AN ABSTRACT OF THE THESIS OF

RODNEY L. SLAGLE for the degree of MASTER OF SCIENCE

in STATISTICS presented on December 18, 1978

Title: MODIFICATION OF BEALE'S METHOD FOR QUADRATIC PROGRAMMING PROBLEMS

Abstract approved: Redacted for Privacy

Jeffrey L. Arthur

This thesis deals with a modification to an existing mathematical programming algorithm called Beale's method. This method is designed to solve only quadratic or linear programming problems. The modification proposed aims to solve some of the problems this method encounters and to shorten the time required for solution by examining a change in decision rules.

Modification of Beale's Method for Quadratic Programming Problems

by

Rodney L. Slagle

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1979

APPROVED:

Redacted for Privacy

Assistant Professor of Statistics
in charge of major

Redacted for Privacy

Chairman of Department of Statistics

Redacted for Privacy

Dean of Graduate School

Date thesis is presented December 18, 1978

Typed by Clover Redfern for Rodney L. Slagle

ACKNOWLEDGMENTS

The author greatly appreciates the time, energy and concern that was received from many professors during his scholastic career. Of these, the members of the Department of Statistics at Oregon State have shown great care and respect for him as a person and the goals he wished to attain.

The author wishes to express his special appreciation to Dr. Dave Butler for his part in the author's introduction to, and stimulation in, the field of operations research. In the area of statistics, Dr. Dave Thomas contributed much valuable material and interest.

Not much can be said in words about the quality and character of the person to whom the author is most deeply grateful to for his enormous contribution to this thesis. The understanding and patience required to direct the author in such positive ways came from the person of Dr. Jeffrey Arthur. It need not be said that this thesis would not have been conceived if Dr. Arthur's qualities as a professor were not of the highest caliber.

This thesis required unsponsored research funds from the Milne Computer Center of Oregon State, all of which was provided very generously. Many thanks are due to all those people involved.

Outside the scholastic area, the author is very grateful to many people. His friends, though not comprehending his work, gave him much encouragement and support, even if this support was in nontraditional forms. In this respect many thanks to Ed Veovich for his abundant supply of good times, and for often putting things in their proper perspective.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
	Introduction	1
	Quadratic Programming Problems	2
	Quadratic Programming Methods	3
	Existing Methods	3
	Comparison of Existing Methods	6
	Objectives of Thesis	7
II.	BEALE'S METHOD AND PROPOSED MODIFICATION	8
	Beale's Method	8
	Proposed Modification	13
	Illustrative Example Problem	15
	Solution by Beale's Method	16
	Solution by Modified Version of Beale's Method	20
	A Counterexample	24
III.	COMPUTATIONAL STUDY	27
	Methods Used for Study	27
	Problems Used in Study	28
	Initial Results	29
	Additional Study	37
IV.	CONCLUSIONS	40
	BIBLIOGRAPHY	42
	APPENDICES	44
	Appendix I: Problem Generator Code	44
	Appendix II: Example Problem Input	47

LIST OF TABLES

Table		Page
I.	Results of computational study.	30
II.	Average completion summary.	32
III.	Estimated average execution times for modified Beale's method.	34

MODIFICATION OF BEALE'S METHOD FOR QUADRATIC PROGRAMMING PROBLEMS

I. INTRODUCTION

Introduction

Mathematical programming is generally applied to solve problems dealing with the use or allocation of scarce resources in a "best" way. These resources can be material, time, labor, machines, capital, etc., and typically their best use is in such a way as to minimize total cost or maximize net profit. The scarcity of these resources yields a set of constraints relating their combined use to a set of preset limits. When this measure of effectiveness--called the objective function--and the constraints can be written as algebraic inequalities and equations, techniques of mathematical programming can be applied to solve the allocation problem.

The simplest form of mathematical programming problem is one where the objective function and the constraints can be expressed as linear functions of the decision variables. This type of problem is called a linear programming (LP) problem and specific techniques have been developed for solving them, most notably different variations of the LP simplex method [5] and the complementary techniques [6].

