R. COROIU

Simplex Methods for Solving Quadratic Programming Problems

Quadratic programming simplex algorithms are extensions of the simplex method and differ but little from one another.

In this paper two quadratic programming algorithms are presented, which seem to be of wider utility owing to the manner in which they are built and the hypotheses employed.

The problems of non-linear programming have become object of study starting from 1951 when H.W. Kuhn and A.W. Tucker published their work "Nonlinear Programming", in which the conditions that are necessary and sufficient for the solutions to such problems to be optimum are determined. The study of this domain has quickly developed consequently to the ever more complex problems raised by economy and by the development of electronic computers.

Among them, a well defined place is held by quadratic programming, a domain of study to which a series of mathematicians have brought their contribution, and particularly E. Barankin, R. Dorfman, G.B. Dantzig, E.M.L. Beale, C. Van de Panne, A. Whinston and P. Wolfe.

A FEW BASIC NOTIONS

Let us consider a second degree function in the variables $x_1, \ldots, x_n$

$$f(x_1, x_2, \ldots x_n) = \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$

where

$$c_{ij} = c_{ji}.$$

We note by

$$k(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$

the quadratic form of the function $f(x_1, ..., x_n)$ which can be written in the form of an algebraic sum of squares of some linear, homogeneous and independent expressions. We note by k^+ the number of squares with positive coefficients, and by k^- the number of squares with negative coefficients. We obviously have relation

$$k^+ + k^- \leqslant n$$

We shall say that the quadratic form is *defined positively* if

$$k^+ = n \text{ and } k^- = 0$$

For

$$k^+ < n \text{ and } k^- = 0 \text{ the square form}$$

$f(x_1, x_2, ..., x_n)$ is *semidefined positively.*

One defines analogously the quadratic forms *defined negatively or semidefined negatively* for $k^+ = 0$ and $k^- = n$, respectively, $k^+ = 0$ and $k^- \leqslant n$.

If $k^+ > 0$ and $k^- > 0$ the quadratic form is undefined.

Is is proved that any quadratic form defined positively is strictly convex and any quadratic form semidefined positively is convex.

A problem of quadratic programming is stated in the general form as follows:

Let an n dimensional vector $x = (x_1, x_2, ..., x_n)$ be determined, which minimizes the convex quadratic form (or maximizes the concave form).

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j \tag{1}$$

satisfying the conditions

$$\sum_{j=1}^{n} a_{kj} x_j \leqslant b_k \qquad k = 1 \ldots m \tag{2}$$

$$x_j \geqslant 0 \qquad j = 1 \ldots n \tag{3}$$

In the matricial form:

Let the following convex quadratic form be minimized (or concave quadratic form be maximized)

$$f(x) = p'x + 1/2x'Cx \tag{4}$$

on the set of solutions to the system

$$Ax \leqslant b \tag{5}$$

$$x \geqslant 0 \tag{6}$$

where

$p = [p_1, p_2, ..., p_n]$ is an n dimensional column vector

$C = (c_{ij})$, $i, j = 1, ... n$, a symmetric matrix of size $n \times n$,

called quadratic form matrix.

$x = [x_1, ..., x_n]$ is an n dimensional column vector and x' its transpose.

KUHN AND TUCKER'S THEOREM

Let us assume we are in the case where the minimum of a convex quadratic function is to be determined. Kuhn and Tucker's theorem is stated as follows:

The necessary and sufficient condition for a vector $x^* = (x_1^*, ..., x_n^*)$ to be the optimum solution to the quadratic programming is for x^* to satisfy restrictions (2), (3) and for a vector $u = (u_1, ..., u_m)$ to exist, satisfying the conditions

$$v_i = 2\sum_{j=1}^{n} c_{ij}x_j^* + p_i + \sum_{k=1}^{m} u_k a_{ki} \tag{7}$$

$$v_i \geqslant 0, \quad u_k > 0 \quad i = 1, 2, ..., n, \quad k = 1, 2, ..., m \tag{8}$$

$$\sum_{i=1}^{n} v_i x_i^* = 0 \tag{9}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n} u_k(a_{kj}x_j^* - b_k) = 0 \tag{10}$$

Relations (7), (8), (9), (10), are Kuhn and Tucker's conditions.

This condition is sufficient.

