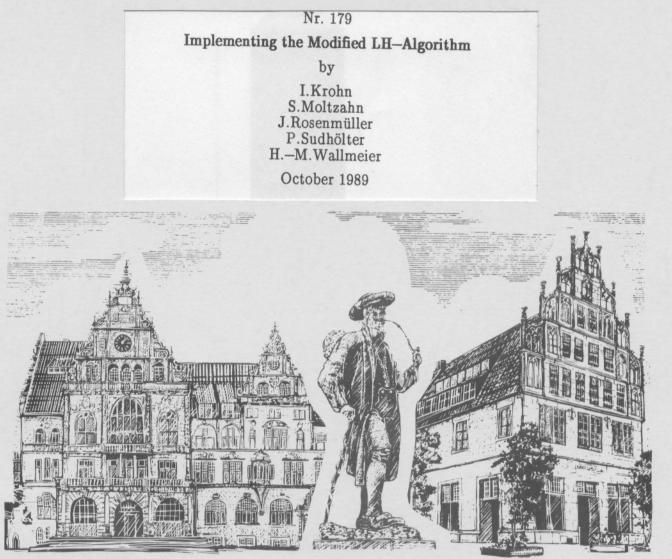# Universität Bielefeld/IMW

Working Papers
Institute of Mathematical Economics

Arbeiten aus dem
Institut für Mathematische Wirtschaftsforschung

Nr. 179

Implementing the Modified LH—Algorithm

by

I. Krohn
S. Moltzahn
J. Rosenmüller
P. Sudhölter
H.—M. Wallmeier

October 1989

H. G. Bergenthal

ABSTRACT

The modified *Lemke–Howson algorithm* is a constructive procedure which enables us to compute equilibrium points of a bimatrix game. The algorithm as described by one of the authors (see ROSENMÜLLER [9]) is based on the original version invented by LEMKE–HOWSON [5]. However, it differs from this version with respect to several features. It works directly with the matrices defining the bimatrix game A and B. It has an easy and very direct geometrical interpretation, hence for small games we can follow the development of the algorithm geometrically. Finally, instead of being "bilinear", the algorithm behaves rather like a "piecewise linear program".

This presentation closes a gap: although the algorithm has been described geometrically (and with a flow diagram) in [9], there has been no constructive procedure that can be implemented on a computer. This is provided by the present paper.

We give all necessary proofs and computations in order to establish the following facts: There are <u>two</u> tableaus accompanying the proceeding of the algorithm. As the algorithm changes, moving alternatingly in the simplices of mixed strategies, so does the computational procedure alternatingly dealing with the two different tableaus.

Both tableaus contain six regions depending on the various ways of "transitions" the procedure has to perform.

While this all is in marked difference to linear programming, there is also consolation: The well known rectangle rule of linear programming can be modified easily (that is, there is a family of rectangle rules) such that changing the tableau alternatingly amounts to applying the appropriate rectangle rule. Thus, there is also close similarity to the familiar LP–procedure.

Thus, a complete description of the modified LH–algorithm is provided that can immediately be implemented on any computer, in particular we supply an APL–program that, e.g., can be run on an IBM PC (IBM is a registered trademark of International Business Machines Cooperation).

SECTION 1

Introduction

Let $I = \{1,...,m\}$ and $J = \{1,...,n\}$. A *bimatrix game* (in mixed strategies) is a quadruple

(1) $$\Gamma = (X,Y,A,B)$$

such that $A = (a_{ij})_{i \in I, j \in J}$ and $B = (b_{ij})_{i \in I, j \in J}$ are $m \times n$ matrices and

(2) $$X = \{x \in \mathbb{R}^m \mid x = (x_1,...,x_m) \geq 0, \sum_{i \in I}^{m} x_i = 1\}$$

(3) $$Y = \{y \in \mathbb{R}^n \mid y = (y_1,...,y_n) \geq 0, \sum_{j \in J}^{n} y_j = 1\}$$

are the (mixed) strategies of player 1 and 2. If player 1 chooses $x \in X$ and player 2 chooses $y \in Y$, then *payoffs* are defined by

$$x A y = \sum_{i \in I} \sum_{j \in J} x_i a_{ij} y_j$$

for player 1, and $x B y$ for player 2.

A pair $(\overline{x},\overline{y}) \in X \times Y$ is said to be an *equilibrium point* if

$$\overline{x} A \overline{y} \geq x A \overline{y} \qquad (x \in X)$$

and

$$\overline{x} B \overline{y} \geq \overline{x} B y \qquad (y \in Y)$$

holds true; thus, in equilibrium, no player has an incentive to deviate for his payoff cannot be improved upon. If Γ is a *zero−sum* game (i.e., $B = -A$), then an equilibrium consists of a pair of optimal strategies and vice versa.

The *Lemke−Howson−algorithm* as devised in LEMKE–HOWSON [5] is a procedure that (for "nondegenerate games") yields an equilibrium point within finitely many steps.

The procedure works by transforming the bimatrix game into a bilinear program; whereafter the algorithm, starting with an "unbounded edge" proceeds by moving along

a certain system of polyhedral edges of dimension 1 to search for an equilibrium point. An implementation of this version of the *LH–algorithm* in the sense that the geometrical behavior of the algorithm is represented by a sequence of tableaus, to be computed consecutively and leading to a numerical evaluation of an equilibrium point, has been presented in PARTHASARATHY – RAGHAVAN [6]; however, a formal proof (and an established computer program) for a neatly working version of the algorithm on a contemporal computer is lacking.

The *Lemke–Howson–algorithm* also yields some insight into the structure of equilibria. It shows that the number of equilibrium points (for nondegenerate games) is odd. It is also known that not every equilibrium point can necessarily be reached in any case; even if the initial "unbounded edge" is being changed, there are equilibria not to be reached by the LH–algorithm (for further literature we refer to AGGARVAL [1], BASTIAN [2], PARTHASARATHY–RAGHAVAN [6], SHAPLEY [7], TODD [10, 11]).

An alternative version of the algorithm (the modified LH–algorithm) has been presented in ROSENMÜLLER [9], Chapter I, Section 1. This version works directly with the matrices with A and B constituting the bimatrix game. The algorithm is not bilinear but rather "piecewise linear": it works effectively in the simplices X and Y, alternating performing steps in each of them. There is a flow diagram established in [9] which, however, requires the computation of solutions of certain linear equations after each step and, hence, is not in the spirit of traditional linear programming. In practice the procedure suggested by the flow diagram is rather slow and the capacity of most computers is not sufficient, even for small problems.

As the procedure is not a standard optimization problem it is not clear how to exactly define a sequence of "tableaus" corresponding to the geometrical movement of the *Lemke–Howson–algorithm* as presented in [9]. This is exactly the goal of the present paper. We suggest the correct parametrization of edges of certain subpolyhedra of the simplices of mixed strategies X and Y. Using this parametrization, we define a pair of tableaus (corresponding to the alternating behavior of the modified LH–algorithm) such that alternating performing the rectangle rule in each of the tableaus actually yields an equilibrium point. The procedure can thus be implemented on a computer and, for the sake of completeness, we are adding an APL–version of such a program.

Let $A_{i\cdot}$, $A_{\cdot j}$ denote the i–th row and j–th column of the matrix A respectively. Introduce the convex polyhedra

(4)
$$K_i = \{y \in Y \mid A_{i\cdot}\, y \geq A_{k\cdot}\, y \; (k \in I)\} \qquad (i \in I)$$

$$L_j = \{x \in X \mid x\, B_{\cdot j} \geq x\, B_{\cdot l} \; (l \in J)\} \qquad (j \in J)$$

as well as

(5)
$$K_T = \bigcap_{i \in T} K_i \qquad\qquad T \subseteq I, \; T \neq \emptyset$$

$$L_R = \bigcap_{j \in R} L_j \qquad\qquad R \subseteq J, \; R \neq \emptyset$$

E.g. K_i denotes the mixed strategies of player 2 against which the (pure) strategy $i \in I$ (or the mixed strategy e^i) of player 1 is "best reply". It is not hard to see that $(\bar{x}, \bar{y})$ is an equilibrium point of Γ if and only if

$$\bar{y} \in K_{\{i \mid \bar{x}_i > 0\}} \text{ and } \bar{x} \in L_{\{j \mid \bar{y}_j > 0\}}.$$

Thus, in equilibrium, the positive coordinates of $\bar{x}$ and the polyhedra K_i containing $\bar{y}$ correspond to each other (in fact uniquely if nondegeneracy prevails) – this is of course an analogue to the familiar "optimality condition" of L.P. theory. We are thus motivated to introduce polyhedra

(6)
$$H_{T,U} = K_T \cap \{y \in Y \mid y_j = 0 \; (j \in U)\}$$

$$G_{R,V} = L_R \cap \{x \in X \mid x_i = 0 \; (i \in V)\}.$$

The game is called *nondegenerate* if

(7) $\dim H_{T,U} = n - |T| - |U|, \quad \dim G_{R,V} = m - |R| - |V|, \text{ for } H_{T,U} \neq \emptyset \neq G_{R,V}$

(cf. Definition 1.11, SEC.1, CH.1, of [9]).

We shall assume that the game we are dealing with is nondegenerate.

In this case we have the following characterization of equilibrium points:

(8)
$$\text{Let } (\bar{x}, \bar{y}) \in X \times Y \text{ and put } T = \{i \mid \bar{x}_i > 0\} \subseteq I \text{ and}$$
$$R = \{j \mid \bar{y}_j > 0\} \subseteq J. \text{ Then } (\bar{x}, \bar{y}) \text{ is an equilibrium point}$$
$$\text{if and only if } |T| = |R| \text{ and } \{(\bar{x}, \bar{y})\} = H_{T,R^c} \times G_{R,T^c}.$$

For the details, see [9], and in particular Corollary 1.13 in SEC.1, CH.1.

