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

Postan - A Package for Postoptimal Analysis (An Extension of Minos)

Dobrowolski, G., Hajduk, K., Korytowski, A. and Rys, T.

IIASA Collaborative Paper
July 1984

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

POSTAN – A PACKAGE FOR POSTOPTIMAL ANALYSIS (AN EXTENSION OF MINOS)

G. Dobrowolski
K. Hajduk
A. Korytowski
T. Ryś

July 1984
CP-84-32

Collaborative Papers report work which has not been performed solely at the International Institute for Applied Systems Analysis and which has received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute, its National Member Organizations, or other organizations supporting the work.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

PREFACE

This paper presents a new software package which has been developed in collaboration with IIASA. The new package, POSTAN, is designed for postoptimal analysis of linear programming problems, and is embedded in the well-known linear and nonlinear programming code MINOS. POSTAN is composed of a number of FORTRAN subroutines which may be called by adding some new keywords to the original list of MINOS specifications. The main function of POSTAN is to determine the ranges in which certain parameters may be changed without affecting the optimal solution and/or the optimal basis.

In this paper the authors outline the general form of the linear programming problems studied, describe the six new subroutines in some detail, and illustrate this description with a printout obtained in the solution of a sample problem. The mathematical theory behind the software package is given in an Appendix.

ANDRZEJ WIERZBICKI
Chairman
System and Decision Sciences

POSTAN – A PACKAGE FOR POSTOPTIMAL ANALYSIS (AN EXTENSION OF MINOS)

G. Dobrowolski[1], *K. Hajduk*[2], *A. Korytowski*[2], *and T. Ryś*[1]

1. INTRODUCTION

POSTAN is a postoptimal analysis package for linear programming problems. It is composed of a number of FORTRAN routines which are incorporated into MINOS, the well-known linear and nonlinear programming code developed by Murtagh and Saunders [1]. The postoptimal analysis of a linear programming problem is performed after MINOS has found an optimal solution, and is initiated by adding particular specifications to the original list of MINOS specifications.

As the output of the unmodified version of MINOS includes sensitivity coefficients, the objectives of POSTAN are confined to *ranging*, i.e., determining the ranges in which certain parameters (or groups of parameters) may be changed without affecting the optimal solution and/or the optimal basis.

The formulation of the linear problem analyzed by POSTAN is the same as for MINOS: Minimize (or maximize) a linear cost function

$$F(x) = a_0 x \qquad (1)$$

subject to m row constraints:

$$d_i \leq a_i x \leq g_i \; , \quad i = 1,\ldots,m \qquad (2)$$

and n constraints on separate variables:

$$d_{m+i} \leq x_i \leq g_{m+i} \; , \quad i = 1,\ldots,n \quad . \qquad (3)$$

[1]Systems Research Department of the Institute of Automatics, Systems Engineering and Telecommunications (Academy of Mining and Metallurgy, Cracow) and of the Industrial Chemistry Research Institute, Warsaw, Poland.
[2]Institute of Automatics, Systems Engineering and Telecommunications, Academy of Mining and Metallurgy, Cracow, Poland.

Here x is an n-dimensional column vector of decision variables, a_0 is an n-dimensional row vector of cost coefficients (also called the *objective row*), the a_i, $i = 1,...,m$, are n-dimensional row vectors, the lower bounds d_i, $i = 1,...,m+n$, are real numbers or $-\infty$, and the upper bounds g_i, $i = 1,...,m+n$, are real numbers or $+\infty$. Of course, if the bounds take the values $+\infty$ or $-\infty$ the corresponding relation (2) or (3) must be replaced by a strict inequality. If $d_i = g_i$, then the variable x_i is said to be *fixed*. If $d_i = -\infty$ and $g_i = +\infty$ the variable x_i is said to be *free*. Analogous terms are used to describe the rows $a_i x$.

It should be recalled that in MINOS the two-sided inequality constraints (2) are not stated explicitly, but rather specified using *ranges*. More precisely, a one-sided inequality is introduced in the form $a_i x \le g_i$ (type L) or $a_i x \ge d_i$ (type G), together with a real number r_i called the *range*. In the first case, the difference between the right-hand side g_i and this number yields the lower bound ($d_i = g_i - r_i$); in the second case the sum of the right-hand side d_i and the real number r_i gives the upper bound ($g_i = d_i + r_i$).

The linear programming problem is transformed by MINOS into the following internal form: Minimize (or maximize) the variable

$$-\tilde{x}_{n+1+\mathrm{obj}} \tag{4}$$

subject to equality constraints:

$$\tilde{A}\tilde{x} = 0 \tag{5}$$

and inequality constraints:

$$\tilde{l} \le \tilde{x} \le \tilde{u} \quad . \tag{6}$$

Here $\tilde{A}$ is an $(m+1) \times (n+m+2)$-matrix:

$$\tilde{A} = \begin{bmatrix} \tilde{a}_1 & \tilde{b}_1 & \\ \cdot & \cdot & \\ \cdot & \cdot & I \\ \cdot & \cdot & \\ \tilde{a}_{m+1} & \tilde{b}_{m+1} & \end{bmatrix} , \tag{7}$$

where I denotes the $(m+1)\times(m+1)$ identity matrix and

$$\tilde{a}_i = a_i \quad \forall i < \text{obj}\ , \quad \tilde{a}_{\text{obj}} = a_0\ , \quad \tilde{a}_i = a_{i-1} \quad \forall i > \text{obj}\ , \tag{8}$$

$$\tilde{b}_i = b_i \quad \forall i < \text{obj}\ , \quad \tilde{b}_{\text{obj}} = 0\ , \quad \tilde{b}_i = b_{i-1} \quad \forall i > \text{obj}\ ,$$

where

$$b_i = \begin{cases} 0 \text{ if } d_i = -\infty \text{ and } g_i = +\infty \\ d_i \text{ if } d_i \text{ is finite and } g_i = +\infty \\ g_i \text{ if } g_i \text{ is finite} \end{cases} .$$

The first n components of the extended vector of decision variables $\tilde{x} \in R^{n+m+2}$ form a subvector identical to x; these components are described as *structural*. Element $\tilde{x}_{n+1}$ is called the *right-hand-side component*; it is fixed at -1. The remaining components of $\tilde{x}$ are called *slack* or *logical components*. The objective variable $\tilde{x}_{n+1+\text{obj}}$ is free. The vector of lower bounds $\tilde{l}$ and the vector of upper bounds $\tilde{u}$ are defined as follows:

$$\tilde{l}_i = d_{m+i} \quad \forall i = 1,\dots,n\ , \quad \tilde{l}_{n+1} = -1, \quad \tilde{l}_{n+1+\text{obj}} = -\infty \ , \tag{9}$$

$$\tilde{u}_i = g_{m+i} \quad \forall i = 1,\dots,n\ , \quad \tilde{u}_{n+1} = -1, \quad \tilde{u}_{n+1+\text{obj}} = +\infty \ .$$

Now let $i = n+1+j$, $j = 1,\dots,m$. Then

$$\tilde{l}_i = h_i\ , \quad \tilde{u}_i = k_i \ \text{ for } j < \text{obj} \text{ and } \tilde{l}_i = h_{i-1}\ , \quad \tilde{u}_i = k_{i-1} \text{ for } j > \text{obj} \ , \tag{10}$$

where

$h_i = k_i = 0$ if the j-th row constraint is fixed (i.e., of type E) (11)

$h_i = 0$, $k_i = +\infty$ if $d_j = -\infty$ and g_j is finite (one-sided constraint of type L)

$h_i = -\infty$, $k_i = 0$ if d_j is finite and $g_j = +\infty$ (one-sided constraint of type G)

$h_i = 0$, $k_i = g_j - d_j$ if d_j and g_j are finite

$h_i = -\infty$, $k_i = +\infty$ if the j-th row constraint is free .

It should be noted that for practical reasons all quantities greater than or equal to 10^{15} are taken as equal to infinity in POSTAN, and all quantities whose absolute value is less than 10^{-9} are regarded as equal to zero.