Since x^* is the solution to the quadratic programming, we have

$$f(x^*) = \sum_{i=1}^{n} p_i x_i^* + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_i^* x_j^* \tag{11}$$

hence

$$f(x) - f(x^*) = \sum_{i=1}^{n} p_i(x_i - x_i^*) + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(x_i x_j - x_i^* x_j^*) \tag{12}$$

Since the quadratic function is convex we obtain

$$h(x - x^*) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(x_i - x_i^*)(x_j - x_j^*) \geqslant 0 \tag{13}$$

From (12) and (13) there results

$$f(x) - f(x^*) \geqslant \sum_{i=1}^{n} p_i(x_i - x_i^*) + 2\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_j^*(x_i - x_i^*)$$

and, on the basis of relations (7), (9) and (10)

$$f(x) - f(x^*) \geqslant \sum_{i=1}^{n} p_i(x_i - x_i^*) + \sum_{i=1}^{n} \left(v_i - p_i - \sum_{k=1}^{m} u_k a_{ki} \right)(x_i - x_i^*)$$

Developing the last parenthesis we have

$$f(x) - f(x^*) \geqslant \sum_{i=1}^{n} \left(v_i - \sum_{k=1}^{n} u_k a_{ki} \right)(x_i - x_i^*)$$

and

$$f(x) - f(x^*) \geqslant \sum_{i=1}^{n} v_i x_i - \sum_{k=1}^{n} \sum_{i=1}^{n} u_k (a_{ki} x_i - b_k)$$

but $a_{ki} x_i - b_k \leqslant 0$
and hence

$$f(x) - f(x^*) \geqslant 0$$

which implies

$$f(x) \geqslant f(x^*)$$

and therefore the required condition is sufficient.

To prove the necessity of the condition we use Minkovsky-Farkas's lemma which states:

If we have m linear and homogeneous inequalities with n variables

$$I_k = \sum_{j=1}^{n} a_{ij} x_j \geqslant 0 \qquad k = 1 \ldots m \tag{14}$$

which implies the linear and homogeneous inequality

$$I_0 = \sum_{j=1}^{n} a_{0j} x_j \tag{15}$$

the necessary and sufficient condition for I_0 to be a linear combination of the forms I_k is for m non-negative numbers u_k to exist, so that

$$I_0 = \sum_{k=1}^{m} u_k I_k \tag{16}$$

In the problem we are dealing with let us admit that $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is the optimum solution and that

$$x_j^* = 0 \text{ for } j \in N_1 = \{1, 2, \ldots, n_1\} \tag{17}$$

$$x_j^* > 0 \text{ for } j \in N_2 = \{n_1 + 1, \ldots, n\} \tag{18}$$

We also admit that some restrictions are satisfied with the sign (=) and the others with the sign (<). Let be:

$$\sum_{j=1}^{n} a_{ij}x_j^* = b_i \qquad i \in M_1 = \{1, \ldots m_1\} \tag{19}$$

$$\sum_{j=1}^{n} a_{ij}x_j^* < b_i \qquad i \in M_2 = \{m_1 + 1, \ldots m\} \tag{20}$$

Therefore, from relations (17) and (18) it becomes obvious that for any non-negative vector $x(x_1, \ldots, x_n)$ satisfying the problem restrictions we have

$$x_j - x_j^* \geqslant 0 \text{ for } j \in N_1 \tag{21}$$

and from (19), and (20) there results

$$\sum_{j=1}^{n} a_{ij}(x_j - x_j^*) \leqslant 0 \text{ for } i \in M_1 \tag{22}$$

On the other hand, based on the convexity of $f(x)$, we have relation

$$\sum_{j=1}^{n} \frac{\partial f(x^*)}{\partial x_j}(x_j - x_j^*) \geqslant 0 \tag{23}$$

According to Minkovsky-Farkas's lemma, applied to relations (21), (22), and (23) there are $n_1 + m_1$ non-negative numbers $v_1, v_2, \ldots, v_{n_1}$; $u_1, u_2, \ldots u_{m_1}$ so that

$$\sum_{j=1}^{n} \frac{\partial f(x^*)}{\partial x_j}(x_j - x_j^*) = \sum_{j=1}^{n_1} v_j(x_j - x_j^*) - \sum_{i=1}^{m_1} u_i \sum_{j=1}^{n} a_{ij}(x_j - x_j^*)$$

$$\sum_{j=1}^{n}\left(\frac{\partial f(x_j^*)}{\partial x_j} + \sum_{i=1}^{m_1} u_i a_{ij}\right)(x_j - x_j^*) = \sum_{j=1}^{n_1} v_j(x_j - x_j^*)$$

If we take $u_i = 0$ for $i = m_1 + 1 \ldots m$, and $v_j = 0$ for $j = n_1 + 1, \ldots n$ the sums may be extended, i acquiring values from 1, ..., m and $j = 1, \ldots, n$

$$\sum_{j=1}^{n}\left(\frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^{m} u_i a_{ij}\right)(x_j - x_j^*) = \sum_{j=1}^{n} v_j(x_j - x_j^*)$$

hence

$$\frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^{m} u_i a_{ij} = v_j$$

the necessity being thus proved.