The statement formalized in (8) can be interpreted geometrically as follows: the simplices X and Y of mixed strategies are *decomposed* by the polyhedra L_j ($j \in J$) and K_i ($i \in$ I) respectively. Among the subfaces of such polyhedra we distinguish *vertices* $H_{T,U}$, $|T| + |U| = n$ and edges $H_{T,U}$, $|T| + |U| = n-1$ (for some $K_i \subseteq Y$; the situation is analogously described in X). A vertex $H_{T,U} = \{\bar{y}\}$ has "labels" assigned to by the polyhedra it is adjacent to (i.e. labels $i \in T$ with $\bar{y} \in K_i$) and by the positive coordinates of $\bar{y}$ (i.e., $\bar{y}_j > 0$ for $j \in U^c$). If $(\bar{x}, \bar{y})$ is an equilibrium point, then the labels of $\{\bar{x}\} = G_{R,V}$ and $\{\bar{y}\} = H_{T,U}$ correspond to each other in a unique way.
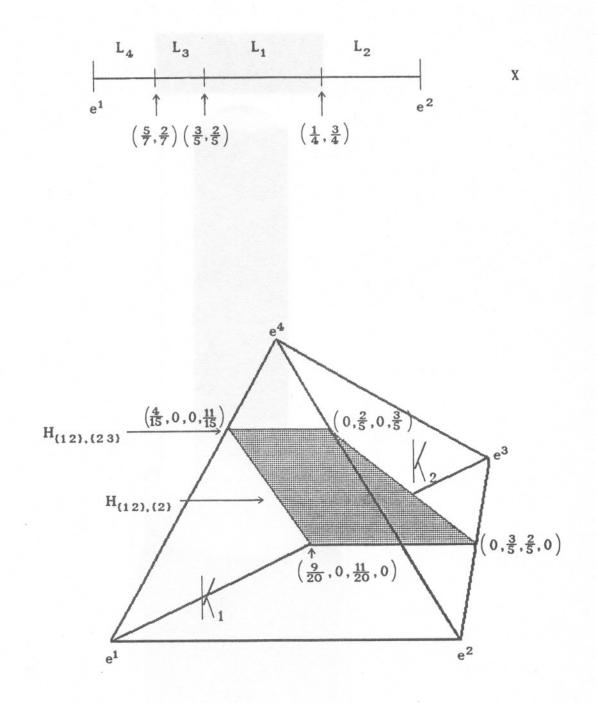
Example 1.1:

Consider the matrices
$$A = \begin{bmatrix} 5 & 3 & -4 & -1 \\ -6 & -3 & 5 & 3 \end{bmatrix}$$
and
$$B = \begin{bmatrix} -1 & -4 & 7 & 11 \\ 3 & 4 & -9 & -19 \end{bmatrix},$$

then the following sketch illustrates the decomposition of X into polyhedra L_1, L_2, L_3, L_4 and the decomposition of Y into polyhedra K_1, K_2 (cf. Fig.1). An equilibrium point is given by

$$\bar{x} = \begin{bmatrix} \frac{3}{5}, \frac{2}{5} \end{bmatrix}, \{\bar{x}\} = G_{13,\emptyset} = G_{13,\{1,2\}^c}$$

$$\bar{y} = \begin{bmatrix} \frac{9}{20}, 0, \frac{11}{20}, 0 \end{bmatrix}, \{\bar{y}\} = H_{12,24} = H_{12,\{1,3\}^c}$$

where the indices ("labels") are matching in the appropriate way: $\bar{x}$ has positive coordinates 1,2 and $\bar{y} \in K_1 \cap K_2$; analogously $\bar{y}_1 > 0$, $\bar{y}_3 > 0$ while $\bar{x} \in L_1 \cap L_3$.

Fig. 1

The modified LH–algorithm is explained in detail in SEC.1, CH.1 of ROSENMÜLLER [9], see also ROSENMÜLLER [8] for the n–person game version (WILSON [12] describes the "multilinear" n–person version of the "original" LH–algorithm). We would like to assume the reader is slightly familiar with the presentation in [9].

For our present purpose we shall describe the modified LH–algorithm with the aid of Example 1.1 as follows: Use e^i to denote the i'th unit vector.

The algorithm starts with a vertex, say

$$\{e^4\} = H_{\{2\}, \{1\ 2\ 3\}} \subsetneq Y$$

in Y. As $e^4 \in K_2$, we move to simplex X and choose

$$\{e^2\} = G_{\{2\}, \{1\}}$$

as the first vertex in X. Now, $e^2 \in L_2$ means that, in Y, we should admit for positive 2nd coordinates, i.e. move "towards" $e^2 \in Y$. That is, we delete index 2 from the labels describing $\{e^4\} \in Y$, thus moving along the edge

$$H_{\{2\}, \{1,3\}} .$$

The endpoint of which is

$$y^1 = (0, \tfrac{1}{3}, 0, \tfrac{2}{3})$$

defining a vertex

$$\{y^1\} = H_{\{1,2,3\}, \{1,3\}}$$

Here the new lable i = 1 appeared, thus in X, we leave e^2 along the edge

$$G_{\{2\}, \emptyset}$$

arriving at $x^1 = (\tfrac{1}{4}, \tfrac{3}{4})$ where

$$\{x^1\} = G_{\{1,2\}, \emptyset} .$$

Hence the next edge in Y is $H_{\{1,2\},\,\{3\}}$ which leads to $y^2 = (\frac{1}{4}, 0, 0, \frac{3}{4})$, We have

$$\{y^2\} = H_{\{1,2\},\,\{2,3\}} \;.$$

The next steps are along the edge $G_{\{1\},\,\emptyset}$ towards $x^2 = (\frac{3}{5}, \frac{2}{5})$; $\{x^2\} = G_{\{13\},\,\emptyset}$ and along $H_{\{1,2\},\,\{2\}}$ towards $y^3 = (\frac{1}{2}, 0, \frac{1}{2}, 0)$. Now $\{y^3\} = H_{\{1,2\},\,\{2,4\}}$ and all labels match in the required manner as we have explained above. Thus we have reached the equilibrium point.

The main purpose of this paper is to develop the computational procedure that accompanies the geometrical picture we have just studied. To this end we shall explain what kind of "movement along an edge" we should adopt for the rigorous mathematical representation. In other words we shall define the "canonical parametrization" of edges depending, however, on what kind of movement along an edge we have in mind. For, (speaking in Y) according to whether we leave a polyhedron K_i (i.e. delete a label $i \in H_{T,U}$) or whether we leave a subface of Y (i.e. delete an index $j \in H_{T,U}$) there are two ways of departing from a vertex in order to move along an edge. Similarly, there are two ways of arriving at a vertex after having travelled along an edge. This yields for different types of a journey and the "canonical parametrization" of this journey along an edge must be chosen accordingly. The appropriate choice is then reflected by the "appropriate" definition of the two tableaus corresponding to a pair of edges, each one located in a simplex X or Y respectively.

The development of our presentation is intended as follows. In Section 1 we will again mention the four ways of travelling along an edge (the detailed discussion has been performed already in [8]). We shall then extensively discuss the case which is most typical for the modified LH–algorithm. The other three cases will not be treated in detail. Hence, Section 2 is devoted to developing the "canonical parametrization" for "case 1a" and for explaining the introductory data of the tableau corresponding to a vertex. In Section 3 we define the tableau (actually a pair of tableaus) and introduce the well known rectangle procedure (which – though in structure resembling the one used in linear programming procedures – is quite different in its detailed appearance). We then prove that the rectangle rule, applied to the tableaus, accompanies the journey between two edges; again the proof is being presented in detail for just one particular

case while the other cases are being treated superficially. Section 4 is then intended to collect the pieces: we present a detailed instruction for using the algorithm. That is, given the matrices A and B, it is explained how to set up the initial tableaus and perform the necessary steps in order to reach a final tableau. This eventually yields a pair of vectors constituting an equilibrium point of the game $\Gamma = (X, Y, A, B)$. Finally, in Section 5, for the sake of completeness, we add a computer program that actually performs the necessary computations. The program has been written in APL and was running on the IBM 6150 RT computer (IBM 6150 is a trademark of International Business Machines Cooperations). However, it can be implemented on any personal computer endowed with APL.

Let us finish this section by introducing the necessary notational conventions.

The matrices A and B are fixed throughout our presentation. In order to avoid indices (coordinates) m+1, n+1, we put

$$\mathbf{I} = \{1,...,m, \square\} = I \cup \{\square\}$$

$$\mathbf{J} = \{1,...,n, *\} = J \cup \{*\}$$

and similarly for $T \subseteq I$, $U \subseteq J$

$$\mathbf{T} = T \cup \{\square\}, \ \mathbf{U} = U \cup \{*\}.$$

Next, vectors $x \in \mathbb{R}^m$ are also repeated as functions $x : I \to \mathbb{R}$, thus we denote the restriction of x onto $T \subseteq I$ by x_T, this is of course to be identified with the vector $x_T = (x_i)_{i \in T} \in \mathbb{R}^T$. For convenience we write

$$x_{-T} := x_{T^c} = x_{I-T}$$

such that for $z = (x_1,...,x_m,\lambda) \in \mathbb{R}^I$ we have e.g.

$$z_{-T} : I - T \to \mathbb{R}$$

$$z_{-T} = (x_i)_{i \notin T} \ .$$

Frequently singletons $\{i\} \subseteq I$ and their elements are identified, thus

$$x_i = x_{\{i\}} \text{ and } x_{-i} = x_{-\{i\}} = (x_1,...,x_{i-1},x_{i+1},...,x_m)$$

for $x \in \mathbb{R}^m$. In this context, "+" is used for "$\cup$" in case of a disjoint union, e.g.,

$$T + i = T \cup \{i\} \text{ (for } i \notin T \subseteq I)$$

$$T + \square - i_0 = (T \cup \{\square\}) - \{i_0\} \text{ (for } i_0 \in T \subseteq I)$$

etc.

The disjoint union of subsets of I, say, is accompanied by the formation of a "direct sum" of functions (vectors) defined on these subsets.

E.g., if T', $T'' \subseteq I$, $T' \cap T'' = \emptyset$, and $z' : T' \to \mathbb{R}$, $z'' : T'' \to \mathbb{R}$, then $z = z' \oplus z''$ denotes the function on $T = T' + T''$ $(= T' \cup T'')$ defined by

$$z : T' + T'' \to \mathbb{R}$$

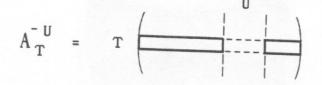$$z_i = z'_i \ (i \in T'), \ z_i = z''_i \ (i \in T'')$$

or (less precise in notation) the vector

$$z = (z', z'').$$

An analogous notation is employed for matrices. E.g., matrix A can be seen as a mapping $A : I \times J \to \mathbb{R}$ and for $T \subseteq I$, $U \subseteq J$ we denote by A_T^U the restriction on $T \times U$ – which is represented by the matrix

Similarly

We write $A_T := A_T^J$; however, the i'-th row of A is $A_{i.}$ and the j'-th column is $A_{.j}$, thus

$$A_{i.} = A_{\{i\}}^J = A_i^J$$

but A_i is avoided.

Next let $e = (1,...,1)$ (used for $e \in \mathbb{R}^m$ and $e \in \mathbb{R}^n$). Write

$$\alpha = \begin{bmatrix} & & & -1 \\ & A & & \vdots \\ & & & -1 \\ 1 & \cdot & \cdot \cdot 1 & 0 \end{bmatrix}$$
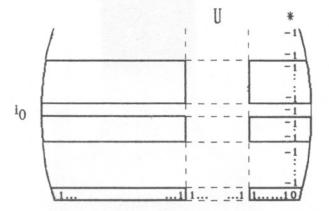
and

$$\mathcal{B} = \begin{bmatrix} & & & 1 \\ & B & & \vdots \\ & & & 1 \\ -1 & \cdot & \cdot \cdot -1 & 0 \end{bmatrix}$$

Thus, $\alpha : \mathbf{I} \times \mathbf{J} \to \mathbb{R}$ and if $i_0 \in T \subseteq I \subseteq \mathbf{I}$ and $U \subseteq J \subseteq \mathbf{J}$, then it is seen that

$$\alpha_{T-i_0}^{-U} = \alpha_{T+\square-i_0}^{J-U} = \alpha_{-(T^c+i_0)}^{-U}$$

is represented by

SECTION 2

The Canonical Parametrization

Let us focus our interest on the motion which the modified LH–algorithm performs in Y. Basically, there are 4 types of "transitions" that occur when algorithmic procedure leaves a vertex, moves along an edge and reaches the next vertex – geometrically speaking. These four transitions can be classified according to whether a subface of Y is being left (reached) or a polyhedron K_i is being left (reached) upon departure (arrival).

Again, the details are explained in [9], Ch.1, SEC.1, hence, for our present purpose we illustrate the four types of transitions for the case that A and B are 3 × 3 matrices by Figure 2.

Here, $H_{T,U} = \{\bar{y}\} \subseteq Y$ denotes the "departure" vertex while the "arrival" vertex varies accordingly, e.g., in case 1a) we have $H_{\hat{T},\hat{U}} = \{\hat{y}\} = H_{T-i_0+i_1,U}$, etc.