The first applications of mathematical programming on nonlinear problems was naturally in the area of quadratic programming [3, 9]. This problem involves the minimization of a convex quadratic objective function of the variables, subject to a set of linear constraints. This definition can be extended to finding the local minimum of any quadratic objective function subject to the linear constraints. However this extension cannot always be applied, because many present methods require a strictly convex function. The method discussed in this paper has the advantage of the more general definition of the problem.

Quadratic Programming Problems

The general quadratic programming (QP) problem can be stated in mathematical form as:

minimize

1) $f(x) = c'x + x'Qx$

subject to

2) $Ax \geq b; \quad x \geq 0$

where c is an $n \times 1$ cost vector, x is an $n \times 1$ solution vector, Q is an $n \times n$ symmetric matrix, A is an $m \times n$ constraint matrix and b is an $m \times 1$ positive vector.

The solution of any quadratic problem must satisfy the well known Kuhn-Tucker conditions for that problem. The Kuhn-Tucker conditions for the above general (QP) problem are as follows:

3) $v - 2Qx + A'u = c$

4) $Ax - y = b$

5) $v'x = 0$

6) $u'y = 0$

7) $x, y, u, v \geq 0 \quad x, v \in \mathbb{R}^n, \quad u, y \in \mathbb{R}^m$

where x and y are primal variables with y an $m \times 1$ primal slack vector and u and v are the dual variables with u an $m \times 1$ dual solution vector and v an $n \times 1$ dual slack vector.

Quadratic Programming Methods

Existing Methods

Since quadratic programming is the natural first step past linear programming in complexity, much attention has been devoted to it and many methods devised for the solution of the QP problem. Of these methods, probably the most widely used and accepted are Dantzig's, Lemke's, Wolfe's and Beale's.

Some methods, such as Lemke's, Wolfe's and Dantzig's are directly based on the solution of the Kuhn-Tucker conditions, whereas

Beale's is based on modifications to the LP simplex method. Only a short review of these methods will be given, but references for a more detailed description of each method are included.

Lemke's method [6] solves the QP problem by forming and solving from the Kuhn-Tucker conditions, a "complementary problem" of the form,

$$w = Mz + q; \quad w'z = 0; \quad w, z \geq 0$$

where

$$w = \begin{bmatrix} v \\ y \end{bmatrix}; \quad M = \begin{bmatrix} 2Q & -A' \\ A & 0 \end{bmatrix}; \quad q = \begin{bmatrix} c' \\ -b \end{bmatrix}$$

and

$$z = \begin{bmatrix} x \\ u \end{bmatrix}.$$

w is set equal to q except that a w_i which equals the minimum q_i is replaced by an artificial variable z_0. The next step is to continue through a sequence of "almost" complementary solutions until the artificial variable is driven from the basis, whence the method terminates with a complementary solution and thus the QP solution. For a more detailed description see [8].

Wolfe's method [13] starts by adding a set of artificial variables to equations 3 and 4 of the Kuhn-Tucker conditions and then uses the Phase-1 simplex method [5] to get a solution to equations 3, 4 and 7.

The objective of this procedure is to minimize the sum of these artificial variables. A modification to the Phase-1 simplex method, the "restricted basis entry" is required so that the entering variable does not violate Kuhn-Tucker conditions 5 and 6.

In Dantzig's method [5] the first initial step is the same as in Wolfe's but the objective is to minimize the true objective function of the quadratic problem. This is done by successive LP simplex type iterations except for a modification to the entering and leaving basic variable criterion. These modifications are either to maintain the solution of the K-T conditions or to restore their feasibility.

Beale's method [3] is quite different from the previous methods in that it does not work on the Kuhn-Tucker conditions, rather it works directly on the objective function and the constraint equations. An initial feasible basis is selected and expressions for the basic variables in terms of the nonbasic variables are substituted into the objective function and constraint equations. The objective function is then represented in a symmetric tableau where the first row contains one-half the partial derivative of the objective function with respect to the corresponding nonbasic variable. Then a variable is selected to enter the basis and is increased in value until either its derivative vanishes or it drives some basic variable to zero. If the latter is encountered then a simple simplex iteration is done on the objective function and the constraint matrix. But if the former occurs then a new nonbasic "free variable" is introduced. A restriction on the entering basic variable states that all free variables having nonzero derivatives must be entered before a regular variable is brought in the basis. The method terminates when no variable can be added to the basis to lower the value of the objective function.

