Cowles Foundation Paper 235
Reprinted in Naval Research Logistics Quarterly, 12(2), 1965

THE BOUNDED VARIABLE PROBLEM — AN APPLICATION OF THE DUAL METHOD FOR QUADRATIC PROGRAMMING*†

Andrew Whinston

University of Virginia

ABSTRACT

The paper presents an algorithm to solve the bounded variable quadratic programming problem. The algorithm is a direct extension of an earlier algorithm of H. Wagner (N.R.L.Q. 1958) for the case of a bounded variable linear programming problem.

INTRODUCTION

In this paper we shall develop a compact and efficient algorithm for solving the following problem:

(1) $$\text{Max } p'x - \frac{1}{2}x'Cx,$$

subject to

(2) $$Ax = b,^{‡}$$

(3) $$0 \leqq x \leqq \tilde{b},$$

where

$$x = (x_1, \ldots, x_n)',$$

$$\tilde{b} = (\tilde{b}_1, \ldots, \tilde{b}_n)',$$

$$b = (b_1, \ldots, b_m)',$$

and

$$p = (p_1, \ldots, p_n)'.$$

A is an m×n matrix, and C is a symmetric n×n positive semi-definite matrix. In principle this problem may be handled by the various quadratic programming algorithms by explicitly taking into account the upper bound restrictions of (3). In this note we shall indicate how to solve this problem but without having to work with a tableau which includes the conditions of (3).

The problem of (1), (2), and (3) has been discussed in the case when the matrix C is composed entirely of zeros, i.e., the case of a linear programming problem with bounded

*This paper is an outgrowth of some joint work with Mr. C. van de Panne as reported in [5].
†Research undertaken by the Cowles Foundation for Research in Economics under task NR 047-006 with the Office of Naval Research.
‡This form of the constraints may be obtained by adding appropriate slack variables.

variables.* For this case one very simple and efficient method has been developed by Wagner [6]. Wagner, in effect, was able to show that with several slight modifications of the dual method of Lemke [4] an algorithm was at hand which could solve the problem without enlarging the tableau to account explicitly for the bounded restrictions.

Recently, van de Panne and Whinston have presented a new quadratic programming algorithm which they call "The Dual Method for Quadratic Programming" [5]. The method is referred to as a dual method since among other things for the case where C is composed of zeros, application of this method is equivalent to the dual method of Lemke for linear programming [4]. We shall show that the bounded variable problem can be handled by a slight modification of the dual method presented in [5] without having to include the restrictions (3) in the tableau. The algorithm will also have the feature that in the case where C is composed of zeros it will reduce essentially to Wagner's algorithm. In this sense the algorithm presented here can be considered as an extension of Wagner's to the case of quadratic programming.

THE ALGORITHM

The relevant Kuhn-Tucker conditions [3] for the problem (1), (2), and (3) are:

$$u - \tilde{u} - A'v - Cx = -p, \tag{4}$$

$$Ax = b, \tag{5}$$

$$u'x = 0, \tag{6}$$

$$\tilde{u}'(x - \tilde{b}) = 0, \tag{7}$$

$$x \leqq \tilde{b}, \tag{8}$$

and

$$x \geqq 0, \quad u \geqq 0, \quad \tilde{u} \geqq 0, \tag{9}$$

where

$$u = (u_1, \ldots, u_n)',$$

$$\tilde{u} = (\tilde{u}_1, \ldots, \tilde{u}_n)',$$

and

$$v = (v_1, \ldots, v_m)'.$$

Introducing the variables

$$y = (y_1, \ldots, y_m)',$$

and

$$\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)',$$

we have, replacing several of the conditions

*See [1], [2], and [6].

(5)' $$Ax + y = b,$$

(7)' $$\tilde{u}'\tilde{x} = 0,$$

(8)' $$x + \tilde{x} = \tilde{b},$$

and the added conditions

(10) $$y = 0,$$

and

(11) $$\tilde{x} \geqq 0.$$

Based on these conditions we may form the following Simplex tableau:

TABLEAU I

Value of Basic Variables	u	$\tilde{u}$	v	x	y	$\tilde{x}$
-p	I	-I	-A'	-C	0	0
b	0	0	0	A	I	0
$\tilde{b}$	0	0	0	I	0	I