F. Barankin and R. Dorfman have presented Kuhn and Tucker's theorem in an equivalent form which turns the problem of quadratic programming into a problem of linear programming.

Let us consider a problem of convex quadratic programming written in the form (1) — (3). In this case Kuhn and Tucker's conditions are written

$$v_i = p_i + 2\sum_{j=1}^{n} c_{ij}x_i x_j + \sum_{k=1}^{m} u_k a_{ki} \qquad i = 1, \ldots, n \tag{24}$$

$$y_k = b_k - \sum_{j=1}^{n} a_{kj}x_j \qquad k = 1, \ldots, m \tag{25}$$

$$\sum_{i=1}^{n} v_i x_i = 0 \tag{26}$$

$$\sum_{k=1}^{m} u_k y_k = 0 \tag{27}$$

$$x_i \geqslant 0,\ y_k \geqslant 0,\ v_i \geqslant 0,\ u_k \geqslant 0. \tag{28}$$

Owing to the conditions of non-negativity of vectors x, v, u and y, it can be seen that the products $u_k y_k$ and $v_i x_i$ are non-negative and hence their sum can only become zero if each product is zero. With this remark the problem can be re-written as follows:

Let the solutions to the following system be determined:

$$2\sum_{j=1}^{n} c_{ij}x_j + \sum_{k=1}^{m} u_k a_{ki} - v_i = -p_i \qquad i = 1 \ldots n \tag{29}$$

$$\sum_{j=1}^{n} a_{kj}x_j + y_k = b_k \qquad k = 1 \ldots m \tag{30}$$

$$x_i \geqslant 0,\ y_k \geqslant 0,\ u_k \geqslant 0,\ v_i \geqslant 0 \tag{31}$$

which minimize the function

$$G = \sum_{i=1}^{n} v_i x_i + \sum_{k=1}^{m} u_k y_k \tag{32}$$

a minimum which is obviously equal to zero.

The matrix of the problem written in the form (29) — (32) is of the dimensions $(m + n) \times 2(m + n)$ and its form is

$$A = \left\| \begin{array}{cccc} \overbrace{\quad}^{x} & & \overbrace{\quad}^{u} & \overbrace{\quad}^{v} \quad \overbrace{\quad}^{y} \\ c_{11} \cdots c_{1n} & & a_{11} \cdots a_{m1} & -1 \cdots 0 \quad 0 \cdots 0 \\ \vdots \quad \vdots & & \vdots \quad \vdots & \vdots \quad \vdots \quad \vdots \quad \vdots \\ c_{n1} \cdots c_{nn} & & a_{1n} \cdots a_{mn} & 0 \cdots -1 \quad 0 \cdots 0 \\ a_{11} \cdots a_{1n} & & 0 \cdots 0 & 0 \cdots 0 \quad 1 \cdots 0 \\ \vdots \quad \vdots & & \vdots \quad \vdots & \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{m1} \cdots a_{mn} & & 0 \cdots 0 & 0 \cdots 0 \quad 0 \cdots 1 \end{array} \right\|$$

$\bar{A}$ can also be written

$$\bar{A} = \begin{Vmatrix} C & A' & -E^{(n)} & 0 \\ A & 0 & 0 & E^{(m)} \end{Vmatrix}$$

where $E^{(n)}$ and $E^{(m)}$ are unit matrices of the order n, respectively m.

In the problem written in the form (29)—(32) the vector of the unknowns is $[x, u, v, y]$ and the free term vector is $d = [-p, b]$ and hence, in a matricial form the problem can be written

$$Cx + A'u - v = -p'$$
$$Ax + y = b$$
$$x \geqslant 0, y \geqslant 0, u \geqslant 0,$$
$$(\min)G = v'x + u'y.$$

The problem of quadratic programming has thus been turned into a problem of linear programming.

To every primal variable x, y there corresponds a dual variable (v, respectively u, and vice-versa).

Two kinds of tables can be found in a quadratic programme:

— *the standard table* has the property that if a primal variable stands in its basis, then the corresponding dual variable does not and vice-versa;

— *the non-standard table* is deprived of the above property.

In a non-standard table at most one variable x can stand in the basis together with its dual v; this pair of values is called *basic pair*. Likewise, there is at most one pair of corresponding variables x and v that do not both have a basis, and they form a non-basic pair.

The *simplex algorithm for quadratic programming*, also known under the name of *Whinston's algorithm*, is a simple generalization of the simplex method in linear programming.