2. POSTAN SUBROUTINES

In its present form the package contains six subroutines, which can be divided into two groups. CRAN, RHSRAN and BRAN perform ordinary ranging (by elements) while DIRRAN, DRHSRN and DBRAN perform directional ranging. In this section we describe the input required by each subroutine and the output that it produces, and give an explanation of the results. The mathematical theory is presented in the Appendix.

2.1. CRAN

CRAN performs ordinary ranging on the costs. For each cost component a_0^i, $i = 1,...,n$, the subroutine determines the largest range in which a_0^i may vary without affecting the optimal solution. While the range for a_0^i is being determined, all other components a_0^j, $j \neq i$, remain fixed at their original values. CRAN also gives some information on the change of state of variables at the boundaries.

This subroutine does not require any input data.

The output is entitled COST RANGING. The following information is then given for each cost component, $i = 1,...,n$:

NUMBER	Number of structural variable
COLUMN	Name of structural variable
OBJ GRADIENT	Cost component
LOWER LIMIT	Lower boundary of the range in which the cost component may vary without affecting the optimal solution
UPPER LIMIT	Upper boundary of this range
CHANGE AT LOWER LIMIT (OR OPT SOL VANISHES)	Name of the nonbasic variable which changes its state at the lower boundary; this is printed only if the lower boundary is finite. (Beware: CRAN does not know if there is an optimal solution beyond the boundary so that the name of a nonbasic variable may be printed even if the optimal solution vanishes)
CHANGE AT UPPER LIMIT (OR OPT SOL VANISHES)	Name of the nonbasic variable which changes its state at the upper boundary (other explanations as above)
M+J	NUMBER $+m+1$

2.2. RHSRAN

RHSRAN performs ordinary ranging on the right-hand sides (rhs). For each component $\tilde{b}_i$, $i = 1,\ldots,m+1$, of the vector of right-hand sides (except for the objective row, $i \neq$ obj), this subroutine determines the maximum range in which $\tilde{b}_i$ may vary without affecting the optimal basis. While the range for $\tilde{b}_i$ is being determined, all other components $\tilde{b}_j$, $j \neq i$, are fixed at their original values. It should be noted that the rhs vector $\tilde{b}$ is not always the right-hand side of a constraint system in the original formulation (1)–(3); the user should refer to (5)–(11). In addition, RHSRAN gives some information on the change of state of variables at the boundaries.

This subroutine does not require any input data.

The output is entitled RHS RANGING. The following information is then given for each rhs component, $i = 1,\ldots,m+1$, except for the objective row, $i \neq$ obj:

NUMBER	$n+i+1$
ROW	Name of row
RHS	Right-hand-side component $\tilde{b}_i$
LOWER LIMIT	Lower boundary of the range in which the rhs component may vary without affecting the optimal basis
UPPER LIMIT	Upper boundary of this range
CHANGE AT LOWER LIMIT (OR OPT SOL VANISHES)	Name of the basic variable which becomes nonbasic at the lower boundary. LL is printed if this variable reaches its lower bound and UL if it reaches its upper bound; the name is printed only if the boundary is finite. (Beware: RHSRAN does not know if there is an optimal solution beyond the boundary and so a variable name may be printed even if the optimal solution vanishes)
CHANGE AT UPPER LIMIT (OR OPT SOL VANISHES)	Name of the basic variable which becomes nonbasic at the upper boundary (other explanations as above)
M+J	Number of row

2.3. BRAN

BRAN performs ordinary ranging on the bounds. For each lower bound $\tilde{l}_i$ and each upper bound $\tilde{u}_i$, $i = 1,\ldots,n+m+2$, the subroutine determines two ranges: range A, which is the maximum range in which the bound may vary without affecting the optimal solution, and range B, which is the maximum range in which the bound may vary without affecting the optimal basis. While these ranges are being determined for $\tilde{l}_i$ (or $\tilde{u}_i$), all other bounds remain fixed at their original values. BRAN also gives some information on the change of state of variables at the boundaries. This analysis is not performed for fixed variables, i.e., if $\tilde{u}_i = \tilde{l}_i$.

This subroutine does not require any input data.

The output is entitled BOUND RANGING. It is divided into two parts, A and B, which will now be discussed separately.

Part A

Part A is entitled A. NO SOLUTION CHANGE and is divided into two subsections, SECTION 1 – ROWS and SECTION 2 – COLUMNS, which correspond to the sections of the same name in the final output of MINOS.

SECTION 1 – ROWS contains the following information for each slack variable $\tilde{x}_i$, $i = n+2,\ldots,n+m+2$ (or for each row constraint), except for the slack variable $\tilde{x}_{n+1+\text{obj}}$ which corresponds to the objective row. In the first two columns we have:

NUMBER	Number of slack variable i
ROW	Name of row

If $\tilde{u}_i = \tilde{l}_i$ for the slack variable under consideration, the remaining columns contain only the message FIXED VARIABLE.

In the case when the slack variable $\tilde{x}_i$ is nonbasic at its lower bound the message VARIABLE AT LOWER BOUND appears in the next two columns, which otherwise contain:

LL FOR L BOUND	Lower boundary of range A for $\tilde{l}_i$
UL FOR L BOUND	Upper boundary of range A for $\tilde{l}_i$

The next two columns give similar information about the upper bound. In other words, if the slack variable $\tilde{x}_i$ is nonbasic at its upper bound, the message VARIABLE AT UPPER BOUND is printed; if not the columns contain:

LL FOR U BOUND	Lower boundary of range A for $\tilde{u}_i$
UL FOR U BOUND	Upper boundary of range A for $\tilde{u}_i$

The last column contains:

I	Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable $\tilde{x}_i$, $i = 1,\ldots,n$. All of the information may be interpreted in the same way as in SECTION 1 – ROWS, with the following exceptions:

NUMBER	Number of structural variable
COLUMN	Name of structural variable
M+J	$m + 1 + i$

Part B

Part B is entitled B. NO BASIS CHANGE. It is also divided into two subsections, SECTION 1 – ROWS and SECTION 2 – COLUMNS.

SECTION 1 – ROWS contains the following information for each slack variable $\tilde{x}_i$, $i = n+2,\ldots,n+m+2$, except for the slack variable $\tilde{x}_{n+1+\mathrm{obj}}$ which corresponds to the objective row. The first two columns contain:

NUMBER	Number of slack variable
ROW	Name of row

If $\tilde{u}_i = \tilde{l}_i$ for the slack variable under consideration, the remaining columns contain only the message FIXED VARIABLE.

In the case when the slack variable $\tilde{x}_i$ is nonbasic at its lower bound the message VARIABLE AT LOWER BOUND appears in the next two columns, which otherwise contain:

LL FOR L BOUND	Lower boundary of range B for $\tilde{l}_i$
UL FOR L BOUND	Upper boundary of range B for $\tilde{l}_i$

The next two columns give similar information about the upper bound. In other words, if the slack variable $\tilde{x}_i$ is nonbasic at its upper bound, the message VARIABLE AT UPPER BOUND is printed; if not the columns contain:

LL FOR U BOUND	Lower boundary of range B for $\tilde{u}_i$
UL FOR U BOUND	Upper boundary of range B for $\tilde{u}_i$

The columns which follow all appear under the heading CHANGES AT BDRIES (OR OPT SOL VANISHES). These are used only for nonbasic slack variables, remaining blank for basic variables.

If the slack variable $\tilde{x}_i$ is at its lower bound then the columns contain the names of the basic variables which change their state at the boundaries of range B for $\tilde{l}_i$, given that the solution does not vanish. The message LL indicates that the variable has reached its lower bound, while UL shows that the upper bound has been reached. The first column, headed LOWER, gives the name of the variable which changes its state at the lower boundary of range B for $\tilde{u}_i$; the second column, headed UPPER, gives the name of the variable which changes its state at the upper boundary. The name of $\tilde{x}_i$ may also appear under the heading UPPER. This means that $\tilde{u}_i$ is the upper boundary of range B for $\tilde{l}_i$ and the set of feasible solutions is then empty beyond this boundary.