Let us first start out with an extensive discussion of case 1a). We shall define a certain version of a parametrization of the edge $H_{T-i_0,U}$ joining $\bar{y}$ and $\hat{y}$, called the "canonical" one. This will suggest (at least partially) the form of the corresponding "tableau" and the way the tableau changes when the algorithm switches from $\bar{y}$ to $\hat{y}$.

To this end, let us now fix an extreme point or vertex

$$Y \supseteq H_{T,U} = \{\bar{y}\}$$

such that $|T| \geq 2$ and $|T| + |U| = n$; put

$$\bar{\lambda} := A_{i\cdot} \bar{y} \quad (i \in T)$$

and define, for some fixed $i_0 \in T$

$$L_{T-i_0}^{-U} := \{\mu = (\gamma,\nu) \in \mathbb{R}^{-U} \times \mathbb{R} \mid \alpha_{T-i_0}^{-U} \mu = 0\} .$$

Then we have

Lemma 2.1:

1. $L_{T-i_0}^{-U}$ is a linear subspace of $\mathbb{R}^{-U} \times \mathbb{R}$ with dimension 1.

2. $\alpha_{i_0\cdot}^{U} \mu \neq 0$ for all $\mu = (\gamma,\nu) \in L_{T-i_0}^{-U}$ with $\gamma \neq 0$.

1 a)

$T, U \quad \longrightarrow \quad T - i_0 + i_1 \ , \ U$

1 b)

$\leftarrow Y_{j_1} = \{ y \in Y \mid y_{j_1} = 0 \}$

$T, U \quad \longrightarrow \quad T - i_0 \ , \ U + j_1$

2 a)

$Y_{j_0} = \{ y \in Y \mid y_{j_0} = 0 \}$

$T, U \quad \longrightarrow \quad T + i_1 \ , \ U - j_0$

2 b)

$T, U \quad \longrightarrow \quad T , \ U - j_0 + j_1$

Fig. 2

Proof:

Follows immediately since the game is nondegenerate.

Definition 2.2:

For $i_0 \in T$ let

$$\overline{\mu}^{i_0} = (\overline{\gamma}^{i_0}, \overline{\nu}^{i_0}) \in L_{T-i_0}^{-U}$$

be defined by the requirement that

(1)
$$\alpha_{i_0}^U \overline{\mu}^{i_0} = 1$$

holds true.

Let us observe that the quantities of generic types $\mu = (\gamma, \nu)$, as considered so far, can be naturally extended to vectors of $\mathbb{R}^n \times \mathbb{R}$ by adding zero coordinates for all $i \in U$. More precisely

$$L_{T-i_0}^{-U} \oplus O_U \subseteq \mathbb{R}^n \times \mathbb{R}$$

is as well a linear subspace of $\mathbb{R}^n \times \mathbb{R}$ with dimension 1 and $\overline{\mu}^{i_0} \oplus O_U$ is a distinctive element of this subspace.
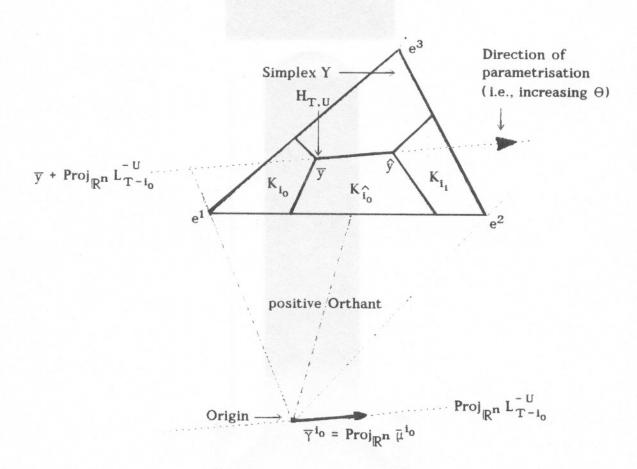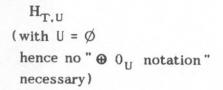
Accordingly,

$$(\overline{y}, \overline{\lambda}) + (L_{T-i_0}^{-U} \oplus O_U)$$

is an affine subspace of $\mathbb{R}^n \times \mathbb{R}$ with distinctive elements $(\overline{y}, \overline{\lambda})$ and $(\overline{y}, \overline{\lambda}) + (\overline{\mu}^{i_0} \oplus O_U)$. In view of Definition 2.2, we have obviously

(2)
$$\alpha_T(\overline{y}, \overline{\lambda}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^T$$

(3)
$$\alpha_T((\overline{y}, \overline{\lambda}) + (\overline{\mu}^{i_0} \oplus O_U)) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} i_0 \qquad \in \mathbb{R}^T$$

If we consider the projection of $\mathbb{R}^n \times \mathbb{R}$ onto $\mathbb{R}^n$, then the situation may be viewed by Fig.3 – assuming that A and B have 3 columns. Also, Fig.3 represents the case in which $\bar{y}$ has positive coordinates – hence $U = \emptyset$.

Simplex Y

$H_{T,U}$

Direction of parametrisation (i.e., increasing Θ)

e^3

e^1

e^2

$\bar{y} + \text{Proj}_{\mathbb{R}^n} L_{T-i_0}^{-U}$

$\bar{y}$

$\hat{y}$

K_{i_0}

$K_{\hat{i}_0}$

K_{i_1}

positive Orthant

Origin ⟶

$\bar{\gamma}^{i_0} = \text{Proj}_{\mathbb{R}^n} \bar{\mu}^{i_0}$

$\text{Proj}_{\mathbb{R}^n} L_{T-i_0}^{-U}$

$H_{T,U}$
(with $U = \emptyset$
hence no " $\oplus 0_U$ notation "
necessary)

Fig. 3

Definition 2.3:

The *canonical parametrization* of $(\bar{y}, \bar{\lambda}) + (L_{T-i_0}^{-U} \oplus O_U)$ is the mapping

$$\theta \longrightarrow (\bar{y}, \bar{\lambda}) - \theta\,(\bar{\mu}^{i_0} \oplus O_U)$$

(4)

$$\mathbb{R} \longrightarrow \mathbb{R}^n \times \mathbb{R}\,.$$

We write $(y^\theta, \lambda^\theta) := (\bar{y}, \bar{\lambda}) - \theta\,(\bar{\mu}^{i_0} \oplus O_U)$ for $\theta \in \mathbb{R}$. (Actually, an additional index i_0 would be appropriate but will be omitted for the sake of not overburdening our notation.)

Theorem 2.4:

Let $\theta \longrightarrow (y^\theta, \lambda^\theta)$ be the canonical parametrization described in Definition 2.3.

1. There is $\bar{\theta}^{i_0} > 0$ such that

$$H_{T-i_0, U} = \{y^\theta \mid 0 \leq \theta \leq \bar{\theta}^{i_0}\}$$

2. $\bar{\theta}^{i_0}$ is explicitly computed by

$$\bar{\theta}^{i_0} = \min\left\{ \frac{\bar{\lambda} - A_{i\cdot}\,\bar{y}}{\bar{\nu}^{i_0} - A_{i\cdot}^{-U}\,\bar{\gamma}^{i_0}} \;\middle|\; i \in T^c,\; \bar{\nu}^{i_0} > A_{i\cdot}^{-U}\,\bar{\gamma}^{i_0} \right\}$$

$$\bigwedge \min\left\{ \frac{\bar{y}_j}{\bar{\gamma}_j^{i_0}} \;\middle|\; j \in U^c,\; \bar{\gamma}_j^{i_0} > 0 \right\}$$

3. For $i \in T - i_0$ and $i' \in T^c + i_0$, $0 < \theta < \bar{\theta}^{i_0}$ we have

$$A_{i\cdot}\,y^\theta = \lambda^\theta > A_{i'\cdot}\,y^\theta$$

4. $y^{\bar{\theta}^{i_0}} =: \hat{y}$ is the second vertex (apart from $\bar{y}$) adjacent to the edge $H_{T-i_0, U}$.

Note that in statement 2 the minimizer decides as to whether case 1a) or case 1b) is prevailing. I.e., if the arg min is some $i_1 \in T^c$ and

$$\theta^{i_0} = \frac{\bar{\lambda} - A_{i_1} \cdot \bar{y}}{\bar{\nu}^{i_0} - A_{i_1}^{-U} \cdot \bar{\gamma}^{i_0}}$$

then we are dealing with case 1a), etc.

Proof of Theorem 2.4:

In view of our previous construction the affine one–dimensional subspace of

$$(\bar{y}, \bar{\lambda}) + (L_{T-i_0}^{-U} \oplus O_U),$$

which is parametrized by

$$\theta \longrightarrow y^\theta \quad (\theta \in \mathbb{R})$$

contains the edge $H_{T-i_0, U}$. In particular, for $\theta = 0$ we have $y^0 = \bar{y} \in H_{T,U}$. In view of the definition property (1) of $\bar{\gamma}^{i_0} = (\bar{\mu}^{i_0}, \bar{\nu}^{i_0})$ we have clearly

$$A_{i_0} \cdot (\bar{\gamma}^{i_0} \oplus O_U) = \bar{\nu}^{i_0} + 1.$$

Also, exploring the $-$ sign in the canonical representation, we come up with

$$A_{i_0} \cdot y^\theta = A_{i_0} \cdot (\bar{y} - \theta (\bar{\gamma}^{i_0} \oplus O_U))$$

$$= \bar{\lambda} - \theta (A_{i_0} \cdot (\bar{\gamma}^{i_0} \oplus O_U))$$

$$(5) \qquad = \bar{\lambda} - \theta (\bar{\nu}^{i_0} + 1)$$

$$= \lambda^\theta - \theta < \lambda^\theta = A_i \cdot y^\theta$$

whenever $\theta > 0$ and $i \in T-i_0$. This implies

$$y^\theta \notin K_{i_0} \qquad (\theta > 0)$$

Hence, for sufficiently small $\theta > 0$ it turns out that $y^\theta \in H_{T-i_0, U}$ and $y^\theta \notin H_{T,U}$.

By a compactness argument, statements 1., 3., and 4. of our theorem follow at once; it remains to show 2.

Now, clearly $y^\theta \in H_{T-i_0, U}$ for all θ that satisfy

$$(6) \qquad y^\theta \geq 0, \; A_{i.} \; y^\theta \leq \lambda^\theta, \; (i \in T^c),$$

and θ^{i_0} is the smallest θ that violates one of conditions (6), i.e., the smallest θ violating either

$$(7) \qquad \bar{y}_j - \theta \, \bar{\gamma}_j^{i_0} > 0$$

for some j with $j \in U^c$, $\bar{\gamma}_j^{i_0} > 0$, or

$$(8) \qquad A_{i.} \; (\bar{y} - \theta \, (\bar{\gamma}^{i_0} + O_U)) < \bar{\lambda} - \theta \, \bar{\nu}^{i_0}$$

i.e.

$$\theta \, (\bar{\nu}^{i_0} - A_i^{-U} \, \bar{\gamma}^{i_0}) < \bar{\lambda} - A_{i.} \; \bar{y}$$

for some $i \in T-i_0$ with $\bar{\nu}^{i_0} < A_i^{-U} \, \bar{\gamma}^{i_0}$.

Obviously, the θ we are looking for is the one given by 2., $\qquad\qquad$ q.e.d.

So far our presentation has just been dealing with the "departure vertex", which, in cases 1a) and 1b) is obtained by sacrificing a condition "$\bar{y} \in K_{i_0}$", i.e., by leaving K_{i_0}.