Comparison of Existing Methods

Some comparisons of these methods have been done which point out their differences and relative effectiveness. Van de Panne and Whinston [11] compare the solution paths (i.e., the direction taken toward optimization of the objective function) and effectiveness of Dantzig's method with Beale's method. They point out that solutions of successive iterations in Dantzig's method do appear in Beale's method but Beale's may have extra iterations between these solutions. This would correspond to when, in Beale's method, free variables are introduced and later removed. So they conclude that based on the number of iterations, Dantzig's method has a definite advantage over Beale's method.

A comparison of the relative effectiveness of Dantzig, Wolfe, Beale and a modified version of Wolfe's was done by Braitsch [4]. Braitsch notes that based on average iteration numbers, both Dantzig and Beale are more efficient than both versions of Wolfe. He also concludes that although Dantzig has a slight edge over Beale, this advantage is decreased greatly as the quadratic effect is reduced, i.e., the problem becomes more linear.

In a more recent paper by Ravindran and Lee [9] a comparison was done of Lemke, Wolfe, complex simplex, quadratic differential and a general nonlinear program, SUMT. They draw the conclusion that Lemke's method is superior over the other methods included in their study. The criterion for their conclusion was the average number of iterations and mean execution time. Since their paper dealt with a comparison of more recently developed methods and the classical method of Wolfe, no direct comparison can be made of Lemke and Dantzig or Beale because these methods are also considered more efficient than Wolfe's.

Objectives of Thesis

Some statement should be made here about the direction and objectives of this paper. Basically this paper has two main objectives. The first is to examine Beale's method in detail and develop a modification to the method so as to solve some of the problems this method encounters.

The second objective of the paper is to do a small experimental computational study on the computer to test and compare this modified method with the original Beale's method and some other quadratic programming methods.

II. BEALE'S METHOD AND PROPOSED MODIFICATION

Beale's Method

Beale outlines in [3] two new versions of his original method which first appeared in Beale [1]. Since the first method introduced is the version applied in the computer algorithm used for this study, it will be explained in more detail than already given. The practical version of Beale's method, originally presented briefly in Beale [2] is discussed in detail in [3]. A detailed description of this method here is unnecessary for the purpose of this paper, but basically this version is the same as the original except the objective function is written as a sum of squares.

The second version Beale presents deals with incorporating the product form of the inverse matrix method into Beale's method, and is also outlined in [3]. This revision can be applied to either the original or practical version of Beale's method.

Beale's original method is one application of the ordinary LP simplex method extended to quadratic problems. One advantage of Beale's method is that it reduces to the LP simplex method for purely linear problems, which most QP simplex methods do not.

In general the simplex method deals with the minimization or local minimization of an objective function C, of n variables x_j that are restricted to be nonnegative and must satisfy a set of m

linear independent equations, called constraints, of the form

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad (i = 1, \ldots, m) \quad m \leq n$$

If a feasible solution exists (i.e., a set of x_j, $j = 1, \ldots, n$, such that all the constraint equations are satisfied) then we can find some basic feasible solution in which m of the n variables take on non-negative values and the rest are zero. Using the constraints we can now express these basic variables in terms of the nonbasic ones to give us the equations

$$x_b = \tilde{a}_{b0} + \sum_{k=1}^{n-m} \tilde{a}_{bk} x_{m+k} \qquad (b = 1, \ldots, m)$$

where x_b is the b^{th} basic variable.

This solution is generally expressed in tableau form called the $\tilde{A}$ or constraint matrix. At this solution the basic variables x_b take on the value $\tilde{a}_{b0}$ and all the nonbasic variables are zero.

