General Mathematics Vol. 8 /2000 Nr. 3–4

A Simplex Version of Theil–Van de Panne Algorithm

Dumitru Barac

Abstract

In this paper, after we have introduced the generalized Jordan step notion, we presented a tabular form of the Theil-Van de Panne algorithm of the quadratic programming, which we consider a simplex version of this algorithm.

1991 Mathematical Subject Classification. Primary 90C20.

1 Generalized Jordan step

Let us consider a system of linear equations having the explicit (basic) form:

$$x_{n+i} = a_{i1}x_1 + \cdots + a_{in}x_n, \quad n+i \in B,$$

where $B = \{n+1, \ldots, n+m\}$. We write this under the form

$$\text{(1.1)} \qquad \begin{array}{ccc} & & \begin{array}{ccccc} x_1 & \ldots & x_j & \ldots & x_n \end{array} \\ x_{n+1} & = & \\ \ldots & & \\ x_{n+i} & = & \\ \ldots & & \\ x_{n+m} & = & \end{array} \begin{array}{|ccccc|} \hline a_{11} & \ldots & a_{1j} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{i1} & \ldots & a_{ij} & \ldots & a_{in} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{mj} & \ldots & a_{mn} \\ \hline \end{array}$$

or, briefly, $\begin{array}{ccc} & & x_{\bar{B}} \\ x_B & = & \boxed{A} \end{array}$, where $A = (a_{ij})$ is a $m \times n$ real matrix,

and $\bar{B} = \{1, \ldots, n\}$.

The usual terminology names $x_h, h \in B$ as *basic variables,* and $x_h, h \in \bar{B}$ as *nonbasic variables* (corresponding to the basis B).

We select now the subsets $R \subset B$ and $S \subset \bar{B}$, $card(R) = card(S) = p$, such that the matrix A_{RS} whose rows (resp. columns) correspond to the indices of R (resp. S) is non-singular. Denoting $\bar{R} = B \setminus R$, and $\bar{S} = \bar{B} \setminus S$, we can write (1.1) as

$$
(1.2)\qquad
\begin{array}{rc|cc|}
 & & x_S & x_{\bar S} \\
\cline{3-4}
x_R & = & A_{RS} & A_{R\bar S} \\
x_{\bar R} & = & A_{\bar R S} & A_{\bar R\bar S} \\
\cline{3-4}
\end{array}\ .
$$

We are proposing now to change the nonbasic variables x_j, $j \in S$, by the basic variables $x_{n+i},\ n+i \in R$. From the equalities

$$
\begin{cases}
x_R \ = \ A_{RS}x_S + A_{R\bar S}x_{\bar S} \\
x_{\bar R} \ = \ A_{\bar R S}x_S + A_{\bar R\bar S}x_{\bar S}
\end{cases},
$$

are immediately obtained the followings:

$$
\begin{cases}
x_S \ = \ A_{RS}^{-1}x_R - A_{RS}^{-1}A_{R\bar S}x_{\bar S} \\
x_{\bar R} \ = \ A_{\bar R S}A_{RS}^{-1}x_R + (A_{\bar R\bar S} - A_{\bar R S}A_{RS}^{-1}A_{R\bar S})x_{\bar S}
\end{cases}.
$$

Therefore, from the tableau (1.2) corresponding to the basis $B = R \cup \bar R$, one deduces the tableau

$$
(1.3)\qquad
\begin{array}{rc|cc|}
 & & x_R & x_{\bar S} \\
\cline{3-4}
x_S & = & A'_{SR} & A'_{S\bar S} \\
x_{\bar R} & = & A'_{\bar R R} & A'_{\bar R\bar S} \\
\cline{3-4}
\end{array}
$$

corresponding to the basis $B' = S \cup \bar R$, where

$$
(1.4)\qquad
\begin{cases}
A'_{SR} = A_{RS}^{-1},\ A'_{S\bar S} = -A_{RS}^{-1}A_{R\bar S},\ A'_{\bar R R} = A_{\bar R S}A_{RS}^{-1} \\
A'_{\bar R\bar S} = A_{\bar R\bar S} - A_{\bar R S}A_{RS}^{-1}A_{R\bar S}
\end{cases}.
$$

We call the passage from(1.2) to (1.3) (p–)*generalized Jordan step*, and the relations (1.4) are called *the formulas for changing the basis*. The matrix A_{RS} is called *pivot matrix*.

