Section 3.2) is printed.
3. On the occasion of a cold start, the input of the MPS file is reported. Error diagnostics and warnings are also issued, if applicable. This information should be self-explanatory, and therefore is not included in the examples presented in the Appendices.
4. If the modification option is activated the relevant information and possibly some diagnostics are provided (see Section 3.4 and Appendix 2).
5. A summary of input data and problem statistics is printed.
6. If a user overestimates the core required, a reallocation procedure is called and a report is printed.
7. If scaling is performed, this is reported.
8. The setting up of the problem is reported.
9. The values of the parameters set by the PARAM procedure (see Section 5.7) are reported.

The storage allocation information issued after the problem has been set up refers to two parts of the communication region:

1. The fixed part (for a given version of HYBRID), which contains the values of all the control statements.
2. The working area, which contains the rest of the information and the data for the problem being solved.

Additional information may be placed in different files (see Appendices 3 and 4 for examples of the display of matrices by rows and columns).

4.2 Information provided during optimization

Information may be provided at several levels of detail. The user is advised not to change the default values of the parameters which control the level of detail: these default values produce the information presented in Appendix 1. This information is issued every time step number 3 in the augmented Lagrangian minimization algorithm (see Section 5.4) is executed or the multipliers are updated (see Section 5.5).

The abbreviations used (see Appendices 1 and 2) are explained below.

ITER	Number of iterations
RINF	Norm of gradient (L_∞)
C	Norm of gradient (L_2)
GOAL	Value of goal function
NINF	Number of infeasibilities
SINF	Sum of absolute values of infeasibilities
MAXINF	Maximum value of infeasibilities (the name of the row concerned follows)
SITC	Small iteration (i.e., number of conjugate gradient iterations)
RO	Value of penalty parameter
EPSRO	Value of EPS/RO
COM.TIME	Computation time
GDUAL	Value of the gradient of the dual function
MULT. NORM	Value of the norm of the multipliers
FDUAL	Value of the dual function
ACT. ROWS	Number of active rows
BASIC COLUMNS	Number of columns that are not equal to a given bound
ALF	Step length

In addition, a report is issued each time the communication region is stored.

Finally, exit from the optimization routine and the status of the solution is reported.

4.3 Results

Results are reported in standard MPS format with an additional column that contains (in the appropriate sections) scaling coefficients for each row and column.

5. SOLUTION TECHNIQUE

5.1 General remarks

The most popular methods for solving linear programming problems are based on the simplex algorithm. However, a number of other iterative approaches have recently been developed [5–7].

HYBRID belongs to this group of newer methods. The solution technique is based on the minimization of an augmented Lagrangian using a modification of the conjugate gradient method. The Lagrange multipliers are updated using a modified version of the multiplier method [8] (see Sections 5.4 and 5.5).

This method is useful not only for linear programming problems but also for other purposes, as described in Section 1.2. In addition, the method may be used to solve problems with non-unique solutions (a result of regularization – see Section 5.6).

5.2 Formulation of the problem

We will consider a linear programming problem (P) in the following standard form (see, e.g., [9]):

$$\min \; cx$$

$$b - r \le Ax \le b \qquad \text{(P)}$$

$$l \le x \le u$$

where $x, c, l, u \in R^n$, $b, r \in R^m$ and A is an $m \times n$ matrix.

The following notation will be used:

a_i denotes the i-th row of matrix A

x_j denotes the j-th component of vector x

$\|x\|$ denotes the Euclidean norm of vector x

$(u)_+$ denotes the vector composed of the non-negative elements of vector u (negative elements are replaced by zeros)

5.3 The multiplier method

We shall first explain how the multiplier method may be applied directly to LP problems.

Consider the problem (PO), which is equivalent to the problem (P):

$$\min\ cx$$

$$Bx \leq d \qquad \text{(PO)}$$

where $d \in R^p$, B is a $p \times n$ matrix, and $m \leq p \leq 2(m+n)$. To apply the multiplier method to this problem we proceed as follows:

Select initial multipliers y^0 (e.g., $y^0 = 0$) and $RO \in R$, $RO > 0$. Then for $k = 0,1,\ldots$, determine successive values of x^{k+1}, y^{k+1} where

$$x^{k+1} = \underset{x}{\operatorname{argmin}}\ L(x,y^k)$$

and

$$y^{k+1} = (y^k + RO(Bx-d))_+$$

where

$$L(x,y^k) = cx + (\|(y^k + RO(Bx-d))_+\|^2 - \|y\|^2)/(2RO)$$

until a stopping criterion is satisfied.

The method has the following basic properties:

1. A piecewise quadratic differentiable convex function is minimized at each iteration.
2. The algorithm terminates in a finite number of iterations for any positive RO.
3. There exists a constant ROM such that for any $RO \geq ROM$ the algorithm terminates in one iteration.

Note that it is assumed above that the function $L(\cdot,y^k)$ is minimized exactly and that the value of the penalty parameter RO is fixed. Less accurate minimization may be performed provided that certain conditions are fulfilled (see, e.g., [7,8]). For numerical reasons a non-decreasing sequence of penalty parameters $\{RO(k)\}$ is generally used instead of a fixed RO.