These expressions for the basic variables in terms of the non-basic ones are now used in the objective function to express it in terms of the nonbasic variables only. With the objective function in this form, the partial derivative of C with respect to any nonbasic variable can be considered by keeping all other nonbasic variables at zero.

In examining this partial derivative, if we find $\partial C/\partial x_{m+j} \geq 0$, then a small increase in x_{m+j} will not decrease the value of C. But if, however, $\partial C/\partial x_{m+j} < 0$, then some increase in x_{m+j} will reduce the value of C.

In the linear programming problem where C is linear and $\partial C/\partial x_{m+j} < 0$, then it will be profitable to increase x_{m+j} to the point where one has to stop before some basic variable is driven negative.

In the case where C is a nonlinear function, when $\partial C/\partial x_{m+j} < 0$, it will be profitable to increase x_{m+j} until either

1) some basic variable is about to be made negative, or

2) $\partial C/\partial x_{m+j}$ vanishes and is about to become positive.

No problem is encountered in the first case, one simply makes a change of basis. If we are examining, say, the partial $\partial C/\partial x_p$, x_p nonbasic, and it is profitable to increase its value from 0 until the basic variable x_q is about to go negative, then the change of basis is made by making x_p basic and x_q nonbasic, using the equation

$$x_q = \tilde{a}_{q0} + \sum_{\ell=1}^{n-m} \tilde{a}_{q\ell} x_{m+\ell}$$

to substitute for x_p in terms of x_q and the other nonbasic variables in the constraint equations and objective function. The geometric interpretation of this transformation is to move (along an extreme edge) from one extreme point of the constraint space to an adjacent extreme point.

When the second case is encountered, one has a more difficult problem if the objective function is generally nonlinear but when (as in our case) it is at most quadratic, the problem becomes very manageable. Because C is quadratic the partial derivative of C, $\partial C/\partial x_p$, is a linear function of the nonbasic variables.

The most useful way to express the objective function (which shall be used for this research) is to write C in a symmetric form

$$C = \sum_{k=0}^{n-m} \sum_{\ell=0}^{n-m} C_{k\ell} x_{m+k} x_{m+\ell}$$

where $x_{m+k} = x_{m+\ell} = 1$ when k or $\ell = 0$.

With the objective function expressed in this form the partial derivative of C is directly observable in the matrix as

$$\frac{1}{2} \frac{\partial C}{\partial x_p} = C_{p0} + \sum_{k=1}^{n-m} C_{pk} x_{m+k} .$$

When, as in the second case pointed out earlier, this $\partial C/\partial x_p$ vanishes before some basic variable is driven to zero then x_p is made basic and a new nonbasic variable u_i is introduced such that

$$u_i = C_{p0} + \sum_{k=1}^{n-m} C_{pk} x_{m+k}$$

where the subscript i designates the i^{th} such variable introduced.

These u variables are not restricted to nonnegative values and are therefore called free variables in contrast to the original x variables (which are so restricted and are now called restricted variables).

Geometrically, what happens in this second case when a u variable is introduced, is that instead of moving from one extreme point of the constraint space to another, one now moves to a new feasible point that is not an extreme point of the original problem. As a u variable is introduced we are generating a new constraint to move us in a direction conjugate to the previous directions with respect to the objective function. The problem here occurs, as Beale points out in [3], that having introduced u_i and forcing $\partial C/\partial x_p = 0$ we would like to have it remain so, and continue in these conjugate directions. But Beale's method (in any form) as it is now, cannot maintain these properties. If, when moving along one of these conjugate directions, another original constraint is encountered, then one has to remove all the free variables introduced thus far before continuing the iterations. This involves making all previous u_i's basic and discarding them as they are now of no use to us. This fact is explained in more detail by Beale in [3]. Even with this problem the process will terminate in a finite number of steps and a proof of this is also given in [3].

Proposed Modification

It is with this problem outlined in the previous section that the main purpose of this paper is concerned. When searching for some nonbasic x variable to enter the basis, different criteria can be used without any real modification to the method of Beale. Some of the criteria that have been used [4] are

1) The first nonbasic x variable encountered with its partial with respect to C, negative

2) The x variable that will decrease the value of C at the fastest initial rate.