If the slack variable $\tilde{x}_i$ is at its upper bound, these columns contain the names of the basic variables which change their states at the boundaries of range B for $\tilde{u}_i$, given that the solution does not vanish. The messages LL and UL have the same meaning as above. The first column, headed LOWER, gives the name of the variable which changes its state at the lower boundary of range B for $\tilde{u}_i$; the second column, headed UPPER, gives the name of the variable which changes its state at the upper boundary. If the name of $\tilde{x}_i$ appears under the heading LOWER, then $\tilde{l}_i$ is the lower boundary of range B for $\tilde{u}_i$ and the set of feasible solutions is then empty beyond this boundary.

The last column contains:

I	Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable $\tilde{x}_i$, $i = 1,\dots,n$. All of the information may be interpreted in the same way as in SECTION 1 – ROWS, with the following exceptions:

NUMBER	Number of structural variable
COLUMN	Name of structural variable
M+J	$m+1+i$

Beware: in most cases BRAN does not know if there is an optimal solution beyond the boundaries of range B and so a variable name may be printed under

CHANGES AT BDRIES (OR OPT SOL VANISHES) even if the optimal solution vanishes. The question of whether the optimal solution exists may be answered (in the negative) only if the name of the nonbasic variable $\tilde{x}_i$ appears in the appropriate column of the output.

2.4. DIRRAN

DIRRAN performs directional ranging on the costs. For a given increment $\Delta a_0 \in R_n$ of the cost vector a_0, this subroutine determines the largest real $t_{max} \geq 0$ such that for every cost vector of the form $a_0 + t\Delta a_0$, $t \in [0, t_{max}]$, the optimal solution is the same as at the point a_0 (i.e., at $t = 0$). The boundary cost components $a_0^i + t_{max}\Delta a_0^i$, $i = 1,\ldots,n$, and some information on the change of state of variables at the boundary are also given. Beware: if a structural variable, say $\tilde{x}_i$, is fixed, then Δa_0^i is automatically set to zero, regardless of the value given in the data.

Data: see Section 3.2.

The output is entitled DIRECTIONAL COST RANGING. It takes one of two forms, depending on the value of t_{max}. If $t_{max} < 10^{15}$, we have the finite range case, while if $t_{max} \geq 10^{15}$ we have the infinite range case. Let us consider the finite range case first.

In this case the sub-heading FINITE RANGE is printed below the main title, with the corresponding value of t_{max} in brackets. Next, the following information is given for each structural variable $\tilde{x}_i$, $i = 1,\ldots,n$:

NUMBER	Number of structural variable
COLUMN	Name of structural variable
DIRECTION	Increment component Δa_0^i
OBJ GRADIENT	Cost component a_0^i
BOUNDARY VALUE	Boundary value of cost component $(a_0^i + t_{max}\Delta a_0^i)$
M+J	$m + 1 + i$

At the boundary $t = t_{max}$ either the optimal solution vanishes or one of the nonbasic variables changes its state. The name and original state of this variable are given in the last row of the output in the form: AT BOUNDARY VARIABLE "name" CEASES TO BE AT "bound" OR OPTIMAL SOLUTION VANISHES. The letters LL are substituted for "bound" if the variable is no longer at its lower bound, while UL appears if the variable is no longer at its upper bound.

In the infinite range case ($t_{max} \geq 10^{15}$) the message INFINITE RANGE (TMAX.GE.1.E15) is displayed. Beneath this the same information is given for each structural variable as in the finite range case, except for the BOUNDARY VALUE, which is no longer relevant.

2.5. DRHSRN

DRHSRN performs directional ranging on the right-hand sides. For a given vector of increments $\Delta\tilde{b} \in R^{m+1}$ of the rhs vector $\tilde{b}$, DRHSRN determines the largest real $t_{max} \geq 0$ such that for every rhs of the form $\tilde{b} + t\Delta\tilde{b}$, $t \in [0, t_{max}]$, the optimal basis is the same as at the point $\tilde{b}$ (i.e., at $t = 0$). $\Delta\tilde{b}_{obj}$ is automatically set to zero.

Data: see Section 3.2.

The output is entitled DIRECTIONAL RHS RANGING. It takes one of two forms, depending on the value of t_{max}. If $t_{max} < 10^{15}$, we have the finite range case, while if $t_{max} \geq 10^{15}$ we have the infinite range case. Let us consider the finite range case first.

In this case the sub-heading FINITE RANGE is printed below the main title, with the corresponding value of t_{max} in brackets. Next, the following information is given for each row (or each slack variable $\tilde{x}_i$, $i = n+2,\ldots,n+m+2$), except for the objective row (or slack variable $\tilde{x}_{n+1+obj}$):

NUMBER	Number of slack variable
ROW	Name of row
DIRECTION	Component $\Delta\tilde{b}_i$ of the increment vector
RHS	Right-hand-side component $\tilde{b}_i$
BOUNDARY VALUE	Boundary value of rhs component $(\tilde{b}_i + t_{max}\Delta\tilde{b}_i)$
I	Row number

At the boundary $t = t_{max}$ either the optimal solution vanishes or one of the basic variables changes its state. The name and type of change are given in the last row of the output in the form: AT THE BOUNDARY VARIABLE "name" PASSES FROM THE BASIS TO "bound" OR OPTIMAL SOLUTION VANISHES. The letters LL are substituted for "bound" if the variable reaches its lower bound and UL if it reaches its upper bound.

In the infinite range case ($t_{max} \geq 10^{15}$) the message INFINITE RANGE (TMAX.GE.1.E15) is displayed. Beneath this the same information is given for each non-objective row as in the finite range case, except for the BOUNDARY VALUE, which is no longer relevant.

2.6. DBRAN

DBRAN performs directional ranging on the bounds. For a given vector of increments col $(\Delta\tilde{l},\Delta\tilde{u}) \in R^{2(n+m+2)}$ of the vector of bounds col $(\tilde{l},\tilde{u})$, this subroutine determines two real numbers:

- $t_{maxa} \geq 0$, the largest real number such that for every bound vector of the form col $(\tilde{l},\tilde{u})$ + t col $(\Delta\tilde{l},\Delta\tilde{u})$, $t \in [0,t_{maxa}]$, the optimal solution is the same as for the bound vector col $(\tilde{l},\tilde{u})$, i.e., at $t = 0$.
- $t_{maxb} \geq 0$, the largest real number such that for every bound vector of the form col $(\tilde{l},\tilde{u})$ + t col $(\Delta\tilde{l},\Delta\tilde{u})$, $t \in [0,t_{maxb}]$, the optimal basis is the same as for the bound vector col $(\tilde{l},\tilde{u})$, i.e., at $t = 0$.

The bound increments $\Delta\tilde{l}_i,\Delta\tilde{u}_i$ which correspond to fixed variables are automatically set to zero regardless of the values given in the data.

Data: see Section 3.2.

The output is entitled DIRECTIONAL BOUND RANGING. Information on t_{maxa} is given under the heading A. NO CHANGE IN THE OPTIMAL SOLUTION. If $t_{maxa} < 10^{15}$ the message FINITE RANGE is displayed, with the corresponding value of t_{maxa} in brackets. If $t_{maxa} \geq 10^{15}$, INFINITE RANGE (TMAXA.GE.1.E15) is printed. Similar information on t_{maxb} is given under the heading B. NO CHANGE IN THE OPTIMAL BASIS. The rest of the output is divided into two sections: SECTION 1 – ROWS and SECTION 2 – COLUMNS.