5.4 Modification of the conjugate gradient method for the minimization of an augmented Lagrangian

The conjugate gradient method has been modified to take advantage of the formulation of the problem. The method may be understood as an extension of the techniques developed by Polyak [10], O'Leary [11] and Hestenes [12] for minimization of a quadratic function on an interval using the conjugate gradient method.

The problem (P) may be reformulated as follows:

$$\begin{aligned} &\min\ cx \\ &Ax + z = b \\ &l \le x \le u \\ &0 \le z \le r \end{aligned} \qquad \text{(PS)}$$

where $z \in R^m$ are slack variables.

Formulation (PS) has a number of advantages over the initial formulation (PO):

1. The dimension of matrix A in (PS) is usually much smaller than that of matrix B in (PO).
2. The problem is one of minimization of a quadratic function in (PS), and of minimization of a piecewise quadratic in (PO).
3. Some computations only have to be performed for subsets of variables. Note that slack variables are introduced only for ease of interpretation and do not have to be computed.

In (PS) the augmented Lagrangian is defined by

$$L(x,z,y) = cx + (\|y + RO(Ax+z-b)\|^2 - \|y\|^2)/(2RO) \ .$$

We shall first discuss the problem of minimizing $L(x,z,y)$ for given y,RO, subject to lower and upper bounds for x and z. The gradient of L is defined by

$$\frac{\partial L}{\partial x} = c + A^T(y + RO(Ax+z-b))$$

$$\frac{\partial L}{\partial z} = y + RO(Ax+z-b) = g + RO\cdot z$$

where

$$g = y + \mathrm{RO}(Ax - b) \ .$$

From the Kuhn–Tucker optimality condition the following relations hold for the minimum point (x^*, z^*):

$$\frac{\partial L}{\partial x_j} \geq 0 \ \text{ if } \ x_j^* = l_j \ , \quad \frac{\partial L}{\partial x_j} \leq 0 \ \text{ if } \ x_j^* = u_j \ ,$$

$$\frac{\partial L}{\partial z_i} \geq 0 \ \text{ if } \ z_i^* = 0 \ , \quad \frac{\partial L}{\partial z_i} \leq 0 \ \text{ if } \ z_i^* = r_i \ ,$$

and

$$\frac{\partial L}{\partial x_j} = 0 \ \text{ if } \ l_j < x_j^* < u_j$$

$$\frac{\partial L}{\partial z_i} = 0 \ \text{ if } \ 0 < z_i^* < r_i \ .$$

For any given point such that $l \leq x \leq u$ it is possible to determine slack variables z in such a way that the optimality conditions with respect to z are obeyed. Variables z are defined by

$$z_i = \begin{cases} 0 & \text{if } g_i > 0 \qquad (\partial L / \partial z_i > 0) \\ r_i & \text{if } g_i < -\mathrm{RO}\cdot r_i \qquad (\partial L / \partial z_i < 0) \\ -g_i / \mathrm{RO} & \text{if } -\mathrm{RO}\cdot r_i \leq g_i \leq 0 \quad (\partial L / \partial z_i = 0) \end{cases} \ .$$

We shall use the following notation and definitions. The vector of variables x with indices that belong to a set J will be denoted by x^J, and analogous notation will be used for variables g. We shall let q denote minus the gradient of the Lagrangian reduced to x-space $(q = -(\partial L / \partial x))$. The following sets of indices are defined for a given point x:

The set of indices I of active constraints, i.e.,

$$I = \{i : g_i \leq -\mathrm{RO}\cdot r_i\} \cup \{i : g_i \geq 0\} \ .$$

$\bar{I}$ is the complement of I, i.e.,

$$\bar{I} = \{1,2,\ldots,m\} \setminus I \ .$$

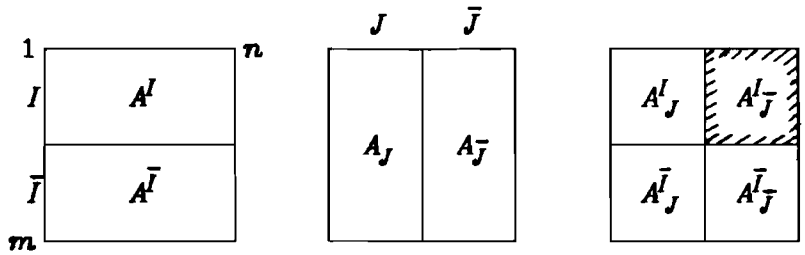
The set of indices J of variables that should be equal to either the upper or the lower bound, i.e.,

$$J = \{j : x_j = l_j \text{ and } q_j \leq 0\} \cup \{j : x_j = u_j \text{ and } q_j \geq 0\} .$$

$\bar{J}$ is the complement of J, i.e.,

$$\bar{J} = \{1,2,.....,n\} \setminus J .$$

For clarity the matrix A may be split up as follows:



In essence, the Lagrangian is minimized using the conjugate gradient method with the following modifications:

1. During the minimization process x and z satisfy simple constraints and z enters the augmented Lagrangian in the form defined on p. 17.
2. The conjugate gradient routine is run until no new constraint becomes active, i.e., neither set I nor set J increases in size. If this occurs, the computed step length is shortened to reach the next constraint, the corresponding set (I or J) is enlarged and the conjugate gradient routine is re-entered with the direction set equal to minus the gradient.
3. Sets J and I are defined before entering the procedure discussed in point 2 and may be enlarged only before the minimum is found. When the minimum with respect to the variables with indices in sets $\bar{J}$ and I has been found, sets J and I are redefined.
4. Minimization is performed subject only to those components of variables x whose indices belong to set $\bar{J}$, i.e., variables that are not currently equal to a bound value.