Neither of these criteria are aimed at solving the problem described although they do effect the average number of steps to complete the problem.

In this paper the author proposes a new criterion which attempts to solve this problem so that when a free variable is introduced, no other original constraints will be encountered before a solution is found. This new criterion is simply to search the nonbasic x variables that have a negative partial with respect to the objective function until one is found that can be increased in value until it is about to drive some basic variable negative. When this type of iteration is not available then some other criterion is used for the entering basic variable.

This proposed criterion is now presented in detail. From the very first iteration on, the nonbasic variables are searched until one is found such that its partial is negative (i. e., $\partial C/\partial x_p < 0$) and x_p can be increased, say to some value $D_1 > 0$, before the partial vanishes. Now looking to see to what value x_p can be increased before it drives some basic x variable to zero, we find that we can increase x_p to say, $D_2 > 0$. Now if $D_2 \leq D_1$ then x_p is made basic with the value D_2 and some x variable is taken out of the basis and the process starts again. But if $D_1 < D_2$, then x_p is not allowed to enter the basis and the search for an entering variable continues until one is found such that $D_2 < D_1$. If after searching the nonbasic variables, none are found that meet the criteria of $D_2 \leq D_1$, then a variable is entered that has $D_1 < D_2$. At this point we are now creating a free variable and hopefully all subsequent iterations will be of this type until a solution is found, if one exists.

A geometric explanation would be to always move from one extreme point of the constraint space to an adjacent extreme point until this type of movement is no longer available. By doing this first you will now hopefully be at an extreme point that will allow you to move in conjugate directions towards the solution without encountering another original constraint.

Illustrative Example Problem

An example is given here to demonstrate the steps taken by both the original and modified Beale's method. This example is presented by Beale and solved using Beale's method in [3].

The example problem is

minimize

$$C = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to the constraints

$$x_1 + x_2 + 2x_3 \leq 3 \qquad x_1, x_2, x_3 \geq 0.$$

The first step is to introduce the slack variable x_4 and get

$$x_4 = 3 - x_1 - x_2 = 2x_3 .$$

Now the objective function is written as a symmetric matrix as follows:

$$\begin{aligned} C = \; & (-9 - 4x_1 - 3x_2 - 2x_3) \\ & + (-4 + 2x_1 + x_2 + x_3)x_1 \\ & + (-3 + x_1 + 2x_2 \quad)x_2 \\ & + (-2 + x_1 \quad + x_3)x_3 \; . \end{aligned}$$

Solution by Beale's Method

From the symmetric matrix of the objective function given in the previous section, we can see that an increase in any of the variables x_1, x_2 or x_3 will decrease the value of C. But x_1 is chosen because it is the variable that will decrease the value of the objective function at the fastest rate. Increasing x_1 decreases x_4 but x_4 will stay positive until $x_1 = 3$. But now looking at the partial of x_1 we find $\frac{1}{2}\partial C/\partial x_1 = -4 + 4x_1 + x_2 + x_3$, and this becomes zero when $x_1 = 2$. So a free variable u_1 is introduced such that $u_1 = -4 + 2x_1 + x_2 + x_3$ and is now our new nonbasic variable.

Since x_1 is now basic we have

$$x_1 = 2 + \frac{1}{2}u_1 - \frac{1}{2}x_2 - \frac{1}{2}x_3$$

$$x_4 = 1 - \frac{1}{2}u_1 - \frac{1}{2}x_2 - \frac{3}{2}x_3 \; .$$

Solving our objective function in terms of our new nonbasic variables, we now have

$$
\begin{aligned}
C = \ & (1 \qquad\qquad - x_2 \qquad\qquad) \\
& + (\quad \tfrac{1}{2} u_1 \qquad\qquad\qquad)u_1 \\
& + (-1 \qquad + \tfrac{3}{2} x_2 - \tfrac{1}{2} x_3)x_2 \\
& + (\qquad\qquad - \tfrac{1}{2} x_2 + \tfrac{1}{2} x_3)x_3 \ .
\end{aligned}
$$

