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

PLP - A Package for Parametric Programming

Golebiowski, A.

IIASA Working Paper

December 1988

WORKING PAPER

PLP - A PACKAGE FOR PARAMETRIC PROGRAMMING

A. Golebiowski

December 1988
WP-88-118

IIASA
International Institute
for Applied Systems Analysis

PLP - A PACKAGE FOR PARAMETRIC PROGRAMMING

A. Golebiowski

December 1988
WP-88-118

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

This paper is one of the series of 11 Working Papers presenting the software for interactive decision support and software tools for developing decision support systems. These products constitute the outcome of the contracted study agreement between the System and Decision Sciences Program at IIASA and several Polish scientific institutions. The theoretical part of these results is presented in the IIASA Working Paper WP-88-071 entitled *Theory, Software and Testing Examples in Decision Support Systems* which contains the theoretical and methodological bacgrounds of the software systems developed within the project.

This paper presents the PLP package for parametric linear programming. This package constitutes the extension to MINOS, the well known linear and nonlinear programming code developed at Stanford University, and uses the MINOS as the solver of optimization problems. The PLP gives a complete parametric programming analysis for one, or more, of the following vectors: cost, rhs and bounds. In the same run several problems of this kind can be solved and for each, the starting point may be the original optimal solution obtained in the last problem. This property makes the PLP especially interesting for multiple criteria analysis using the reference point approach.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

PLP - A PACKAGE FOR PARAMETRIC PROGRAMMING

Adam Golebiowski

Institute of Automatics, Academy of Mining and Metallurgy,
Cracow, Poland.

CONTENTS

INTRODUCTION

PLP is a software package for parametric linear programming. It is an extension of MINOS, the well-known linear and nonlinear programming code developed by Saunders and Murtagh *. PLP is initiated by adding some specifications to the original list of MINOS specifications.

The package PLP uses MINOS as the solver of optimization problems. It includes sections which create an iterative framework for parametric programming and perform ranging and housekeeping procedures.

The formulation of the linear problem analyzed by PLP is similar as for MINOS.

Optionally, PLP gives a complete parametric programming analysis for one, or more, of the following vectors: cost, rhs and bounds. Of course such analysis can also be performed for single elements of these vectors. In the same run, several problems of this kind can be solved and for each, the starting point may be the original optimal solution or the final solution obtained in the last problem.

The last current complete solution in MINOS format is printed or stored with frequency specified by the user. Additionally, the user specifies the frequency of printing of a short message about current changes of optimal basis.

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

A. THEORETICAL GUIDE

1. GENERAL INFORMATION.

As options of PLP can be expressed in terms of the internal formulation of the linear problem used by MINOS we shall begin with recalling this concept.

The external formulation (supplied by the user) of the linear problem to be solved by MINOS is: Minimize (or maximize) a linear cost function

$$F(x) = a_0 x \tag{1}$$

subject to m row constraints:

$$d_i \le a_i x \le g_i\ , \quad i = 1,\dots,m \tag{2}$$

and n constraints on separate variables:

$$d_{m+i} \le x_i \le g_{m+i}\ , \quad i = 1,\dots,n \quad . \tag{3}$$

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, $i = 1,\dots,m$, are n-dimensional row vectors, the lower bounds d_i, $i = 1,\dots,m+n$, are real numbers or $-\infty$, and the upper bounds g_i, $i = 1,\dots,m+n$, 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

$$\tilde{a}_i = a_i \quad i < \text{obj} \ , \quad \tilde{a}_{\text{obj}} = a_0 \ , \quad \tilde{a}_i = a_{i-1} \quad i > \text{obj} \ , \tag{8}$$
$$\tilde{b}_i = b_i \quad i < \text{obj} \ , \quad \tilde{b}_{\text{obj}} = 0 \ , \quad \tilde{b}_i = b_{i-1} \quad i > \text{obj} \ ,$$

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:

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

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.

2. MATHEMATICAL THEORY

This section presents elements of ranging theory for the linear programming problem (4)-(6). Some nonconventional notation will be used in order to avoid discussion of many particular cases. The sign $\leq$ will denote "less than or equal to" if the expressions on its both sides are finite and "less than" otherwise. Similarly, $\geq$ will denote "greater than or equal to" or "greater than". The notation $[t_1,t_2]$ will be used for the closure of the open interval (t_1,t_2); that is, t_1 and/or t_2 do not belong to the interval if they are not finite. For the sake of simplicity we shall assume that obj $= m+1$, i.e., the objective row is the last row in matrix $\tilde{A}$. 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 $y_i = \tilde{x}_i$ $i = n+1,\dots,n+m$. We also define an $m \, x \, (n+m)$-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 $l_i = \tilde{l}_i$, $u_i = \tilde{u}_i$ $i = 1,\dots,n$ and $l_i = h_{i+1}$, $u_i = k_{i+1}$ $i = n+1,\dots,n+m$; and a row vector $c \in R_{n+m}$, where $c_i = a_0^i$

$i = 1,...,n$ and $c^i = 0 \;\; i = n+1,...,n+m$.

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

$$F(y) = cy \tag{12}$$

subject to:

$$Ay = b \tag{13}$$

$$l \leq y \leq u \quad . \tag{14}$$

We denote the optimal solution of this problem by z and decompose it in the obvious way 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 is 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

$$z_B = B^{-1}b \; - \; B^{-1}(Lz_l + Uz_u + Sz_s) \tag{15}$$

for the basic vector and

$$F(z) = c_B B^{-1}b + (c_l - c_B B^{-1}L)z_l + (c_u - c_B B^{-1}U)z_u + (c_s - c_B B^{-1}S)z_s \quad . \tag{16}$$

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

$$D = B^{-1} \quad . \tag{17}$$

2.1. Parametric analysis of cost.

In every iteration of PLP COST the ranging problem has to be solved in the first place. Let Δc be a given nonzero row vector in R_{n+m}, where $\Delta c^i = 0$ for $i = n+1,...,n+m$ and for fixed variables. We consider programming problems (12)–(13) with the cost vector c replaced by $\bar{c}(t)$, where

$$\bar{c}(t) = c + t\Delta c \tag{18}$$

and t is a real number, $t \in R^1$. We wish to determine the largest range $[0, t_{\max}]$ 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 (16) that the optimal solution remains unchanged and equal to z for all values of t such that

$$\epsilon(\; \bar{c}_l(t) - \bar{c}_B(t)DL) \leq 0 \tag{19}$$

and

$$\epsilon(\ \bar{c}_u(t) - \bar{c}_B(t)DU) \geq 0 \quad , \tag{20}$$

where

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

Hence

$$t\epsilon(\Delta c_l - \Delta c_B DL) \leq \epsilon(c_B DL - c_l) \tag{21}$$

$$t\epsilon(\Delta c_u - \Delta c_B DU) \geq \epsilon(c_B DU - c_u) \quad .$$

We shall use the following notation:

$$T_j = -c^j + c_B DA^j \ , \quad \Delta T_j = -\Delta c^j + \Delta c_B DA^j \ , \quad j \in I_u \cup I_l \quad . \tag{22}$$

In the case of maximization we then have

$$t_{\max} = \min \{-T_j/\Delta T_j\} \quad , \tag{23}$$

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 $\Delta T_j > 0$.

In the case of minimization $t_{\max}$ is determined from (23) 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 $\Delta T_j < 0$.

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

If $t_{\max}$ is finite, two situations are possible: either the optimal solution vanishes for all $t > t_{\max}$ or a new optimal solution exists for some $t > t_{\max}$. This change of the optimal solution is determined by MINOS in the following way.