5. Minimization is performed subject only to those components of variables z whose indices belong to set I, i.e., slack variables that correspond to violated constraints. Note that formally this only requires the use of different formulae for z. In actual fact it is sufficient to know only the members of set I.

We may now present the algorithm for minimization of the augmented Lagrangian in a more formal way. The algorithm consists of the following steps:

1. For given y and RO choose a point x such that $l \leq x \leq u$
2. Compute $g = y + \mathrm{RO}(Ax - b)$
3. Determine sets I and $\bar{I}$
4. Compute g defined as follows:

$$g_i := \begin{cases} g_i + \mathrm{RO}\cdot r_i & \text{if } g_i < -\mathrm{RO}\cdot r_i \\ g_i & \text{if } g_i > 0 \\ 0 & \text{if } -\mathrm{RO}\cdot r_i \leq g_i \leq 0 \end{cases}$$

5. Compute minus the gradient reduced to x-space:

$$q = -(c + (A^I)^T g^I)$$

6. Determine sets J and $\bar{J}$
7. If $q_j = 0$ for all $j \in \bar{J}$ then x is a minimum point of the Lagrangian
8. Set $p^J = q^J$
9. Compute

$$s = A_{\bar{J}} p^J$$

$$c = \| q^J \|^2$$

$$d = \| s^I \|^2$$

$$\mathrm{ALF}(1) = c / d$$

Note that ALF(1) is the conjugate gradient step length in direction p^J

10. Find the step length that would violate the nearest non-active constraint, i.e.,

$$\mathrm{ALF}(2) = \min_{s_i > 0} \{ g_i / s_i \}, \quad i \in \bar{I}$$

11. Find the step length that would enable a variable to reach a bound, i.e.,

$$\text{ALF(3)} = \min_j \left\{ \frac{(l_j - x_j)}{p_j} \; : \; p_j < 0, \quad j \in \bar{J} \right\}$$

$$\text{ALF(4)} = \min_j \left\{ \frac{(u_j - x_j)}{p_j} \; : \; p_j > 0, \quad j \in \bar{J} \right\}$$

12. Determine step length ALFA = $\min_i$ (ALF(i)) and compute the new point $x^J := x^J + \text{ALFA} \cdot p^J$ and minus the gradient at that point:

$$g_i := g_i - \text{ALFA} \cdot s_i$$

g_i as in step 4

$$q^{\bar{J}} = -((A_J^I)^T g^I + c^{\bar{J}})$$

13. If $q = 0$ go to step 2

14. If ALFA = ALF(1) continue with the conjugate gradient step

$$\text{BETA} = \| q^J \| / c$$

$$c = \| q^J \|^2$$

$$p^J = q^J + \text{BETA} \cdot p^J$$

and go to step 9

15. If ALFA = ALF(2) add the index for which ALF(2) had its smallest value to set I and remove that index from set $\bar{I}$. Go to step 4

16. Add the index for which min (ALF(3),ALF(4)) had the smallest value to set J and remove that index from set $\bar{J}$. Go to step 8

Note that this step is only reached if ALFA = ALF(3) or ALFA = ALF(4)

Note that the condition $q = 0$ is in practice replaced by $\| q^{\bar{I}} \| \leq \text{EPS}(k)$. The value of EPS($k$) may be quite large in the first few iterations; it then decreases as the number of iterations increases.

5.5 Modification of the multiplier method

The method will be presented in algorithmic form.

1. Compute an initial vector of multipliers on the basis of the particular option chosen (i.e., either $y^0 = 0$ or y^0 corresponding to the constraints violated at starting point x)

2. Find x^{k+1} which minimizes the augmented Lagrangian (see Section 5.4) with accuracy EPS(k). It is assumed that EPS(k) : = min (EPS(k) , gx·EPS(k)), where the sequence EPS(k) → 0, and gx is the norm of the violated constraints. In addition, EPS(k) ≥ EPSM, where EPSM is the assumed maximum accuracy

3. If $\| q^J \| \leq$ EPS(k) and more than one conjugate gradient iteration has been performed, go to step 6 ($\| q^J \|$ is the norm of the gradient of the augmented Lagrangian)

4. Set RO(k) : = min (RO(k)·ROST , ROMX) (ROMX is the assumed maximum value of the penalty parameter and ROST is assumed to be constant)

5. If RO(k) = ROMX then set EPS(k) : = max (EPS(k)·EPSS,EPSM) where EPSS and EPSM are assumed parameters. Go to step 2

6. Compute new multipliers

$$y_i^{k+1} : = \begin{cases} y_i^k + \text{RO}(k)(g_i x^{k+1} - b_i) & \text{if } y^k + \text{RO}(k)(a_i x^{k+1} - b_i) \geq 0 \\ y_i^k + \text{RO}(k)(g_i x^{k+1} - b_i + r_i) & \text{if } y^k + \text{RO}(k)(a_i x^{k+1} - b_i + r_i) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