By transforming the function objective yielded by (23) on the basis of formula (26), we obtain:

$$\begin{aligned} f(x) &= p'x + 1/2\ x'(v - A'u - p) \\ &= 1/2\ p'x + 1/2\ v'x - 1/2\ u'Ax \\ &= 1/2\ p'x + 1/2\ v'x - 1/2\ u'(b - y) = \\ &= 1/2\ p'x + 1/2\ v'x + 1/2\ u'b + 1/2\ u'y \end{aligned}$$

For any standard form table

$$v'x + u'y = 0$$

hence

$$f(x) = 1/2\ p'x + 1/2\ u'b \qquad (33)$$

and we shall note

$$F(x) = p'x + u'b \tag{34}$$

where

$$F(x) = 2f(x) \tag{35}$$

The simplex table for quadratic programming is of the following form:

Basic variables	Value of basic variables	v	u	x	y	F
v	$-p$	I	$-A'$	$-C$	0	0
y	b	0	0	A	I	0
F	0	0	$-b'$	$-p'$	0	1

As in the simplex method for linear programming one starts from an initial basic solution. As it can be seen from the table, the solution to the problem is: $y = b$, $x = 0$, $F = 0$, for the variables in the basis; all the variables not belonging to the basis (x and u) are zero in value.

If all the components of vector b are non-negative the solution $y = b$, $x = 0$, $F = 0$ can be reached and the initial solution has thus been determined. In the opposite case this solution must be determined by means of certain procedures.

We consider as known an initial basic solution to the problem and then, like in the case of a problem of linear programming, it is necessary to determine the vector entering in the basis according to the following rules:

— If the given table is in the standard form, it will enter in the variable basis x, to which corresponds the negative dual variable v, with the highest absolute value. In the opposite case, that is of the table in a non-standard form, one introduces in the basis the dual variable corresponding to the non-basic pair. One thus notices that the dual variables play the part of the differences $z_j - c_j$ of the primary simplex method used in linear programming.

— One determines the vector coming out of the basis by choosing

$$\min_i \ (b_i/a_{ik}; \quad b_i/a_{ik} \geqslant 0, \quad a_{ik} \neq 0) \tag{36}$$

where b_i is the i order component of the vector representing the value of the variables in the basis, and a_{ik} is the component in line i of the new basic variable.

The procedure, according to which the vector coming out of the basis is determined, is the same both for standard and for non-standard form tables. Since the function objective increases in any iteration starting from a table in the standard form and does not decrease in any iteration starting from a non-standard table, a fact that is quite easy to prove, the programme ends in a finite number of iterations.

As an illustration to the above let us solve the following example:

Let the function

$$f = 2x_1 + x_2 + 3x_3 - 2x_1^2 - 3x_2^2 - x_3^2 + 2x_1x_3$$

be maximized on the set of solutions of the system

$$2x_1 + x_2 + x_3 \leqslant 2$$

$$x_1, x_2, x_3 \geqslant 0$$

It is easy to verify that the function objective is positively semi-defined, hence, it is convex. According to the notations used

$$p = [2, 1, 3]' \qquad C = \begin{pmatrix} 4 & 0 & -2 \\ 0 & 6 & 0 \\ -2 & 0 & 2 \end{pmatrix} \qquad b = [2]'$$
$$A = [2, 1, 1]$$

Vector b has positive components and, by forming the initial table on basis of the given model, the solution in the zero iteration can be reached and hence the algorithm may be applied.

The variable x_3 goes in the basis since the most negative of the dual variables, $v_3 = -3$, corresponds to it.

To determine the vector coming out of the basis

$$\min \{-3/-2, 2/1\} = 3/2$$

s to be computed and hence x_3 replaces v_3 in the basis, the pivot being —2. The computations are made with the formulae known from linear programming.

Iteration 1. The new table obtained is of standard form and hence the method in the previous iteration will be used. Vector x_1, which replaces y, goes in the basis since

$$\min \left\{-5/-2, \frac{5}{2/3}\right\} = \frac{5}{6}$$

and the pivot is 3.

Iteration 2. After effecting the computation we obtain a new table in which the basis $B_2 = \{v_1, v_2, v_3, x_1 F\}$ contains a basic pair formed of the variable x_1 and its dual v_1; it is therefore a table in a non-standard form. According to the foregoing, the dual variable corresponding to the non-basic pair, that is u, will go in the basis, and v_2 will come out of the basis since

$$\min \left\{\frac{-14/3}{-10/3}, \frac{-1}{-1}, \frac{5/3}{1/3}\right\} = 1$$

and the pivot is —1, hence the table obtained is in a non-standard form. For this reason the value of the function objective which is 16/3 should be corrected, the product $v_1 x_1 = -7/9$ should be deducted and $F^* = 16/3 + 7/9 + 55/9$ is obtained.

