MATHEMATICA — REVUE D'ANALYSE NUMÉRIQUE
ET DE THÉORIE DE L'APPROXIMATION
L'ANALYSE NUMÉRIQUE ET LA THÉORIE DE L'APPROXIMATION
Tome 13, N° 2, 1984, pp. 125–142

A SIMPLEX-LIKE TECHNIQUE FOR THE MAXIMIZATION OF A CONVEX QUADRATIQUE FUNCTION UNDER LINEAR CONSTRAINTS

by
I. MARUŞCIAC
(Cluj-Napoca)

1. Introduction. We consider the following maximization problem:

$$\max\ \{x^T Cx + 2cx:\ \ Ax = a,\ x \geqslant 0\}, \tag{1}$$

where A, C are $m \times n$ and $n \times n$ matrices respectively and a, c are vectors of the appropiate dimention. Matrix C is assumed to be simetric and positive semidefinite.

In the present paper a simplex-like technique is used to establish simple optimality criteria for the solution to the problem (1) via general bilinear programming problem studied by the author in [5]. Then a simplex-like algorithm is described to find local and global maximum of the problem (1) respectively.

The general bilinear programming problem considered here is the following:

$$\max\{f(x, y) = x^T Cy + cx + dy\} \tag{2}$$

subject to the linear constraints

$$Ax = a,\ x \geqslant 0 \tag{3}$$

$$By = b,\ y \geqslant 0, \tag{4}$$

where A, B, C are $m \times n,\ p \times q,\ n \times q$ — matrices respectively and $a,\ b,\ c,\ d,\ x,\ y$ are vectors of the appropiate dimension.

In what follows we shall use the following Altman's result given in [1].

THEOREM 1. *If (x^*, y^*) is an optimal solution of the bilinear programming problem* (2)—(4), *then there is a basic feasible one.*

The convex quadratique maximization p·oblem (1) can be considered as a special case of a bilinear programming problem in which

$$B = A,\ b = a,\ d = c:$$

(5) $$\max\ \{x^TCy + c(x + y)\}$$

subject to

(6) $$Ax = a, \quad Ay = a, \quad x \geqslant 0,\ y \geqslant 0.$$

THEOREM 2. [1]. *Let (x^*, y^*) be an optimal solution of the bilinear programming problem* (5), (6). *Then x^* and y^* are optimal solutions of the quadratique programming problem* (1) *and*

$$x^{*T}Cy^* = x^{*T}Cx^* = y^{*T}Cy^*,$$
$$cx^* = cy^*$$

If an aldition C is positive definite then $x^* = y^*$.

2. Jordan elimination in bilinear programming. Let us consider

(7) $$f(x, y) = \sum_{i=1}^{n}\sum_{j=1}^{p} c_{ij}x_iy_j + \sum_{i=1}^{n} c_ix_i + \alpha + \sum_{j=1}^{p} d_jy_j + \beta$$

(8) $$z_k = \sum_{i=1}^{n} a_{ki}x_i + a_k, \quad k = 1, 2, \ldots, m$$

(9) $$u_h = \sum_{j=1}^{p} b_{hj}y_j + b_h, \quad h = 1, 2, \ldots, q$$

and let $b_{rs} \neq 0$ be the pivot element în (9). Then after substituting

$$y_r = \frac{1}{b_{rs}}\Big(-\sum_{j\neq s} b_{rj}y_j + u_r - b_r\Big)$$

in (7) we obtain

$$f(x, y\,;\, u) = \sum_{i=1}^{n} x_i\Big(\sum_{j\neq s} c'_{ij}y_j + c'_{is}u_r\Big) + \sum_{i=1}^{n}(c_i + \delta'_i)x_i + \alpha + \sum_{j\neq s} d'_jy_j + d'_su_r + \beta',$$

where

(10) $$d'_j = d_j - \frac{d_sb_{rj}}{b_{rs}},\ j \neq s$$
$$c'_{ij} = c_{ij} - \frac{c_{ij}b_{rj}}{b_{rs}}, \quad j \neq s,\ i = 1, 2, \ldots, n$$
$$c'_{js} = \frac{c_{is}}{b_{rs}},\ i = 1, 2, \ldots, n$$
$$\beta' = \beta - \frac{d_sb_r}{b_{rs}}$$
$$\delta'_i = \frac{c_{is}b_r}{b_{rs}}, \quad i = 1, 2, \ldots, n.$$