A shifted value of the cost vector is determined

$$c(t') = c + (t_{\max} + \Delta')\Delta c \tag{24}$$

Δ' is an appropriately chosen increment (see below). For this cost vector, MINOS finds the corresponding optimal solution. Next, the value of the cost vector and optimal cost at $t = t_{\max}$ are calculated

$$\bar{c}(t_{\max}) = \bar{c}(t') - \Delta'\Delta c \tag{25}$$

$$F(t_{\max}) = \bar{c}(t_{\max})z \tag{26}$$

where z is the right-hand limit of the optimal solution for $t = t_{\max}$ and Δ' is computed from:

$$\Delta' = DELTA * x \tag{27}$$

where DELTA is given by the user in the keyword PLP INCREMENT and x is the greatest real for which the following inequality is satisfied

$$-x\,\Delta T_i \leq f(x) \tag{28}$$

where

$$f(x) = \begin{cases} TOLD * \|\ (\ \bar{c}_B t_{\max} + \Delta\bar{c}_B x)B^{-1}\ \|\ , & \|\ (\ \bar{c}_B t_{\max} + \Delta\bar{c}_B x)B^{-}\ \| > 1 \\ TOLD & , \text{ otherwise} \end{cases} \tag{29}$$

This inequality is solved for all values of the subscript i which belong to the set $I_{\sum}$ (see section 3.1 point 3), yielding a sequence $\{x_i\}$. The maximum value of elements x_i is used in formula (27).

2.2. Parametric analysis of rhs.

In every iteration of PLP RHS the ranging problem has to be solved in the first place. Let Δb be a given nonzero column vector in R^m. We consider the family of linear programming problems (12)–(14) with the rhs vector b replaced by $\bar{b}(t)$, where

$$\bar{b}(t) = b + t\Delta b \tag{30}$$

and $t \in R^1$. We wish to determine the largest range $[0,t_{\max}]$ 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

$$\bar{z}_B(t) = z_B + tB^{-1}\Delta b \quad . \tag{31}$$

It is clear that the nonbasic variables do not change for values of $t \in [0,t_{\max}]$. The range $[0,t_{\max}]$ is determined by the feasibility constraint on the basic variables:

$$l_B \le \bar{z}_B(t) \le u_B \tag{32}$$

or

$$l_B - z_B \le tD\Delta b \le u_B - z_B \quad . \tag{33}$$

Define

$$t_1 = \min_{j=1,\dots,m} \left\{ \frac{u_{Bj} - z_{Bj}}{D_j\Delta b} : D_j\Delta b > 0 \right\} \tag{34}$$

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

We then have

$$t_{\max} = \min\,\{t_1,t_2\} \tag{34}$$

If $D_i\Delta b \le 0$ for all i, $i = 1,\dots,m$, then we set $t_1 = +\infty$. Similarly, if $D_i\Delta b \ge 0$ for all i, $i = 1,\dots,m$, then we set $t_2 = -\infty$.

If $t_{\max}$ is finite, two situations are possible: either the optimal solution vanishes for $t >$ tmax or a new optimal solution exists for some $t > t_{\max}$. This change of optimal solution is determined by MINOS in the following way.

A shifted rhs vector is determined

$$\bar{b}(t') = b + (t_{\max} + \Delta')\Delta b \tag{35}$$

Δ' is an appropriately chosen increment (see below). For this rhs vector, MINOS finds the corresponding optimal solution. Next, the value of the rhs vector, the basic vector and the optimal cost at $t = t_{\max}$ are calculated

$$\bar{b}(t_{\max}) = \bar{b}(t') - \Delta'\Delta b \tag{36}$$

$$z_B = B^{-1}\bar{b}(t_{\max}) - B^{-1}(Lz_l + Uz_u + Sz_s) \tag{37}$$

$$F(z) = c_B B^{-1}\,\bar{b}(t_{\max}) + (c_l - c_B B^{-1}L)z_l + (c_u - c_B B^{-1}U)z_u \tag{38}$$
$$+ (c_s - c_B B^{-1}S)z_s$$

where; z_l, z_u, z_s and z_B are the decomposition of the right hand side limit of the optimal solution for $t = t_{\max}$. The matrices B, L, U, S are the decomposition of constraint matrix A valid for the optimal solution for $t = t_{\max}$.
Δ' is computed from

$$\Delta' = DELTA \, ^* \, x \tag{39}$$

where $DELTA$ is given by the user in the keyword PLP INCREMENT and x is the greatest real for which the following inequality is satisfied

$$x \mid D_{j\max} \Delta b \mid \leq TOLX \tag{40}$$

where $j_{\max}$ is the subscript for which $t_{\max}$ is calculated in formula (34).

2.3. Ranging of bounds.

In every iteration of PLP BOUND the ranging problem has to be solved in the first place. 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

$$\bar{l}(t) = l + t\Delta l, \qquad \bar{u}(t) = u + t\Delta u \tag{41}$$

and $t \in R^1$. We wish to determine two ranges, $[0, t_{\text{maxa}}]$ and $[0, t_{\text{maxb}}]$. 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 boundary t_{maxa} is easily determined from the following conditions: for every $t \in [0, t_{\text{maxa}}]$

$$
\begin{aligned}
& t\Delta l_i = 0 \text{ if } i \in I_l \\
& t\Delta u_i = 0 \text{ if } i \in I_u \\
& l_i + t\Delta l_i \leq u_i \text{ if } i \in I_u \\
& u_i + t\Delta u_i \geq l_i \text{ if } i \in I_l \\
& l_i + t\Delta l_i \leq z_i \leq u_i + t\Delta u_i \text{ if } i \in I_B.
\end{aligned} \tag{42}
$$

The first two conditions imply that $t_{\text{maxa}} = 0$ if Δl_i 0 for some $i \in I_l$ and/or Δu_i 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

$$
\begin{aligned}
& \Delta z_l = \Delta l_l \, , \; \Delta z_u = \Delta u_u \\
& \Delta z_B = -D(L\Delta l_l + U\Delta u_u)
\end{aligned} \tag{43}
$$

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

$$
\begin{aligned}
& l_l + t\Delta l_l \leq u_l + t\Delta u_l \, , \; l_u + t\Delta l_u \leq u_u + t\Delta u_u \\
& l_B + t\Delta l_B \leq z_B + t\Delta z_B \leq u_B + t\Delta u_B
\end{aligned} \tag{44}
$$

or

$$
\begin{aligned}
& t(\Delta l_l - \Delta u_l) \leq u_l - l_l \\
& t(\Delta l_u - \Delta u_u) \leq u_u - l_u
\end{aligned} \tag{45}
$$

$$t(\Delta l_B + DL\Delta l_l + DU\Delta u_U) \leq z_B - l_B$$

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

Define

$$t_1 = \min_{j \in B} \left\{ \frac{u_j - l_j}{\Delta l_j - \Delta u_j} : \Delta l_j - \Delta u_j > 0 \right\} \tag{46}$$

$$t_2 = \min_{j=1,\ldots,m} \left\{ \frac{z_{B_j} - l_{B_j}}{\Delta u_{B_j} + D_j(L\Delta l_l + U\Delta u_u)} : \text{denominator} < 0 \right\}$$

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

Finally,

$$t_{\text{maxb}} = \min\{t_1, t_2, t_3\} \tag{47}$$

If the set of indices j over which a minimum is taken is empty, we substitute $+\infty$ for t_1 , t_2 , or t_3 in (46). For instance, if $\Delta l_j - \Delta u_j \leq 0$ for all $j \in_B$, we take $t_1 = +\infty$, and so on.

If $t_{\max}$ is finite, two situations are possible: either the optimal solution vanishes for $t > t_{\text{maxb}}$ or a new optimal solution exists for some $t > t_{\text{maxb}}$. This change of optimal solution is determined by MINOS in the following way.