Iteration 3. The new table obtained is also in the non-standard form and v_2 goes in the basis. Since

$$\min\left\{\frac{-4/3}{-10/3};\ \frac{4/3}{1/3}\right\} = \frac{2}{5}$$

the pivot is $\frac{-10}{3}$ and vector v_1 comes out of the basis. The value of the function objective should again be corrected; it will then be 64/9.

Iteration 4. The new table obtained has as a basis $B_4 = \{v_2, u, x_3, x_1, F\}$; hence, it is of a standard form and, since it no longer contains any dual variable of negative value, the solution is optimum.

The optimum solution sought is then

$$x_1 = 2/5, \quad x_2 = 0, \quad x_3 = 6/5.$$

The value $F = 36/5$ gives us the double of the function objective, hence

$$f = 18/5.$$

Iterations for Whinston's method

Iteration	Basis	Value of basic variables	v_1	v_2	v_3	x_1	x_2	x_3	y	u	F	F^*
0	v_1	—2	1			—4	0	2		—2		
	v_2	—1		1		0	—6	0		—1		
	v_3	—3			1	2	0	—2*		—1		
	y	2				2	1	1	1	0		
	F	0				—2	—1	—3		—2	1	
1	v_1	—5	1		1	—2	0			—3		
	v_2	—1		1	0	+0	—6			—1		
	x_3	3/2			—1/2	—1	0	1		1/2		
	y	1/2			1/2	3*	1		1	—1/2		
	F	9/2			—3/2	—5	—1			—1/2	1	
2	v_1	—14/3	1		4/3		2/3		2/3	—10/3		
	v_2	—1		1	0		—6		0	—1*		
	x_3	5/3			—1/3		1/3	1	1/3	1/3		
	x_1	1/6			1/6	1	2/3		1/3	—1/6		
	F	16/3			—2/3		2/3		5/3	—4/3	1	55/9
3	v_1	—4/3	1	—10/3*	4/3		62/3		2/3			
	u	1		—1	0		6		0	1		
	x_3	4/3		1/3	—1/3		—5/3	1	1/3			
	x_1	1/3		—1/6	1/6	1	4/3		1/3			
	F	20/3		—4/3	—2/3		26/3		5/3		1	64/9
4	v_2	2/5	—3/10	1	—2/5		—31/5		—1/5			
	u	7/5	—3/10		—2/5		— 1/5		—1/5	1		
	x_3	6/5	1/10		—1/5		2/5	1	2/5			
	x_1	2/5	—1/20		1/10	1	3/10		3/10			
	F	36/5	—2/5		—6/5		2/5		7/5		1	

WOLFE'S ALGORITHM

A problem of quadratic programming of the form

$$\min f(x, \lambda) = \lambda px + 1/2 \quad x'Cx \tag{37}$$

is given, with the conditions

$$Ax = b \tag{38}$$
$$x \geqslant 0$$

where λ is a real, non-negative parameter and the other magnitudes have the meaning shown above.

On the basis of the results obtained by Barankin and Dorfman, Philip Wolfe has established a simplex method for solving problems of quadratic programming. This method has two forms: The *short form*, positively defined for the case where $\lambda = 0$ or C. The *long form*, positively semi-defined for the case $\lambda \geqslant 0$ or C.

SHORT FORM OF WOLFE'S ALGORITHM

The results obtained by Barankin and Dorfman give the possibility of replacing the problem of quadratic programming by the following problem of linear programming.

$$\min \sum_i w_i \tag{39}$$

with the conditions

$$\begin{aligned} &Ax + w = b \\ &v - A'u - Cx + \alpha - \beta = -\lambda p' \\ &x, v, w, \alpha, \beta \geqslant 0 \end{aligned} \tag{40}$$

This problem constitutes a first stage in solving the stated problem, since it does not contain the condition $u'x = 0$, a condition that is necessary if x is to be the solution to the initial problem. This first stage is called the initial phase and its aim is to determine a solution that can be achieved for the $Ax = b$ system.

The simplex algorithm is used for minimizing the objective function. Since $b \geqslant 0$ an initial basis is formed for this system by means of the vectors α, β and w, taking u and v as equal to zero. The problem thus has $m + n$ equations and $m + 3n$ variables.