Now, let us turn to the "arrival", that is, as we want to treat case 1a), the entrance into some new K_{i_1}.

In other words, let us consider the situation in which there is $i_1 \in T^c$ satisfying

$$(9) \qquad \bar{\theta}^{i_0} = \frac{\bar{\lambda} - A_{i_1.} \; \bar{y}}{\bar{\nu}^{i_0} - A_{i_1}^{-U} \, \bar{\gamma}^{i_0}} = \frac{\alpha_{i_1.} (\bar{y}, \bar{\lambda})}{\alpha_{i_1}^U \, \mu^{i_0}}$$

This means that the vertex adjacent to $H_{T-i_0,U}$ (apart from $\overline{y}$) is

$$\{y^{\theta^{i_0}}\} = H_{T-i_0+i_1,\,U}.$$

Let us write $\hat{y} := y^{\theta^{i_0}}$.

Suppose now that, for $\hat{i}_0 \in T-i_0 + i_1$, we want to perform the same procedure as previously, yielding the "canonical parametrization" of $H_{T-i_0+i_1-\hat{i}_0,U}$. This way we obtain the vector

$$\hat{\mu}^{\hat{i}_0} \in L^{-U}_{T-i_0+i_1-\hat{i}_0},$$

which, given $\hat{y}$, is defined by a requirement analogously to (1), i.e., by

$$(10) \qquad \alpha_{\hat{i}_0}^{-U} \hat{\mu}^{\hat{i}_0} = 1 .$$

Define a quantity

$$(11) \qquad \overline{c}^{\,i_0}_{i_1} := - \alpha_{i_1}^{-U} \overline{\mu}^{i_0} .$$

Then it turns out that this quantity may be used to establish a direct relation between $\overline{\mu}^{i_0}$ and $\hat{\mu}^{\hat{i}_0}$ as follows:

Corollary 2.5:

Let $i_1 \in T^c$ satisfy (9) and suppose that $\hat{\mu}^{\hat{i}_0} \in L^{-U}_{T-i_0+i_1-\hat{i}_0}$ is given via (10). Then (using (11)) we have

$$(12) \qquad \hat{\mu}^{\hat{i}_0} = \overline{\mu}^{\hat{i}_0} - \frac{\overline{c}^{\,\hat{i}_0}_{i_1}}{\overline{c}^{\,i_0}_{i_1}} \overline{\mu}^{i_0} \quad \text{for } \hat{i}_0 \neq i_1$$

and

(13)
$$\hat{\mu}^{i_1} = -\frac{\bar{\mu}^{i_0}}{\bar{c}^{i_0}_{i_1}} \ .$$

Proof:

By definition of $\bar{\mu}^{i_0}$ we have

$$\alpha^U_{T-i_0} \bar{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{T-i_0},$$

hence

(14)
$$\alpha^U_{T-i_0+i_1} \bar{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha^U_{i_1} \bar{\mu}^{i_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{c}^{i_0}_{i_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 \ \in \mathbb{R}^{T-i_0+i_1}.$$

$$-\alpha^U_{T-i_0+i_1} \frac{\bar{\mu}^{i_0}}{\bar{c}^{i_0}_{i_1}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 \ \in \mathbb{R}^{T-i_0+i_1}$$

$-\dfrac{\bar{\mu}^{i_0}}{\bar{c}^{i_0}_{i_1}}$ satisfies the defining properties of $\hat{\mu}^{i_1}$ – which proves (13).

Similarly, consider now the case $\hat{i}_0 \neq i_1$. We have

(15)
$$\alpha^U_{T-i_0+i_1-\hat{i}_0} \bar{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha^U_{i_1} \bar{\mu}^{i_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{c}^{i_0}_{i_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 \ \in \mathbb{R}^{T-i_0+i_1-\hat{i}_0}$$

Next, the *canonical parametrization at $\bar{y}$ with respect to $\hat{\imath}_0$* (which is an element of T!) yields the quantity $\bar{\mu}^{\hat{\imath}_0}$, which is uniquely defined by the requirement that it satisfies

$$(16) \qquad \alpha_T^{-U} \, \bar{\mu}^{\hat{\imath}_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow \hat{\imath}_0 \\ \\ \\ \end{matrix} \in \mathbb{R}^T .$$

Thus

$$(17) \qquad \alpha_{T-i_0+i_1-\hat{\imath}_0}^{-U} \, \bar{\mu}^{\hat{\imath}_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha_{i_1}^{-U} \bar{\mu}^{\hat{\imath}_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_1 \\ \\ \\ \end{matrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{c}_{i_1}^{\hat{\imath}_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_1 \\ \\ \\ \end{matrix} \in \mathbb{R}^{T-i_0+i_1-\hat{\imath}_0} .$$

Multiplying (15) with the appropriate factor and subtracting the result from (17), we come up with

$$(18) \qquad \alpha_{T-i_0+i_1-\hat{\imath}_0}^{-U} \left(\bar{\mu}^{\hat{\imath}_0} - \frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{\hat{\imath}_0}} \, \bar{\mu}^{i_0} \right) = 0 \in \mathbb{R}^{T-i_0+i_1-\hat{\imath}_0} .$$

Moreover, using (16) and Definition 2.2, we find

$$\alpha_{\hat{\imath}_0}^{-U} \left(\bar{\mu}^{\hat{\imath}_0} - \frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{\hat{\imath}_0}} \, \bar{\mu}^{i_0} \right)$$

$$(19) \qquad = 1 - \frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{\hat{\imath}_0}} \, \alpha_{\hat{\imath}_0}^{-U} \, \bar{\mu}^{i_0}$$

$$= 1 - 0 = 1 .$$

Concluding, we realize that (18) and (19) show $\bar{\mu}^{\hat{i}_0} - \dfrac{\bar{c}_{i_1}^{\hat{i}_0}}{\bar{c}_{i_1}^{i_0}}\,\bar{\mu}^{i_0}$ to satisfy the conditions

defining $\hat{\mu}^{\hat{i}_0}$ uniquely; this indeed verifies (12), $\hspace{3cm}$ q.e.d.

Corollary 2.6:

For $i_1 \in T^C$ let

$$(20) \qquad \Theta_{i_1} := -\alpha_{i_1} \cdot (\bar{y}, \bar{\lambda}) = -A_{i_1} \cdot \bar{y} + \bar{\lambda}.$$

Then, for $j_1 \in U^C$

$$(21) \qquad \hat{y}_{j_1} = \bar{y}_{j_1} - \dfrac{\bar{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\,\bar{\gamma}_{j_1}^{i_0}$$

and

$$(22) \qquad \bar{\lambda} = \bar{\lambda} - \dfrac{\bar{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\,\bar{\nu}^{i_0}.$$

Proof:

Indeed, since

$$\bar{c}_{i_1}^{i_0} = -\alpha_{i_1}^U\,\bar{\mu}^{i_0} = \bar{\nu}^{i_0} - A_{i_1}^{U^C}\,\bar{\gamma}^{i_0},$$

we can use (9) and (4) in order to obtain

$$\hat{y} = \bar{y} - \bar{\theta}^{i_0}(\bar{\gamma}^{i_0}\,0\,O_U) = \bar{y} - \dfrac{\bar{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\,(\bar{\gamma}^{i_0} \oplus O_U).$$

Let us pause for some reflection.

The development as presented so far describes the transition from the vertex $\bar{y}$ to the adjacent vertex $\hat{y}$ (assuming we consider case 1a, that is $(T, U) \rightarrow (T-i_0+i_1, U)$). For-

mulae (12) and (13), with some good will, may be interpreted as an analogue to the well known "rectangle rule" of linear programming.

Indeed, in order to compute $\hat{y}$ by means of $\overline{y}$ we need certain quantities $\overline{\gamma}_{\cdot}$, $\overline{c}_{\cdot}$, $\overline{\Theta}_{\cdot}$. Moreover, in order to compute the next adjacent vertex, we have to start with $\hat{y}$ and use the corresponding quantities, say $\hat{\gamma}_{\cdot}$, $\hat{c}_{\cdot}$, and $\hat{\Theta}_{\cdot}$. Hence, we have to find a computational rule for the transition of these quantities. To this end, we focus on corollary 2.5 which indeed presents a version of the rectangle rule for a transformation of $\overline{\mu}$ to $\hat{\mu}$. This transformation in turn depends on the quantities c: as indicated by the result of Corollary 2.5. This means that we have to establish the rectangle rule for the quantities c and Θ as well. It seems advisable to combine all necessary quantities in what is usually called the "tableau" assigned to the vertex $\overline{y}$. This tableau should at least contain quantities $\overline{y}$, $\overline{\gamma}$, $\overline{c}$, $\overline{\Theta}$.

There is, however, a further obstacle: So far we have only discussed case (1a). There are four other cases which conceivably would yield additional quantities to be represented in our tableau to be constructed. At this instant, therefore, we prefer to present the tableau without further motivation. Rather, the quantities that will appear shall be justified by further computation and transformational arguments following in the next sections.

SECTION 3

The Tableau

The peculiar pattern of the LH–algorithm as presented in SECTION 1 asks for a slightly more complicated version of the tableau attached to a certain vertex $\{\overline{y}\} = H_{T,U}$. It should be noted that we still are discussing the situation in Y only. There is obviously a similar tableau attached to any vertex in the corresponding simplex X.
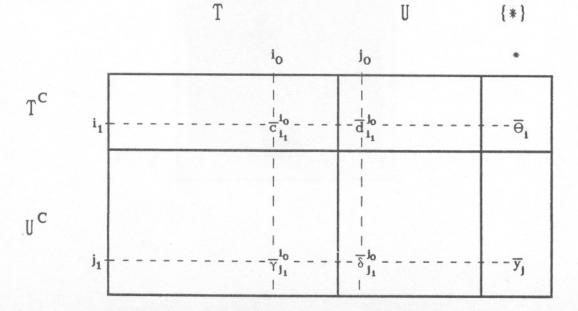
The tableau to be presented below, contains 6 different regions, four of them corresponding to the defining subsets T and U and their complements respectively. According to what kind of transition (corresponding to the cases 1a) to 2b)) is necessary, the "rectangle rule" will switch the coefficients depending on the positions in the various regions of the tableau. Ideally, in order to compute the transition formula (that is to verify the "rectangle rule"), we would have to consider the behavior of each of the coefficients in the 6 regions depending on four possible cases of transition; that is, we would have to perform 24 computational procedures. To proceed with this task explicitly would put some strain on the reader and is not actually necessary in all instances. We will hence concentrate on a few dominant computational procedures and leave the remaining ones to the reader.

Definition 3.1:

Let $H_{T,U} = \{\overline{y}\} \subseteq Y$ be a vertex in Y. The *tableau* corresponding to $\overline{y}$ is the mapping

$$\Upsilon_{\overline{y}} : (T^C \cup U^C) \times (T \cup U \cup \{*\}) \to \mathbb{R}$$

defined by $\Upsilon_{\overline{y}}(s,r) = \overline{T}_{sr}$ ($s \in T^C \times U^C$, $r \in T \times U \times \{*\}$) where $\overline{T}$ is the $m \times (n+1)$–matrix

The entries of the matrix are defined as follows.