SECTION 1 – ROWS contains the following information for each slack variable $\tilde{x}_i$, $i = n+2,\ldots,n+m+2$ (or for each row), except for the slack variable $\tilde{x}_{n+1+obj}$ which corresponds to the objective row:

NUMBER	Number of slack variable
ROW	Name of row
LL DIRECTION	Component $\Delta\tilde{l}_i$ of the lower bound increment vector $\Delta\tilde{l}$
LL BOUNDARY A	Boundary value of the lower bound $\tilde{l}_i + t_{maxa}\Delta\tilde{l}_i$; this is printed only if $t_{maxa} < 10^{15}$
LL BOUNDARY B	Boundary value of the lower bound $\tilde{l}_i + t_{maxb}\Delta\tilde{l}_i$; this is printed only if $t_{maxb} < 10^{15}$
UL DIRECTION	Component $\Delta\tilde{u}_i$ of the upper bound increment vector $\Delta\tilde{u}$

UL BOUNDARY A	Boundary value of the upper bound $\tilde{u}_i + t_{\text{maxa}}\Delta\tilde{u}_i$; this is printed only if $t_{\text{maxa}} < 10^{15}$
UL BOUNDARY B	Boundary value of the upper bound $\tilde{u}_i + t_{\text{maxb}}\Delta\tilde{u}_i$; this is printed only if $t_{\text{maxb}} < 10^{15}$
I	Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable $\tilde{x}_i$, $i = 1,...,n$, with the following differences:

NUMBER	Number of structural variable
COLUMN	Name of structural variable
M+J	$m + 1 + i$

The last two rows of the output contain information on the change of state of variables at the boundaries. If $t_{\text{maxa}} < 10^{15}$, the message AT THE BOUNDARY A THE VARIABLE "name" HITS "bound" is displayed. The letters LL are substituted for "bound" if the variable hits its lower bound and letters UL if it hits the upper bound. If $t_{\text{maxb}} < 10^{15}$, the message AT THE BOUNDARY B THE BASIC VARIABLE "name" BECOMES NONBASIC AT "bound" OR OPTIMAL SOLUTION VANISHES is displayed in the next row. Once again, LL is used to denote the lower bound and UL the upper bound.

3. IMPLEMENTATION OF POSTAN FOR MINOS

In order to insert the POSTAN procedures into MINOS, and to allow them to be used in the same way as other MINOS facilities, we have made the changes outlined below.

3.1. New key-words in the SPECS file

BY-ELEMS COST RANGING (default: off)	This activates the postoptimal analysis of cost ranges. Subroutine CRAN is called (see Section 2.1). Insensitivity ranges for each cost coefficient are calculated under the assumption that the values of the others are kept constant. There is no request for data
BY-ELEMS RHS RANGING (default: off)	A similar procedure is carried out for each component of the rhs vector. Subroutine RHSRAN is called (see Section 2.2). There is no request for data
BY-ELEMS BOUND RANGING (default: off)	This command initiates the computation of insensitivity ranges for the upper and lower bounds of each structural and logical variable. Subroutine BRAN is called (see Section 2.3). Insensitivity ranges are produced for two

	cases: NO SOLUTION CHANGE and NO BASIS CHANGE. There is no request for data
DIRECTIONAL COST RANGING (default: off)	This is the first of three commands which enable the user to perform postoptimal directional analysis. The cost vector is shifted along the direction indicated by the data while the optimal basis is retained. Subroutine DIRRAN is called (see Section 2.4). The length of the insensitivity interval thus obtained is printed out
DIRECTIONAL RHS RANGING (default: off)	This command activates postoptimal analysis of the rhs vector. Subroutine DRHSRN is called (see Section 2.5). The direction of change has to be specified
DIRECTIONAL BOUND RANGING (default: off)	Postoptimal analysis of the upper and lower bounds of all variables is activated. Subroutine DBRAN is called (see Section 2.6). The user has to provide data to define the direction of alteration of both upper and lower bounds
DATA RANGING FILE n (default: $n = 5$)	This key-word specifies the logical number of the data file for directional ranging procedures. File n is read after the processing of other MINOS input files has been completed. It is obligatory to declare a data ranging file if at least one of the ranging procedures is used. If none of these procedures is called, this key-word will be ignored if it is present

3.2. Data ranging file – input format

The data for the postoptimization procedures are prepared in an MPS-like format and placed in the file specified by the MINOS key-word DATA RANGING FILE. The data sets for different directional ranging procedures may be given in any order. The beginning of the data set for each procedure is identified by the line NAME and its end by the line ENDATA. If it occurs, the line 'SET' must be given immediately after the line NAME in each data set; this line defines the default values of all the variables which are not explicitly defined. Every data set is identified by the name given in the line NAME.

The records in the DATA RANGING FILE should have the following (basic) form, which is analogous to MPS format:

Columns:	1–4,	5–12,	15–22,	25–36,	40–47,	50–61
Fields:	f1,	f2,	f3,	f4,	f5,	f6

Below we give a detailed description of the data set for each directional ranging procedure.

Directional Bound Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DBOU			
2.	'SET'	Comments		Value		
3.		LOWER	Row/col. name	Value	Row/col. name	Value
4.		UPPER	Row/col. name	Value	Row/col. name	Value
5.	ENDATA					

Remarks:

- If field f2 in a given record is empty, this means that it is the same as in the previous record. Field f2 must not be empty in the first data record.
- The records with identifiers UPPER and LOWER may appear in any order.
- LOWER is used for increments of the lower bounds and UPPER for increments of the upper bounds.

Directional Cost Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DCOS			
2.	'SET'	Comments		Value		
3.			Col. name	Value	Col. name	Value
4.	ENDATA					

Directional RHS Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DRHS			
2.	'SET'	Comments		Value		
3.			Row name	Value	Row name	Value
4.	ENDATA					

The following general rules apply to all data sets:

- One of the fields f3,f4 may be empty.
- If 'SET' appears, it must follow immediately after NAME. If 'SET' does not occur, the default for all variables whose values are not specified is zero. This has the same effect as

  'SET'   0.

- Comments may be entered in arbitrary positions in the data set. They are identified by an asterisk * in the first column.

- The DATA RANGING FILE is read once to find the necessary data set (one cycle is performed).

3.3. Alterations to MINOS

Several of the original MINOS procedures have been altered in order to accommodate the POSTAN package.

Workspace for the ranging procedures is located within array Z of MINOS.

4. AN EXAMPLE

We shall now illustrate the performance of POSTAN using a simple example. The linear programming problem is as follows: Maximize

$$F(x) = x_1 - x_2 + 0.5x_3 + 2x_4 + 3x_5$$

subject to

$$1.1x_1 + 1.2x_2 + 1.3x_3 \le 7$$

$$0.1x_3 + 0.2x_4 - x_5 \ge -7$$

$$-10.6 \le 2.1x_3 + 2.2x_4 \le 10.7$$

$$x_1 + x_2 + x_3 = 0.01$$

$$x_1 \le 1.5\,, \quad x_2 \ge -1.4\,, \quad 0 \le x_3 \le 10\,, \quad x_5 = 2 \quad .$$

An additional constraint is introduced as the fourth row:

$$-\infty < 5x_1 + 5x_2 + 5x_3 + 5x_4 < +\infty$$

in order to demonstrate the effect of a free constraint on the POSTAN output.

Below we give the MINOS and POSTAN specifications used to solve this problem. Notice that all the ranging routines of POSTAN are called. The MPS file and the data ranging file for POSTAN are then presented. It should be noted that in the DBRAN data the increment components for those bounds of the slack variables to which MINOS assigns a default value of zero are also set equal to zero. This is normal in most applications.