If we consider the simplex table

(11)

	x	y	1
$z =$	A	0	a
$u =$	0	B	b
$f =$	c	0	α
	0	d	β
	0	C	0

tnen after a Jordan elimination step (J.e.s.) we get the table

(12)

	x	$y_1 \ u_r \ y_n$	1
$z =$	A	0	a
$u_1 =$			
$\vdots$			
$y_s =$	0	B'	b'
$\vdots$			
$u_q =$			
$f =$	c	0	α
	0	d'	β'
	0	C'	δ'

where C', b', d', β', δ' are formed by the elements given in (10) and B' is the matrix obtained from B after a standard Jordan step.

From (10)—(12) it is seen that in bilinear programming a Jordan elimination step should be carried out according to the usual rules to which one adds

Additional rule: if the pivot element is an element of the matrix B then

$$c' = c + \delta'.$$

If we take a pivot element in the matrix A then instead of table (11) we consider the table

	x	y	1
$z =$	A	0	a
$u =$	0	B	b
	c	0	α
$f =$	0	d	β
	C^{T}	0	0

after a J.e.s. with pivot element $a_{rs} \neq 0$ we get the table

	$x_1 \ldots z_r \ldots x_n$	y	1
$z_1 =$			
$\vdots$			
$x_s =$	A'	0	a'
$\vdots$			
$z_m =$			
$u =$	0	B	b
	c'	0	α'
$f =$	0	d	β
	C'^{T}	0	γ'

Now the additional rule consists in:

$$d' = d + \gamma'$$

(here "$'$" means the transformed matrix (vector) after a J.e.s.).

Now if we consider the simplex table

(13)

	x	x	1
$z =$	A	0	a
$u =$	0	A	a
	c	0	α
$f =$	0	c	α
	0	C	0

corresponding to the quadratique programming (1), then after a J.e.s. we get the table

	x	$x_1 \ldots u_r \ldots x_n$	1
$z =$	A	0	a
$u_1 =$			
$\vdots$			
$x_s =$	0	A'	a'
$\vdots$			
$u_m =$			
	$c + \delta'$	0	α
$f =$	0	c'	α'
	0	C'	0

After a new J.e.s., taking the same pivot element a_{rs} in the first matrix A and the corresponding matrix C'^T, then (14) will be transformed into the table

(15)

	$x_1 \ldots z_r \ldots x_n$	$x_1 \ldots z_r \ldots x_n$	1
$z_1 =$ ⋮ $x_s =$ ⋮ $z_m =$	A'	0	a'
$z_1 =$ ⋮ $x_s =$ ⋮ $z_m =$	0	A'	a'
$f =$	$(c+\delta')'$	0	β'
	0	$c'+\gamma'$	α'
	C''^T	0	γ'

here $C'' = (C')'$.

DEFINITION 1. *The pair of the two J.e.s. by which the table* (13) *is transformed into* (15) *trough* (14) *will be called a double Jordan elimination step (d.J.e.s.) of the quadratique programming problem* (1).

REMARK 1. A d.J.e.s. with the pivor element a_{rs} can be carried out as follows: starting from the table

(16)

	x	1
$z =$	A	a
$f =$	c	α
	C	0

(that must be read: $z = Ax + a$, $f(x) = x^TCx + 2cx + 2\alpha$

we carry out a J.e.s. and we obtain the table

(17)

	$x_1 \ldots z_r \ldots x_n$	1
$z_1 =$ ⋮ $x_s =$ ⋮ $z_m =$	A'	a'
$f =$	c'	α'
	C'	δ'

In view of the additional rule another J.e.s. with the same pivot element a_{rs} will be carried out starting from the table