If $p = 1$, the passage is called *Jordan step*, in accordance with [4] and [5]. Taking into account the Steinitz theorem [5], a (p–)generalized Jordan step is equivalent to p Jordan steps.

The tableaux (1.2) and (1.3) are called *simplex tableaux*.

Using the elementary properties of the determinants, in the case $m = n$ we deduce

$$(1.5) \qquad det\begin{pmatrix} A'_{SR} & A'_{S\bar{S}} \\ A'_{\bar{R}R} & A'_{\bar{R}\bar{S}} \end{pmatrix} = \frac{1}{det(A_{RS})} det(A_{\bar{R}\bar{S}}) \ .$$

It is well-known the fact that A is nonsingular if and only if $A'_{\bar{R}\bar{S}}$ is nonsingular.

Let suppose now that the matrix A of $n \times n$ order is symmetric. We consider $p_1, \ldots, p_k$ natural numbers such that $1 \leq p_1 < \cdots < p_k \leq n$. We define the matrices $A^{(0)}, \ldots, A^{(k-1)}$ and $A^0, \ldots, A^{k-1}$, if it is possible, according to the following recurrent scheme:

1) $A^{(0)} := A$ and A^0 is the minor of A corresponding to the rows and columns $1, \ldots, p_1$;

2) if A^{h-1} is nonsingular, then $A^{(h)}$ is obtained from $A^{(h-1)}$ following a generalized Jordan step with A^{h-1} as pivot matrix,

and A^h is the minor of $A^{(h)}$ corresponding to the rows and columns $p_h + 1, \ldots, p_{h+1}$ $(h = 1, \ldots, k-1)$.

Proposition 1 *a) The matrix A is nonsingular if and only if the previous scheme is possible for $k = n$.*

b) The matrix A is positively defined if and only if the previous scheme is possible for $k = n$ and the numbers $A^h, h = 0, \ldots, n-1$ are positive.

Proof. The second assertion is a direct consequence of the well-known Jacobi's result and of the relation (1.5) expressed in the case $p = 1$.

Taking into account this proposition, we are able to state a general result.

Proposition 2 *a) The matrix A is nonsingular if and only if the previous scheme is generally possible.*

b) The matrix A is positively defined if and only if the previous scheme is generally possible and the matrices $A^h, h = 0, \ldots, k-1$ are positively defined.

2 Brief exposition of the classical Theil–Van de Panne algorithm

The Theil–Van de Panne algorithm is a specifically combinatorial method to minimize a quadratic strict convex function under the affine constraints:

$$(2.1)\qquad \begin{cases} e + d^\top x + \dfrac{1}{2} x^\top C x \to min \\ b + Ax \geq 0\ , \end{cases}$$

where $e \in \mathbf{R}$, $b \in \mathbf{R}^{m\times 1}$, $d \in \mathbf{R}^{n\times 1}$, $A \in \mathbf{R}^{m\times n}$, $C \in \mathbf{R}^{n\times n}$, and it is assumed that C is a symmetric and positively defined matrix.

Let denote $M = \{1, \ldots, m\}$ and $a_{i\cdot} = (a_{i1}, \ldots, a_{in}) \in \mathbf{R}^{1\times n}$ for each $i \in M$.

The basic idea of the algorithm, exposed in [1], [2] and [3], is to determine the set $S^\star \subset M$ of the indices whose constraints are precisely satisfied with equality by the optimal solution $x^\star$ and then to obtain $x^\star$ as the solution of an optimization problem where the constraints are only these equalities.