The shifted vectors of lower and upper bounds are determined

$$\bar{l}(t') = l + (t_{\text{maxb}} + \Delta')\Delta l \tag{48}$$

$$\bar{u}(t') = u + (tsub\text{maxb} + \Delta')\Delta u$$

where Δ' is an appropriately chosen increment (see below). For these bound vectors, MINOS finds the corresponding optimal solution. Next, the values of the bound vectors, the basic vector and the optimal cost at $t = t\text{maxb}$ are calculated

$$\bar{l}(t_{\text{maxb}}) = \bar{l}(t') - \Delta'\Delta l \tag{49}$$

$$\bar{u}(t_{\text{maxb}}) = \bar{u}(t') - \Delta'\Delta u$$

$$z_B = B^{-1}b - B^{-1}(Lz_l(t_{\text{maxb}}) + Uz_u(t_{\text{maxb}}) + Sz_s) \tag{50}$$

$$F(z) = c_B B^{-1} b + (c_l - c_B B^{-1} L) z_l(t_{\text{maxb}}) + (c_u - c_B B^{-1} U) z_u(t_{\text{maxb}}) + (c_s - c_B B^{-1} S) z_s \tag{51}$$

where z_l, z_u, z_s and z_B are the decomposition of the right-hand side limit of the optimal solution for $t = t_{\text{maxb}}$. The matrices B, L, U, S are the decomposition of the constraint matrix A valid for the optimal solution $t = t_{\text{maxb}}$ and Δ' is computed from:

$$\Delta' = DELTA * x \tag{52}$$

where $DELTA$ is given by the user in the keyword PLP INCREMENT and

$$x = \frac{TOLX}{|f|} \tag{53}$$

f is the denominator of that fraction in the two last definitions (46) which is equal to t_{maxb}.

3. THE METHODS.

3.1. The method of PLP COST.

The algorithm of PLP COST is as follows:

1. Set $i := 0$, $t_i := 0$.
2. MINOS finds the optimal solution for t_i with the basic matrix B and the basic vector z_B.
3. The boundary value of the parameter t_{i+1} is calculated, such that for all $a_0(t)$, $t \in [t_i, t_{i+1})$ the optimal solution is constant. The set I_∞ of nonbasic variables is determined, containing all nonbasic variables for which reduced costs:

$$a_0{}^k(t) - \tilde{a}_B(t)B^{-1}\tilde{A}^k \tag{54}$$

where $\tilde{A}_k$ is the k-th column of the constant matrix $\tilde{A}$ (see (7)), reach zero for some t in the interval $[t_i, t_i + 10^{-9})$. These variables are nonbasic in the decomposition valid for $t = t_i$.

4. Next, the value t' of the parameter is determined:

$$t' = t_{i+1} + \Delta', \Delta' = DELTA{*}\Delta, DELTA > 1 \tag{55}$$

where Δ is the greatest increment of the parameter such that for $t = t_{i+1} + \Delta$ the nonbasic variable whose reduced cost reaches zero at t_{i+1} is still recognized by MINOS as optimal.

5. New cost vector is computed:

$$a_0(t') = a_0(t_i) + (t' - t_i)\Delta a_0 \tag{56}$$

6. MINOS finds the new optimal solution for the new cost vector $a_0(t')$.
7. Set $t_i := t_{i+1}$ and shift the cost vector back to t_i

$$a_0(t_i) := a_0(t') - \Delta'\Delta a_0 \tag{57}$$

8. Set $i := i+1$ and go to 3.

3.2. The method of PLP RHS.

The algorithm of PLP RHS is as follows:

1. Set $i := 0$ and $t_i := 0$.
2. MINOS finds the optimal solution for t_i with the basic matrix B and basic vector z_B. At the same time it finds the optimal decomposition into basic and nonbasic variables.
3. The boundary value of the parameter t_{i+1} is calculated (see section 2.2), such that for all $\tilde{b}(t)$, $t \in [t_i, t_{i+1})$ the optimal basis (basic matrix) is constant and equal to B. The set I_Σ of the basic variables is determined containing all basic variables which reach their bounds for some value of t in the interval $[t_{i+1}, t_{i+1} + 10^{-9}]$. These variables are basic in the decomposition valid for $t = t_i$.
4. Next, the value t' of the parameter is determined

$$t' = t_{i+1} + \Delta' \ , \ \Delta' = DELTA{*}\Delta, DELTA > 1 \tag{58}$$

where Δ is the greatest increment of the parameter such that for $t = t_{i+1} + \Delta$ the basic variable which reaches its bound at t_{i+1} is still recognized by MINOS as feasible.

5. New rhs vector is computed

$$\tilde{b}(t') = \tilde{b}(t_i) + (t' - t_i)\Delta\tilde{b} \tag{59}$$

and the corresponding starting basic solution

$$z_B(t') = z_B(t_i) + (t' - t_i)B^{-1}\Delta\tilde{b} \tag{60}$$

6. MINOS finds the optimal solution for the new rhs vector $\tilde{b}(t')$, starting from the shifted basic solution (60) which is infeasible. The new optimal basis is denoted by B and the new basic vector by $z_B(t')$

7. Set $t_i := t_{i+1}$ and shift the solution back to t_i,

$$z_B(t_i) = z_B(t') - (t' - t_i)B^{-1}\Delta\tilde{b} \tag{61}$$

also

$$\tilde{b}(t_i) = b(t') - (t' - t_i)\Delta\tilde{b} \tag{62}$$

8. Set $i := i + 1$ and go to (3).

3.3. The method of PLP BOUND

The algorithm of PLP BOUND is as follows:

1. Set $i := 0$ and $t_i := 0$

2. MINOS finds the optimal solution for t_i with the basic matrix B and the basic vector z_B. At the same time it finds the optimal decomposition into the basic and nonbasic variables.

3. The boundary value of the parameter t_{i+1} is calculated (see section 2.3), such that for all $\tilde{l}(t)$ and $\tilde{u}(t)$, $t \in [t_i, t_{i+1})$ the optimal basis (basic matrix) is constant and equal to B. The set $I_{\sum}$ of basic variables is determined, containing all basic variables which reach their bounds for some value of t in the interval $[t_{i+1}, t_{i+1} + 10^{-9}]$. These variables are basic in the decomposition valid for $t = t_i$.

4. Next, the value t' of the parameter is determined

$$t' = t_{i+1} + \Delta' \ , \ \Delta' = DELTA^*\Delta, \ DELTA > 1 \tag{63}$$

where Δ is the greatest increment of the parameter such that for $t = t_{i+1} + \Delta$ the basic variable which reaches its bound at t_{i+1} is still recognized by MINOS as feasible.

5. New bound vectors are computed:

$$\tilde{l}(t') = \tilde{l}(t_i) + (t' - t_i)\Delta\tilde{l} \tag{64}$$

$$\tilde{u}(t') = \tilde{u}(t_i) + (t' - t_i)\Delta\tilde{u}$$

and the corresponding starting basic solution:

$$z_B(t') = z_B(t_i) + (t' - t_i)B^{-1}(L\Delta\tilde{l} + U\Delta\tilde{u}). \tag{65}$$

MINOS finds the optimal solution for the new bound vectors, starting from the shifted basic vector (65) (which is infeasible). The optimal basis is denoted by B and the basic vector by $z_B(t')$.

7. Set $t_i := t_{i+1}$ and shift the solution back to t_i

$$z_B(t_i) = z_B(t') - (t'-t)B^{-1}(L\Delta\tilde{l} + U\Delta\tilde{u}). \tag{66}$$

8. Set $i := i+1$ and go to 3.

B. USER MANUAL

1. BRIEF CHARACTERIZATION OF BASIC FUNCTIONS OF PLP.

1.1. Parametric analysis of cost (PLP COST).

The cost vector $a_0 = (a_0^1, a_0^2, \ldots, a_0^n)$ (see (1)) is changed along a direction given by the user, $\Delta a_0 = (\Delta a_0^1, \Delta a_0^2, \ldots, \Delta a_0^n)$ according to the formula:

$$a_0(t) = a_0(0) + t\Delta a_0 \,,\; t \geq 0 \qquad (67)$$

where $a_o(0)$ is the starting value of cost. If the 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.

PLP determines a sequence of values of the parameter denoted by $t_0, t_1, \ldots, t_k$, such that $0 = t_0 < t_1 < t_2 < \cdots < t_k$ and in each of the intervals $[t_i, t_{i+1}), i = 0, \ldots, k-1$ the optimal solution is constant and in each case the optimal basis is different. The integer k : (1) may be defined by the user as the maximum number of iterations, (2) may be determined by the condition that the optimal solution is constant for every $t \geq t_k$ and different from that in $[\, t_{k-1}, t_k)$, (3) may be determined by the condition that there are no optimal solutions for every $t > t_k$.