(18)

	x	1
$z =$	A	a
$f =$	$c+\delta'$	α
	C'^T	0

After a J.e.s. with pivot element a_{rs} we obtain the table

(19)

	$x_1 \ldots z_r \ldots x_n$	1
$z_1 =$ ⋮ $x_s =$ ⋮ $z_m =$	A'	a'
$f =$	$(c+\delta')'$	β'
	C''^T	γ'

Therefore, after a d.J.e.s. with pivot element a_{rs} table (16) is transformed onto the table

(20)
$$\begin{array}{r|c|c|}
 & x_1 \ldots z_r \ldots x_n & 1 \\ \hline
z_1 = & & \\
\vdots & & \\
x_s = & A' & a' \\
\vdots & & \\
z_m = & & \\ \hline
f = \Big\{ & c'' = (c + \delta')' & \alpha'' = \dfrac{\alpha' + \beta'}{2} \\ \cline{2-3}
 & C'' & 0 \\ \hline
\end{array}$$

THEOREM 3. *If $C^T = C$ and C is positive semidefinite, then*

(i) $$(c + \delta')' = c' + \gamma'$$

(ii) $$C''^T = C''.$$

Proof. According to the relations (10) we have

$$\delta'_j = -\frac{c_{js} a_r}{a_r}, \quad c'_{ij} = c_{ij} - \frac{c_{is} a_{rj}}{a_{rs}}, \quad j \neq s, \quad c'_{is} = \frac{c_{is}}{a_{rs}},$$

$$c'_j = c_j - \frac{c_s a_{rj}}{a_{rs}}, \quad \gamma'_j = -\frac{c'_{sj} a_r}{a_{rs}} = -\left(c_{sj} - \frac{c_{ss} a_{rj}}{a_{rs}}\right) a_r / a_{rs}.$$

Therefore

(21)
$$c''_j = (c_j + \delta'_j)' = c_j + \delta'_j - \frac{(c_s + \delta'_s) a_{rj}}{a_{rs}} = c_j - \frac{c_s a_{rj}}{a_{rs}} - \frac{c_{js} a_r}{a_{rs}} + \frac{c_{ss} a_r a_{rj}}{a_{rs}^2} = c'_j - \left(c_{js} - \frac{c_{ss} a_{rj}}{a_{rs}}\right) a_r / a_{rs} = c'_j + \gamma'_j, \quad j \neq s,$$

since $c_{js} = c_{sj}$.

(22) $$c''_s = (c_s + \delta'_s)' = \frac{c_s + \delta'_s}{a_{rs}} = \frac{c_s}{a_{rs}} - \frac{c_{ss} a_r}{a_{rs}^2} = c'_s + \gamma'_s,$$

i.e. (i) holds.

If we denote: $B = C'^T$, then from (10) it follows that

$$c''_{ij} = b'_{ij} = b_{ij} - \frac{b_{is} a_{rj}}{a_{rs}} = c'_{ji} - \frac{c'_{si} a_{rj}}{a_{rs}} =$$

$$= c_{ji} - \frac{c_{js} a_{ri}}{a_{rs}} - \left(c_{si} - \frac{c_{ss} a_{ri}}{a_{rs}}\right) \frac{a_{rj}}{a_{rs}},$$

hence

(23) $$c''_{ij} = c_{ji} - \frac{c_{js} a_{ri}}{a_{rs}} - \frac{c_{si} a_{rj}}{a_{rs}} + \frac{c_{ss} a_{ri} a_{rj}}{a_{rs}^2}, \quad i, j \neq s.$$

Similarly we obtain

$$c''_{ij} = c_{ij} - \frac{c_{is} a_{rj}}{a_{rs}} - \frac{c_{sj} a_{ri}}{a_{rs}} + \frac{c_{ss} a_{rj} a_{ri}}{a_{rs}^2}, \quad i, j \neq s.$$

Since $c_{ij} = c_{ji}$ it follows that $c''_{ij} = c''_{ji}$.