7. If $\| y^{k+1} - y^k \| >$ EPSD then set RO(k + 1) = min (RO(k)·ROST,ROMX), set k : = k + 1 and go to step 2

8. Check the feasibility of the current point. If it is feasible ($gx^{k+1} \leq$ FEAS), minimize the augmented Lagrangian with the vector of multipliers fixed at y^{k+1} and with accuracy EPS(k + 1), and then stop

9. If the point tested at step 8 was infeasible and RO(k) < ROMX then set RO(k + 1) = min (RO(k)·ROST,ROMX), set k : = k + 1 and go to step 2

10. If step 9 was omitted, check the feasibility of the problem by minimizing the square Euclidean norm of the violated constraints (provided that such a test has not already been performed – see option GETF). If the problem is infeasible, then stop

11. Take the feasible solution found in step 10 as the current point, set $k := k + 1$, update EPS(k) = max (EP(k)·EPSS,EPSM) and go to step 2

5.6 Regularization

It is possible that a linear programming problem may not have a unique optimal solution. Although this is theoretically rare, in practice many problems actually have a large set of widely varying basic solutions for which the objective values differ very little [7].

In some cases the simplex algorithm will stop when a basic solution is recognized as optimal for a given set of tolerances. For problems with a non-unique optimum the first optimal solution found is accepted, so that one may not even be aware of the non-uniqueness of the solution reported as optimal.

Thus we are faced with the problem of choosing an optimal (or, in most cases, to be more accurate, a suboptimal) solution that possesses certain additional properties required by the user. This problem may be overcome by applying an approach called ***regularization***. Regularization is a means of finding the optimal solution with either minimum Euclidian norm or minimum distance from a given reference point. The second of these options has not yet been implemented; the first may be activated by a REGZERO statement in the specification file.

The minimum norm solution is obtained by carrying out a sequence of minimizations of regularized augmented Lagrangians rather than one minimization of an "ordinary" augmented Lagrangian. Thus minimization of $L(\cdot,y^k)$ in problem (PO) is replaced by

$$x^{k+1} = \underset{x}{\operatorname{argmin}}\, L(x,y^k) + \frac{1}{2\,\mathrm{RETA}(k)}\|x\|^2$$

where

$$\mathrm{RETA}(k) \to \infty, \quad \sum_{i=1}^{\infty} (\mathrm{RO}(k)/\mathrm{RETA}(k))^{1/2} < \infty .$$

5.7 Setting parameters

Various parameters occur in the algorithms presented in the preceding two sections. Most of them play an important role and have to be chosen very carefully. Moreover, the values of some of these parameters are (or should be) interrelated.

The values of any of the parameters may be reset by the user. If this is done, the PARAM procedure checks only whether the parameter meets certain general requirements, e.g., that it is positive. Thus the user should be very careful when making changes in parameters that affect tolerances.

Some parameters have a non-zero default value. This is generally the case for parameters that do not depend on the problem being solved. If the user specifies an unacceptable value for such a parameter, the default value is restored.

Other parameters default to an initial value of zero; the parameter values are then recomputed according to the rules given below as the program proceeds. If a user specifies a non-zero initial value which becomes unacceptable during the course of the calculation, the values computed from the following rules are restored:

EPS = FEAS*abs(AMXMAT), where

EPS is the initial tolerance for minimization of the augmented Lagrangian

FEAS is the feasibility tolerance

AMXMAT is the largest of the matrix elements

EPSD = FEAS, where

EPSD is the stopping criterion for multiplier iteration

RETA = 1./abs(AMXMAT), where

RETA is the initial regularization parameter

RSETA = ROST**2, where

RSETA is the coefficient for increasing RETA

RMETA = 1.e+4/sqrt(FEAS), where

RMETA is the maximum value of the regularization parameter

MITER = 2*(N+M), where

MITER is the maximum number of multiplier iterations

N is the number of variables

M is the number of constraints

MSITER = (M+N)*N, where

MSITER is the maximum number of iterations during minimization of the augmented Lagrangian

6. CONCLUSIONS

Although HYBRID was made operational during our visit to IIASA in July 1982, a lot of work still remains to be done before the package can be used without involving some consultation with the authors. This is due to the fact that less than one man-year has so far been assigned directly for the development of HYBRID. However, the results of the tests performed so far seem to justify our choice of method.

HYBRID provides very useful diagnostics for any LP problem and is also useful for problem verification. These functions are not performed by other linear programming packages, e.g., by MINOS – it is interesting to note that the authors of MINOS actually advise the user to debug and verify the problem with another package before using MINOS. In fact MINOS may generate an irrelevant problem for the MPS file, even including some minor mistakes (like duplicated lines) without any error messages.

It is true that for some problems HYBRID performs much worse than the commercial packages FMPS and MINOS. Although we have solved many small-scale problems using HYBRID, these small examples do not provide useful information on the time required for solution because of the large proportion of 'user time' that depends on the current load of the computer. To solve a 'Mangasarian-type' problem [5] (a full dense matrix of 57 rows and 57 columns) takes HYBRID 1.38 minutes and MINOS 0.7 minutes. On the other hand, HYBRID would need more than 10 minutes (stopping the run by setting a time limit) to solve a medium-sized problem (a model of the agricultural sector with about 200 rows) while MINOS requires only about 2 minutes. This is due to the fact that HYBRID is (for some problems) very sensitive to the values of certain parameters and to the options used. We expect further work on HYBRID to result in greatly improved performance in this field.