We then give the standard MINOS printout, followed by the output of the subroutines DIRRAN, DBRAN, DRHSRN, CRAN, BRAN, and RHSRAN.

```
begin  (T  E  S  T)
      max
      rows 10
*
      by-elems cost   ranging
      by-elems rhs    ranging
      by-elems bound  ranging
*
      directional cost   ranging
      directional rhs    ranging
      directional bound  ranging
*
      data ranging file 7
      mps          file 3
*
      log frequency 1
end
```

```
name           test
rows
 n  ob
 l  r1
 g  r2
 l  r3
 n  r4
 e  r5
columns
    x1         ob                  1.
    x1         r1                  1.1
    x1         r4                  5.
    x1         r5                  1.
    x2         ob                 -1.
    x2         r1                  1.2
    x2         r4                  5.
    x2         r5                  1.
    x3         ob                  .5
    x3         r1                  1.3
    x3         r2                  .1
    x3         r3                  2.1
    x3         r4                  5.
    x3         r5                  1.
    x4         ob                  2.
    x4         r2                  .2
    x4         r3                  2.2
    x4         r4                  5.
    x5         ob                  3.
    x5         r2                 -1.
rhs
    rh         r1                  7.
    rh         r2                 -7.
    rh         r3                 10.7
    rh         r5                  .01
ranges
    ra         r3                 21.3
bounds
 mi bo         x1
 up bo         x1                  1.5
 lo bo         x2                 -1.4
 up bo         x3                 10.
 fr bo         x4
 fx bo         x5                  2.

endata
```

```
name            dcos
'set'           comm                 1.
*
* empty set of data for directional cost ranging
*
* ---------------------------------------------
*
* this is a comment
*
endata
name            dbou
'set'                                1.
    lower       x4                   0.
    lower       r1                   0.
    lower       r2                   0.
                r3                   0.
    lower       r4                   0.
    lower       r5                   0.
    upper       x4                   0.    r1                 0.
                r2                   0.
                r5                   0.    r4                 0.
endata
name            drhs
'set'     #########                  1.
                r4                   0.
endata
1
```

```
                m i n o s    ---    version  4.0    mar  1981
                = = = = =

                *********************************************
                S T A R T   at 14:08:45
                CED 1984   Wed Jun 20
                *********************************************
```

```
specs file
----------
           begin  (T  E  S  T)
                max
                rows 10
           *
                by-elems cost  ranging
                by-elems rhs   ranging
                by-elems bound ranging
           *
                directional cost  ranging
                directional rhs   ranging
                directional bound ranging
           *
                data ranging file 7
                mps          file 3
           *
                log frequency 1
           end
1
```

```
parameters
----------
mps input data.
row limit.............         10     list limit............          0     lower bound default....  0.  e+00
column limit..........         30     error message limit....        10     upper bound default....  1.00e+20
elements limit (coeffs)       150     phantom elements.......         0     aij tolerance..........  1.00e-10

files.
mps file (input file)..         3     old basis file (map)...         0     (card reader)..........         5
solution file..........         0     new basis file (map)...         0     (printer)..............         6
insert file............         0     backup basis file......         0     (scratch file).........         8
punch file.............         0     load file..............         0     dump file..............         0
data ranging file .....         7

frequencies.
log iterations.........         1     check row error........        30     cycle limit............         1
save new basis map.....       100     factorize (invert).....        60     cycle tolerance........  0.  d+00

lp parameters.
iterations limit.......        30     feasibility tolerance..  1.00d-05     partial price factor...         1
crash option...........         1     dj tolerance...........  1.00d-06     multiple price.........         0
weight on objective....  0.  d+00     pivot tolerance........  5.27d-09

nonlinear problems.
nonlinear constraints..         0     superbasics limit......         1     derivative level.......         3
nonlinear jacobian vars         0     hessian dimension......         0     verify level...........         0
nonlinear objectiv vars         0     linesearch tolerance...   0.01000     difference interval....  1.05d-08
problem number.........         0     reduced-gradient tol...   0.20000     conjugate-gradnt method         1

augmented lagrangian.
jacobian...............    sparse     major iterations limit.        20     radius of convergence..  1.00d-02
lagrangian.............       yes     minor iterations limit.        40     row tolerance..........  1.00d-05
penalty parameter......  0.  d+00     completion.............      full

miscellaneous.
lu row tolerance.......  1.00d-04     print level..(jflxi)...         1     imbed..................       yes
lu col tolerance.......   0.10000     debug level............         0     print spikes...........        no
lu mod tolerance.......   0.90000     unbounded objectv value  1.00d+20     unbounded step size....  1.00d+10

post-optimal analysis.
direct. cost ranging...       yes     direct. bound ranging..       yes     direct. rhs ranging....       yes
by-elems cost ranging..       yes     by-elems bound ranging.       yes     by-elems rhs ranging...       yes
matrix element ranging.        no     matrix row ranging.....        no     matrix column ranging..        no

number of words of core available for workspace   10000
1

mps file
--------
      1    name          test
      2    rows
      9    columns
     30    rhs
     35    ranges
     37    bounds
     44    endata
```

```
names selected
--------------
objective       ob                  (max)         1
rhs             rh                                4
ranges          ra                                1
bounds          bo                                6



matrix statistics
-----------------
              total      normal        free        fixed      bounded
rows            6           2            2            1           1
columns         5           0            1            1           3

no. of matrix elements       20      density      55.556
no. of rejected coeffs        0        aijtol 1.00000e-10
biggest and smallest coeffs  5.00000e+00   1.00000e-01  (excluding obj and rhs)

length of row-name hash table          101
collisions during table lookup           0

partition size for partial pricing       5
1
iterations
----------


crash option  1
free rows     2    free cols     1    pass2 (e rows)    1    pass3    0    remainder    2

factorize    1   demand     0   iteration     0    infeas      1    objectv  0.         d+00
slacks       4   linear     2   nonlinear     0    elems      12    density  33.3
p4 bumps     0   spikes     0   core reqd   207    l limit  2133    u limit  4266
lu bumps     0   spikes     0   aij elems     8    l elems     3    u elems     1    f elems    0   0.

-itn    0 -- feasible solution.  objective =    1.853727230d+01


biggest dj =  0.    d+00   norm rg =  0.    d+00   norm pi =  1.188d+00
1
exit -- optimal solution found.

norm of x                        4.864d+00      norm of pi                        1.188d+00
1
problem name    test                  objective value    1.8537272295d+01

status          optimal soln          iteration    0    superbasics    0

objective       ob                  (max)
rhs             rh
ranges
```

section 1 - rows

number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
7	ob	bs	18.53727	-18.53727	none	none	1.00000	1
8	r1	bs	-0.12900	7.12900	none	7.00000	0.	2
9	r2	bs	-1.02727	-5.97273	-7.00000	none	0.	3
10	r3	ul	10.70000	0.	-10.60000	10.70000	-0.90909	4
11	r4	bs	24.36818	-24.36818	none	none	0.	5
12	r5	eq	0.01000	0.	0.01000	0.01000	-1.00000	6

1
section 2 - columns

number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
1	x1	bs	1.41000	1.00000	none	1.50000	0.	7
2	x2	ll	-1.40000	-1.00000	-1.40000	none	-2.00000	8
3	x3	ll	0.	0.50000	0.	10.00000	-2.40909	9
4	x4	bs	4.86364	2.00000	none	none	0.	10
5	x5	eq	2.00000	3.00000	2.00000	2.00000	3.00000	11
6	rh	eq	-1.00000	0.	-1.00000	-1.00000	-9.73727	12

1

p o s t a n --- postoptimal analysis package , version 1.2 jun 1984

1
data ranging file

```
      1  name          dcos
      2  'set'         comm      0.100000d+01
      3  *
      4  * empty set of data for directional cost ranging
      5  *
      6  * ----------------------------------------------
      7  *
      8  * this is a comment
      9  *
xxx empty ranging file. all variables are set to 0.100000d+01
     10  endata
```

1directional cost ranging

infinite range (tmax.ge.1.e15)

number	.column.	...direction..	.obj.gradient.	m+j
1	x1	1.00000	1.00000	7
2	x2	1.00000	-1.00000	8
3	x3	1.00000	0.50000	9
4	x4	1.00000	2.00000	10
5	x5	1.00000	3.00000	11

```
1
 data ranging file
 ----------------
       11  name          dbou
       12  'set'                  0.100000d+01
       22  endata


1directional bound ranging
 ------------------------


 a. no change in the optimal solution
    finite range (tmaxa=   0.     d+00)

 b. no change in the optimal basis
    finite range (tmaxb=   0.23763d+02)

 section 1 - rows

  number  ...row..  .ll direction.  .ll boundary a  .ll boundary b  .ul direction.  .ul boundary a  .ul boundary b   ..i

       8  r1             0.             0.              0.              0.                  none            none    2
       9  r2             0.                 none            none        0.              0.              0.          3
      10  r3             0.             0.              0.              1.00000         21.30000        45.06334    4
      11  r4             0.                 none            none        0.                  none            none    5
      12  r5             0.             0.              0.              0.              0.              0.          6

 section 2 - columns

  number  .column.  .ll direction.  .ll boundary a  .ll boundary b  .ul direction.  .ul boundary a  .ul boundary b   m+j

       1  x1             1.00000            none            none        1.00000         1.50000         25.26334    7
       2  x2             1.00000        -1.40000        22.36334        1.00000             none            none    8
       3  x3             1.00000        0.              23.76334        1.00000         10.00000        33.76334    9
       4  x4             0.                 none            none        0.                  none            none   10
       5  x5             1.00000         2.00000        25.76334        1.00000         2.00000         25.76334   11

 at the boundary a the variable x2       hits 11

 at the boundary b the basic variable  r1       becomes nonbasic at 11 or optimal solution vanishes
1
 data ranging file
 ----------------
       23  name          drhs
       24  'set'         ####     0.100000d+01
       26  endata
```