1.2. Parametric analysis of rhs (PLP RHS).

The right-hand side $\tilde{b} = \mathrm{col}\,(\tilde{b}_1, \ldots, \tilde{b}_{m+1})$ $\tilde{b} = \mathrm{col}\,(\tilde{b}_1, \ldots, \tilde{b}_{m\pm 1})$ (see (7) and (8)) is changed along a direction given by the user, $\Delta\tilde{b} = \mathrm{col}(\Delta\tilde{b}_1, \ldots, \Delta\tilde{b}_{m+1})$, according to the formula:

$$\tilde{b}(t) = \tilde{b}(0) + t\Delta\tilde{b} \;\;,\; t \geq 0 \qquad (68)$$

where $\tilde{b}(0)$ is the starting value of rhs. The component of $\Delta\tilde{b}$ which corresponds to the objective row is automatically set to zero, $\Delta\tilde{b}_{obj} = 0$.

PLP determines a sequence of values of the parameter denoted by $t_0, t_1, \ldots, t_k$ such that $0 = t_0 < t_1 < t_2 < \cdots < t_k$ and in each of the intervals $[t_i, t_{i+1}), i = 0, \ldots, k-1$ the optimal basis is constant and in each case different. The integer k : (1) may be defined by the user as the maximum number of iterations, (2) may be determined by the condition that the optimal basis is constant for every $t \geq t_k$ and different from that in $[t_{k-1}, t_k)$, (3) may be determined by the condition that there are no feasible solutions for every $t > t_k$.

1.3. Parametric analysis of bounds (PLP BOUND).

The vector of bounds $\mathrm{col}(\tilde{l}, \tilde{u}) \in R^{2(n+m+2)}$ (see (9)) is changed along a direction given by the user, $\mathrm{col}(\Delta\tilde{l}, \Delta\tilde{u})$, according to the formula:

$$\mathrm{col}(\tilde{l}(t), \tilde{u}(t)) = \mathrm{col}(\tilde{l}(0), \tilde{u}(0)) + t\,\mathrm{col}(\Delta\tilde{l}, \Delta\tilde{u}) \;,\; t \geq 0 \qquad (69)$$

where $\mathrm{col}(\tilde{l}(0), \tilde{u}(t))$ is the starting value of bounds. 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.

If there is no lower and/or upper bound for the i-th variable $\tilde{x}_i$ (see (6)) the corresponding increment $\Delta\tilde{l}_i$ and/or $\Delta\tilde{u}_i$, respectively, is also automatically set to zero.

PLP determines a sequence of values of the parameter denoted by $t_o, t_1, \ldots, t_k$ such that $0 = t_0 < t_1 < t_2 < \cdots < t_k$ and in each of the intervals $[t_i, t_{i+1})$, $i = 0, \ldots, k-1$ the optimal basis is constant and in each case different. The integer k : (1) may be defined by the user as the maximum number of iterations, (2) may be determined by the condition that the optimal basis is constant for every $t \geq t_k$ and different from that in $[t_{k-1}, t_k)$, (3) may be determined by the condition that there are no feasible solutions for every $t > t_k$.

Each interval $[t_i, t_{i+1}]$ is optionally divided into two subintervals $[t_i, t_i^a]$, $[t_i^a, t_{i+1}]$. The interval $[t_i, t_i^a]$ is the maximum interval where the optimal solution remains constant and not only the optimal basis. It often happens that $t_i = t_i^a$.

1.4. Dependent and independent work.

All three kinds of analysis can be performed in one run. The starting point for the next kind of analysis may be either the original starting optimal solution (The Independent Work) or the last optimal solution obtained in the preceding analysis (The Dependent Work). The continuation is impossible if the optimal solution vanishes.

1.5. Controlling output.

In each of the three kinds of analysis the following information is available. The user has to specify the frequency of printing the complete current optimal solution in MINOS format. This means that the complete printout is given for the values of parameters t equal to $t_{0+}, t_{p+}, t_{2p+}, \ldots,$ and $t_{(k-1)+}$ or t_{k+} depending on whether the optimal solution exists for $t > t_k$. The notation t_{i+} means that we take the right-hand limit of the optimal solution at t_i. The user specifies frequency of printing the so called PLP ITERATION LOG. This is a short message containing most important information about current change of optimal solution (value of the parameter t, change of optimal basis, current value of objective function).

1.6. Tolerances.

The performance of PLP is strongly affected by the choice of tolerances. Especially important are two tolerances determined in MINOS : the tolerance of optimality (TOLD) and the tolerance of feasibility (TOLX). In the proper procedures of the PLP the following general rule is adopted. All quantities greater than or equal to 1.E+15 are taken as equal to infinity and all quantities whose absolute value is less than 1.E-9 are regarded as equal to zero.

2. INPUT

The input contains all necessary elements for MINOS with the conditions given below.

2.1. New key-words in the SPECS file

Key	Default	Meaning
PLP COST ANALYSIS n	off	This keyword activates the parametric analysis of cost. The integer n is the number of iterations to be performed. If no value or a zero value of n is given, all iterations will be performed (until the optimal solution becomes constant or the optimal solution vanishes).
PLP RHS ANALYSIS n	off	This keyword activates the parametric analysis of rhs. The integer n is the number of iterations to be performed. If no value or a zero value of n is given, all iterations will be performed (until the optimal basis becomes constant or the optimal solution vanishes).

PLP BOUND ANALYSIS n	off	This keyword activates the parametric analysis of bounds. The absolute value of integer n is the number of iterations to be performed. If n is less than zero an additional output is printed in each iteration which gives the values of $t_i^a - t_i$ and $t_{i+1} - t_i$ and the corresponding boundary values of bounds.
PLP ORDER	off	This keyword activates the dependent work of PLP. If it does not occur, PLP performs each of the required kinds of analysis only once (keywords PLP COST..., PLP RHS..., PLP BOUND...). In each of these, the starting point is the original optimal solution. If PLP ORDER appears in the SPECS FILE, it must precede the sequence of keywords PLP COST..., PLP RHS..., PLP BOUND..., which define the kinds of analysis to be performed in the same order. For each kind of analysis, the starting point is the last optimal solution obtained in the last analysis. If the optimal solution vanishes, the run stops. Each kind of analysis can be performed up to five times, in an arbitrary order (determined by the sequence of keywords PLP COST..., PLP RHS..., PLP BOUND...). In each repetition of the same kind of analysis, the search direction and the maximum number of iterations must be the same. The value of n given in the last keyword referring to a particular kind of analysis is valid for all its repetitions.
PLP SOLN n	n = 1	This keyword specifies the frequency of printing the current complete solution in the MINOS format. Full solution is printed after every n iterations. If this keyword is omitted or $n = 0$, the effect is the same as for $n = 1$.
PLP FREQUENCY n	n = 1	This command activates the frequency of printing the short message called PLP ITERATION LOG (see section 3 of USER MANUAL). A PLP ITERATION LOG is printed after every n iterations. If this keyword is omitted or $n = 0$, the effect is the same as for $n = 1$.
PLP SOLU-TION n	off	If this (optional) keyword is used with $n \geq 0$ complete outputs of optimal solution will be stored in file n with the frequency given in PLP SOLN m. If n=0 or this keyword does not occur, the complete outputs are stored in the printer file.
PLP FILE n	n = 5	The absolute value of n is the logical number of the data file for parametric programming. This file is read after processing other MINOS files has been completed. The parameter n also controls the output of the search directions. If n is less than zero, the search direction of each PLP analysis is printed. These directions are not printed for any other entry.

PRINT DATA PLP FILE	off	If this keyword is used, the whole DATA PLP FILE will be printed in the output. Otherwise, only the records with comments and the records NAME, SET and ENDATA are printed.
PLP INCREMENT d	d = 1.1	This keyword specifies the value of factor Δ (see sections 3.1, 3.2 and 3.3 of THEORETICAL GUIDE). (4.2), (4.3)).

2.2. DATA PLP file – input format.

The data for the PLP procedures are prepared in an MPS-like format and placed in the file specified by the key-word DATA PLP FILE n. The data sets for different PLP 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 parametric programming procedure.