After solving this first part vector w and all the components of the vectors α and β not belonging to the optimum basis are excluded from the model. The n remaining components are noted by z and the matrix of their coefficient by E. We have thus obtained a solution to the system:

$$\begin{aligned} &Ax = b \\ &Cx + A'u - v + Ez = -\lambda p' \\ &x, v, z > 0 \end{aligned} \tag{41}$$

Recursive phase. A basis being given, and a basic solution satisfying the system (41) and the conditions: $v'x = 0$ and $\sum_{j=1}^{n} z_j > 0$, a change of basis is made by the simplex method, in view of minimizing the linear form

$$\sum_{j=1}^{n} z_i \tag{42}$$

with the following condition:

— if x_k belongs to the basis, v_k cannot go in the basis and reciprocally,
— if v_k belongs to the basis, x_k cannot go in the basis. (43)

The model in the recursive phase has, like the one in the initial phase, $(m + n)$ equations with $(m + 3n)$ unknowns. The minimum of the objective function is zero $(z = 0)$.

LONG FORM OF WOLFE'S ALGORITHM FOR SOLVING A PROBLEM OF QUADRATIC PROGRAMMING

As already mentioned, the long form of Wolfe's algorithm is used to solve problems of the form (37)—(38) where $\lambda \geqslant 0$ and it can be broken up in three stages:

— the solution is determined for $\lambda = 0$;

— the maximum value λ_M is determined, for which this solution keeps its character of optimum solution;

— the solution for $\lambda > \lambda_M$ is determined; hence, a method similar to the one in quadratic programming.

Initial phase.

One determines

$$\min \sum_{i=1}^{n} w_i \tag{44}$$

with the conditions

$$\begin{gathered} Ax + w = b \\ Cx + A'u - v + \alpha - \beta = 0 \\ x, v, \alpha, \beta, w \geqslant 0 \end{gathered} \tag{45}$$

This model was obtained from (39)—(40) for $\lambda = 0$. The minimum value of the function objective must be zero and we have obtained an initial solution for

$$\min \sum_{j=1}^{n} z_j$$

with the conditions

$$Ax = b$$
$$Cx + A'u - v + \mu p' + Ez = 0 \tag{46}$$
$$x, v, z \geqslant 0$$

where $\mu = 0$. Solving this problem, with the supplementary condition $v'x=0$, one obtains the solution for which

$$\min \sum_{j=1}^{n} z_j = 0$$

Recursive phase

Ending the initial phase, a basic solution was obtained for the model

$$\min \sum_{j=1}^{n} \mu_j \tag{47}$$

with the conditions (46). The solution to this problem is obtained by using the simplex algorithm with the following supplementary conditions:

— if x_k belongs to the basis, v_k cannot go in the basis
— if v_k belongs to the basis, x_k cannot go in the basis (48)
— no z_j can go in the bais.

If no change of the basis can be carried out under these conditions, then $\mu = 0$, $F(\lambda) = -\infty$ for all $\lambda > 0$, hence the problem has no finite minimum for $\lambda \neq 0$.

If a change of the basis is possible, the recursive phase will yield a finite succession of values

$$0 = \mu^0 < \mu^1 < \ldots < \mu^k$$

as well as the optimum solutions corresponding to each values of the parameter $x^0, x^1, \ldots, x^k$.

It can be proved that the optimum solution to the problem of quadratic programming for $\mu^l \leqslant \lambda \leqslant \mu^{l+1}$ is obtained by interpolation from the solutions corresponding to the characteristic values μ^l and μ^{l+1}, i.e.

$$x_\lambda = \frac{\mu^{l+1} - \lambda}{\mu^{l+1} - \mu^l} x^l + \frac{\lambda - \mu^l}{\mu^{l+1} - \mu^l} x^{l+1} \tag{49}$$

and for

$$\lambda > \mu^l \tag{50}$$
$$x_\lambda = x^l + (\lambda - \mu^l) x^\infty$$

where the optimum solution for values of the parameter superior to value μ^l were noted by x^∞.

Hence, in the recursive phase of Wolfe's algorithm, the simplex algorithm is used, completed by certain conditions for the going of a vector in the basis, conditions which do not appear in the simplex algorithm for linear programming.

Example;

Let the minimum of the function

$$f = 2x_1 - x_2 - 3x_3 + 2x_1^2 + 3x_2^2 + 5/2x_3^2 - 2x_1x_3$$

be determined, with the conditions

$$2x_1 + x_2 + x_3 \leqslant 2$$

$$x_1 + 3x_2 + 2x_3 \geqslant 1$$

In this example we have

$$p = [2, \quad -1, \quad -3]$$

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \end{pmatrix} \quad b = [2, 1] \quad C = \begin{pmatrix} -4 & 0 & 2 \\ 0 & -6 & 0 \\ 2 & 0 & 5 \end{pmatrix}$$

The fact that matrix C is positively defined can be verified and, consequently, the problem can be included in the conditions required for the short form of Wolfe's method.