1directional rhs ranging

finite range (tmax=0.90000d-01)

number	...row..	...direction..	.........rhs..	boundary value	..i
8	r1	1.00000	7.00000	7.09000	2
9	r2	1.00000	-7.00000	-6.91000	3
10	r3	1.00000	10.70000	10.79000	4
11	r4	0.	0.	0.	5
12	r5	1.00000	0.01000	0.10000	6

at the boundary variable x1 passes from the basis to ul or optimal solution vanishes

1cost ranging

number	.column.	.obj gradient.	..lower limit.	..upper limit.	.change at lower limit. or opt sol vanishes	.change at upper limit. or opt sol vanishes	m+j
1	x1	1.00000	-1.00000	none	x2		7
2	x2	-1.00000	none	1.00000		x2	8
3	x3	0.50000	none	2.90909		x3	9
4	x4	2.00000	0.	none	r3		10
5	x5	3.00000	3.00000	3.00000			11

1bound ranging

a. no solution change

section 1 - rows

number	...row..	.ll for l bound.	.ul for l bound.	.ll for u bound.	.ul for u bound.	..i
8	r1	none	7.12900	7.12900	none	2
9	r2	none	-5.97273	-5.97273	none	3
10	r3	variable at lower bound		0.	none	4
11	r4	none	-24.36818	-24.36818	none	5
12	r5	fixed variable				6

1 section 2 - columns

number	.column.	.ll for l bound.	.ul for l bound.	.ll for u bound.	.ul for u bound.	m+j
1	x1	none	1.41000	1.41000	none	7
2	x2	variable at lower bound		-1.40000	none	8
3	x3	variable at lower bound		0.	none	9
4	x4	none	4.86364	4.86364	none	10
5	x5	fixed variable				11

1 b. no basis change

section 1 - rows

number	...row..	.ll for l bound.	.ul for l bound.	.ll for u bound.	.ul for u bound	...changes at bdries... (or opt sol vanishes) lower		upper		..i
8	r1	none	7.12900	7.12900	none					2
9	r2	none	-5.97273	-5.97273	none					3
10	r3	none	21.30000	0.	none			r3	ul	4
11	r4	none	-24.36818	-24.36818	none					5
12	r5	fixed variable								6

1 section 2 - columns

number	.column.	.ll for l bound.	.ul for l bound.	.ll for u bound.	.ul for u bound	...changes at bdries... (or opt sol vanishes) lower		upper		m+j
1	x1	none	1.41000	1.41000	none					7
2	x2	-1.49000	69.88998	-1.40000	none	x1	ul	r1	ll	8
3	x3	-0.09000	10.00000	0.	none	x1	ul	x3	ul	9
4	x4	none	4.86364	4.86364	none					10
5	x5	fixed variable								11

1rhs ranging
----

number	...row..	.....rhs......	..lower limit.	..upper limit.	.change at lower limit. (or opt sol vanishes)		.change at upper limit. (or opt sol vanishes)		m+j
8	r1	7.00000	-0.12900	none	r1	ll			2
9	r2	-7.00000	none	-1.02727			r2	ul	3
10	r3	10.70000	-55.00000	none	r2	ul			4
11	r4	0.	none	none					5
12	r5	0.01000	none	0.10000			x1	ul	6

endrun

APPENDIX: MATHEMATICAL THEORY

This Appendix presents the elements of ranging theory necessary to solve the linear programming problem (4)–(6). For the sake of simplicity we shall assume that obj = $m+1$, i.e., the objective row is the last row in matrix A. As the value of variable $\tilde{x}_{n+1}$ is fixed at -1 we may remove it from the problem formulation, defining a new column vector of decision variables $y \in R^{n+m}$, where $y_i = \tilde{x}_i \ \forall i = 1,\ldots,n$ and $y_i = \tilde{x}_{i+1} \ \forall i = n+1,\ldots,n+m$. We also define an $m \times (n+m)$-matrix

$$A = \begin{bmatrix} a_1 & \\ \cdot & \\ \cdot & I \\ \cdot & \\ a_m & \end{bmatrix} ;$$

column vectors $b \in R^m$ (see (8)), $l, u \in R^{n+m}$, where $l_i = \tilde{l}_i$, $u_i = \tilde{u}_i \ \forall i = 1,\ldots,n$ and $l_i = h_{i+1}$, $u_i = k_{i+1} \ \forall i = n+1,\ldots,n+m$; and a row vector $c \in R_{n+m}$, where $c^i = a_0^i \ \forall i = 1,\ldots,n$ and $c^i = 0 \ \forall i = n+1,\ldots,n+m$.

The linear programming problem now takes the form: Minimize (or maximize) the linear cost function

$$F(y) = cy \tag{A.1}$$

subject to

$$Ay = b \tag{A.2}$$

$$l \le y \le u \quad . \tag{A.3}$$

We denote the optimal solution of this problem by z and decompose it into the following subvectors:

- z_B – basic vector
- z_l – vector of nonfixed, nonbasic variables which are at their lower bounds
- z_u – vector of nonfixed, nonbasic variables which are at their upper bounds
- z_s – vector of fixed variables (i.e., variables for which $u_i = l_i$).

Let I_u be the set of indices of all nonbasic variables at their upper bounds and let I_l be the set of indices of all nonbasic variables at their lower bounds. Fixed variables are not included in I_u or I_l. We shall let I_B denote the set of indices of all basic variables. This decomposition is also applied to the other vectors, yielding, for example, c_B, c_l, c_u; l_B, l_l, l_u; u_B, u_l, u_u. It is clear that $z_l = l_l$, $z_u = u_u$, $z_s = u_s$. Thus the constraint matrix A may be decomposed into the basic matrix B and matrices L, U, S such that

$$Bz_B + Lz_l + Uz_u + Sz_s = b \quad .$$

Hence we have

$$z_B = B^{-1}b - B^{-1}(Lz_l + Uz_u + Sz_s) \tag{A.4}$$

for the basic vector and

$$F(z) = c_B B^{-1}b + (c_l - c_B B^{-1}L)z_l + (c_u - c_B B^{-1}U)z_u + (c_s - c_B B^{-1}S)z_s \tag{A.5}$$

for the optimal cost.

Here and elsewhere we shall denote the i-th row of a matrix H by H_i and the j-th column by H^j. Define

$$D = B^{-1} \quad . \tag{A.6}$$

A1. Ranging of costs

Let Δc be a given nonzero row vector in R_{n+m}, where $\Delta c^i = 0$ for $i = n+1,\ldots,n+m$ and for fixed variables. We consider the family of linear programming problems (A.1)–(A.3) with the cost vector c replaced by $\bar{c}(t)$, where

$$\bar{c}(t) = c + t\Delta c \tag{A.7}$$

and t is a real number, $t \in R^1$. We wish to determine the largest range $[t_{\min}, t_{\max}]$ in which the coefficient t may vary without affecting the optimal solution, i.e., the range of t values for which the optimal solution is equal to z.