Parametric analysis of cost (PLP COST)

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

Parametric analysis of rhs (PLP RHS)

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

Parametric analysis of bounds (PLP BOUNDS)

	f1	f2	f3	f4	f5	f6
1.	NAME		PLPB			
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.

The following general rules apply to all data sets:

- One of the fields f3, f5, (f4, f6) 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 values should be written as real numbers in a format accepted by FORTRAN.

2.3. Specification of zeros in the MPS file.

In two kinds of parametric analysis, PLP COST and PLP RHS the user has to specify explicitly some of the zero values of the objective row elements (vector a_0) and/or the rhs column elements (vector $\tilde{b}$), exactly in the same way as the nonzero values specified in the data (MPS file). This refers to those elements of the vector a_0 and/or $\tilde{b}$ for which the corresponding elements of Δa_0 and/or $\Delta \tilde{b}$, respectively, are different from zero.

Example $a_0 = (1.,0.,0.,3.,5.)$, $\Delta a_0 = (-1.,0.,0.1,1.,0.)$
In this case the element a_0^3 has to be explicitly specified in MPS

x3 obj 0.

where x_3 is the name of the third column (structural variable) and obj is the name of the objective row.

3. OUTPUT

The title of the output of PLP is:

P L P VERSION 1.0 JUNE 1986

In the case of dependent work of PLP the subtitle is printed:

DEPENDENT WORK OF PLP

Otherwise, this subtitle is omitted.

The output may be sent either to the printer file or to the file defined by the keyword PLP SOLUTION FILE. Only the output produced by the procedure SOLN of MINOS can be stored in the latter one.

Since the SOLN output is described in MINOS manuals we will confine ourselves to the output of PLP sent to the printer file, and so we will also skip the messages given by MINOS.

Each kind of parametric analysis procedures produces a printout containing the following information.

Title:

PLP COST - for parametric analysis of cost

PLP RHS - for parametric analysis of rhs

PLP BOUND - for parametric analysis of bounds

Search direction (optionally):

For PLP COST it has the following format. For each structural variable $\tilde{x}_i$, $i = 1,\ldots,n$ the following information is given:

NUMBER - Number of structural variable

COLUMN - Name of structural variable

DIRECTION - Increment component Δa_0^i

OBJ GRADIENT- Cost component a_0^i

M+J - $m + 1 + i$

In the case of PLP RHS 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 increment vector

RHS - Right-hand-side component $\tilde{b}_i$

I - Row number

For PLP BOUND this printout is divided into two sections:

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 inc. vector $\Delta\tilde{l}$

LOWER LIMIT- Lower bound $\tilde{l}_i$

UL DIRECTION- Component $\Delta\tilde{u}_i$ of the upper bound inc. vector $\Delta\tilde{u}$

UPPER LIMIT- Upper bound $\tilde{u}_i$

I - Row number

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

NUMBER - Number of structural variable

COLUMN - Name of structural variable

M+J - $m+1+i$

PLP iteration log printing:

Printing frequency is given in the keyword PLP FREQUENCY. It takes one of the following forms:

If only one variable in the optimal basis has been exchanged and none of the non-basics has changed its state, the following message is printed:

PITN - Number of iteration of current parametric analysis

OBJ - Objective value

TMAX - Current boundary value of parameter t

VARIABLE "name" (number of the variable) FROM "bound" REPLACES BASIC VARIABLE "name" (number of the variable) WHICH PASSES TO "bound"

(LL is substituted for "lower bound" and UL for "upper bound")

In other cases the first three items are the same as above and the last row is replaced by the appropriate number of the following sentences:

VARIABLE"name" (number of the variable) FROM "bound" ENTERS THE BASIS

BASIC VARIABLE"name" (number of the variable) PASSES TO "bound"

VARIABLE"name" (number of the variable) FROM "bound" PASSES TO "bound"

If a variable which does not belong to $I_{\sum}$ has changed its state, this row is preceded by the following message:

WITHIN THE GIVEN TOLERANCE ONLY THE FOLLOWING INFORMATION IS AVAILABLE

Special messages

1. If in the final iteration the situation arises in which the optimal basis is constant for every $t > t_{\max}$, the following message appears in the printer file:

   PITN - Number of iteration of current parametric analysis - FOR THE VALUE OF THE PARAMETER = Value of t_i INFINITE RANGE (TMAX.GE.1.E15)

   where TMAX $= t_{i+1} - t_i$. In this case the last optimal solution is stored in the printer file or in the file defined by the keyword PLP SOLUTION FILE.

2. If the optimal solution vanishes, one of the following MINOS messages is printed:
   - in the case of PLP COST:

     EXIT - PROBLEM IS UNBOUNDED

     this is followed by:

     PITN - Number of iteration of current parametric analysis
     TMAX= Boundary value of the parameter t
   - for PLP RHS and PLP BOUND:

     EXIT - PROBLEM IS INFEASIBLE
     NO. AND SUM OF INFEASIBILITIES "number" and "value"

     This is followed by:

     PITN = Number of iteration of current kind of analysis
     TMAX = Boundary value of parameter

   In both cases the SOLN output corresponding to the value t_{i+1} of parameter t is printed or stored in the file defined by the user in the keyword PLP SOLUTION FILE.

3. If MINOS cannot find the next optimal solution because of tolerances defined in MINOS, the following printout is displayed:

   WITHIN THE GIVEN TOLERANCE NO NEW BASIS IS FOUND

   This is a failure of the package. In order to continue the analysis, the user should decrease the appropriate tolerance (tolerances) in MINOS or to increase the factor DELTA in keyword PLP INCREMENT.

4. If the keyword PLP BOUND ANALYSIS n is less than zero an additional output is printed. It gives the values: $t_i^a - t_i = t_{\mathrm{maxa}}$,$t_{i+1} - t_i = t_{\mathrm{maxb}}$ and the corresponding boundary values of bounds:

   PITN = Number of iteration of current kind of analysis
   TMAX = boundary value of parameter

   This is followed by the information on t_{maxa}.

4. EXAMPLES

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

Maximize

$$F(x) = 0.1x_1 + x_2$$

subject to:

$$x_1 + x_3 = 3.$$

$$0.7065\,(x_1 + x_2) + x_4 = 3.826$$

$$x_2 + x_5 = 3.$$
$$-0.7065(x_1 - x_2) + x_6 = 1.$$
$$-x_1 + x_7 = -1$$
$$0 \le x_1 \ , \ x_2 \le 5 \ , \ 0 \le x_i \le 2 \ ,i = 3,\ldots,7.$$

Two runs of PLP are presented. The first shows the independent work of PLP. It contains all three kinds of parametric analysis: PLP COST, PLP RHS, PLP BOUND. In the second, we have the results of dependent work of PLP. The task for PLP was to perform one iteration of PLP RHS, then all iterations of PLP COST and then all iterations of PLP BOUND.

Below we give the MPS file common for both runs and then we give the MINOS and PLP specifications used to solve each of these problems.