However, we shall use the following Tableau II which is associated with the problem (1) and (2) where we have as a starting solution

$$u = -p$$

and

$$y = b.$$

Without loss of generality we may assume the vector $-p \geqq 0$.* For each i, we refer to the pairs (x_i, u_i), (v_i, y_i), and $(\tilde{u}_i, \tilde{x}_i)$ as complementary variables. Before giving the rules of the algorithm we make certain preliminary remarks. The tableau is transformed in the usual simplex manner once the vector to come in and leave the basis is determined. Let $\{z_i\}$ be the set of variables which are permitted to leave the basis in a particular iteration and $\{\ell_{ij}\}$ be

TABLEAU II

Basic Variable	Value of Basic Variables	u	v	x	y
u	-p	I	-A'	-C	0
y	b	0	0	A	I

*See [5] for a discussion of starting solutions.

the elements in the j^{th} column of the tableau where we assume that this is the column of the variable to come in the basis. Then the variable to leave the basis is that z_i for which

$$\underset{i}{\text{Min}} \left\{ k \frac{z_i}{\ell_{ij}} \,\middle|\, k \frac{z_i}{\ell_{ij}} \geqq 0 \quad \ell_{ij} \neq 0 \right\}$$

is achieved where $k = 1$ if the new variable is introduced in a positive amount and $k = -1$ if it is introduced in a negative amount. The algorithm proceeds in the following manner:*

1. Among the set of x, $\tilde{x}$, and $-|y|$ variables choose the most negative. If there are none, the algorithm is terminated.

2. Introduce into the tableau the complementary u, $\tilde{u}$, or v variable to the one chosen in step 1 with the opposite sign to its complement. Choose the variable to be removed from among the u and $\tilde{u}$ variable in the basis and the variable designated in step 1. If the variable designated in step 1 is removed return to step 1; if not, go to step 3.

3. Introduce the x or $\tilde{x}$ variable corresponding to the u or $\tilde{u}$ variable which just left the basis. The variable to be eliminated is chosen from the u and $\tilde{u}$ variables in the basis and the variable designated in step 1. If a u or $\tilde{u}$ variable is eliminated go back to step 3, if not go to step 1.

4. At any stage of the algorithm if an x or $\tilde{x}$ variable exceeds its upper bound (i.e., some x_i or $\tilde{x}_i$ is greater than the corresponding $\tilde{b}_i$) perform the following operations on the tableau before proceeding to the designated step in the process: In the row of the tableau associated with the $x(\tilde{x})$ variable change all the signs except for the elements in the column of the $x(\tilde{x})$ variable. Change all the signs in the corresponding $u(\tilde{u})$ variable's column. In the columns designating the $x_i(\tilde{x}_i)$ variable and its corresponding $u_i(\tilde{u}_i)$ variable change their headings to $\tilde{x}_i(x_i)$ and $\tilde{u}_i(u_i)$.

The proof of convergence of the algorithm can be obtained directly from [4] as soon as we make certain observations on properties of Tableau I. In effect, we shall indicate that using rules (1), (2), (3), and (4) on Tableau II is equivalent to applying rules (1), (2), and (3) to Tableau I. Since convergence has been proved for Tableau I in [5], which includes all the constraints, the method proposed here will also converge to the optimum.

When rules (1), (2), and (3) are applied to Tableau I it is clear that the tableau elements in the columns of u_1 and $\tilde{u}_i$ will have opposite signs. The second observation can be best put in terms of a lemma concerning Tableau I.

LEMMA 1: Consider an iteration when both x_j and $\tilde{x}_j$ for some value of j are in the basis at values K_j and $\tilde{K}_j$, respectively. Let $\{h_i\}$ be the set of nonbasic variables, $\{a_{ij}\}$ the tableau elements opposite x_j in the tableau and $\{\tilde{a}_{ij}\}$ the elements opposite $\tilde{x}_j$. Then we will have

$$a_{ij} = -\tilde{a}_{ij} \quad \text{for each } i.$$

PROOF: We know that at any step in the procedure

*Rules (1), (2), and (3) are based on the rules for the dual method as described in [5].