It is clear from (A.5) that the optimal solution remains unchanged and equal to z for all values of t such that

$$\varepsilon(\bar{c}_l(t) - \bar{c}_B(t)DL) \le 0 \tag{A.8}$$

and

$$\varepsilon(\bar{c}_u(t) - \bar{c}_B(t)DU) \geq 0 \quad , \tag{A.9}$$

where

$$\varepsilon = \begin{cases} +1 & \text{in the case of maximization} \\ -1 & \text{in the case of minimization} \end{cases}$$

Hence

$$t\,\varepsilon(\Delta c_l - \Delta c_B DL) \leq \varepsilon(c_B DL - c_l) \tag{A.10}$$

$$t\,\varepsilon(\Delta c_u - \Delta c_B DU) \geq \varepsilon(c_B DU - c_u)$$

We shall use the following notation:

$$T_j = -c^j + c_B DA^j \;, \quad \Delta T_j = -\Delta c^j + \Delta c_B DA^j \;, \quad j \in I_u \cup I_l \tag{A.11}$$

In the case of maximization we then have

$$t_{\max} = \min \{-T_j / \Delta T_j\} \quad , \tag{A.12}$$

where the minimum is taken over all values of j from I_l such that $\Delta T_j < 0$ and all values of j from I_u such that $\Delta T_j > 0$, and

$$t_{\min} = \max \{-T_j / \Delta T_j\} \quad , \tag{A.13}$$

where the maximum is taken over all values of j from I_l such that $\Delta T_j > 0$ and all values of j from I_u such that $\Delta T_j < 0$.

In the case of minimization $t_{\max}$ is determined from (A.12) but with the minimum taken over all values of j from I_l such that $\Delta T_j > 0$ and all values of j from I_u such that $\Delta T_j < 0$; $t_{\min}$ is determined from (A.13) with the maximum taken over all values of j from I_l such that $\Delta T_j < 0$ and all values of j from I_u such that $\Delta T_j > 0$.

In all cases, if the set of indices over which the maximum (or minimum) is taken is empty, then $t_{\min} = -\infty$ (or $t_{\max} = +\infty$).

From these general results it is not difficult to derive formulae for the cost ranging routines of POSTAN. Imposing the condition $t \geq 0$ and dropping the relations for $t_{\min}$, we obtain results that may be used for the directional cost ranging routine (DIRRAN). Setting $\Delta c = e_i$, where e_i is the i-th unit vector (which has all components equal to zero except for the i-th component, which is equal to one), we obtain formulae for the ordinary cost ranging routine (CRAN). In this case we formulate the results directly in terms of the cost component $\bar{c}^i = c^i + t$. For nonbasic components we have

$$\varepsilon\bar{c}^i \le \varepsilon c_B DA^i \quad \text{if } i \in I_l \tag{A.14}$$

$$\varepsilon\bar{c}^i \ge \varepsilon c_B DA^i \quad \text{if } i \in I_u \quad . \tag{A.15}$$

If $i \in I_B$, we have, by virtue of (A.11):

$$\Delta T_j = D_i A^j \tag{A.16}$$

and

$$c^i + t_{\min} \le \bar{c}^i \le c^i + t_{\max} \quad , \tag{A.17}$$

where $t_{\max}$ and $t_{\min}$ are determined from (A.12) and (A.13).

At each boundary of the interval $[t_{\min}, t_{\max}]$ a nonbasic variable changes its state. The number of this variable and the kind of change are determined by the component on the left-hand side of (A.8) or (A.9) that changes its sign at the boundary. If

$$\varepsilon(\bar{c}^i(t) - \bar{c}_B(t)DA^i) > 0 \quad \forall t > t_{\max} \tag{A.18}$$

for some $i \in I_l$, then at the upper boundary $t = t_{\max}$ the i-th variable passes from I_l to either I_B or I_u, or the optimal solution vanishes. If (A.18) holds but for all $t < t_{\min}$, then an equivalent statement may be made for the lower boundary $t_{\min}$.

If

$$\varepsilon(\bar{c}^i(t) - \bar{c}_B(t)DA^i) < 0 \quad \forall t > t_{\max} \tag{A.19}$$

for some $i \in I_u$, then at the upper boundary $t = t_{\max}$ the i-th variable passes from I_u to either I_B or I_l, or the optimal solution vanishes. If (A.19) holds but for all $t < t_{\min}$, then an equivalent statement may be made for the lower boundary $t_{\min}$.

A2. Ranging of right-hand sides

Let Δb be a given nonzero column vector in R^m. We consider the family of linear programming problems (A.1)–(A.3) with the rhs vector b replaced by $\bar{b}(t)$, where

$$\bar{b}(t) = b + t\Delta b \tag{A.20}$$

and $t \in R^1$. We wish to determine the largest range $[t_{min}, t_{max}]$ in which the coefficient t may vary without affecting the optimal basis, i.e., the range of t values for which the optimal basis is equal to B.

Letting $\bar{z}_B(t)$ denote the vector of basic variables in the optimal solution corresponding to the rhs vector $\bar{b}(t)$, we have

$$\bar{z}_B(t) = z_B + tB^{-1}\Delta b \quad . \tag{A.21}$$

It is clear that the nonbasic variables do not change for values of $t \in [t_{min}, t_{max}]$. The range $[t_{min}, t_{max}]$ is determined by the feasibility constraint on the basic variables:

$$l_B \leq \bar{z}_B(t) \leq u_B \tag{A.22}$$

or

$$l_B - z_B \leq tD\Delta b \leq u_B - z_B \quad . \tag{A.23}$$

Define

$$t_1 = \min_{j=1,\dots,m} \left\{ \frac{u_{Bj} - z_{Bj}}{D_j \Delta b} : D_j \Delta b > 0 \right\} \tag{A.24}$$

$$t_2 = \max_{j=1,\dots,m} \left\{ \frac{l_{Bj} - z_{Bj}}{D_j \Delta b} : D_j \Delta b > 0 \right\}$$

$$t_3 = \min_{j=1,\dots,m} \left\{ \frac{l_{Bj} - z_{Bj}}{D_j \Delta b} : D_j \Delta b < 0 \right\}$$

$$t_4 = \max_{j=1,\dots,m} \left\{ \frac{u_{Bj} - z_{Bj}}{D_j \Delta b} : D_j \Delta b < 0 \right\} \quad .$$

We then have

$$t_{max} = \min\{t_1, t_3\} \, , \quad t_{min} = \max\{t_2, t_4\} \quad . \tag{A.25}$$

If $D_i \Delta b \leq 0$ for all i, $i = 1,\ldots,m$, then we set $t_1 = +\infty$ and $t_2 = -\infty$. Similarly, if $D_i \Delta b \geq 0$ for all i, $i = 1,\ldots,m$, then we set $t_3 = -\infty$ and $t_4 = +\infty$.

To obtain results that may be used for the directional ranging routine (DRHSRN) it suffices to assume that $t \geq 0$ and to drop the relations for $t_{\min}$. To obtain formulae for the ordinary ranging routine (RHSRAN) we take $\Delta b = e_i$, where e_i is the i-th unit vector. We then have

$$D_j \Delta b = D_j^i \tag{A.26}$$

in (A.24).

At each boundary of the interval $[t_{\min}, t_{\max}]$ a basic variable changes its state or the optimal solution vanishes. The number j of the basic variable which becomes nonbasic at the upper boundary is determined by

$$t_{\max} = \frac{u_{Bj} - z_{Bj}}{D_j \Delta b} \quad \text{if } t_{\max} = t_1 \tag{A.27}$$

$$t_{\max} = \frac{l_{Bj} - z_{Bj}}{D_j \Delta b} \quad \text{if } t_{\max} = t_3 \tag{A.28}$$

In the first case the j-th basic variable reaches its upper bound, while in the second it passes to its lower bound. The number j of the basic variable which changes its state at the lower boundary $t = t_{\min}$ is determined by

$$t_{\min} = \frac{l_{Bj} - z_{Bj}}{D_j \Delta b} \quad \text{if } t_{\min} = t_2 \tag{A.29}$$

$$t_{\min} = \frac{u_{Bj} - z_{Bj}}{D_j \Delta b} \quad \text{if } t_{\min} = t_4 \tag{A.30}$$

In the first case the basic variable passes to its lower bound and in the second it reaches its upper bound.