Then we give the standard MINOS printout, followed by two outputs of PLP.

```
name          test
rows
 n  ob
 e  r1
 e  r2
 e  r3
 e  r4
 e  r5
columns
    x1        ob             0.1
    x1        r1             1.
    x1        r2             .7065
    x1        r4            -.7065
    x1        r5            -1.
    x2        ob             1.
    x2        r2             .7065
    x2        r3             1.
    x2        r4             .7065
    x3        r1             1.
    x4        r2             1.
    x5        r3             1.
    x6        r4             1.
    x7        r5             1.
rhs
    rh        r1             3.
    rh        r2             3.826
    rh        r3             3.
    rh        r4             1.
    rh        r5            -1.
bounds
 mi bo        x2
 up bo        x2             5.
 up bo        x3             2.
 up bo        x4             2.
 up bo        x5             2.
 up bo        x6             2.
 up bo        x7             2.
endata
```

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

= = = = =

specs file

```
begin Second P L P Test
     maximize
     plp rhs analysis
     plp cost analysis
     plp bound analysis -10
     plp increment  1.5
     plp frequency 1
     data plp file -9
     print data plp file
end
```

```
problem name    test                objective value    3.2415428215d+00

status          optimal soln        iteration   1     superbasics   0

objective       ob                  (max)
rhs             rh
ranges
bounds          bo
```

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.24154	-3.24154	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2
	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.14154	3
	12	r3	eq	3.00000	.00000	3.00000	3.00000	-.90000	4
a	13	r4	eq	1.00000	.00000	1.00000	1.00000	.00000	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
1	x1	bs	2.41543	.10000	.00000	none	.00000	7
2	x2	bs	3.00000	1.00000	none	5.00000	.00000	8
3	x3	bs	.58457	.00000	.00000	2.00000	.00000	9
4	x4	ll	.00000	.00000	.00000	2.00000	-.14154	10
5	x5	ll	.00000	.00000	.00000	2.00000	-.90000	11
6	x6	bs	.58700	.00000	.00000	2.00000	.00000	12
7	x7	bs	1.41543	.00000	.00000	2.00000	.00000	13
8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-3.24154	14

P L P   - - -   version 1.0   june 1986,

= = =

data plp file

```
 1  name         plpc
 2  'set'                     .000000d+00
 3               x1          1.00000d+00              .00000d+00
 4               x2         -1.00000d+00              .00000d+00
 5  *
 6  *   Note:
 7  *   Declaration of dummy coefficients (=0 in MPS file) of the objective
 8  *   is not necessary because the above direction is defined in the x1-x2
 9  *   subspace of cost vectors.
10  *
```

```
    11  endata


plp cost
--------

 number  .column.  ...direction..  .obj gradient.  m+j

     1  x1              1.00000          .10000     7
     2  x2             -1.00000         1.00000     8
     3  x3               .00000          .00000     9
     4  x4               .00000          .00000    10
     5  x5               .00000          .00000    11
     6  x6               .00000          .00000    12
     7  x7               .00000          .00000    13

pitn=  1       obj=  0.29784856d+01       tmax=  0.45000d+00       variable x5      (  5) from 11
replaces basic variable  x3      (  3) which passes to 11

problem name   test                  objective value    2.9784855632d+00

status         optimal soln          iteration   1     superbasics   0

objective      ob               (max)
rhs            rh
ranges
bounds         bo

section 1 - rows

 number  ...row..  at  ...activity...  slack activity  ..lower limit.  ..upper limit.  .dual activity   ..i

     9  ob        bs        2.97849        -2.97849          none            none          1.00000       1
    10  r1        eq        3.00000          .00000         3.00000         3.00000        0.00000       2
    11  r2        eq        3.82600          .00000         3.82600         3.82600        -.77848       3
    12  r3        eq        3.00000          .00000         3.00000         3.00000        0.00000       4
a   13  r4        eq        1.00000          .00000         1.00000         1.00000         .00000       5
a   14  r5        eq       -1.00000          .00000        -1.00000        -1.00000         .00000       6

section 2 - columns

 number  .column.  at  ...activity...  .obj gradient.  ..lower limit.  ..upper limit.  reduced gradnt   m+j

     1  x1        bs        3.00000          .55000          .00000           none          .00000       7
     2  x2        bs        2.41543          .55000           none          5.00000         .00000       8
     3  x3        ll         .00000          .00000          .00000         2.00000        0.00000       9
     4  x4        ll         .00000          .00000          .00000         2.00000        -.77848      10
     5  x5        bs         .58457          .00000          .00000         2.00000        0.00000      11
     6  x6        bs        1.41300          .00000          .00000         2.00000         .00000      12
d    7  x7        bs        2.00000          .00000          .00000         2.00000         .00000      13
     8  rh        eq       -1.00000          .00000        -1.00000        -1.00000       -2.97849      14

pitn=  2       obj=  0.32999950d+01       tmax=  0.10000d+01       variable x4      (  4) from 11
replaces basic variable  x6      (  6) which passes to ul

problem name   test                  objective value    3.2999950096d+00

status         optimal soln          iteration   2     superbasics   0

objective      ob               (max)
rhs            rh
ranges
bounds         bo

section 1 - rows
```

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.30000	-3.30000	none	none	1.00000	1
	10	r1	eq	3.00000	.00000	3.00000	3.00000	-1.09999	2
	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.00001	3
a	12	r3	eq	3.00000	.00000	3.00000	3.00000	.00000	4
	13	r4	eq	1.00000	.00000	1.00000	1.00000	0.00000	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	3.00000	1.10000	.00000	none	.00000	7
	2	x2	bs	1.58457	0.00000	none	5.00000	.00000	8
	3	x3	ll	.00000	.00000	.00000	2.00000	-1.09999	9
	4	x4	bs	.58700	.00000	.00000	2.00000	-.00001	10
	5	x5	bs	1.41543	.00000	.00000	2.00000	.00000	11
	6	x6	ul	2.00000	.00000	.00000	2.00000	0.00000	12
d	7	x7	bs	2.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-3.30000	14

pitn=   3 for the value of the parameter=   1.00000 infinite range (tmax.ge.1.e15)

data plp file

```
     12  name           plpr
     13  'set'                       .000000d+00
     14                 r3          1.00000d+00                 .00000d+00
     15  *
     16  *   Note that all components of rhs vector are defined in MPS file.
     17  *
     18  endata
```

plp rhs
----

number	...row..	...direction..	.........rhs..	..i
10	r1	.00000	3.00000	2
11	r2	.00000	3.82600	3
12	r3	1.00000	3.00000	4
13	r4	.00000	1.00000	5
14	r5	.00000	-1.00000	6

pitn= 1   obj= 0.36154282d+01   tmax= 0.41543d+00   variable x5   (5) from 11
replaces basic variable x6   (6) which passes to 11

problem name   test   objective value   3.6154281824d+00

status   optimal soln   iteration   1   superbasics   0

objective   ob   (max)
rhs   rh
ranges
bounds   bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.61543	-3.61543	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2

	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.77849	3
a	12	r3	eq	3.41544	.00000	3.41544	3.41544	.00000	4
	13	r4	eq	1.00000	.00000	1.00000	1.00000	-.63694	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	2.00000	.10000	.00000	none	.00000	7
	2	x2	bs	3.41543	1.00000	none	5.00000	.00000	8
	3	x3	bs	1.00000	.00000	.00000	2.00000	.00000	9
	4	x4	ll	.00000	.00000	.00000	2.00000	-.77849	10
d	5	x5	bs	0.00000	.00000	.00000	2.00000	.00000	11
	6	x6	ll	.00000	.00000	.00000	2.00000	-.63694	12
	7	x7	bs	1.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-3.61543	14

exit -- problem is infeasible.
no. and sum of infeasibilities 1 2.57313880d-05

pitn= 2 tmax= 0.24154d+01

data plp file

```
     19 name          plpb
     20 'set'                     .000000d+00
     21      upper    x2          -1.00000d+00               .00000d+00
     22 endata
```

plp bound
----

section 1 - rows

number	...row..	.ll direction.	.lower limit..	.ul direction.	.upper limit..	..i
10	r1	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	6

section 2 - columns

number	.column.	.ll direction.	.lower limit..	.ul direction.	.upper limit..	m+j
1	x1	.00000	.00000	.00000	none	7
2	x2	.00000	none	-1.00000	5.00000	8
3	x3	.00000	.00000	.00000	2.00000	9
4	x4	.00000	.00000	.00000	2.00000	10
5	x5	.00000	.00000	.00000	2.00000	11
6	x6	.00000	.00000	.00000	2.00000	12
7	x7	.00000	.00000	.00000	2.00000	13

a. no change in the optimal solution
finite range (tmaxa= 0.20000d+01)

b. no change in the optimal basis
finite range (tmaxb= 0.20000d+01)