One of the advantages of HYBRID is illustrated by the fact that, for a medium-sized problem (185 rows, 236 columns, 2236 elements) HYBRID uses only 16.7% of the virtual and 47% of the real memory required by MINOS. This proportion could be greatly improved in the optimization phase either through segmentation of the package (for computers designed to allow such action) or by splitting the package into two parts, the first of which will perform the diagnostics and prepare the communication region, while the second carries out the optimization.

The advantages introduced by the scaling option should also not be overlooked. It is generally agreed that scaling is essential for numerical reliability – the problem used by Dr. Fortuna of the IIASA Computer Services group to demonstrate the efficiency of scaling provides a good illustration. (Incidentally, this problem is not solved correctly by MINOS).

The future development of HYBRID will follow two main paths:

First, considerable effort will be devoted to better evaluation of the parameters and control options that are critical to the performance of HYBRID. The data processing that will allow the treatment of quadratic problems will be made operational when the necessary debugging has been completed. It is possible that we shall implement a numerically stable procedure for the minimization of quadratic functions – however, this would require more memory.

Second, user-oriented options will be emphasized, with particular importance being attached to procedures for generation, interactive modification and solution of a sequence of multiobjective problems. A REVISE procedure that will allow the introduction of new columns and rows, reclassification of rows and the introduction of new elements will be encoded. In addition, we hope to examine the possibility of linking a problem generator to HYBRID to provide a tool for formulating problems in a more straightforward way than standard MPS format.

We hope that, despite the reservations outlined above, HYBRID will eventually be a useful tool with many practical applications. We would be grateful for any criticisms or comments that would help us to improve the package.

ACKNOWLEDGEMENTS

The authors wish to thank the Chairman of the System and Decision Sciences Program at IIASA, Prof. A.P. Wierzbicki, for his continued encouragement and support of the work reported here. Thanks are also due to Dr. Z. Fortuna

and Dr. A. Lewandowski for many helpful comments and suggestions. Last but by no means least, the authors are greatly indebted to Helen Gasking for editing and processing this report.

REFERENCES

1. M. Kallio, A. Lewandowski and W. Orchard-Hays. An implementation of the reference point approach for multiobjective optimization. WP-80-35. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980.
2. M. Makowski and J. Sosnowski. Coordination of sectoral production planning using prices and quotas. CP-81-38. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1981.
3. A.P. Wierzbicki. A methodological guide to multiobjective decision making. WP-79-122. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1979.
4. M. Makowski and J. Sosnowski. Implementation of an algorithm for scaling matrices and other programs useful in linear programming. CP-81-37. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1981.
5. O.L. Mangasarian. Iterative solution of linear programs. *SIAM Journal for Numerical Analysis*, 18(4): 606–614, 1976.
6. B.T. Polyak and N.V. Tretiyakov. An iterative method for linear programming and its economic interpretation. *Economic and Mathematical Methods*, 8: 740–751, 1972 (in Russian).
7. J.S. Sosnowski. Linear programming via augmented Lagrangian and conjugate gradient methods. In S. Walukiewicz and A.P. Wierzbicki (Eds.), *Methods of Mathematical Programming*, Proceedings of a 1977 Conference in Zakopane. Polish Scientific Publishers, Warsaw, 1981.
8. D.P. Bertsekas. Multiplier methods: a survey. *Automatica*, 12: 133–145, 1976.
9. B.A. Murtagh and M.A. Sanders. MINOS – A large-scale nonlinear programming system (for problems with linear constraints). User guide. Technical Report, Systems Optimization Laboratory, Stanford University, 1977.
10. B.T. Polyak. The conjugate gradient method in extremal problems. *Computational Mathematics and Mathematical Physics*, 9: 94–112, 1969.
11. D.P. O'Leary. A generalized conjugate gradient algorithm for solving a class of quadratic problems. *Linear Algebra and its Applications*, 34: 371–399, 1980.
12. M.R. Hestenes. *Conjugate Gradient Methods in Optimization*. Springer Verlag, Berlin, 1980.