For $S \subset M$ we denote by x^S the (most unique) solution of the problem:

$$(2.2)\qquad \begin{cases} e + d^\top x + \dfrac{1}{2} x^\top C x \to min \\ \qquad b_i + a_{i\cdot} x = 0\ \ , i \in S. \end{cases}$$

Tacking into account the Kuhn–Tucker conditions, the solution of

(2.2) may be obtained from the system

$$(2.3)\qquad \begin{cases} d + Cx - A_S^\top v^S = 0 \\ b_S + A_S x = 0 \end{cases}$$

(b_S, respectively A_S, are the minors of b, respectively A, corresponding to S, and v^S is the vector of Lagrange multipliers). If the rows of A_S are linearly independent, then a standard calculus leads to

$$(2.4)\qquad \begin{cases} x^S = x^\phi - C^{-1}A_S^\top(A_S C^{-1} A_S^\top)^{-1}(b_S + A_S x^\phi) \\ v^S = -(A_S C^{-1} A_S^\top)^{-1}(b_S + A_S x^\phi). \end{cases}$$

The feasibility of x^S is stated by calculating

$$V(x^S) = \{i \in \bar{S} = M \setminus S : b_i + a_{i\cdot}x^S < 0\},$$

that $V(x^S)$ is the set of those indices of $\bar{S}$ corresponding to the negative elements of the vector

$$[2.5]\qquad b_{\bar{S}} + A_{\bar{S}}x^S = (b_S + A_{\bar{S}}x^\phi) -$$

$$-(A_{\bar{S}}C^{-1}A_S^\top)(A_S C^{-1}A_S^\top)^{-1}(b_S + A_S x^\phi).$$

Step 0. Compute $AC^{-1}A^\top$ and $b + Ax^\phi = b - AC^{-1}d$.

Step 1. $\mathcal{S} := \{\phi\}; \mathcal{T} := \phi$.

Step 2. Select $S \in \mathcal{S}$; $\mathcal{S} := \mathcal{S} \setminus \{S\}$.

Step 3. Compute x^S from (2.4). If this is impossible, pass to step 5. Compute $b_{\bar{S}} + A_{\bar{S}}x^S$ from (2.5) (using those from step 0) and $V(x^S)$.

Step 4. If $V(x^S) = \phi$, then

if $V(x^{S\setminus\{i\}}) \neq \phi$ for each $i \in S$, then

$S^\star := S$ and stop the algorithm,

else pass to step 5,

else $\mathcal{T} := \mathcal{T} \cup \{S \cup \{i\} : i \in V(x^S)\}$.

Step 5. If $\mathcal{S} \neq \phi$, then pass to step 2.

If $\mathcal{T} = \phi$, then (2.1) has no solution and stop the algorithm,

else $\mathcal{S} := \mathcal{T}$, $\mathcal{T} := \phi$ and pass to step 2.

3 Our simplex version of the algorithm

We use the notations of the previous section.

The classical version does not use the Kuhn–Tucker multipliers for inactive constraints. The solution of (2.2) may be obtained in a simplex tableau, using the Jordan step.

Denoting $y := b + Ax$ (≥ 0), first we write the tableau

$$\begin{array}{rc|c|cc|} & & 1 & x & v \\ \cline{3-5} 0 & = & d & C & -A^\top \\ y & = & b & A & O \\ \cline{3-5} \end{array} , \tag{3.1}$$

which contains the Kuhn–Tucker conditions of (2.1) (moreover $y, v \in \mathbf{R}_+^{m\times 1}, y_i v_i = 0,\ i \in M$, and v_i are the Kuhn–Tucker multipliers). To compute x^{ϕ} we use a n-generalized Jordan step with C as a pivot matrix or a sequence of n Jordan steps with the succesive elements of the principal diagonal of C as pivot elements. In the second way, applying Proposition 1, we check up that C is really positively defined.

Step 0. Change 0 by x, and eliminate them.

Then we get the simplex tableau

$$(3.2) \qquad y = \begin{array}{cc} 1 & v \\ \hline \multicolumn{1}{|c|}{q} & \multicolumn{1}{c|}{P} \\ \hline \end{array} ,$$

where $q = b - AC^{-1}d$, $P = AC^{-1}A^{\top}$ (and $x = C^{-1}A^{\top}v - C^{-1}d$). Let's denote by

$$(3.3) \qquad z = \begin{array}{cc} 1 & w \\ \hline \multicolumn{1}{|c|}{q^S} & \multicolumn{1}{c|}{P^S} \\ \hline \end{array}$$

the simplex tableau obtained by changing y_S by v_S, where $(z_S, w_S) = (v_S, y_S)$, and $(z_{\bar{S}}, w_{\bar{S}}) = (y_{\bar{S}}, v_{\bar{S}})$.

The other steps of the algorithm are the followings:

Step 1. $\mathcal{S} := \{\phi\}; \mathcal{T} := \phi$.