Initial phase

We turn the restriction system into an equation system

$$2x_1 + x_2 + x_3 + w_1 = 2$$

$$x_1 + 3x_2 + 2x_3 + w_2 - w_3 = 1$$

The initial phase aims to determine an achievable solution to this system, hence the function objective is

$$(\min)\ w = w_2$$

the conditions being written in matricial from

	x_1	x_2	x_3	u_1	u_2	v_1	v_2	v_3	α_1	α_2	α_3	β_1	β_2	β_3	w_1	w_2	w_3	
C	−4	0	2	2	1	−1			1			−1						−2
	0	−6	0	1	3		−1			1			−1					1
	2	0	5	1	+2			−1			1			−1				3
A	2	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
	1	3	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

The optimum solution for the initial phase was obtained after a single change of basis and min $w_2 = 0$.

The variables α_2, α_3 and β_1 belong to the optimum basis in the initial phase and in the recursive phase they will be noted $\beta 1 = z_1$, $\alpha_2 = z_2$, $\alpha_3 = z_3$ and the variables α_1, β_2, β_3 which do not belong to the optimum basis no longer appear in the model for the recursive phase. The variable w_2 disapperas as well.

Simplex table for initial phase

$c_j \rightarrow$			0			0		0			0			0			0		1	0
C_B	*Basis*	Constants	x_1	x_2	x_3	u_1	u_2	v_1	v_2	v_3	α_1	α_2	α_3	β_1	β_2	β_3	w_1	w_2	w_3
0	β_1	2	4	0	−2	−2	−1	1			−1			1					
0	α_2	1	0	−6	0	1	3		−1			1			−1				
0	α_3	4	2	0	5	1	+2			−1			1			−1			
0	w_1	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	w_2	1	1	3	2	0	0	0	0	0	0	0	0	0	0	0	0	1	−1
		1	1	3	2	0	0	0	0	0	0	0	0	0	0	0	0	1	−1
0	β_1	2	4		−2	−2	−1	1			−1			1				0	0
0	α_2	3	2		−4	1	3		−1			1			−1			2	−2
0	α_3	4	2		5	1	+2			−1			1			−1		0	0
0	w_1	5/3	5/3		5/3	0	0										1	−1/3	1/3
0	x_2	1/3	1/3	1	2/3	0	0											1/3	−1/3
			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Recursive phase

Let

$$z = \min \sum_{j=1}^{3} z_i \tag{51}$$

be determined, with the conditions

$$\begin{vmatrix} x_1 & x_2 & x_3 & u_1 & u_2 & v_1 & v_2 & v_3 & z_1 & z_2 & z_3 & w_1 & w_3 \\ 4 & & -2 & -2 & -1 & 1 & & & 1 & & & & 0 \\ 2 & & -4 & 1 & 3 & & -1 & & & 1 & & & -2 \\ 2 & & 5 & 1 & +2 & & & -1 & & & 1 & & 0 \\ 5/3 & & 5/3 & 0 & 0 & & & & & & & 1 & 1/3 \\ 1/3 & 1 & 2/3 & 0 & 0 & & & & & & & & -1/3 \end{vmatrix} = \begin{vmatrix} z_1 \\ z_2 \\ z_3 \\ w_1 \\ x_2 \end{vmatrix}$$

In the recursive phase the so-called factors of the cost relation for function (51) must again be computed, and the columns α_1, β_2, β_3 and w_2 will be crossed out. The computation will then be carried out, using the simplex algorithm of linear programming.

Recursive phase

C_j			0	0	0	0	0	0	0	0	1	1	1	0	0
C_B	Basis	Constants	x_1	x_2	x_3	u_1	u_2	v_1	v_2	v_3	z_1	z_2	z_3	w_1	w_3
1	z_1	2	—4		—2	—2	—1	1			1				0
1	z_2	3	2		—4	1	3		—1			1			—2
1	z_3	4	2		5	1	2			—1			1		0
0	w_1	5/3	5/3		5/3	0	0							1	1/3
0	x_2	1/3	1/3	1	2/3	0	0								—1/3
		9	00		—1	0	4	1	—1	—1	0	0	0	0	—2
0	v_1	3	—10/3		—10/3	—5/3		1	—1/3		1	1/3			0
0	v_2	1	2/3		—4/3	1/3	1		—1/3			1/3			—2/3
1	z_3	2	2/3		23/3	1/3			2/3	—1		—2/3		1	0
0	w_1	5/3	5/3		5/3	0			0		0			1	1/3
0	x_2	1/3	1/3	1	2/3	0			0		0				—1/3
		2	—8/3	0	13/3	— 4/3	0	1	1/3	—1	0	—4/3	0	0	2/3
0	v_1	89/23	—70/23			—35/23		1	—1/23	—10/23		1/23	10/23	12/5	4/5
0	u_2	31/23	—18/23		1	9/23	1		—5/23	—4/23		1/23	4/23	—6/5	0
0	x_3	6/23	2/23			1/23			2/23	—3/23		—2/23	3/23	—6/5	2/5
0	x_1	85/69		1		— 5/69			—10/69	5/23		10/69	—5/23	3/5	1/5
0	x_2	11/69	16/69			— 2/69			—4/69	2/23		4/69	—2/23	—1/5	—2/5
		0	0	0	0	0	0	0	0	0	—1	—1	—1	—12/5	—14/5