APPENDIX 1 Sample run (cold start)

```
                h y b r i d   -   version 1.0   july 1982

executed on  22.09.83  at 21:44.05

files
recovery....    3       mps input...    2       modificat..    4
multicriter.    8       column.list.    9       row list...    9
diagnostic..    6       solution....    6       secure.....   11
bsecure.....   12       mps output..   13       byrows out.   14
bycols out..   15       userfile....   10

options
recovery....   no       modification   no       multicr....   no
getfeasible.   no       userfile....   no       mpsout.....   no
reg.(ref)...   no       reg.(zero)..   no       reg.(prec.)   no
accept......   no       byrows......  yes       bycolumns..  yes
scaling.....  yes       paral. rows.   no       comp.multp.  yes
lower bound.  yes       upper bound.   no       spirit.....   no

names
problem.....                objective...              (maximize)
rhs,ranges..                bounds......              solution...opti
columns.....    nzer        rows........     nact

dimensions
rows........  100       columns.....  300       elements...  1500
diff. elem..  1500

integer parameters
iter. log...    0       sec. freq...  1500      istop......    1
iswtch......    1       inorm.......     1      max. errors   50
imup........    1       nbcl........     0      irotx......    1
miter.......    0       mtime.......    10      iter count.    0
infeas......    2       initi.......     0

real parameters
big number.  0.1000e+31  zero  tol..  0.1000e-29  small number  0.1000e-05
lower boun.  0.     e+00  upper boun.  0.1000e+31  feas. tol...  0.1000e-03
tpara......  0.1000e-05  eps........  0.     e+00  epsm........  0.     e+00
epss.......  0.     e+00  epsd.......  0.     e+00  ro..........  0.1000e+01
rost.......  0.2000e+01  ro max.....  0.2000e+03  ro step2....  0.4000e+01
sbeta......  0.5000e+00  seta.......  0.5000e+00  seps........  0.9850e+00
seps1......  0.1000e-01
reta.......  0.     e+00  rmeta......  0.     e+00  rseta.......  0.     e+00
em.........  0.     e+00  ems........  0.7600e+00  emmin.......  0.     e+00

  9291  4-bytes words is assigned for working area


input of mps file from unit   2
 card    section
    1    name          exhybri
    2    rows
    7    columns
                              updated max. number of rows is      4
   18    rhs
   21    ranges
   23    bounds
   27    endata
```

```
 input summary
 =============
 objective       goal        (max)
 rhs and ranges  rhl
 bounds          bdl

number of
 rows                    4   (   1 eq,    1 le,    1 ge,    1 n)
 columns                 3   (max. spec.     300)
 matrix elements        10   (max. spec.    1500)
 total dif. magn.       17   (max. spec.    1500)
 rhs                     2
 ranges                  1
 bounds                  3

                    matrix only      matrix, rhs, ranges and bounds
 density            83.333%                -
 v                  0.32557                  2.6672
 mean                1.3450                -
 var                0.46241                -
 min. elem.         1.00000                 1.00000
 max. elem.          2.0700                  15.000


realocation of memory assignement
actual numer of columns  is      4
actual numer of elements is     10

scaling begins
                                    matrix only     matrix,rhs,ranges,bounds
iter     gn          v          min.coef   max.coef   min.coef   max.coef
 0  19.14       1.409      0.4474       1.030      0.4474       7.211
 1  2.714       1.144      0.6339       1.553      0.6339       7.725        g
 2  2.868       1.026      0.5266       1.398      0.5266       5.411        ccd
 3  1.663      0.6410      0.5793       1.607      0.3978       3.356        ccd
 4 0.6232d-01 0.4863      0.6036       1.224      0.3356       2.950        ccd
 5 0.2493d-02 0.4871      0.6193       1.191      0.3248       2.930        ccd
optimal scaling   iter=   5 stop= 1.00174

after scaling
                    matrix only      matrix, rhs, ranges and bounds
 v                  0.1142                   0.5854
 mean                1.141                 -
 var                0.3109                 -
 min. elem.         1.0000                   0.2500
 max. elem.          2.070                   3.750


scaling coef.     minimal      maximal
rows             0.12500      0.25000
columns           4.0000       4.0000


end with seting up the problem (   0.0250 min. execution time so far)
  319 4-bytes words are finnally assigned for working area

fixed part of communication region contains  632 bytes
recovery information is saved on unit no   11
   eps       2.07000e-04
   epss     0.750000
   epsm      1.00000e-06
   epsd     1.00000e-04
   reta    0.483092
   rmeta     1.00000e+06
   rseta     4.00000
   nbol    3
   miter    14
   msiter    21
```

```
exit spirit routine


iter =    0 grad[inf]=  2.29752     grad[1**2]= 2.68502     goal=  7.31000
ninf =    2 sinf      =  13.0525     maxinf    =  12.793      (a1      )
sitc =    0 ro        =  1.00000     epsro = 0.20700e-03 comp. time  0.30667e-01


iter =    3 grad[inf]=  1.61773     grad[1**2]= 1.61773     goal=  36.2703
ninf =    2 sinf      =  1.27764     maxinf    = 0.92593      (a3      )
sitc =    6 ro        =  1.00000     epsro = 0.20700e-03 comp. time  0.31667e-01
gdual     =  4.6732     mult. norm =  2.3366     fdual =  30.093
act. rows =    2    basic columns =    2
grad[inf] = 0.28054e-06 grad[1**2] =  0.14439d-12 alf       =  1.7664
recovery information is saved on unit no   12


iter =    4 grad[inf]=  1.01000     grad[1**2]= 1.42130     goal=  48.7856
ninf =    2 sinf      =  2.87935     maxinf    =  1.9534      (a1      )
sitc =    2 ro        =   2.0000     epsro = 0.77625e-04 comp. time  0.36667e-01
gdual     = 0.41222e-08 mult. norm = 0.11683e-01 fdual =  30.093
act. rows =    2    basic columns =    2
grad[inf] = 0.34530e-06 grad[1**2] =  0.22984d-12 alf       = 0.43330
recovery information is saved on unit no   11

fixed part of communication region contains  632 bytes
recovery information is saved on unit no   12


iter =    5 grad[inf]= 0.692006e-06 grad[1**2]=0.960741d-06 goal=  30.0928
ninf =    0 sinf      =       0.     maxinf    =      0.     (        )
sitc =    2 ro        =     4.00     epsro = 0.77625e-04 comp. time  0.42667e-01
itmul =    2 itotal =     10


exit optimization routine - status: optimal solution
```