If $j = s$ then we have

(24) $$c''_{is} = b'_{is} = \frac{b_{is}}{a_{rs}} = \frac{c_{si}}{a_{rs}} = \left(c_{si} - \frac{c_{ss} a_{ri}}{a_{rs}}\right) / a_{rs}$$

respectively

$$c''_{si} = b'_{si} = b_{si} - \frac{b_{ss} a_{ri}}{a_{rs}} = c'_{is} - \frac{c'_{ss} a_{ri}}{a_{rs}} =$$

$$= \frac{c_{is}}{a_{rs}} - \frac{c_{ss}}{a_{rs}} \cdot \frac{a_{ri}}{a_{rs}} = \left(c_{is} - \frac{c_{ss} a_{ri}}{a_{rs}}\right) / a_{rs},$$

hence $c''_{is} = c''_{si}$.

REMARK 1. Formulae (21)—(24) can be looked as rules of a d.J.e.s. for the quadratique programming problem. Hence a d.J.e.s. is a transformation of the table (16) onto table (29), where matrix A' and vector a' are obtained from A and a respectively by a usual J.e.s. and C'', c'' and α'' are calculated as follows:

(25)
$$c''_{ij} = c_{ij} - \frac{c_{is} a_{rj}}{a_{rs}} - \frac{c_{sj} a_{ri}}{a_{rs}} + \frac{c_{ss} a_{ri} a_{rj}}{a_{rs}^2}, \quad i,j \neq s$$

$$c''_{is} = \frac{c_{is}}{a_{rs}} - \frac{c_{ss} a_{ri}}{a_{rs}^2}, \quad i \neq s, \quad c''_{ss} = \frac{c_{ss}}{a_{rs}^2}.$$

(26)
$$c''_j = c_j - \frac{c_s a_{rj}}{a_{rs}} - \frac{c_{js} a_r}{a_{rs}} + \frac{c_{ss} a_{rj} a_r}{a_{rs}^2}, \quad j \neq s$$

$$c''_s = \frac{c_s}{a_{rs}} - \frac{c_{ss} a_r}{a_{rs}^2}$$

(27) $$\alpha'' = \alpha - \frac{c_s a_r}{a_{rs}} + \frac{c_{ss} a_r^2}{2a_{rs}^2}.$$

REMARK 2. The rules of a d.J.e.s do not change if instead of the simplex table (16) we start with the table

$$
\begin{array}{r|c|c|}
 & -x & 1 \\ \hline
z = & A & a \\ \hline
\multirow{2}{*}{$f =$} & c & \alpha \\ \cline{2-3}
 & C & 0 \\ \hline
\end{array}
$$

In this case we have to use the modified Jordan elimination step (i.e. the pivot line rests unchanged but pivot column changes sign).

3. Optimality criteria. To obtain a basic feasible solution (b.f.s.) to the problem (1), we shall use the d.J.e.s. described at the section 2. To simplify the notation we assume that matrix A is of the full rank. Then startind from the table

$$
\text{(28)}\qquad
\begin{array}{r|c|c|}
 & -x & 1 \\ \hline
0 = & A & a \\ \hline
\multirow{2}{*}{$f =$} & -c & 0 \\ \cline{2-3}
 & C & 0 \\ \hline
\end{array}
$$

and assuming (without loss of generality) that the pivot elements were taken from the first m column of A, then after m d.J.e.s. we get the table

$$
\text{(29)}\qquad
\begin{array}{r|c|c|}
 & -x_{m+1} \; \ldots \; -x_n & 1 \\ \hline
\begin{array}{r} x_1 = \\ \vdots \\ x_m = \end{array} & B & b \\ \hline
\multirow{2}{*}{$f =$} & p & P \\ \cline{2-3}
 & D & 0 \\ \hline
\end{array}
\; , \; b \geqslant 0.
$$

REMARK 3. Simplex table (29) corresponds to the following canonical form of the convex quadratic programming problem:

$$
(1') \qquad \max \{f(x) = X^T DX - 2pX + 2P\}
$$

subject to

$$
BX \leqslant b, \; X \geqslant 0,
$$

where $X = (x_{m+1}, \ldots, x_n)^T$.

Therefore, if the convex quadratique programming problem is given in canonical form (1′) then we start direclty by a simplex table like (29).