The last column contains

$$(2) \qquad \Theta = -\mathcal{O}_{-T}(\bar{y},\lambda) = A_{-T}\,\bar{y} + \lambda\,e_{-T}$$

(see Corollary 2.6) and the vector $\bar{y}_{-U}$ (i.e. the positive coordinates of $\bar{y}$). Next, $\bar{\gamma}$: is obtained via $\bar{\mu}^{i_0} = (\bar{\gamma}^{i_0}, \bar{\nu}^{i_0}) \in \mathbb{R}^{J-U}$ ($i_0 \in T$) and the requirement that

$$(3) \qquad \mathcal{O}_T^{-U}\,\bar{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = e_T^{i_0}$$

(cf. Lemma 2.1 and Definition 2.2).

Similarly, $\bar{c}$: is obtained by inspecting (11) in SEC.2, that is

$$(4) \qquad \bar{c}^{i_0} = -\mathcal{O}_{-T}^{-U}\,\bar{\mu}^{i_0} \in \mathbb{R}^{I-T}$$

Finally, the quantities $\bar{d}$: and $\bar{\delta}$: have not been motivated as yet, the formal definition is given as follows.

For $j_0 \in U$ vectors $\bar{\delta}^{j_0} \in \mathbb{R}^{J-U}$ and $\bar{\rho}^{j_0} = (\bar{\delta}^{j_0}, \bar{\sigma}^{j_0}) \in \mathbb{R}^{J-U}$ are defined by the requirement that

$$(5) \qquad \mathcal{O}_T^{j_0} = \mathcal{O}_T^{-U}\,\bar{\rho}^{j_0}$$

holds true. By non–degeneracy, $\bar{\rho}^{j_0}$ is indeed well defined (this is in fact the normal paradigm of changing the base in the L.P.–case). Accordingly, we define for $j_0 \in U$, the vector $\bar{d}^{j_0} \in \mathbb{R}^{I-T}$ by

$$(6) \qquad \bar{d}^{j_0} = A_{-T}^{j_0} - \mathcal{O}_{-T}^{-U}\,\bar{\rho}^{j_0}.$$

Remark 3.2:

There is no harm in visualizing $\Upsilon_{\bar{y}}$ by $\overline{\Upsilon}$ – however, with respect to a matrix the *ordering* of rows and columns sometimes is important – thus, in a rigorous representation, $\Upsilon_{\bar{y}}$ is actually an *equivalence class* of matrices – to be obtained by permuting rows and columns of $\overline{\Upsilon}$ (including the "row and column indices").

Given the definition of "the tableau" for $H_{T,U} = \{\hat{y}\}$, let us turn to the rectangle rule, which is a mapping of transforming general tableaus.

Fix $U \subseteq J$ and $T \subseteq I$.

Let Υ be a mapping (the "(T,U)–tableau")

$$\Upsilon : (T^c \cup U^c) \times (T \cup U) \to \mathbb{R}$$

and let $\qquad i_0 \in T, i_1 \in I - T$.

Let

$$\hat{T} := T - i_0 + i_1$$
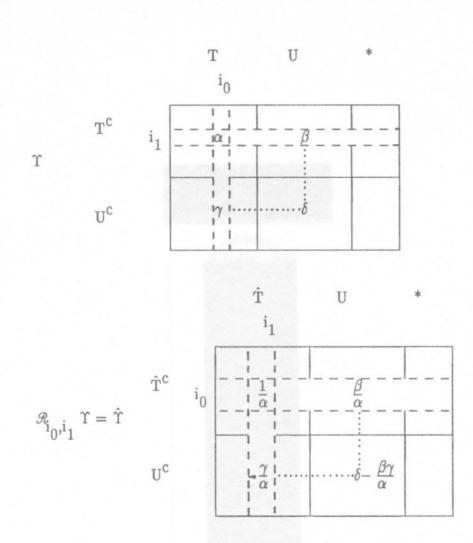
and let $\hat{\Upsilon}$ denote a mapping $((\hat{T},U)$–tableau)

$$\hat{\Upsilon} : (\hat{T}^c \cup U^c) \times (\hat{T} \cup U) \to \mathbb{R} .$$

The "rectangle rule" (for (i_0, i_1)) is a mapping that sends (T,U)–tableaus into $(\hat{T},U)$–tableaus, say

$$\mathscr{R}_{i_0, i_1} : \{\Upsilon\} \to \{\hat{\Upsilon}\}$$

as indicated by the familiar figure

Of course, application of this kind of rectangle rule will correspond to a transfer in case
1a)

$$H_{T,U} \to H_{T-i_0+i_1,U} = H_{\hat{T},U}$$

If we have to deal with a transfer

$$H_{T,U} \to H_{T-i_0,U+j_1} = H_{\hat{T},\hat{U}}$$

(corresponding to case 1b)), then there is a corresponding $\mathcal{R}_{i_0,j_1}$. Here, $\mathcal{R}_{i_0,j_1} \Upsilon = \hat{\Upsilon}$ is

a mapping as indicated via

The ordering of rows and columns is, of course, arbitrary – which is why a "tableau" perhaps is better thought of as a "mapping". The fact that we have four kinds of transitions – and hence four kind of tableaus – must be taken care of extensively while implementing the algorithm.

Theorem 3.3:

Let $H_{T,U} = \{\bar{y}\} \subseteq Y$ be a vertex in Y and let $i_0 \in T$, $|T| \geq 2$. Suppose, $\hat{y}$ is the vertex adjacent to $H_{T-i_0,U}$ other than $\bar{y}$ and assume that $\{\hat{y}\} = H_{T-i_0+i_1,U}$. Then the corresponding tableaus satisfy

$$(7) \qquad \Upsilon_{\hat{y}} = \mathscr{R}_{i_0 i_1} \Upsilon_{\bar{y}}$$

Proof:

We have to compute the transition for 6 types of entries in $\Upsilon_{\hat{y}}$ and $\Upsilon_{\bar{y}}$.

Let us first concentrate on "the block $U^C \times T$" of $\Upsilon_{\bar{y}}$, the transitions should be

$$\gamma \to -\frac{\gamma}{\alpha} \qquad \text{for } j_1,\, i_0$$

$$\delta \to \delta - \frac{\beta\gamma}{\alpha} \qquad \text{for } j_1,\, \hat{i}_0 \neq i_0$$

The entries of $\Upsilon_{\bar{y}}$ are

$$\gamma = \bar{\mu}^{i_0}_{j_1} \qquad \alpha = \bar{c}^{i_0}_{i_1}$$

$$\beta = \bar{c}^{\hat{i}_0}_{i_1} \qquad \delta = \bar{\mu}^{\hat{i}_0}_{j_1}$$

Thus, the rectangle rule requires

$$\frac{1}{\alpha} = \frac{1}{\bar{c}^{i_0}_{i_1}} \qquad \frac{\beta}{\alpha} = \frac{\bar{c}^{\hat{i}_0}_{i_1}}{\bar{c}^{i_0}_{i_1}}$$

$$-\frac{\gamma}{\alpha} = -\frac{\bar{\mu}^{i_0}_{j_1}}{\bar{c}^{i_0}_{i_1}} \qquad \delta - \frac{\beta\gamma}{\alpha} = \bar{\mu}^{\hat{i}_0}_{j_1} - \bar{\mu}^{i_0}_{j_1}\frac{\bar{c}^{\hat{i}_0}_{i_1}}{\bar{c}^{i_0}_{i_1}}$$

As for row j_1, consult Corollary 2.5. Clearly, (13) in SEC.2 tells us that $-\frac{\gamma}{\alpha}$ is indeed the (j_1, i_1) entry of $\Upsilon_{\hat{y}}$ while (12), SEC.2 indicates that $\delta - \frac{\beta\gamma}{\alpha}$ is the (j_1, i) entry of $\Upsilon_{\hat{y}}$ (in "the block $\hat{U}^C \times U$").

The remaining computations, though sometimes tedious, are a mere formalism. By virtue of our considerations in SEC.2 we know that the tableau entries of $\Upsilon_{\bar{y}}$ determine $\hat{y}$. Now, as the entries of some $\Upsilon_{.}$ are defined formally by (1)...(6), we just have to verify the rectangle rule via the definitions (1)...(6).

To this end, fix $\hat{i}_0 \in \hat{T}$, $\hat{i}_1 \in \hat{T}^C$. Also, denote the entries of $\Upsilon_{\hat{y}}$ with a $\hat{\ }$, e.g., $\hat{\mu}$, $\hat{\rho}$,... etc. the same notation has been employed in SEC.2.

First of all, let us take the block $U^C \times U$, i.e., consider $\rho = (\delta, \sigma)$.

As $\hat{\rho}^{j_0}$ (for $j_0 \in U = \hat{U}$) is defined via

$$(8) \qquad \sigma_{\hat{T}}^{-U}\,\hat{\rho}^{j_0} = \sigma_{\hat{T}}^{j_0}$$

we compute

$$(9) \qquad \sigma_{\hat{T}}^{-U}\,(\bar{\rho}^{j_0} - \frac{d_{i_1}^{j_0}}{\bar{c}_{i_1}^{i_0}}\,\bar{\mu}^{i_0})$$

the coordinates $i \in \hat{T}$ of (9) are given by

$$\sigma_i^{-U}\,\bar{\rho}^{j_0} + \frac{a_{i_1 j_0} - \sigma_i^{-U}\,\bar{\rho}^{j_0}}{\sigma_{i_1}^{-U}\,\bar{\mu}^{i_0}}\,\sigma_i^{-U}\,\bar{\mu}^{i_0}$$

$$= \begin{cases} \sigma_i^{j_0} + \dfrac{a_{i_1 j_0} - \sigma_i^{-U}\,\bar{\rho}^{j_0}}{\sigma_{i_1}^{-U}\,\bar{\mu}^{i_0}}\cdot 0 & \text{for } i \neq i_1 \\[3em] \sigma_{i_1}^{-U}\,\bar{\rho}^{j_0} + \dfrac{a_{i_1 j_0} - \sigma_i^{-U}\,\bar{\rho}^{j_0}}{\sigma_{i_1}^{-U}\,\bar{\mu}^{i_0}}\,\sigma_{i_1}^{-U}\,\bar{\mu}^{i_0} & \text{for } i = i_1 \end{cases}$$

$$= \quad \sigma_i^{j_0}$$

that is, the coordinates are those of the right hand side of (8). Thus, the term in parenthesis of (9) must be the left hand side of (8) – this takes care of the $\bar{\delta}$:–entries in the block $U^C \times U$.

Next, the $\bar{y}_.$ –entries, i.e. the block $U^c \times \{*\}$, are obviously taken care of by Corollary 2.6, i.e., by formula (20) of SEC.2.