problem name exhybri
status optimal solution
objective value 30.093
iteration count 5

section 1 - rows

number	...row..	at	...activity...	slack activity	.lower limit..	..upper limit.	dual activity.	.scaling coef.
1	a3	eq	.	.	.	.	-0.92586	0.25000
2	a1	ul	12.00000	.	2.00000	12.00000	0.97671	0.12500
3	a2	bs	-12.18328	2.81672	-15.00000	none	.	0.25000
4	goal		30.09276	••••••••••••	objective	••••••••••••		

section 2 - columns

number	.column.	at	...activity...	obj. gradient.	.lower limit..	..upper limit.	reduced cost..	.scaling coef.
1	x1	bs	12.05941	1.01000	1.00000	none	-0.13329e-03	4.00000
2	x2	ul	6.00000	1.05000	2.00000	6.00000	3.06203	4.00000
3	x3	bs	11.61276	1.00000	.	none	0.13840e-03	4.00000

end of run - total execution time 0.0507 min.
finished on 22.09.83 at 21:44.12

APPENDIX 2 Sample run from recovery file with modifications

```
  following specs cards are read
recovery 3


recovery from file no        3
problem name            exhybri
saved on                22.09.83
at                      20:43.53
status          optimal solution
iteration count                5
size                         160
c byrows
c bycols
modify
fmodify 5
romx 200.
finput 2
fsoluti 6
*
1

                h y b r i d   -   version 1.0   july 1982

executed on  22.09.83  at 21:31.49

files
recovery....    3       mps input...    2       modificat..    5
multioriter.    8       column.list.    9       row list...    9
diagnostic..    6       solution....    6       secure.....   11
bsecure.....   12       mps output..   13       byrows out.   14
bycols out..   15       userfile....   10

options
recovery....  yes       modification  yes       multicr....   no
getfeasible.   no       userfile....   no       mpsout.....   no
reg.(ref)...   no       reg.(zero)..   no       reg.(prec.)   no
accept......   no       byrows......  yes       bycolumns..  yes
scaling.....  yes       paral. rows.   no       comp.multp.  yes
lower bound.  yes       upper bound.   no       spirit.....   no

names
problem..... exhybri      objective...goal          (maximize)
rhs,ranges..rhl           bounds......bdl           solution...opti
columns.....    nzer      rows........     nact

dimensions
rows........    4       columns.....    4       elements...   10
diff. elem..   21

integer parameters
iter. log...    0       sec. freq...  1500       istop......    1
iswtch......    1       inorm.......     1       max. errors   50
imup........    1       nbcl........     3       irotx......    1
miter.......   14       mtime.......    10       iter count.    5
infeas......    2       initi.......     0

real parameters
big number.  0.1000e+31  zero  tol..  0.1000e-29  small number  0.1000e-05
lower boun.  0.    e+00  upper boun.  0.1000e+31  feas. tol...  0.1000e-03
tpara.......  0.1000e-05  eps........  0.1552e-03  epsm........  0.1000e-05
epss........  0.7500e+00  epsd.......  0.1000e-03  ro..........  0.4000e+01
rost........  0.2000e+01  ro max.....  0.2000e+03  ro step2....  0.4000e+01
sbeta.......  0.5000e+00  seta.......  0.5000e+00  seps........  0.9850e+00
seps1.......  0.1000e-01
reta........  0.4831e+00  rmeta......  0.1000e+07  rseta.......  0.4000e+01
em..........  0.    e+00  ems........  0.7600e+00  emmin.......  0.    e+00
```

```
    320  4-bytes words is assigned for working area
1


 modification cards are read from unit no  5
   following modification cards are read
 columns
 columns  section is modified
    x1   goal  12.
 x1       goal      old coef.    1.0100000      is replaced by    12.000000
 noprint
   printing of cards is supressed
 bounds   section is modified
 lower bound for x1        = 1.00000000      is replaced by           0.
 upper bound for x1        =  17.000000      is introduced
 lower bound for x2        =
 upper bound for x2        =
 columns  section is modified
 x2       a2        old coef.    2.0699999      is replaced by    2.0200000
 x1       a1        old coef.    2.0200000      is replaced by    2.0000000
 ranges   section is modified
 range for a1         =  10.0000000      is removed
 range for a2         =   12.000000      is introduced
 rhs      section is modified
 rhs for a1           =   12.000000      is replaced by   12.000000
 rhs for a2           =  -15.000000      is replaced by  -2.0699999
 upper   bound for a2         =     -3.0000      is replaced by  9.9300003
 problem name  exhybri is changed to test              (due to existing range)
    17 cards have been processed

  problem summary (after modification)
  ====================================
  name             test
  objective        goal        (max)
  rhs and ranges   rhl
  bounds           bdl

 number of
  rows                   4    (   1 eq,    1 le,    1 ge,    1 n)
  columns                3    (max. spec.      4)
  matrix elements       10    (max. spec.     10)
  total dif. magn.      21    (max. spec.     21)
  rhs                    2
  ranges                 1
  bounds                 4

                    matrix only      matrix, rhs, ranges and bounds
  density           83.333%                -
  v                 1.6006                   3.7762
  mean              2.4370                 -
  var               3.2177                 -
  min. elem.       1.00000                   1.00000
  max. elem.        12.000                    17.000


 scaling is supressed
 scaling coef. from recovery file are applied
1


 end with seting up the problem (   0.0593 min. execution time so far)
   320 4-bytes words are finnally assigned for working area

 fixed part of communication region contains  632 bytes
 recovery information is saved on unit no   11



 exit spirit routine
```

```
 iter =    5 grad[inf]=  4.95046     grad[1**2]= 6.14165     goal=  162.626
 ninf =    1 sinf      =  2.45332     maxinf    =  2.4533     (a2      )
 sitc =    0 ro        =     4.00     epsro = 0.77625e-06 comp. time  0.64000e-01
 gdual     =  229.04     mult. norm = 0.72684     fdual =  93.534
 act. rows =    3    basic columns =    3
 grad[inf] = 0.18169e-06 grad[1**2] =  0.89840d-13 alf       =  2.9481
 recovery information is saved on unit no   12


 iter =    7 grad[inf]= 0.550476e-01 grad[1**2]=0.571372d-01 goal=  98.6845
 ninf =    3 sinf      = 0.128105     maxinf    = 0.55754e-01 (a1      )
 sitc =    4 ro        =     8.00     epsro = 0.10000e-05 comp. time  0.70000e-01
 gdual     = 0.20651e-08 mult. norm = 0.72684     fdual =  93.534
 act. rows =    3    basic columns =    3
 grad[inf] = 0.25082e-06 grad[1**2] =  0.10102d-12 alf       =  1.2518
 recovery information is saved on unit no   11

 fixed part of communication region contains  632 bytes
 recovery information is saved on unit no   12


 iter =    8 grad[inf]= 0.683330e-06 grad[1**2]=0.928557d-06 goal=  93.5335
 ninf =    0 sinf      =       0.     maxinf    =       0.     (        )
 sitc =    3 ro        =     8.00     epsro = 0.10000e-05 comp. time  0.74667e-01
 itmul =    2 itotal =       7


 exit optimization routine - status: optimal solution
1problem name      test
 status            optimal solution
 objective value       93.534
 iteration count        8
```