LEMMA 1. *If in* (29) $p > 0$, *then* b.f.s. $x^0 = (b, 0) \in \mathbf{R}^n$ *is a local maximum of the quadratique programming* (1).

Proof. Form (29) it is seen that $f(x^0) = 2P$ and

$$
f(x) = -2 \sum_{i=m+1}^{n} p_i x_i + \sum_{i=m+1}^{n} \sum_{i=m+1}^{n} d_{ij} x_i x_j + 2P
$$

Therefore

$$
(30)\; f(x) - f(x^0) = \sum_{i=m+1}^{n} x_i \left(\frac{1}{2} \sum_{j=m+1}^{n} d_{ij} x_j - p_i \right) + \sum_{j=m+1}^{n} x_j \left(\frac{1}{2} \sum_{i=m+1}^{n} d_{ij} x_i - p_j \right)
$$

Now, if $p > 0$ then (30) shows that

$$
f(x) - f(x^0) \leqslant 0
$$

for each $x_i \geqslant 0$, $i = m+1, \ldots, n$, sufficiently small, i.e. x^0 is a local maximum for f in

$$
\Omega = \{x \in \mathbf{R}^n : Ax = a, \; x \geqslant 0\}.
$$

THEOREM 4. *Let* $x^0 = (b, 0)$ *be a nondegenerate b.f.s. Then* x^0 *is a local maximum of* f *on* Ω *if and only if*

(*i*)
$$
p \geqslant 0;
$$

(*ii*)
$$
d_{ii} \leqslant 0, \quad \forall\, i \in I^0,
$$

where

$$
I^0 = \{i \in I : p_i = 0\}, \quad I = \{m+1, m+2, \ldots, n\}.
$$

Proof. ($\Rightarrow$) Let x^0 be a local maximum of f on Ω and consider

$$
(31) \qquad x^i = (b, 0, \ldots, x_i, \ldots, 0), \; x_i = t > 0, \; i \in I.
$$

It is easy to see form (30) thet

$$
(32) \qquad f(x^i) - f(x^0) = \begin{cases} d_{ii} t^2, & i \in I^0 \\ (d_{ii} t - 2p_i) t, & i \notin I^0 \end{cases}
$$

and therefore $f(x^i) - f(x^0) \leqslant 0$ for each $x_i > 0$, $i \in I$, small enough, implies (i) — (ii).

($\Leftarrow$) Assume that (i)—(ii) hold. Then (32) shows that

(33) $$f(x^i) - f(x^0) \leqslant 0, \quad \forall\, i \in I,$$

for each $t > 0$ small enough. Now we choose $t > 0$ such that the convex hull $V = \text{conv}\ \{x^0, x^{m+1}, \ldots, c^n\} \subset \Omega$, where x^i is defined as in (31). Any $x \in V$ has the representation

$$x = \lambda^0 x^0 + \lambda_1 x^{m+1} + \ldots + \lambda_{n-m} x^n, \quad \lambda_i \geqslant 0, \quad \sum_{i=0}^{n-m} \lambda_i = 1.$$

Consider the convex function $g : \mathbf{R}^n \to \mathbf{R}$,

$$g(x) = f(x) - f(x^0).$$

Hence

$$f(x) - f(x^0) = g(x) \leqslant \lambda_0 g(x^0) + \sum_{i=1}^{n-m} \lambda_i g(x^{m+i}) \leqslant$$

$$\leqslant \max\, \{g(x^0),\ g(x^{m+1}), \ldots, g(x^n)\}.$$

Obviously $g(x_0) = 0$ and (33) shows that

$$g(x^i) \leqslant 0, \qquad i \in I.$$

Thus $f(x) - f(x^0) \leqslant 0$, $\forall\, x \in V$, i.e. x^0 is a local maximum of f on Ω.