We proceed with the $\bar{c}$:–entries in the $T^c \times T$ block, using the fact that the rectangle rule has already been established for $\bar{\mu}$ vs. $\hat{\mu}$. Hence, using the definition as provided in (4), we proceed as follows: First, for all $\hat{i}_0 \neq i_1$:

$$\hat{c}^{\hat{i}_0}_{i_1} = -\alpha^{-U}_{\hat{i}_1} \hat{\mu}^{\hat{i}_0} \qquad \text{by (4)}$$

(10)
$$= -\alpha^{-U}_{\hat{i}_1} (\bar{\mu}^{\hat{i}_0} - \frac{c^{\hat{i}_0}_{i_1}}{\bar{c}^{\hat{i}_0}_{i_1}} \bar{\mu}^{i_0})$$

<div align="center">(by Corollary 2.5, i.e., by (12), SEC.2)</div>

$$= \begin{cases} \bar{c}^{\hat{i}_0}_{i_1} - \dfrac{\bar{c}^{\hat{i}_0}_{i_1}}{\bar{c}^{\hat{i}_0}_{i_1}} \bar{c}^{i_0}_{i_1} & \text{for } \hat{i}_1 \neq i_0 \qquad \text{(by (4))} \\[3em] \dfrac{c^{\hat{i}_0}_{i_1}}{\bar{c}^{i_0}_{i_1}} & \text{for } \hat{i}_1 = i_0 \qquad \text{(by (3))} \end{cases}$$

Similarly, for $\hat{i}_0 = i_1$

$$\hat{c}^{i_1}_{i_1} = -\alpha^{-U}_{\hat{i}_1} \hat{\mu}^{i_1}$$

$$= -\alpha^{-U}_{\hat{i}_1} (-\frac{\bar{\mu}^{i_0}}{\bar{c}^{i_0}_{i_1}})$$

(11)

$$= \begin{cases} -\dfrac{\bar{c}_{i_1}^{\,i_0}}{\bar{c}_{i_1}^{\,i_0}} & \text{for } \hat{\imath}_1 \neq i_0 \qquad \text{by (4)} \\[3em] \dfrac{1}{\bar{c}_{i_1}^{\,i_0}} & \text{for } \hat{\imath}_1 = i_0 \qquad \text{by (3)} \end{cases}$$

Obviously, (10) and (11) establish the rectangle rule for the $T^C \times T$ block. As for the d: in block $T^C \times U$; we have by (6)

$$\hat{d}_{\hat{\imath}_1}^{\,\hat{\jmath}_0} = a_{\hat{\jmath}_0 \hat{\imath}_1} - \alpha_{\hat{\imath}_1}^{-U}\left(\bar{\rho}^{\,\hat{\jmath}_0} - \dfrac{\bar{d}_{\hat{\imath}_1}^{\,\hat{\jmath}_0}}{\bar{c}_{i_1}^{\,i_0}}\bar{\mu}^{\,i_0}\right)$$

(12)

$$= \begin{cases} \bar{d}_{\hat{\imath}_1}^{\,\hat{\jmath}_0} - \dfrac{\bar{c}_{i_1}^{\,i_0}\,\bar{d}_{i_1}^{\,\hat{\jmath}_0}}{\bar{c}_{i_1}^{\,i_0}} & \text{for } \hat{\imath}_1 \neq i_0 \\[3em] \dfrac{\bar{d}_{i_1}^{\,\hat{\jmath}_0}}{\bar{c}_{i_1}^{\,i_0}} & \text{for } \hat{\imath}_1 = i_0 \end{cases}$$

using (6), (4), (3), and (5).

Finally, the $\bar{\Theta}$ in blocks $T^C \times \{*\}$ are transformed by using (2), thus, for $\hat{\imath}_1 \neq i_0$:

$$\hat{\Theta}_{\hat{i}_1} = -\alpha_{\hat{i}_1 \cdot}(\hat{y}, \hat{\lambda})$$

$$(13) \qquad = \bar{\Theta}_{\hat{i}_1} + \alpha_{\hat{i}_1 \cdot}^U \frac{\bar{\Theta}_{\hat{i}_1}^{i_0}}{\bar{c}_{\hat{i}_1}^{i_0}} \bar{\mu}^{i_0} \qquad \text{(by Corollary 2.6)}$$

$$= \bar{\Theta}_{\hat{i}_1} - \frac{\bar{\Theta}_{\hat{i}_1} \bar{c}_{\hat{i}_1}^{i_0}}{\bar{c}_{\hat{i}_1}^{i_0}} \qquad \text{(by (4))}$$

and for $\hat{i}_1 = i_0$:

$$\hat{\Theta}_{\hat{i}_1} = -\alpha_{i_0 \cdot}(\hat{y}, \hat{\lambda})$$

$$= \underbrace{-\alpha_{i_0 \cdot}(\bar{y}, \lambda)}_{0} + \frac{\bar{\Theta}_{\hat{i}_1}}{\bar{c}_{\hat{i}_1}^{i_0}} \alpha_{i_0 \cdot}^U \bar{\mu}^{i_0} \qquad \text{(by Corollary 2.6)}$$

$$(14) \qquad \qquad 0$$

$$= \frac{\bar{\Theta}_{\hat{i}_0}}{\bar{c}_{\hat{i}_1}^{i_0}} \qquad \text{(by (3))} \qquad\qquad \text{q.e.d.}$$

The further development is rather straight forward. There are four kinds of possible transitions $H_{T,U} \to H_{\hat{T}, \hat{U}}$ when passing from one vertex to an adjacent one via some edge. To each of these transitions, there corresponds a rectangle rule – we have explicitly indicated two of them. Now we have

Theorem 3.4:

Let $H_{T,U} = \{\bar{y}\}$ and $H_{\hat{T}, \hat{U}} = \{\hat{y}\}$ be adjacent vertices. Suppose, $\Upsilon_{\bar{y}}$ is the tableau corresponding to $\bar{y}$. Then $\Upsilon_{\hat{y}}$ is obtained by the rectangle rule (i.e., the one corresponding to $H_{T,U} \to H_{\hat{T}, \hat{U}}$).

Proof:

We should discuss briefly all four cases 1a) – 2b). Now, 1a) has already been dealt with. As for 1b) we return to the presentation exhibited in 2.4 and 2.5; here we have to replace formula (9) in SEC.2 by

$$(15) \qquad \eta^{i_0} = \frac{\overline{y}_{j_1}}{\overline{\gamma}^{i_0}_{j_1}}$$

thus assuming that a transition

$$H_{T,U} \rightharpoonup H_{T-i_0,U+j_1}$$

takes place. Again we compute $\hat{y} = y^{\eta^{i_0}}$.

In doing so, we realize that the quantities of the tableau $\tau_{\overline{y}}$ are sufficient in order to perform all necessary computations. Hence, it suffices to again check the rectangle rule ($\mathscr{R}_{i_0,j_1}$, that is) for case 1b). This amounts to joggling around the quantities specified by formulae (2) – (6). As the details are to be perceived by walking the way parallel to the one that led to the treatment of 1a), we shall not offer a further discussion.

As to cases 2), we abbreviate the discussion – in principle we have to introduce another canonical parametrization. Consider the vertex

$$\{\overline{y}\} = H_{T,U}$$

and let $\overline{\lambda} = A_{i.} \, \overline{y} \ (i \in T)$. Pick $j_0 \in U$; it follows from nondegeneracy that

$$(16) \qquad L_T^{j_0-U} = \{\rho = (\delta,\sigma) \in \mathbb{R}^{-(U-j_0)} \times \mathbb{R} \mid \alpha_T^{-(U-j_0)}(\delta,\sigma) = 0\}$$

is a linear subspace of $\mathbb{R}^{-U} \times \mathbb{R}$ of dimension 1.

Again in view of nondegeneracy it is clear that equation (16), i.e.

$$\alpha_T^{j_0} = \alpha_T^{-U} \, \rho_{-j_0}^{j_0}$$

defines the vector $\bar{\rho}^{j_0}$ uniquely and the mapping

(17) $\qquad \theta \to (\bar{y},\lambda) - \theta\,(\bar{\rho}^{j_0} \oplus O_{U-j_0})$

$\qquad\qquad \mathbb{R} \to \mathbb{R}^n \times \mathbb{R}$

defines the canonical parametrization of the affine subspace

$$(\bar{y},\lambda) + (L_T^{j_0-U} \oplus O_{U-j_0}).$$

Of course, the projection

$$\theta \to y^\theta = \bar{y} - \theta\,(\bar{\delta}^{j_0} \oplus O_{U-j_0})$$

also parametrizes an affine subspace of $\mathbb{R}^n$; this latter one contains $H_{T,U-j_0}$ (and

$H_{T,U}$). Thus, the analogue of Theorem 2.4 is given by:

(18) $\qquad$ There is $\bar{\theta}^{j_0} > 0$ such that

$$H_{T,U-j_0} = \{y^\theta \mid 0 \le \theta \le \bar{\theta}^{j_0}\},$$

(19) $\qquad \bar{\theta}^{j_0}$ is explicitly computed by

$$\bar{\theta}^{j_0} = \min\left\{ \left. \frac{\lambda - A_{i\cdot}\,\bar{y}}{\bar{\sigma}^{j_0} - A_{i\cdot}^{-U}\,\bar{\delta}^{j_0}} \;\right|\; i \in T^c, \bar{\sigma}^{j_0} > A_{i\cdot}^{-U}\,\bar{\delta}^{j_0} \right\}$$

$$\bigwedge\; \min\left\{ \left. \frac{\bar{y}_j}{\bar{\delta}_j^{j_0}} \;\right|\; j \in U^c, \bar{\delta}_j^{j_0} > 0 \right\}$$

From this vantage point the reader now views the scene that has so extensively been discribed in case 1. We will leave him there to his own efforts – if necessary.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ q.e.d.
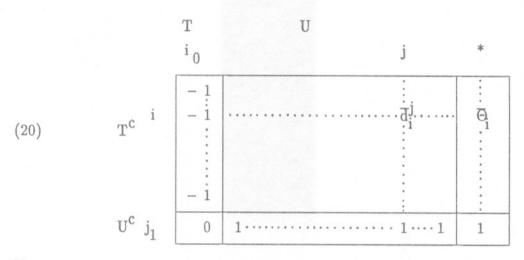
As to this section, our final task is to shortly consider the *initial tableau*. This turns out to be of a nice and simple shape.

Theorem 3.5:

Let $j_1 \in J$ and $\bar{y} = e^{j_1} = (0,\ldots,0,1,0,\ldots,0) \in \underline{Y}$.

Suppose $i_0 \in I$ is such that

$$a_{i_0 j_1} = \max_{i \in I} a_{i j_1}.$$

Then $\{e^{j_1}\} = H_{T,U}$ with $T = \{i_0\}$ and $U = J - \{j_1\}$ is a vertex in Y and the corresponding tableau is given by $\Upsilon_{e^{j_1}}$ which is indicated by the following matrix

(20)

		T	U		
		i_0		j	$*$
T^C	i	-1 -1 -1		$\bar{d}_i^j$	$\bar{\Theta}_i$
U^C	j_1	0	$1 \cdots\cdots\cdots 1 \cdots 1$		1

Here

(21)
$$\bar{\Theta}_i = a_{i_0 j_1} - a_{i j_1} \quad (i \in T^C)$$

and

(22)
$$\bar{d}_i^j = a_{ij} - a_{i j_1} + a_{i_0 j_1} - a_{i_0 j} \quad (i \in T^C, j \in U)$$

Proof:

Clearly, $\{e^{j_1}\} = H_{T,U}$ is a vertex. Note that

$$\bar{\lambda} = a_{i_0 j_1}$$

holds true. All we have to do is verifying the entries of the matrix using formulae (2) – (6).

In view of (2) we have

$$\Theta_i^j = \bar{\lambda} - A_{i \cdot} \; e^{j_1} = a_{j_0 j_1} - a_{i j_1}$$

which shows (21). Next, exploit (5) in order to obtain

$$\bar{\rho}^j = (\alpha_T^{-U})^{-1} \, \alpha_T^j$$

$$= \begin{bmatrix} a_{i_0 j_1} & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{i_0 j} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ -1 & a_{i_0 j_1} \end{bmatrix} \begin{bmatrix} a_{i_0 j} \\ 1 \end{bmatrix}$$

$$= (1, \; a_{i_0 j_1} - a_{i_0 j})$$

The first coordinate of $\bar{\rho}^j$ is $\bar{\delta}_{j_1}^j$, which equals 1.

Next, (6) leads to

$$\bar{d}_i^j = a_{ij} - \alpha_i^U \, \bar{\rho}^j$$

$$= a_{ij} - (a_{i j_1}, -1) \, (1, \; a_{i_0 j_1} - a_{i_0 j})$$

and thus (22).