Now it can be seen that by increasing x_2 we decrease the value of C. We can increase x_2 to 2 before we drive x_4 to zero, but $\frac{1}{2} \partial C / \partial x_2 = -1 + \frac{3}{2} x_2 - \frac{1}{2} x_3$, so we can only increase x_2 to $2/3$ before the partial vanishes and again we are stopped by the partial going to zero. So again we introduce a free variable

$$
u_2 = -1 + \frac{3}{2} x_2 - \frac{1}{2} x_3 \ .
$$

By making u_2 a new nonbasic variable and x_2 our new basic variable we have

$$
\begin{aligned}
x_2 &= \frac{2}{3} \qquad\qquad + \frac{2}{3} u_2 + \frac{1}{2} x_3 \\
x_1 &= \frac{5}{3} + \frac{1}{2} u_1 - \frac{1}{3} u_2 - \frac{2}{3} x_3 \\
x_4 &= \frac{2}{3} - \frac{1}{2} u_1 - \frac{1}{2} u_2 - \frac{5}{3} x_3 \ ,
\end{aligned}
$$

and C becomes

$$
\begin{aligned}
C = \;& (\tfrac{1}{3} && && - \tfrac{1}{3} x_3) \\
& + (&& \tfrac{1}{2} u_1 &&)u_1 \\
& + (&& \quad \tfrac{2}{3} u_2 &&)u_2 \\
& + (-\tfrac{1}{3} && && + \tfrac{1}{3} x_3) x_3 \,.
\end{aligned}
$$

Again we notice that C can be decreased by increasing x_3, but now we have a problem.

This time x_3 can be increased to $2/5$ before x_4 becomes zero but it can be increased to 1 before the partial vanishes. Therefore, performing a simplex pivot yields

$$
\begin{aligned}
x_3 &= \frac{2}{5} - \frac{3}{10} u_1 - \frac{1}{5} u_2 - \frac{3}{5} x_4 \\
x_2 &= \frac{4}{5} - \frac{1}{10} u_1 + \frac{3}{5} u_2 - \frac{1}{5} x_4 \\
x_1 &= \frac{7}{5} + \frac{7}{10} u_1 - \frac{1}{5} u_2 + \frac{2}{5} x_4 \,,
\end{aligned}
$$

and

$$
\begin{aligned}
C = \;& (\frac{3}{25} + \frac{3}{50} u_1 + \frac{1}{25} u_2 + \frac{3}{25} x_4) \\
& + (\frac{3}{50} + \frac{53}{100} u_1 + \frac{1}{50} u_2 + \frac{3}{50} x_4) u_1 \\
& + (\frac{1}{25} + \frac{1}{50} u_1 + \frac{34}{50} u_2 + \frac{1}{25} x_4) u_2 \\
& + (\frac{3}{25} + \frac{3}{50} u_1 + \frac{1}{25} u_2 + \frac{3}{25} x_4) x_4 \,.
\end{aligned}
$$

Now we must remove u_1 and u_2 from the basis. The partial $\partial C/\partial u_1$ becomes zero when $u_1 = -6/53$ and all basic variables remain positive. So by making u_1 basic and introducing u_3 as nonbasic we have

$$u_3 = \frac{3}{50} + \frac{53}{100} u_1 + \frac{1}{50} u_2 + \frac{3}{50} x_4$$

and so,

$$u_1 = -\frac{6}{53} + \frac{100}{53} u_3 - \frac{2}{53} u_2 - \frac{6}{53} x_4$$

$$x_3 = \frac{23}{53} - \frac{30}{53} u_3 - \frac{10}{53} u_2 - \frac{30}{53} x_4$$

$$x_2 = \frac{43}{53} - \frac{10}{53} u_3 + \frac{32}{53} u_2 - \frac{10}{53} x_4$$

$$x_1 = \frac{70}{53} + \frac{70}{53} u_3 - \frac{12}{53} u_2 + \frac{17}{53} x_4 ,$$

$$\begin{aligned} C = & \left(\frac{6}{53} + \frac{2}{53} u_2 + \frac{6}{53} x_4\right) \\ & + \left(\frac{100}{53} u_3\right) u_3 \\ & + \left(\frac{2}{53} + \frac{36}{53} u_2 + \frac{2}{53} x_4\right) u_2 \\ & + \left(\frac{6}{53} + \frac{2}{53} u_2 + \frac{6}{53} x_4\right) x_4 . \end{aligned}$$