(12) $$x_j + \tilde{x}_j = \tilde{b}_j .$$

(Note that in Tableau I no $\tilde{y}_j$ variables are introduced.) Thus for the iteration under consideration

(13) $$K_j + \tilde{K}_j = \tilde{b}_j .$$

From the tableau we have the following two equations:

(14) $$\sum_i a_{ij} h_i + K_j = x_j$$

and

(15) $$\sum_i \tilde{a}_{ij} h_i + \tilde{K}_j = \tilde{x}_j .$$

By adding (14) and (15) we have

$$\sum (a_{ij} + \tilde{a}_{ij}) h_i + K_j + \tilde{K}_j = x_j + \tilde{x}_j .$$

By combining with (13) and noting that $\{h_i\}$ can be chosen arbitrarily, we have the result

q·e·d.

EXAMPLE OF THE ALGORITHM

We consider the following problem:

(16) $$\max_{x_1, x_2} -6x_1 - 2x_1^2 + 2x_1 x_2 - 2x_2^2 ,$$

(17) $$x_1 + x_2 = 2 ,$$

(18) $$x_1 \leqq 2 ,$$

(19) $$x_2 \leqq 1 ,$$

and

(20) $$x_1 \geqq 0, \quad x_2 \geqq 0 .$$

The simplex tableau format for the problem involving constraints (17) is seen in Tableau III. Applying rule 1 we see that y variable is selected. In the next tableau the v variable is introduced and the u_2 variable removed. Since y was not removed we apply rule 3 to Tableau IV. Because u_2 left the basis we introduce x_2 into the basis and then u_1 is to be removed. Since we have not yet removed y we must apply again rule 3 and in Tableau V we introduce x_1 (since u_1 was just removed). Since no u variables are in the basis we may eliminate y. We note that x_2 now exceeds its upper limit so we apply rule 4 and obtain Tableau VII. In Tableau VII we have $\tilde{x}_2 < 0$ and according to rule 2 we introduce $\tilde{u}_2$. In Tableau VIII

TABLEAU FOR EXAMPLE

Tableau	Basic Variables	Values of Basic Variables	u_1	u_2	v_1	x_1	x_2	y
	u_1	6	1	0	-1	-4	+2	0
Tableau III	u_2	0	0	1	-1	+2	-4	0
	y	2	0	0	0	1	1	1
	u_1	6	1	-1	0	-6	+6	0
Tableau IV	v	0	0	-1	1	-2	+4	0
	y	2	0	0	0	1	1	1
	x_2	+1	1/6	-1/6	0	-1	1	0
Tableau V	v	-4	-2/3	-1/3	1	+2	0	0
	y	+1	-1/6	+1/6	0	+2	0	1
	x_2	3/2	+1/12	-1/12	0	0	1	+1/2
Tableau VI	v	-5	-1/2	-1/2	1	0	0	0
	x_1	1/2	-1/12	+1/12	0	1	0	+1/2
Tableau	**Basic Variables**	**Values of Basic Variables**	u_1	$\tilde{u}_2$	v_1	x_1	$\tilde{x}_2$	y
	$\tilde{x}_2$	-1/2	-1/12	-1/12	0	0	1	-1/2
Tableau VII	v	-5	-1/2	+1/2	1	0	0	0
	x_1	1/2	-1/12	-1/12	0	1	0	1/2
	$\tilde{u}_2$	6	+1	+1	0	0	-12	+6
Tableau VIII	v	-8	-1	0	1	0	-0	-3
	x_1	1	0	0	0	1	-1	1

$\tilde{u}_2$ is introduced and $\tilde{x}_2$ eliminated so that we return to rule 1. Checking the tableau elements we see that the algorithm terminates. The solution is $x_1 = 1$, $x_2 = 1$.

REFERENCES

[1] Charnes, A. and Lemke, C. E., "Computational Theory of Linear Programming I: The Bounded Variables Problem," ONR Research Memo No. 10, Graduate School of Industrial Administration, Carnegie Institute of Technology (Jan. 1954).

[2] Dantzig, G. B., "Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming," Econometrica, 23, 174-183 (1955).

[3] Kuhn, H. W. and Tucker, A. W., Non-Linear Programming, Proc. Second Berkeley Symposium (University of California Press, 1951.

[4] Lemke, C. E., "The Dual Method of Solving the Linear Programming Problem," Naval Research Logistics Quarterly, 1, 36-47 (1954).

[5] Panne, C. van de and Whinston, Andrew, "Simplicial Methods for Quadratic Programming," Naval Research Logistics Quarterly, 11, 273-302 (1964).

[6] Wagner, Harvey M., "The Dual Simplex Algorithm for Bounded Variables," Naval Research Logistics Quarterly, 5, 257-262 (1958).