The remaining computations, easy as they are, will not be carried out explicitly.

<div align="right">q.e.d.</div>

SECTION 4

Implementing the Algorithm

Suppose that, starting out from some vertex $\{\bar{y}\} = H_{T,U}$, we have "left K_{i_0}" ($i_0 \in T$), hence a transition takes places along the edge $H_{T-i_0,U}$ and case 1a) or 1b) will prevail.

From Theorem 4 of SEC.2 and the following presentation we know, that this depends essentially on the minimizing argument that yields $\bar{\theta}^{i_0}$ in 2.4.2. Clearly, the quantities competing for this minimizer are basically available in the tableau $\Upsilon_{\bar{y}}$. For, in solving the definition presented by formulae (2) and (4) of SEC.4 into 2.4.2 it turns out that we have to look for the minimizer that yields the expression

$$\min \left\{ \frac{\Theta_i}{\bar{c}_i^{i_0}} \,\Big|\, i \in T^c, \bar{c}_i^{i_0} > 0 \right\} \wedge \min \left\{ \frac{\bar{y}_j}{\bar{\gamma}_j^{i_0}} \,\Big|\, j \in U^c, \bar{\gamma}_j^{i_0} > 0 \right\}$$

Verbally, this means that we take the quotients of column $*$ and column i_0 of $\Upsilon_{\bar{y}}$ "coordinatewise" and look for the minimizing row. According to whether this yields some $i_1 \in T^c$ or some $j_1 \in U^c$, we end up with 1a) or 1b) respectively. Note that the quotient minimizing row is unique by non–degeneracy.

It is not hard to prove the generalization of this.

Theorem 4.1:

Let $\{\bar{y}\} = H_{T,U}$ be a vertex in $\underline{Y}$ with tableau $\Upsilon_{\bar{y}}$. Denote the last column of $\Upsilon_{\bar{y}}$ by $\tau_{.*} (= (\Theta, \bar{y}_{-U}))$. Let $H_{T',U'}$ be an adjacent edge (so $T' = T-\rho$, or $U' = U-\rho$) and let $\tau_{.\rho}$ be the ρ'th column of $\Upsilon_{\bar{y}}$. Next, let σ be a row of $\Upsilon_{\bar{y}}$ (i.e. $\sigma \in T^c$ or $\sigma \in U^c$) such that

(1)
$$\frac{\tau_{\sigma*}}{\tau_{\sigma\rho}} = \min \left\{ \frac{\tau_{\sigma'*}}{\tau_{\sigma'\rho}} \,\Big|\, \tau_{\sigma'\rho} \neq 0, \sigma' \in \{\text{rows of } \Upsilon_{\bar{y}}\} \right\}$$

Then the following holds true:

1. σ is uniquely determined

2. $\{\hat{y}\} = H_{\hat{T},\hat{U}}$ with $\hat{T} = T' + \sigma$

   or $\hat{U} = U' + \sigma$ (chosen appropriately) is the vertex other than $\bar{y}$ that is adjacent to $H_{T',U'}$.

3. $\Upsilon_{\hat{y}} = \mathscr{R}_{\sigma\rho} \, \Upsilon \, \bar{y}$

Proof:

Obvious.

Finally, we have to ponder about the terminating condition. To this end, consider the version of the LH–Algorithm discussed in [9], CH.1, SEC.1, Theorem 1.14, which is based on the set

$$\mathscr{S}_n = \{x \in X \mid x_i > 0 \quad y \in K_i \ (i \in I)$$
$$y_j > 0 \quad x \in L_j \ (j \in J-n)\}$$

Geometrically, this means that the starting vertex in Y is e^n and that the first $H_{T,U}$ is some $H_{i_0, J-n}$. Now, obviously the process terminates once either n is added to the indices in U as to constitute $H_{T,U}$ with $n \in U$ or n is removed from R such that $G_{R,V}$ satisfies $n \notin R$.

In any case, the algorithm terminates once index n appears afresh the first time. If we complete the rectangle rule, then the equilibrium coordinates can be simply read from the tableau as they are listed in the last row. Concluding, the implementing procedure for the modified LH–algorithm is described as follows:

Given Matrices A and B, perform the following steps

STEP INITIALIZE:

1. Choose $n_0 \in J$ arbitrarily.

2. Choose $i_0 \in I$ such that

(2) $$a_{i_0 n_0} = \max_i a_{i n_0}.$$

3.   Choose $i_0 \in J$ such that

(3)
$$b_{i_0 j_0} = \max_j b_{i_0 j}.$$

If $j_0 = n_0$, then STOP. (e^{i_0}, e^{j_0}) is a (pure) equilibrium point. Otherwise, set up

STEP INITIAL TABLEAUS

The initial tableau arising from matrix A is uniquely described by formulae (20), (22), (23).

This defines $\Upsilon_{\bar{y}}$ with $\bar{y} = e^{n_0}$.

The initial tableau arising from matrix B is obtained by exchanging B^T and A, J and I, n and m etc. That is, we have

Here

(5)
$$\bar{\bar{\Theta}}_j = b^T_{j_0 i_0} - b^T_{j i_0}$$
$$= b_{i_0 j_0} - b_{i_0 j}$$

and

(6)
$$\bar{\bar{d}}^i_j = b^T_{ji} - b^T_{j i_0} + b^T_{j_0 i_0} - b^T_{j_0 i}$$
$$= b_{ij} - b_{i_0 j} + b_{i_0 j_0} - b_{i j_0}$$

Having thus established the initial tableaus, *CONTINUE* with the algorithm.

STEP CONTINUE

Having obtained the information j_0 from the B–tableau, determine i_1 (or j_1) as to be the minimizer of the (well defined) quotients of column * and column j_0 in the A tableau, i.e.,

$$\min_{\sigma'} \frac{\tau_{\sigma'}{}^*}{\tau_{\sigma'j_0}} = \frac{\tau_\sigma{}^*}{\tau_{\sigma j_0}}$$

Traditionally, σ is called the "pivot". Apply the rectangle rule, say $\mathcal{R}_{j_0 i_1}$ to the A–tableau. *CONTINUE* with the B–side.

Generally, the information contained in an index ρ ("the pivot") from the previous side determines a column in the tableau of the present side. The minimizer σ of the quotients of the last column and column ρ is the next pivot. It determines the rectangle rule $\mathcal{R}_{\rho\sigma}$ to be applied to the present tableau. Also, the pivot σ is the information to be used at the next step with the tableau of the other side.

As far as the pivot satisfies $\sigma \neq n_0 \in J$, *CONTINUE* with this step; otherwise move to *TERMINATE*.

STEP TERMINATE

If the pivot satisfies $\sigma = n_0 \in J$, the algorithm STOPS (after the last $\mathcal{R}_{\rho n_0}$ has been performed).

The A–tableau as depicted in (1) of SEC.3 contains the positive coordinates $j \in U^c$ of $\bar{y}$, i.e., the vector $\bar{y}_{-U} = \bar{y}_{U^c}$ in "the block $U^c \times \{*\}$". Correspondingly, the B tableau contains some $\bar{x}_{V^c}$ in the corresponding block. Augmenting these quantities by an appropriate string of zero's yields an equilibrium point $(\bar{x}, \bar{y})$.

Program LH :

START

Choose
Vertex

INPUT
A, B, N0

Strategies

Determine initial
Strategy Y using
N0, X as best reply
to Y and Y̲ as best
reply to X. This
yields I0,J0.

N0=J0 ?   Y

N

Inittab

Construct
Initial Tableaus
TAB1   TAB2

J0 ϵ T∪U ?   N

Y

Switch TAB1

CHANGE TAB 1
get new J0

J0 ϵ R∪V ?   N

Y

Switch TAB2

CHANGE TAB 1
get new J0

Get X,Y   Compute X,Y

END

OUTPUT
X,Y

STOP

Subprogram Inittab :

START

INPUT
MAT,IO,JO,
λ,W

W = 0 ?   Y

N

Tranpose MAT

$$TAB = \left\langle \begin{array}{c|ccc|c} -1 & & & & \\ \vdots & \left[\begin{array}{c} \text{cf. SEC 3,(20)} \\ \text{SEC 4,(4)} \end{array}\right] & & & \lambda-MAT.j_0 \\ -1 & & & & \\ \hline 0 & 1 & \cdots\cdots\cdots & 1 & 1 \end{array} \right\rangle$$

N   W = 0 ?   Y

$R=\{IO\};V=\{M+1,..,M+N\}\backslash\{M+JO\}$
compute R^c and V^c
ROW=$R^c \cup V^c$; COL=$R \cup V$

$T=\{IO+N\};U=\{1,...,N\}\backslash\{JO\}$
compute T^c and U^c
ROW=$T^c \cup U^c$; COL=$T \cup U$

$$TAB = \left\langle \begin{array}{ccc} 0 & COL & 0 \\ R & & \\ O & & TAB \\ W & & \end{array} \right\rangle$$

RETURN
TAB

STOP

Subprogram Switchtab :

The Subprogram is divided into two subprograms:
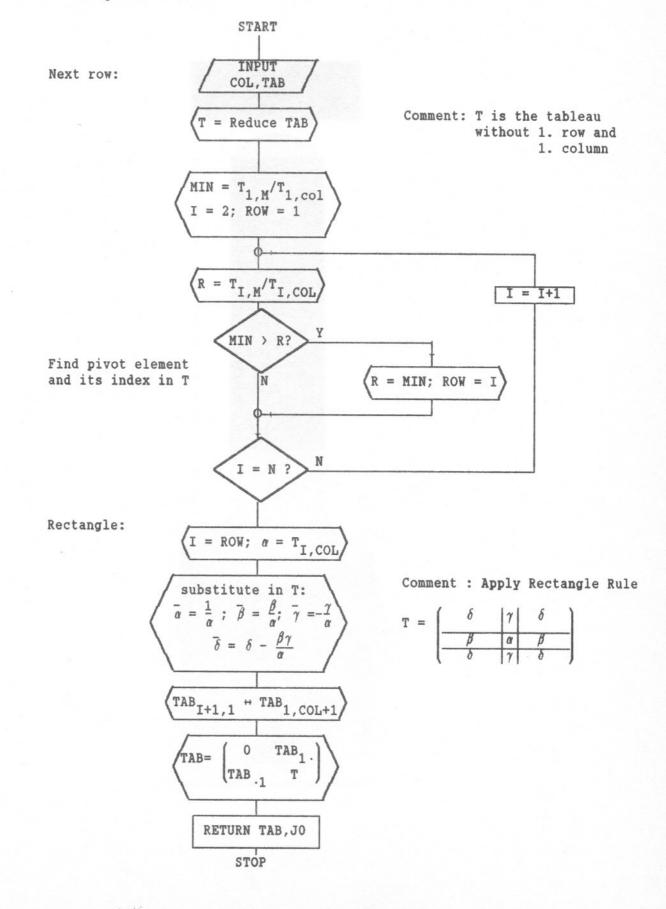1. for given column (COL) compute the row for the pivot element.
2. change the Tableau.