Step 2. Select $S \in \mathcal{S}$; $\mathcal{S} := \mathcal{S} \setminus \{S\}$.

Step 3. Change y_S by v_S. If this is impossible, pass to step 5.

Step 4. If $q^S \geq 0$, then $S^\star := S$ and stop the algorithm,
else $\mathcal{T} := \mathcal{T} \cup \{S \cup \{i\} : q_i^S < 0, i \in \bar{S}\}$.

Step 5. If $\mathcal{S} \neq \phi$, then pass to step 2.
If $\mathcal{T} = \phi$, then (2.1) has no solution and stop the algorithm
else $\mathcal{S} := \mathcal{T}$, $\mathcal{T} := \phi$ and pass to step 2.

Remarks

1. The different steps of the two versions of the algorithm are 0,3 and 4.

2. The calculus of x^S is not necessary at each iteration. For the final $x^\star$ we may use the relation $x^\star = (-C^{-1}d) + (C^{-1}A^\top)v^\star$.

3. If we possess the tableau (3.2) for S, and we must calculate it for $S' \supset S$, we apply the generalized Jordan step by changing only the y_i-variables ($i \in S' \setminus S$) by the corresponding v_i.

4. A revised form of the (generalized) Jordan step is useful after the form (3.2), because only the q^S-vector is important in the algorithm.

5. If in the problem (2.1) the x_j-variables are nonnegative ($x \geq 0$), we replace y_i by $x_{n+i}, i = 1, \ldots, m$, and, obviously, we do not eliminate them.

6. If in the problem (2.1) a part of the constraints are equalities, before the step 0 we eliminate the corresponding zeroes and the real

multipliers by means of the x_j-variables and the first zero-lines. A non-positively defined matrix may become a positively one.

4 Example.

We present now an example exposed in [1, pp.283-286].

$$(4.1)\begin{cases} \frac{1}{2}x_1^2+\frac{5}{2}x_2^2+x_3^2+2x_1x_2-x_1x_3-2x_2x_3-x_1+3x_2+4x_3\rightarrow min \\ x_1\geq 0 \\ -1+x_2\geq 0 \\ 2-2x_1-3x_2+x_3\geq 0 \\ 4-x_1-x_2-2x_3\geq 0 \\ 6-x_1+2x_2-x_3\geq 0\ . \end{cases}$$

According to (3.1), the first simplex tableau is:

$$\begin{array}{cc|r|rrrrrrrr|} & & 1 & x_1 & x_2 & x_3 & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline o_1 & = & -1 & 1 & 2 & -1 & -1 & 0 & 2 & 1 & 1 \\ o_2 & = & 3 & 2 & 5 & -2 & 0 & -1 & 3 & 1 & -2 \\ o_3 & = & 4 & -1 & -2 & 2 & 0 & 0 & -1 & 2 & 1 \\ y_1 & = & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ y_2 & = & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ y_3 & = & 2 & -2 & -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ y_4 & = & 4 & -1 & -1 & -2 & 0 & 0 & 0 & 0 & 0 \\ y_5 & = & 6 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}\ .$$

After the step 0 (changing o_j by x_j, $j = 1, 2, 3$, we see that all pivot elements are positive), we get, in accordance with (3.2) :

$$
(4.2)\qquad
\begin{array}{rc}
 & \\
y_1 & = \\
y_2 & = \\
y_3 & = \\
y_4 & = \\
y_5 & =
\end{array}
\begin{array}{|r|rrrrr|}
\multicolumn{1}{c}{1} & \multicolumn{1}{c}{v_1} & v_2 & v_3 & v_4 & \multicolumn{1}{c}{v_5} \\
\hline
8 & 6 & -2 & -5 & -6 & -11 \\
-6 & -2 & 1 & 1 & 1 & 4 \\
-2 & -5 & 1 & 6 & 6 & 8 \\
7 & -6 & 1 & 6 & 11 & 11 \\
-9 & -11 & 4 & 8 & 11 & 21 \\
\hline
\end{array},
$$

and in subsequent,

$$
(4.3)\qquad
\begin{array}{rc}
 & \\
x_1 & = \\
x_2 & = \\
x_3 & =
\end{array}
\begin{array}{|r|rrrrr|}
\multicolumn{1}{c}{1} & \multicolumn{1}{c}{v_1} & v_2 & v_3 & v_4 & \multicolumn{1}{c}{v_5} \\
\hline
8 & 6 & -2 & -5 & -6 & -11 \\
-5 & -2 & 1 & 1 & 1 & 4 \\
-3 & 1 & 0 & -1 & -3 & -2 \\
\hline
\end{array}.
$$

We put now $\mathcal{S} = \{\phi\}$ and $\mathcal{T} = \phi$.