A3. Ranging of bounds

Let col $(\Delta l, \Delta u)$ be a given column vector in $R^{2(n+m)}$, and be such that $\Delta l_i = \Delta u_i = 0$ if y_i is a fixed variable. We consider the family of linear programming problems (A.1)–(A.3) with the vectors of lower and upper bounds l and u replaced by $\bar{l}(t)$ and $\bar{u}(t)$, respectively, where

$$\bar{l}(t) = l + t\Delta l, \quad \bar{u}(t) = u + t\Delta u \tag{A.31}$$

and $t \in R^1$. We wish to determine two ranges, $[t_{\text{mina}}, t_{\text{maxa}}]$ and $[t_{\text{minb}}, t_{\text{maxb}}]$. The first of these intervals is the largest range in which the coefficient t may vary without affecting the optimal solution (i.e., the range of t values for which the optimal solution remains equal to z); the second is the largest range in which t may vary without affecting the optimal basis (i.e., the range of t values for which the optimal basis remains equal to B).

The boundaries $t_{\text{mina}}, t_{\text{maxa}}$ are easily determined from the following conditions: for every $t \in [t_{\text{mina}}, t_{\text{maxa}}]$

$$t\Delta l_i = 0 \text{ if } i \in I_l \tag{A.32}$$

$$t\Delta u_i = 0 \text{ if } i \in I_u$$

$$l_i + t\Delta l_i \le u_i \text{ if } i \in I_u$$

$$u_i + t\Delta u_i \ge l_i \text{ if } i \in I_l$$

$$l_i + t\Delta l_i \le z_i \le u_i + t\Delta u_i \text{ if } i \in I_B \ .$$

The first two conditions imply that $t_{\text{mina}} = t_{\text{maxa}} = 0$ if $\Delta l_i \ne 0$ for some $i \in I_l$ and/or $\Delta u_i \ne 0$ for some $i \in I_u$.

Let $\bar{z}(t) = z + t\Delta z$ denote the optimal solution corresponding to the vector of bounds col $(\bar{l}(t), \bar{u}(t))$. Then

$$\Delta z_l = \Delta l_l \ , \quad \Delta z_u = \Delta u_u \tag{A.33}$$

$$\Delta z_B = -D(L\Delta l_l + U\Delta u_u) \ .$$

The values of t_{minb} and t_{maxb} may be calculated using the feasibility conditions

$$l_l + t\Delta l_l \le u_l + t\Delta u_l \ , \quad l_u + t\Delta l_u \le u_u + t\Delta u_u \tag{A.34}$$

$$l_B + t\Delta l_B \le z_B + t\Delta z_B \le u_B + t\Delta u_B$$

or

$$t(\Delta l_l - \Delta u_l) \le u_l - l_l \tag{A.35}$$

$$t(\Delta l_u - \Delta u_u) \le u_u - l_u$$

$$t(\Delta l_B + DL\Delta l_l + DU\Delta u_u) \le z_B - l_B$$

$$t(\Delta u_B + DL\Delta l_l + DU\Delta u_u) \ge z_B - u_B$$

Define

$$t_1 = \min_{j \notin I_B} \left\{ \frac{u_j - l_j}{\Delta l_j - \Delta u_j} : \Delta l_j - \Delta u_j > 0 \right\} \tag{A.36}$$

$$t_2 = \max_{j \notin I_B} \left\{ \frac{u_j - l_j}{\Delta l_j - \Delta u_j} : \Delta l_j - \Delta u_j < 0 \right\}$$

$$t_3 = \min_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - l_{Bj}}{\Delta l_{Bj} + D_j(L\Delta l_l + U\Delta u_u)} : \text{denominator} > 0 \right\}$$

$$t_4 = \max_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - l_{Bj}}{\Delta l_{Bj} + D_j(L\Delta l_l + U\Delta u_u)} : \text{denominator} < 0 \right\}$$

$$t_5 = \min_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - u_{Bj}}{\Delta u_{Bj} + D_j(L\Delta l_l + U\Delta u_u)} : \text{denominator} < 0 \right\}$$

$$t_6 = \max_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - u_{Bj}}{\Delta u_{Bj} + D_j(L\Delta l_l + U\Delta u_u)} : \text{denominator} > 0 \right\}$$

Finally,

$$t_{\text{maxb}} = \min\{t_1, t_3, t_5\}, \quad t_{\text{minb}} = \max\{t_2, t_4, t_6\} \tag{A.37}$$

If the set of indices j over which a minimum or maximum is taken is empty, we substitute $+\infty$ for t_1, t_3, or t_5, and $-\infty$ for t_2, t_4, or t_6 in (A.36). For instance, if $\Delta l_j - \Delta u_j \leq 0$ for all $j \notin I_B$, we take $t_1 = +\infty$, and so on.

Results that may be used for the directional ranging routine (DBRAN) may be obtained by assuming that $t \geq 0$ and dropping the relations for t_{mina}, t_{minb}. To obtain the formulae for the ordinary ranging routine (BRAN) we take col $(\Delta l, \Delta u) = e_i$, where e_i is the i-th unit vector in $R^{2(n+m)}$. Expressions which allow us to determine the range $[t_{\text{minb}}, t_{\text{maxb}}]$ for all types of variables are given below.

For $i \in I_l$ we define

$$t_1 = u_i - l_i \tag{A.38}$$

$$t_3 = \min_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - l_{Bj}}{D_j A^i} : D_j A^i > 0 \right\}$$

$$t_4 = \max_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - l_{Bj}}{D_j A^i} : D_j A^i < 0 \right\}$$

$$t_5 = \min_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - u_{Bj}}{D_j A^i} : D_j A^i < 0 \right\}$$

$$t_6 = \max_{j=1,\ldots,m} \left\{ \frac{z_{Bj} - u_{Bj}}{D_j A^i} : D_j A^i > 0 \right\}$$

Hence

$$l_i + t_{\text{minb}} \leq \bar{l}_i \leq l_i + t_{\text{maxb}} \tag{A.39}$$

$$t_{\text{minb}} = \max\,\{t_4, t_6\}$$

$$t_{\text{maxb}} = \min\,\{t_1, t_3, t_5\}$$

and

$$\bar{u}_i \geq l_i \tag{A.40}$$

For $i \in I_u$ we define

$$t_1 = l_i - u_i$$

and t_3, t_4, t_5, t_6 are defined by (A.38). Then

$$u_i + t_{\text{minb}} \leq \bar{u}_i \leq u_i + t_{\text{maxb}} \tag{A.41}$$

$$t_{\text{minb}} = \max\,\{t_1, t_4, t_6\}$$

$$t_{\text{maxb}} = \min\,\{t_3, t_5\}$$

and

$$\bar{l}_i \leq u_i \tag{A.42}$$

If $i \in I_B$ then

$$\bar{l}_i \leq z_i\,, \quad \bar{u}_i \geq z_i \tag{A.43}$$

At each boundary of the interval $[t_{\text{minb}}, t_{\text{maxb}}]$ either a basic variable changes its state or the optimal solution vanishes. If either of the first two inequalities in (A.34) becomes an equality at one of the boundaries, then the feasible set becomes empty at this boundary and the optimal solution vanishes. Now assume that one of the last two inequalities in (A.34) becomes an equality. In this case either the optimal solution vanishes or a basic variable becomes nonbasic. Let $i \in I_B$. If $l_i + t_{\mathrm{minb}} \Delta l_i = z_i + t_{\mathrm{minb}} \Delta z_i$ and $\Delta l_i \neq \Delta z_i$, then at the lower boundary either the optimal solution vanishes or the i-th variable becomes nonbasic at its lower bound. Other cases may be analyzed in a similar way.

REFERENCE

1. B.A. Murtagh and M.A. Saunders. *MINOS – A Large-Scale Linear Programming System. User's Guide.* Technical Report SOL 77-9, Systems Optimization Laboratory, Stanford University, California, 1977.