Now let $x^0 = (b, 0)$ be a degenerate b.f.s. of (1) and let us denote:

$$J = \{i \in \{1, 2, \ldots, m\} : b_i = 0\}.$$

THEOREM 5. *Degenerate b.f.s. $x^0 = (b, 0) \in \Omega$ is a local maximum of f on Ω if and only if*

(i) $$p \geqslant 0$$

(ii) $$d_{ii} \leqslant 0, \quad \forall\, i \in I_d^0$$

where

$$I_d^0 = \{i \in I^0 : b_{ki} \leqslant 0, \quad k \in J\}.$$

Proof. ($\Rightarrow$) If $x^0 = (b, 0)$ is a local maximum for f on Ω, the $f(x) - f(x^0) \leqslant 0$ in a certain neighborhood of x^0. Consider x^i defined as in (31). As $f(x^i) - f(x^0) \leqslant 0$ for $t > 0$ small enough, from (32) it follows that $p_i > 0$, $i \notin I^0$, i.e. $p \geqslant 0$. Hence (i) holds.

LEMMA 2. *Let $x_0 = (b, 0) \in \Omega$ be a degenerate b.f.s. If there is $b_{ki} > 0$, $k \in J$, then for every $t > 0$, x^i is not feasible solution,*

Proof. Consider x^i defined in (31) and let $b_{ki} > 0$, $k \in J$.

Then $x_k = -b_{ki}\, t < 0$, for every $t > 0$, that means $x^i \notin \Omega$, for every $t > 0$,

Now, since $x^i \in \Omega$ it is necessary that

$$b_{ki} \leqslant 0, \quad k \in J,$$

and from (32) it is seen that $f(x^i) - f(x^0) \leqslant 0$ implies

$$d_{ii} \leqslant 0, \quad \forall i \in I_d^0,$$

i.e. (i) — $(ii)_d$ hold.

($\Leftarrow$) The proof of the sufficiency of the conditions (i) — $(ii)_d$ is similar to the proof of the sufficiency of conditions (i) — (ii) on Theorem 4, using in addition Lemma 2.

4. Global maximum. Assume that $x^0 = (b, 0) \in \Omega$) is a local maximum of f on Ω. Then it follows that $b \geqslant 0$ and (i) — (ii) or (i) —$(ii)_d$ hold. For x^i defined as in (31) we have

$$f(x^i) = -2p_i x_i + 2P + d_{ii} x_i^2, \quad \frac{\partial f}{\partial x_i}(x^i) = -2p_i + 2d_{ii} x_i.$$

Condition (i) implies

$$\frac{\partial f(x^0)}{\partial x_i} \leqslant 0, \quad \forall\, i \in I$$

Define

(34) $$x_i^* = \min\, \{t > 0 : f(x^i) - f(x^0) = 0\}$$

where $x_i = t$.

If there is $i \in I$ such that $f(x^i) - f(x^0) = 0$ has no positive solution, then one takes $x_i^* = +\infty$.

Consider the inequality

(35) $$\sum_{i \in I} \frac{x_i}{x_i^*} \leqslant 1$$

(The terms corresponding to $x_i^* = +\infty$ are missing in (35)).

Since function f reaches its maximum in a b.f.s., it follows that

$$\max\, \{f(x) : x \in \Omega_1\} = f(x^0),$$

where

$$\Omega_1 = \Omega \cap \left\{x \in \mathbf{R}^n : \sum_{i \in I} \frac{x_i}{x_i^*} < 1\right\}$$

That is why, in order to determine a new local maximum, we shall find a local maximum of f on $\Omega \setminus \Omega_1$, i.e. adding to the initial constraints

$$Ax = a, \quad x \geqslant 0$$

the constraint

$$\sum_{i \in I} \frac{x_i}{x_i^*} \geqslant 1. \tag{36}$$

With this new quadratique programming problem we proceed similarly until the problem becomes inconsistent. Hence we have |

THEOREM 6. *Let $x^0 = (b, 0) \in \Omega$ be o local maximum of f on Ω and x_i^*, $i \in I$ defined as in* (34). *If*

$$\left\{x \in \mathbb{R}_+^n : Ax = a, \sum_{i \in I} \frac{x_i}{x_i^*} \geqslant 1\right\} = \Phi$$

then x^0 is a global solution to the problem (1).