The optimum solution to the suggested problem is

$$x_1 = 85/69,\ x_2 = 11/69,\ x_3 = 6/23$$

and the value of the function objective is

$$f = \frac{18566}{4761}$$

REFERENCES

1. PANNE, C., VAN DE, and WHINSTON, A., *The Simplex and the Dual Method for Quadratic Programming*, "Operational Research Quarterly", vol. 15, No. 4, 1964, pp. 355—388.
2. WOLFE, P., *The Simplex Method for Quadratic Programming*, "Econometrica", vol. 27, 1959, No. 3-4. pp. 382 — 398.
3. BARANKIN, E., W. and DORFMAN, R., *On Quadratic Programming*, "University of California — Publications in Statistics", vol. 3, No. 13, pp. 285—318
4. BEALE, E. M. L., *On Optimizing a Convex Function Subject to Linear Inequalities*, "Journal of the Royal Statistical Society", No. 17, 1955, pp. 173—184
5. MIHOC, GH. and NADEJDE, I., *Programarea matematică*, Editura Ştiinţifică, 1967
6. DANTZIG, G., B., *Linear Programming and Extensions*, Chap. 24, pp. 471—497
7. KUHN, H. W., and TUCKER A., W., *Nonlinear Programming*, "Proceedings. Second Berkeley Symposium on Mathematical Statistics and Probability", 1951, pp. 481—492.

METHODES SIMPLEX DE SOLUTION DES PROBLEMES DE PROGRAMMATION QUADRATIQUE

Le présent ouvrage expose deux méthodes de solution des problèmes de programmation quadratique, à savoir, les algorithmes de programmation quadratiques qui semblent être d'une large utilité, grâce à leur construction et aux hypothèses avec lesquelles elles travaillent. Les deux algorithmes présentés, aussi bien celui de Whinston que celui de Wolfe (avec ses deux formes, la brève et la longue) sont des extensions de la méthode simplex de programmation linéaire et diffèrent peu entre elles.

La solution des problèmes de programmation quadratique est fondée sur l'ouvrage de H. W. Kuhn et A.W. Tucker, qui fixe les conditions nécessaires et suffisantes pour obtenir les meilleures solutions d'un problème de programmation non-linéaire. E. Barankin et R. Dorfman ont présenté le theorème de Kuhn et Tucker sous une forme équivalente qui fait du problème de programmation quadratique un problème de programmation linéaire.

C'est pourquoi, on a estimé nécessaire de donner, avant de présenter les algorithmes proprement-dits, certaines notions fondamentales, susceptibles de faciliter la compréhension du matériel exposé. C'est toujours à cette fin de clarification qu'on a appliqué chacune des méthodes présentées à un cas précis.

МЕТОД СИМПЛЕКС ДЛЯ РЕШЕНИЯ ЗАДАЧ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

Настоящая работа представляет два метода решения задач квадратичного программирования, а именно: алгоритмы квадратичного программирования, которые кажутся более широко используемыми, благодаря методу, по которому они построены, и гипотезы, по которыми работают. Оба представленных алгоритма, как алгоритм Уинстона, а также Вольфа (в его двух формах, — короткая и длинная — являются расширением метода симплекс линейного программирования и немного отличаются одни от других.

В основе решения задач квадратичного программирования лежит работа Куна и Таккэра, устанавливающая условия, необходимые и достаточные для того чтобы решение задачи нелинейного программирования было бы оптимальным. Е. Баранкин и Р. Дорфман представили теорему Куна и Таккэра в эквивалентной форме, преобразующей задачу квадратичного программирования в задачу линейного программирования.

Исходя из этих мотивов, считается необходимым, чтобы, прежде чем представить собственно алгоритмы, было дано несколько основных понятий, облегчающих понимание предлагаемого материала. Также для уточнения сказанного были применены представленные методы к конкретным случаям.