In the first part of the algorithm we consider the subsets S with $card(S) = 0$. From (4.2) we have $q^{\phi} = (8, -6, -2, 7, -9)^{\top}$, hence $\mathcal{T} = \{\{2\}, \{3\}, \{5\}\}$.

In the second part we consider $card(S) = 1$. Changing in (4.2) y_2 by v_2 (resp. y_3 by v_3, and y_5 by v_5), we obtain
$q^{\{2\}} = (-4, 6, 4, 13, 15)^{\top}, q^{\{3\}} = (19/3, -17/3, 1/3, 9, -19/3)^{\top}$,

and

$q^{\{5\}} = (23/7, -30/7, 10/7, 82/7, 3/7)^\top$, so $\mathcal{T} = \{\{1,2\},\{2,3\},\{2,5\},\{3,5\}\}$.

In the third part we consider $card(S) = 2$. Changing in (4.2) $\{y_1, y_2\}$ by $\{v_1, v_2\}$ (resp. $\{y_2, y_3\}$ by $\{v_2, v_3\}$, etc.), we obtain $q^{\{1,2\}} = (2, 10, -2, 5, 9)^\top$, $q^{\{2,3\}} = (-8/5, 34/5, -4/5, 9, 59/5)^\top$, $q^{\{2,5\}} = (5, 18, -8, -8, -3)^\top$, and $q^{\{3,5\}} = (114/31, -125/31, -15/31, 336/31, 19/31)^\top$, hence $\mathcal{T} = \{\{1,2,3\},\{2,3,5\},\{2,4,5\}\}$.

In the fourth part we consider $card(S) = 3$. Selecting $S = \{1,2,3\}$ and changing $\{y_1, y_2, y_3\}$ by $\{v_1, v_2, v_3\}$ in (4.2) (or changing only y_3 by v_3 in the previous (3.3) simplex tableau for $S = \{1,2\}$), we obtain $q^{\{1,2,3\}} = (8, 18, 4, 1, 7)^\top \geq 0$, therefore $S^\star = \{1,2,3\}$. (For example, $q^{\{2,3,5\}} = (7/3, 250/9, 40/9, -32/3, -59/9)^\top$, $q^{\{2,4,5\}} = (13, 202, -32, 40, -59)^\top$, at this moment $\mathcal{T} = \{\{2,3,4,5\}\}$, but $q^{\{2,3,4,5\}}$ is impossible to be calculated.)

In order to calculate $x^\star$, we call up the formula (4.3):

$$x^\star = \begin{pmatrix} 8 \\ -5 \\ -3 \end{pmatrix} + \begin{pmatrix} 6 & -2 & -5 & -6 & -11 \\ -2 & 1 & 1 & 1 & 4 \\ 1 & 0 & -1 & -3 & -2 \end{pmatrix} (8, 18, 4, 0, 0)^\top = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

References

[1] Dragomirescu, M. and Maliţa, M. *Quadratic Programming. Introduction to Convex Programming* (in Romanian), Ed. Ştiinţifică, Bucharest 1968.

[2] Dragomirescu, M. and Maliţa, M. *Nonlinear Programming* (in Romanian), Ed.Ştiinţifică, Bucarest 1972.

[3] Kuenzi, H.P. et Krelle, W. *La programmation non–linéaire*, Gauthiers–Villards, Paris 1969.

[4] Maruşciac, I. *Solving Methods of Nonlinear Programming Problems* (in Romanian), Ed. Dacia, Cluj 1973.

[5] Zukhovitskii, S.I. and Avdeeva, I.I. *Linear and Convex Programming* (in Russian), Izd. Nauka, Moscow 1964.

Department of Mathematics
Faculty of Sciences ,
University "Lucian Blaga" of Sibiu
Str.I.Raţiu, No.7
2400, Romania