6. Description of the algorithm. From above we conclude with the following algorithm for the global maximum of the convex quadratique programming problem.

Step. 1. Starting from the table (28) find a b.f.s. x^0.

Step 2. If $(i)-(ii)$ or $(i)-(ii)_d$ hold (whether x^0 is a nondegenerate or it is a degenerate b.f.s.) then go to Step 5. Otherwise go to tep 3.

Step 3. Compute

$$D = \frac{d_{ss} b_r^2}{2b_{rs}^2} - \frac{p_s b_r}{b_{rs}} = \max_j \left\{\frac{d_{jj} b_r^2}{2b_{rj}^2} - \frac{p_j b_r}{b_{rj}} > 0\right\} \tag{38}$$

Step 4. Do a d.J.e.s. by choosing a pivot element b_{rs} in the column s and go to Step 2.

Step 5. Compute x_i^* as in (34).

Step 6. Add the inequality (36) to the initial constraints and go to Step 1.

The algorithm is terminated when the new quadratique programming problem becomes inconsistent (Theorem 6).

LEMMA 3. *Let $x^0 = (b, 0) \in \Omega$ be a basic feasible solution of* (1) *and $x' = (b', 0)$ an adjacent b.f.s. obtained from x^0 in Step* 4. *Then*

$$f(x') > f(x^0).$$

Proof. Equality (27) shows that the number D given in (38) is always positive. Indeed, D represents the largest difference between the value of the function f at an adjacent b.f.s. which can be reached from x^0 in one J.e.s. and the value of f at $x^0 : f(x^0) = 2P$. Therefore, if x^0 is not a local maximum b.f.s. then $D > 0$. But we pass Step 4 only when x^0 is not a local maximum b.f.s.

Now, from (27) we have

$$f(x') = 2P - \frac{p_s b_r}{b_{rs}} + \frac{d_{ss} b_r^2}{2b_{rs}^2} = 2P + D = f(x^0) + D.$$

Therefore

$$f(x') > f(x^0).$$

THEOREM 7. *If $\Omega \neq \emptyset$ is bounded then the proposed algorithm halts in finitely many steps generating an optimal b.f.s. to the problem* (1).

P r o o f. It follows immediately from the following facts:

1) By simplex method Step 1 converges in finitely many steps;

2) L e m m a 3 shows that whenever we pass Step 4 the value of the objective function f is improved by $D > 0$.

3) There are only finitely many b.f.s. for Ω.

REMARK 4. In order to improve the cutting plane given in (36) we can use an iterative procedure given by CONNO, H. [2], which generates a cut which is generally deeper than the cut used here. But in the other hand this procedure involves to solve n-m subsidiary linear programming problems at each iterations.

7. Example. To illustrate the algorithm we solve the following example: maximize

$$f(x) = -9x_1 - 15x_2 + 2x_1^2 + 5x_1x_2 + 5x_2^2$$

subject to

$$\begin{aligned} x_1 \quad + x_3 \quad &= 2 \\ x_2 \quad + x_4 &= 2, \quad x_j \geqslant 0,\ j = 1, 2, 3, 4. \end{aligned}$$

Step 1. The initial table is:

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	1
0 =	1	0	1	0	2
0 =	0	1	0	1	2
	9/2	15/2	0	0	0
f =	2	5/2	0	0	0
	5/2	5	0	0	0
	0	0	0	0	0
	0	0	0	0	0

After a J.e.s. with marked pivot element we get the table

	0	$-x_2$	$-x_3$	$-x_4$	1
$x_1 =$	1	0	1	0	2
$0 =$	0	1	0	1	2
	$-9/2$	$15/2$	$-9/2$	0	-9
$f =$	-2	$5/2$	-2	0	-4
	$-5/2$	5	$-5/2$	0	-5
	0	0	0	0	0
	0	0	0	0	0

In view of the additional rule we continue with the table

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	1
$0 =$	1	0	1	0	2
$0 =$	0	1	0	1	2
	$1/2$	$5/2$	0	0	0
$f =$	-2	$-5/2$	0	0	0
	$5/2$	5	0	0	0
	-2	$-5/2$	0	0	0
	0	0	0	0	0