Next we must remove u_2 from the set of nonbasic variables. We find u_2 can be decreased to $-1/18$ until the partial vanishes so we now have

$$u_4 = \frac{2}{53} \qquad + \frac{36}{53} \qquad + \frac{2}{53} x_4$$

and so

$$u_2 = -\frac{1}{18} \qquad + \frac{53}{36} u_4 - \frac{1}{18} x_4$$

$$x_3 = \frac{4}{9} - \frac{30}{53} u_3 - \frac{5}{18} u_4 - \frac{5}{9} x_4$$

$$x_2 = \frac{7}{9} - \frac{10}{53} u_3 + \frac{8}{9} u_4 - \frac{2}{9} x_4$$

$$x_1 = \frac{4}{3} + \frac{70}{53} u_3 - \frac{1}{3} u_4 + \frac{1}{3} x_4$$

and

$$\begin{aligned} C = &\left(\frac{1}{9} \qquad \qquad \frac{1}{9} x_4\right) \\ &+ \left(\frac{100}{53} u_3 \qquad \qquad \right) u_3 \\ &+ \left(\qquad \frac{53}{35} u_4 \qquad \right) u_4 \\ &+ \left(\frac{1}{9} \qquad \qquad + \frac{1}{9} x_4\right) x_4 . \end{aligned}$$

We are now at the optimum solution of $C = 1/9$ with $x_1 = 4/3$, $x_2 = 7/9$ and $x_3 = 4/9$. Note that it took five iterations of the method to reach this optimal solution.

Solution by Modified Version of Beale's Method

As in the example solved using the original Beale's method, we start with x_4 the basic variable and x_1, x_2, x_3 as the nonbasic variables. From the symmetric matrix of the objective function we find that C can be decreased by increasing x_1, x_2 or x_3.

By first examining x_1 we find that it can be increased to 3 before x_4 becomes negative. Examining the objective function we find x_1 can only be increased to a value of 2 before the partial derivative vanishes. So, because we are stopped by the derivative before we are stopped by a constraint (i.e., $2 < 3$ or $D_1 < D_2$), we abandon x_1 and investigate x_2.

Here again we find x_2 is stopped by the partial vanishing because we can increase x_2 to 3 before x_4 is negative but the partial vanishes when x_2 is increased to $3/2$. So again we must abandon x_2 and investigate x_3.

Now investigating x_3 we find that we can increase x_3 to a value of $3/2$ before x_4 becomes negative and to a value of 2 before the partial vanishes. So (because $3/2 < 2$ or $D_2 < D_1$) we can now bring in x_3 as our new basic variable and x_4 becomes the new nonbasic variable.

We now have

$$x_3 = \frac{3}{2} - \frac{1}{2}x_1 - \frac{1}{2}x_2 - \frac{1}{2}x_4$$

and our objective function becomes

$$C = \left(\frac{21}{4} - \frac{9}{4}x_1 - \frac{11}{4}x_2 + \frac{1}{4}x_4\right)$$
$$+ \left(-\frac{9}{4} + \frac{5}{4}x_1 + \frac{3}{4}x_2 - \frac{1}{4}x_4\right)x_1$$
$$+ \left(-\frac{11}{4} + \frac{3}{4}x_1 + \frac{9}{4}x_2 + \frac{1}{4}x_4\right)x_2$$
$$+ \left(\frac{1}{4} - \frac{1}{4}x_1 + \frac{1}{4}x_2 + \frac{1}{4}x_4\right)x_4 .$$