section 1 - rows

number	...row..	at	...activity...	slack activity	.lower limit..	..upper limit.	dual activity.	.scaling coef.
1	a3	eq	.	.	.	.	-8.19433	0.25000
2	a1	ul	12.00000	.	none	12.00000	6.55217	0.12500
3	a2	ll	-2.07000	.	-2.07000	9.93000	7.20169	0.25000
4	goal		93.53352	••••••••••••	objective	••••••••••••		

section 2 - columns

number	.column.	at	...activity...	obj. gradient.	.lower limit..	..upper limit.	reduced cost..	.scaling coef.
1	x1	bs	7.12683	12.00000	.	17.00000	0.80335e-04	4.00000
2	x2	bs	1.09401	1.05000	none	none	0.13667e-03	4.00000
3	x3	bs	6.86287	1.00000	.	none	-0.96734e-04	4.00000

end of run - total execution time 0.0827 min.
finished on 22.09.83 at 21:31.56

APPENDIX 3 Display of a matrix by rows

```
problem summary
===============
name              exhybri
objective         goal        (max)
rhs and ranges    rhl
bounds            bdl

number of
 rows                    4    (   1 eq,    1 le,    1 ge,    1 n)
 columns                 3    (max. spec.       4)
 matrix elements        10    (max. spec.      10)
 total dif. magn.       17    (max. spec.    1500)
 rhs                     2
 ranges                  1
 bounds                  3

                    matrix only      matrix, rhs, ranges and bounds
 density             83.333%                -
 v                  0.32557                  2.6672
 mean                1.3450                -
 var                0.46241               -
 min. elem.         1.00000                 1.00000
 max. elem.          2.0700                  15.000


rows

      1   a3          (eq)    rhs =      0.
      1.04000x1             -1.08000x3

      2   a1          (le)    rhs =  12.000       range = 10.0000
      2.02000x1             -2.06000x2

      3   a2          (ge)    rhs = -15.000       no range
     -1.03000x1             -2.07000x2             1.09000x3

      4   goal        (ne)
      1.01000x1              1.05000x2             1.00000x3


bounds
      1.00000(lo)x1              2.00000(lo)x2              6.00000(up)x2
```

APPENDIX 4 Display of a matrix by columns

```
  problem summary
  ===============
  name              exhybri
  objective         goal        (max)
  rhs and ranges    rhl
  bounds            bdl

 number of
  rows                    4    (   1 eq,    1 le,    1 ge,    1 n)
  columns                 3    (max. spec.       4)
  matrix elements        10    (max. spec.      10)
  total dif. magn.       17    (max. spec.    1500)
  rhs                     2
  ranges                  1
  bounds                  3

                      matrix only      matrix, rhs, ranges and bounds
  density             83.333%                -
  v                  0.32557                  2.6672
  mean                1.3450                -
  var                0.46241               -
  min. elem.         1.00000                 1.00000
  max. elem.          2.0700                  15.000


columns

      1   x1        lower bound       1.00000 upper bound     none
     1.01000goal           2.02000a1            -1.03000a2             1.04000a3

      2   x2        lower bound       2.00000 upper bound       6.00000
     1.05000goal          -2.06000a1            -2.07000a2

      3   x3        lower bound         .       upper bound     none
    -1.08000a3             1.09000a2             1.00000goal
```