After a J.e.s. we get the table

	0	$-x_2$	$-x_3$	$-x_4$	1
$x_1 =$	1	0	1	0	2
$0 =$	0	1	0	1	2
	$-1/2$	$5/2$	$-1/2$	0	-1
$f =$	2	$-5/2$	2	0	4
	$-5/2$	5	$-5/2$	0	-5
	2	$-5/2$	2	0	4
	0	0	0	0	0

After eliminating the O-column and the corresponding line in C′, the new d.J.e.s. starts with the table

	$-x_2$	$-x_3$	$-x_4$	1
$x_1 =$	0	1	0	2
$0 =$	1	0	1	2
	$5/2$	$-1/2$	0	-5
$f =$	5	$-5/2$	0	0
	$-5/2$	2	0	0
	0	0	0	0

After a d.J.e.s. we get

	$-x_3$	$-x_4$	1
$x_1 =$	1	0	2
$x_2 =$	0	1	2
	$9/2$	$15/2$	0
$f =$	2	$5/2$	0
	$5/2$	5	0

This table corresponds to the canonical quadratique programming problem:

$$-9x_3 - 15x_4 + 2x_3^2 + 5x_3x_4 + 5x_4^2 \to \max$$

subject to

$$x_3 \leqslant 2$$

$$x_4 \leqslant 2, \; x_3 \geqslant 0, \; x_4 \geqslant 0.$$

Step 2. It is seen that (i) holds $(9/2 > 0, \; 15/2/2 > 0)$.

Since $I^0 = \emptyset$, *(ii)* is automatically satisfied. Hence $x^0 = (0, 0, 2, 2)$ is a local maximum and $f(x^0) = 0$.

Step 5. We have

$$f(x^3) - f(x') = -9t + (-t \;\; 0)\begin{pmatrix} 2 & 5/2 \\ 5/2 & 5 \end{pmatrix}\begin{pmatrix} -t \\ 0 \end{pmatrix} = -7t.$$

Hence $x_3^* = +\infty$. Similarly we obtain

$$f(x^4) - f(x^0) = -15t + 5t^2 = 5t(t-3),$$

and so $x_4^* = 3$. Therefore the inequality (36) is:

$$x_4 \geqslant 3.$$

Step 6. The new simplex table is the following

	$-x_3$	$-x_4$	1
$x_1 =$	1	0	2
$x_2 =$	0	1	2
$x_5 =$	0	-1	-3
$f =$	9/2	15/2	0
	2	5/2	0
	5/2	5	0

Step. 1. It is clear that the new problem is inconsistent, since inequalities $x_4 \leqslant 2$ and $x_4 \geqslant 3$ are contradictory.

Therefore $x^0 = (0, 0, 2, 2)$ is theoptimal solution to the considered problem and $f(x^0) = 0$.

REFERENCES

[1] Altman, M., *Bilinear programming*, Bull. Acad. Polon. Sci. Math. Astron. Phys **16**, 741–746 (1968).

[2] Konno, H., *Maximization of a convex quadratic function under linear constraints*, Mathematical Programming **11**, 117–127 (1976).

[3] Konno, H., *Maximizing a convex quadratic function over a hypercube*, J. Oper. Res. Soc. Japan **23**, *2*, 171–187 (1980).

[4] Lebedev, V. Iu., *Metod resenia zadač maksimizatzii polozitel'no opredel'onnoi cvadratičnoi formy na mnogogranniki*, Być. Mat. i Mat. Fiz. **22**, *6*, 1344–1351 (1982).

[5] Marusciac, I., *On bilinear programming*, Mathematica-L'Analyse Numér. Théorie Approx. **7**, *1*, 67–79 (1978).

[6] Vandal, A., *Bilinear programming*, Ekonomska Analiza **4**, 1–2, 21–41 (1970).

Received 6.X.1983.

Facultatea de matematică
Universitatea din Cluj-Napoca
Str. Kogălniceanu 1
3400 *Cluj-Napoca*