section 1 - rows

number	...row..	.ll direction.	.ll boundary a	.ll boundary b	.ul direction.	.ul boundary a	.ul boundary b	..i
10	r1	.00000	.00000	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	.00000	.00000	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	.00000	.00000	.00000	.00000	none	none	7
2	x2	.00000	none	none	-1.00000	3.00000	3.00000	8
3	x3	.00000	.00000	.00000	.00000	2.00000	2.00000	9
4	x4	.00000	.00000	.00000	.00000	2.00000	2.00000	10
5	x5	.00000	.00000	.00000	.00000	2.00000	2.00000	11
6	x6	.00000	.00000	.00000	.00000	2.00000	2.00000	12
7	x7	.00000	.00000	.00000	.00000	2.00000	2.00000	13

pitn= 1 obj= 0.32415278d+01 tmax= 0.20000d+01 variable x5 (5) from ll
replaces basic variable x2 (2) which passes to ul

problem name test objective value 3.2415278032d+00

status optimal soln iteration 1 superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.24153	-3.24153	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2
	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.14154	3
a	12	r3	eq	3.00000	.00000	3.00000	3.00000	.00000	4
a	13	r4	eq	1.00000	.00000	1.00000	1.00000	.00000	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	2.41543	.10000	.00000	none	.00000	7
	2	x2	ul	3.00000	1.00000	none	3.00000	.90000	8
	3	x3	bs	.58457	.00000	.00000	2.00000	.00000	9
	4	x4	ll	.00000	.00000	.00000	2.00000	-.14154	10
d	5	x5	bs	0.00000	.00000	.00000	2.00000	.00000	11
	6	x6	bs	.58700	.00000	.00000	2.00000	.00000	12
	7	x7	bs	1.41543	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-.54154	14

a. no change in the optimal solution
finita range (tmaxa= .00000d+00)

b. no change in the optimal basis
finite range (tmaxb= 0.58457d+00)

section 1 - rows

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

10	r1	.00000	.00000	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	.00000	.00000	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	.00000	.00000	.00000	.00000	none	none	7
2	x2	.00000	none	none	-1.00000	3.00000	2.41543	8
3	x3	.00000	.00000	.00000	.00000	2.00000	2.00000	9
4	x4	.00000	.00000	.00000	.00000	2.00000	2.00000	10
5	x5	.00000	.00000	.00000	.00000	2.00000	2.00000	11
6	x6	.00000	.00000	.00000	.00000	2.00000	2.00000	12
7	x7	.00000	.00000	.00000	.00000	2.00000	2.00000	13

pitn= 2 obj= 0.27154131d+01 tmax= 0.25846d+01 variable x4 (4) from ll
replaces basic variable x7 (7) which passes to ul

problem name test objective value 2.7154131457d+00

status optimal soln iteration 2 superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	2.71541	-2.71541	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2
a	11	r2	eq	3.82600	.00000	3.82600	3.82600	.00000	3
a	12	r3	eq	3.00000	.00000	3.00000	3.00000	.00000	4
a	13	r4	eq	1.00000	.00000	1.00000	1.00000	.00000	5
	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.10000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	3.00000	.10000	.00000	none	.00000	7
	2	x2	ul	2.41543	1.00000	none	2.41543	1.00000	8
d	3	x3	bs	.00000	.00000	.00000	2.00000	.00000	9
d	4	x4	bs	0.00000	.00000	.00000	2.00000	.00000	10
	5	x5	bs	.58457	.00000	.00000	2.00000	.00000	11
	6	x6	bs	1.41300	.00000	.00000	2.00000	.00000	12
	7	x7	ul	2.00000	.00000	.00000	2.00000	.10000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-.10000	14

a. no change in the optimal solution
finite range (tmaxa= .00000d+00)

b. no change in the optimal basis
finite range (tmaxb= 0.83086d+00)

section 1 - rows

number	...row..	.ll direction.	.ll boundary a	.ll boundary b	.ul direction.	.ul boundary a	.ul boundary b	..i
10	r1	.00000	.00000	.00000	.00000	.00000	.00000	2

11	r2	.00000	.00000	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	.00000	.00000	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	.00000	.00000	.00000	.00000	none	none	7
2	x2	.00000	none	none	-1.00000	2.41543	1.58457	8
3	x3	.00000	.00000	.00000	.00000	2.00000	2.00000	9
4	x4	.00000	.00000	.00000	.00000	2.00000	2.00000	10
5	x5	.00000	.00000	.00000	.00000	2.00000	2.00000	11
6	x6	.00000	.00000	.00000	.00000	2.00000	2.00000	12
7	x7	.00000	.00000	.00000	.00000	2.00000	2.00000	13

pitn= 3 obj= 0.18845506d+01 tmax= 0.34154d+01 variable x7 (7) from ul
replaces basic variable x6 (6) which passes to ul

problem name test objective value 1.8845506266d+00

status optimal soln iteration 3 superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	1.88455	-1.88455	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2
a	11	r2	eq	3.82600	.00000	3.82600	3.82600	.00000	3
a	12	r3	eq	3.00000	.00000	3.00000	3.00000	.00000	4
	13	r4	eq	1.00000	.00000	1.00000	1.00000	.14154	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	3.00000	.10000	.00000	none	.00000	7
	2	x2	ul	1.58457	1.00000	none	1.58457	1.10000	8
d	3	x3	bs	0.00000	.00000	.00000	2.00000	.00000	9
	4	x4	bs	.58700	.00000	.00000	2.00000	.00000	10
	5	x5	bs	1.41543	.00000	.00000	2.00000	.00000	11
	6	x6	ul	2.00000	.00000	.00000	2.00000	.14154	12
d	7	x7	bs	2.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	.14154	14

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

specs file

```
begin Second P L P Test
     maximize
     plp order
     plp rhs analysis 1
```

```
          plp cost analysis
          plp bound analysis -3
          plp increment  1000.
          plp frequency 1
          data plp file -9
          print data plp file
     end

    P L P    - - -    version 1.0   june 1986,
    = = =
    DEPENDENT  WORK  of   P L P
    ---------  ----       - - -


data plp file
----------------
      12 name          plpr
      13 'set'                    .000000d+00
      14               r3         1.00000d+00              .00000d+00
      15 *
      16 *  Note that all components of rhs vector are defined in MPS file.
      17 *
      18 endata


plp rhs
-------

 number  ...row..  ...direction..  .........rhs..  ..i

     10  r1               .00000        3.00000    2
     11  r2               .00000        3.82600    3
     12  r3              1.00000        3.00000    4
     13  r4               .00000        1.00000    5
     14  r5               .00000       -1.00000    6

pitn=  1       obj=  0.36154282d+01       tmax=  0.41543d+00       variable x5      (  5) from 11
replaces basic variable  x6      (  6) which passes to 11

problem name   test                objective value    3.6154281824d+00

status         optimal soln        iteration    1     superbasics    0

objective      ob           (max)
rhs            rh
ranges
bounds         bo

section 1 - rows

 number  ...row..  at  ...activity...  slack activity  ..lower limit.  ..upper limit.  .dual activity  ..i

      9  ob        bs        3.61543       -3.61543          none            none         1.00000     1
a    10  r1        eq        3.00000         .00000        3.00000         3.00000         .00000     2
     11  r2        eq        3.82600         .00000        3.82600         3.82600        -.77849     3
a    12  r3        eq        3.42251         .00000        3.42251         3.42251         .00000     4
     13  r4        eq        1.00000         .00000        1.00000         1.00000        -.63694     5
a    14  r5        eq       -1.00000         .00000       -1.00000        -1.00000         .00000     6

section 2 - columns

 number  .column.  at  ...activity...  .obj gradient.  ..lower limit.  ..upper limit.  reduced gradnt   m+j

      1  x1        bs        2.00000         .10000          .00000          none          .00000     7
      2  x2        bs        3.41543        1.00000          none          5.00000         .00000     8
```