START

Next row:

INPUT
COL,TAB

$T = \text{Reduce TAB}$

Comment: T is the tableau
without 1. row and
1. column

$\text{MIN} = T_{1,M}/T_{1,col}$
$I = 2; \text{ROW} = 1$

$R = T_{I,M}/T_{I,COL}$

$I = I+1$

MIN > R? **Y**

Find pivot element
and its index in T

N

$R = \text{MIN}; \text{ROW} = I$

I = N ? **N**

Rectangle:

$I = \text{ROW}; \quad \alpha = T_{I,COL}$

substitute in T:
$$\bar{\alpha} = \frac{1}{\alpha} \; ; \; \bar{\beta} = \frac{\beta}{\alpha}; \; \bar{\gamma} = -\frac{\gamma}{\alpha}$$
$$\bar{\delta} = \delta - \frac{\beta\gamma}{\alpha}$$

Comment : Apply Rectangle Rule

$$T = \begin{pmatrix} \delta & \gamma & \delta \\ \hline \beta & \alpha & \beta \\ \delta & \gamma & \delta \end{pmatrix}$$

$TAB_{I+1,1} \leftrightarrow TAB_{1,COL+1}$

$$TAB = \begin{pmatrix} 0 & TAB_{1.} \\ TAB_{.1} & T \end{pmatrix}$$

RETURN TAB,JO

STOP

```
      ∇ R←A LH B;M;N;C;N0;X;Y;I0;LAY;J0;LAX;TAB1;TAB2;ROW;COL;ZEA;ZEB
[1]      M←(ρA)[1] ⍝                          Get dimension of A.
[2]      N←(ρA)[2]
[3]    CHOOSEVERTEX: ⍝                         Select a start strategy for player 1 or 2.
[4]      'Select a column for player 2 out of 1,...,',⍕N
[5]      '               or for player 1 out of ',(⍕N+1),',...,',⍕M+N
[6]      N0←⎕ ◇ C←0 ⍝                          Read input to N0.C=0 means player 2.
[7]      →(~N0∊⍳N+M)/CHOOSEVERTEX ⍝            Is input correct ?
[8]      →(N0≤N)/STRATEGIES ⍝                  Is player 1 selected ?
[9]    TRANSPOSE:⍝                             Then change roles of player 1 and 2,
[10]     C←⍉A ◇ A←⍉B ◇ B←C ⍝                   transpose matrices and
[11]     C←N ◇ N←M ◇ M←C ⍝                     exchange M and N.
[12]     N0←N0-M ◇ C←1 ⍝                       C=1 means player 1 was selected.
[13]   STRATEGIES: ⍝                           N0, I0, J0 are the indices of 1 in strategies
[14]     X←Mρ0 ◇ Y←Nρ0 ⍝                       Y ,X ,Y .Find I0, J0, such that
[15]     I0←A[;N0]⍳LAY←⌈/A[;N0] ⍝              X is best reply to Y
[16]     X[I0]←Y[N0]←1 ⍝                       and Y ist best reply to X.
[17]     J0←B[I0;]⍳LAX←⌈/B[I0;] ⍝              If J0=N0, meaning Y=Y then (X,Y)
[18]     →(J0=N0)/END ⍝                        is an EqP with payoff (LAX,LAY).Then STOP.
[19]   INITTABLEAUS:⍝                          Else construct the initial tableaus for
[20]     TAB1←A INITTAB I0,N0,LAY,0 ⍝          player 1
[21]     TAB2←B INITTAB J0,I0,LAX,1 ⍝          and player 2 (1 means transpose B).
[22]   SWITCHTAB1: ⍝                           J0 was the number of the last used row.
[23]     →(N<COL←¯1+TAB1[1;]⍳J0)/GETXY ⍝        If there is a corresponding column in TAB1
[24]     ROW←TAB1 NEXTROW COL ⍝                then find the row index of the pivot element
[25]     TAB1←(ROW,COL)RECTANGLE TAB1 ⍝        in TAB1 to change TAB1. RECTANGLE claculates the
[26]   SWITCHTAB2: ⍝                           new J0 (global in CHANGE) for TAB2. If there
[27]     →(M<COL←¯1+TAB2[1;]⍳J0)/GETXY ⍝        is no corresponding column to J0, an EqP
[28]     ROW←TAB2 NEXTROW COL ⍝                has been reached.
[29]     TAB2←(ROW,COL)RECTANGLE TAB2 ⍝        If TAB2 was changed correctly
[30]     →SWITCHTAB1 ⍝                         then concider TAB1 again.
[31]   GETXY: ⍝                                Calculate strategies X and Y from
[32]     X←Mρ0 ⍝                               the last columns of TAB1 and TAB2.
[33]     X[((ZEB>N)/ZEB)-N]←((ZEB←1↓TAB2[;1])>N)/1↓TAB2[;M+2]
[34]     Y←Nρ0
[35]     Y[((0<ZEA)∧ZEA<N)/ZEA]←((0<ZEA)∧(ZEA←1↓TAB1[;1])<N)/1↓TAB1[;N+2]
[36]   END: ⍝                                  If player 1 was selected, Y,X is the
[37]     →(C=0)/ANSWER ◇ C←X ◇ X←Y ◇ Y←C ⍝     result, else X,Y.
[38]   ANSWER:'Equilibrium point found with strategies : ',(⍕X),' for player 1'
[39]     '                                    and ',(⍕Y),' for player 2'
[40]     R←X,Y
      ∇
```

```
      ∇ ROW←TAB NEXTROW COL;TA;R;MIN;M
[1]      M←(ρTAB)[2] ◇ ROW←0 ⍝                 If no pivot element can be found return 0.
[2]      →(0=∨/TA←⎕CT<1↓TAB[;COL+1])/END ⍝     Eleminate zeros in COL column,
[3]      R←(TA/1↓TAB[;M])÷TA/1↓TAB[;COL+1] ⍝   divide M+1 column by COL column and
[4]      →((⌊/θ)=MIN←⌊/(R>0)/R)/END ⍝          find the smallest value, if it exists,
[5]      ROW←(TA\R)⍳MIN ⍝                      and its index.
[6]    END: ⍝                                  Return pivot row.
      ∇
```

```
      ∇ R←MAT INITTAB PARAM;I0;J0;LAMBDA;W;DELTA;GAMMA;C;D;TETA;Y;ROW;COL;V;U;N;M
[1]     I0←PARAM[1] ◇ J0←PARAM[2] ◇ LAMBDA←PARAM[3] ◇ W←PARAM[4] ◇ →(W=0)/M1
[2]     A PARAM contains I0, J0, the no. of pos. coordinates in strategies of players 1 and 2,
[3]     A resp, the payoff for this strategy from matrix MAT and boolean variable W that
[4]     A indicates, whether the matrix MAT has to be transposed (W=1).
[5]     MAT←⍉MAT
[6]   M1:M←(ρMAT)[1] ◇ N←(ρMAT)[2] AMAT is an M×N matrix.
[7]     V←Mρ1 ◇ V[I0]←0 A              Take the invers strategies to reduce the matrix.
[8]     U←Nρ1 ◇ U[J0]←0
[9]     C←((M-1),1)ρ⁻1 A               Now calculate the components of the tableau :-------
[10]    D←MAT[I0;J0]+(U/V/MAT)-(⍉((N-1),M-1)ρV/MAT[;J0])+((M-1),N-1)ρU/MAT[I0;] A     |
[11]    TETA←LAMBDA-V/MAT[;J0] A      (   c          d            teta)                |
[12]    GAMMA←0 A                     (gamma     delta          y )                    |
[13]    DELTA←(1,N-1)ρ1 A                                                              |
[14]    Y←1 A                                                                          |
[15]    R←(C,D,TETA),[1]GAMMA,DELTA,Yⁿand order them to the tableau --------------------
[16]    →(W=0)/M2 A                   Calculate the column and row vectors for player 1 or 2:
[17]    ROW←((ι(I0-1)),I0↓ιM),J0+M A RC={1,..,M}\{I0}          VC={J0+M}
[18]    COL←I0,M+(ι(J0-1)),J0↓ιN A    R ={I0}                  V ={M+1,..,M+N}\{J0+M}
[19]    →M3
[20]  M2:ROW←(N+(ι(I0-1)),I0↓ιM),J0ATC={N+1,..,N+M}\{N+I0}  UC={J0}
[21]    COL←(N+I0),(ι(J0-1)),J0↓ιN A T ={N+I0}                U ={1,..,N}\{J0}
[22]  M3:R←(0,COL,0),[1]ROW,R
      ∇


      ∇ NTAB←PIVOT RECTANGLE TAB;M;N;ROW;COL;RO;CO;PI;J1
[1]     M←⁻1+(ρTAB)[1] ◇ N←⁻1+(ρTAB)[2] A Get the dimension of the tableau
[2]     ROW←PIVOT[1] ◇ COL←PIVOT[2] A       and the index of the pivot element.
[3]     NTAB←1 1↓TAB A                     Tableau without row and column vector.
[4]     RO←Mρ1 ◇ RO[ROW]←0 A              Take all rows, exept the pivot row,
[5]     CO←Nρ1 ◇ CO[COL]←0 A              all columns, exept the pivot column.
[6]     NTAB←RO\CO\(NTAB←CO/RO/NTAB)-((RO/NTAB[;,COL])+.×(CO/NTAB[,ROW;]))÷PI←NTAB[ROW;COL]
[7]     NTAB[;COL]←-1↓TAB[;COL+1]÷PI A     Calculate the outer elements (see line 6),
[8]     NTAB[ROW;]←1↓TAB[ROW+1;]÷PI A      the pivot column (see line 7) and pivot row.
[9]     NTAB[ROW;COL]←1÷PI A               Replace the pivot element.
[10]    J1←TAB[1;COL+1] ◇ J0←TAB[ROW+1;1]ATake row and column (global J0),
[11]    NTAB←TAB[1;],[1](1↓TAB[;1]),NTABA copy old row and column vectors to new
[12]    NTAB[ROW+1;1]←J1 A                 tableau and exchange row and column.
[13]    NTAB[1;COL+1]←J0 A                 J0 will be used by LH.
      ∇
```

REFERENCES

[1] Aggarval, V.:   *On the Generation of all Equilibrium Points for Bimatrix Games through the Lemke–Howson Algorithm.* Math. Progr. 1 (1973), pp.232–234

[2] Bastian, M.:   *Another Note on Bimatrix Games.* Math. Progr. 1 (1973), pp.299–300

[3] Borm, P.E.M.:   *On Refinements in Bimatrix Games.* Preprint, Dept. of Mathematics, University of Nijmegen (1989)

[4] Borm, P.,   *A Geometric–Combinatorial Approach to Bimatrix* Gijsberts,A., Tijs S.:   *Games.* Meth. OR 59 (1988)

[5] Lemke E.,   *Equilibrium Points in Bimatrix Games.* SIAM J. Appl.   Howson, T.:   Math. 12 (1964), pp.413–423

[6] Parthasarathy, T.   *Some Topics in Two–Person Games.* Modern Analytic   Raghavan, T.E.S.:   and Computational Methods in Science and Mathematics, No.22, American Elsevier Publ.Co. (1971)

[7] Shapley, L.S.:   *A Note on the Lemke–Howson Algorithm.* Math. Progr. Study 1 (1974), pp.175–189

[8] Rosenmüller, J.:   *On a Generalization of the Lemke–Howson Algorithm to Noncooperative n–Person Games.* SIAM J. Appl. Math. 21 (1971), pp.73–79

[9] Rosenmüller, J.:   *The Theory of Games and Markets.* North Holland Publ. Comp. (1981)

[10] Todd, M.:   *Comments on a Note by Aggarval.* Math. Progr. 10 (1976), pp.130–133

[11] Todd, M.:   *Bimatrix Games – An Addendum.* Math. Progr. 14 (1978), pp.112–115

[12] Wilson, R.:   *Computing Equilibria of n–Person Games.* SIAM J. Appl. Math. 21 (1971), pp.80–87