	3 x3	bs	1.00000	.00000	.00000	2.00000	.00000	9
	4 x4	ll	.00000	.00000	.00000	2.00000	-.77849	10
d	5 x5	bs	0.00000	.00000	.00000	2.00000	.00000	11
	6 x6	ll	.00000	.00000	.00000	2.00000	-.63694	12
	7 x7	bs	1.00000	.00000	.00000	2.00000	.00000	13
	8 rh	eq	-1.00000	.00000	-1.00000	-1.00000	-3.61543	14

data plp file

```
----------------
      1 name          plpc
      2 'set'                   .000000d+00
      3               x1        1.00000d+00               .00000d+00
      4               x2       -1.00000d+00               .00000d+00
      5 *
      6 *   Note:
      7 *   Declaration of dummy coefficients (=0 in MPS file) of the objective
      8 *   is not necessary because the above direction is defined in the x1-x2
      9 *   subspace of cost vectors.
     10 *
     11 endata
```

plp cost

----

number	.column.	...direction..	.obj gradient.	m+j
1	x1	1.00000	.10000	7
2	x2	-1.00000	1.00000	8
3	x3	.00000	.00000	9
4	x4	.00000	.00000	10
5	x5	.00000	.00000	11
6	x6	.00000	.00000	12
7	x7	.00000	.00000	13

pitn= 1 obj= 0.29784856d+01 tmax= 0.45000d+00 variable x6 (6) from ll
replaces basic variable x3 (3) which passes to ll

problem name test objective value 2.9784855632d+00

status optimal soln iteration 1 superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	2.97849	-2.97849	none	none	1.00000	1
	10	r1	eq	3.00000	.00000	3.00000	3.00000	-.00580	2
	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.77849	3
a	12	r3	eq	3.42251	.00000	3.42251	3.42251	.00000	4
a	13	r4	eq	1.00000	.00000	1.00000	1.00000	.00000	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
1	x1	bs	3.00000	.55000	.00000	none	-.00580	7
2	x2	bs	2.41543	.55000	none	5.00000	.00000	8

	3	x3	ll	.00000	.00000	.00000	2.00000	-.00580	9
	4	x4	ll	.00000	.00000	.00000	2.00000	-.77849	10
	5	x5	bs	1.00708	.00000	.00000	2.00000	.00000	11
	6	x6	bs	1.41300	.00000	.00000	2.00000	.00000	12
d	7	x7	bs	2.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-2.99590	14

pitn= 2 obj= 0.33000001d+01 tmax= 0.10000d+01 variable x4 (4) from ll
replaces basic variable x6 (6) which passes to ul

problem name test objective value 3.3000000715d+00

status optimal soln iteration 2 superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.30000	-3.30000	none	none	1.00000	1
	10	r1	eq	3.00000	.00000	3.00000	3.00000	-1.09717	2
	11	r2	eq	3.82600	.00000	3.82600	3.82600	-.00200	3
a	12	r3	eq	3.42251	.00000	3.42251	3.42251	.00000	4
	13	r4	eq	1.00000	.00000	1.00000	1.00000	.00200	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	3.00000	1.10000	.00000	none	0.00000	7
	2	x2	bs	1.58457	0.00000	none	5.00000	.00000	8
	3	x3	ll	.00000	.00000	.00000	2.00000	-1.09717	9
	4	x4	bs	.58700	.00000	.00000	2.00000	-.00200	10
	5	x5	bs	1.83793	.00000	.00000	2.00000	.00000	11
	6	x6	ul	2.00000	.00000	.00000	2.00000	.00200	12
d	7	x7	bs	2.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	-3.29717	14

pitn= 3 for the value of the parameter= 1.00000 infinite range (tmax.ge.1.e15)

data plp file

```
19 name      plpb
20 'set'              .000000d+00
21    upper  x2     -1.00000d+00          .00000d+00
22 endata
```

plp bound
----

section 1 - rows

number	...row..	.ll direction.	.lower limit..	.ul direction.	.upper limit..	..i
10	r1	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	5

14	r5	.00000	.00000	.00000	.00000	6

section 2 - columns

number	.column.	.ll direction.	.lower limit..	.ul direction.	.upper limit..	m+j
1	x1	.00000	.00000	.00000	none	7
2	x2	.00000	none	-1.00000	5.00000	8
3	x3	.00000	.00000	.00000	2.00000	9
4	x4	.00000	.00000	.00000	2.00000	10
5	x5	.00000	.00000	.00000	2.00000	11
6	x6	.00000	.00000	.00000	2.00000	12
7	x7	.00000	.00000	.00000	2.00000	13

a. no change in the optimal solution
finite range (tmaxa= 0.34154d+01)

b. no change in the optimal basis
finite range (tmaxb= 0.34154d+01)

section 1 - rows

number	...row..	.ll direction.	.ll boundary a	.ll boundary b	.ul direction.	.ul boundary a	.ul boundary b	..i
10	r1	.00000	.00000	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	.00000	.00000	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	.00000	.00000	.00000	.00000	none	none	7
2	x2	.00000	none	none	-1.00000	1.58457	1.58457	8
3	x3	.00000	.00000	.00000	.00000	2.00000	2.00000	9
4	x4	.00000	.00000	.00000	.00000	2.00000	2.00000	10
5	x5	.00000	.00000	.00000	.00000	2.00000	2.00000	11
6	x6	.00000	.00000	.00000	.00000	2.00000	2.00000	12
7	x7	.00000	.00000	.00000	.00000	2.00000	2.00000	13

pitn= 1 obj= 0.33000001d+01 tmax= 0.34154d+01 variable x3 (3) from ll
replaces basic variable x2 (2) which passes to ul

problem name	test	objective value	3.3000001013d+00
status	optimal soln	iteration 1	superbasics 0

objective ob (max)
rhs rh
ranges
bounds bo

section 1 - rows

	number	...row..	at	...activity...	slack activity	..lower limit.	..upper limit.	.dual activity	..i
	9	ob	bs	3.30000	-3.30000	none	none	1.00000	1
a	10	r1	eq	3.00000	.00000	3.00000	3.00000	.00000	2
a	11	r2	eq	3.82600	.00000	3.82600	3.82600	.00000	3
a	12	r3	eq	3.42251	.00000	3.42251	3.42251	.00000	4
	13	r4	eq	1.00000	.00000	1.00000	1.00000	1.55697	5
a	14	r5	eq	-1.00000	.00000	-1.00000	-1.00000	.00000	6

section 2 - columns

	number	.column.	at	...activity...	.obj gradient.	..lower limit.	..upper limit.	reduced gradnt	m+j
	1	x1	bs	3.00000	1.10000	.00000	none	.00000	7
	2	x2	ul	1.58457	0.00000	none	1.58457	1.10000	8
d	3	x3	bs	0.00000	.00000	.00000	2.00000	.00000	9
	4	x4	bs	.58700	.00000	.00000	2.00000	.00000	10
	5	x5	bs	1.83793	.00000	.00000	2.00000	.00000	11
	6	x6	ul	2.00000	.00000	.00000	2.00000	1.55697	12
d	7	x7	bs	2.00000	.00000	.00000	2.00000	.00000	13
	8	rh	eq	-1.00000	.00000	-1.00000	-1.00000	1.55697	14

a. no change in the optimal solution
finite range (tmaxa= .00000d+00)

b. no change in the optimal basis
finite range (tmaxb= 0.16207d+00)

section 1 - rows

number	...row..	.ll direction.	.ll boundary a	.ll boundary b	.ul direction.	.ul boundary a	.ul boundary b	..i
10	r1	.00000	.00000	.00000	.00000	.00000	.00000	2
11	r2	.00000	.00000	.00000	.00000	.00000	.00000	3
12	r3	.00000	.00000	.00000	.00000	.00000	.00000	4
13	r4	.00000	.00000	.00000	.00000	.00000	.00000	5
14	r5	.00000	.00000	.00000	.00000	.00000	.00000	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	.00000	.00000	.00000	.00000	none	none	7
2	x2	.00000	none	none	-1.00000	1.58457	1.42251	8
3	x3	.00000	.00000	.00000	.00000	2.00000	2.00000	9
4	x4	.00000	.00000	.00000	.00000	2.00000	2.00000	10
5	x5	.00000	.00000	.00000	.00000	2.00000	2.00000	11
6	x6	.00000	.00000	.00000	.00000	2.00000	2.00000	12
7	x7	.00000	.00000	.00000	.00000	2.00000	2.00000	13

exit -- problem is infeasible.
no. and sum of infeasibilities 1 9.99999046d-03

pitn= 2 tmax= 0.35775d+01