Now by reexamining x_1 and x_2 we find both have negative partials but both are stopped by their partials vanishing before driving x_3 to zero. Since there are no other variables with negative partials x_1 will be made basic and a free variable u_1 will be introduced. This gives us

$$x_1 = \frac{9}{5} + \frac{4}{5}u_1 - \frac{3}{5}x_2 + \frac{1}{5}x_4$$
$$x_3 = \frac{3}{5} - \frac{2}{5}u_1 - \frac{1}{5}x_2 - \frac{3}{5}x_4$$

and

$$C = \left(\frac{30}{25} + \frac{8}{5}u_1 - \frac{7}{5}x_2 + \frac{3}{5}x_4\right)$$
$$+ \left(\frac{8}{5} + \frac{4}{5}u_1 - \frac{8}{25}x_2 + \qquad\right)u_1$$
$$+ \left(-\frac{7}{5} - \frac{8}{25}u_1 + \frac{9}{5}x_2 + \frac{2}{5}x_4\right)x_2$$
$$+ \left(\frac{3}{5} \qquad + \frac{2}{5}x_2 + \frac{1}{5}x_4\right)x_4 .$$

Looking at the objective function matrix we see that x_2 is the only nonbasic variable with a negative partial, but again this partial vanishes before a basic variable is driven negative. So x_2 will be made basic and another free variable, u_2 will be introduced.

Doing this we now have

$$x_2 = \frac{7}{9} + \frac{8}{45} u_1 + \frac{5}{9} u_2 - \frac{2}{9} x_4$$

$$x_1 = \frac{4}{3} + \frac{52}{75} u_1 - \frac{1}{3} u_2 + \frac{1}{3} x_4$$

$$x_3 = \frac{4}{9} - \frac{98}{225} u_1 - \frac{1}{9} u_2 - \frac{5}{9} x_4$$

and our objective function matrix is

$$\begin{aligned} C = & \left(\frac{1}{9} + \frac{304}{225} u_1 \qquad\qquad + \frac{41}{45} x_4 \right) \\ & + \left(\frac{304}{225} + \frac{836}{1125} u_1 \qquad\qquad + \frac{16}{225} x_4 \right) u_1 \\ & + \left(\qquad\qquad \frac{5}{9} u_2 \qquad\qquad \right) u_2 \\ & + \left(\frac{41}{45} + \frac{16}{225} u_1 \qquad\qquad + \frac{1}{9} x_4 \right) x_4 \end{aligned}$$

which indicates we are at the optimal solution with $C = 1/9$, $x_1 = 4/3$, $x_2 = 7/9$ and $x_3 = 4/9$, the same values obtained when solved using the original Beale's method. An important feature of this method is that we reached this solution in only three steps of the modified method whereas the original required five.

A Counterexample

Further investigation has proven that the proposed advantages of the new criterion do not always hold. It is possible that while using the new criterion, a free variable is introduced and then another original constraint is encountered. This was the problem the new criterion was proposed to solve. An illustrative example of this follows.

Consider the following problem:[1]

Minimize

$$C = -16x_1 - 16x_2 + x_1^2 + x_2^2$$

subject to

$$4x_1 + 7x_2 \leq 70$$

$$3x_1 + x_2 \leq 27 \qquad x_1, x_2 \geq 0 .$$

The proposed method begins by introducing slack variables x_3 and x_4 as basic variables to get

$$x_3 = 70 - 4x_1 - 7x_2$$

$$x_4 = 27 - 3x_1 - x_2 .$$

The objective function corresponding to this basis is

[1] Problem by Dr. Butler

$$C = (\quad -8x_1 - 8x_2)$$
$$+(-8 + x_1 \qquad) x_1$$
$$+(-8 \qquad + x_2) x_2 .$$

Both x_1 and x_2 can be made basic but both are restricted in value by their partial derivative with respect to C, so a free variable must be introduced. By making x_1 basic and introducing u_1, the constraint equations become

$$x_1 = 8 + u_1$$
$$x_3 = 38 - 4u_1 - 7x_2$$
$$x_4 = 3 - 3u_1 - x_2 .$$

The objective function is now

$$C = (-64 \qquad - 8x_2)$$
$$+(\quad u_1 \qquad) u_1$$
$$+(-8 \qquad + x_2) x_2 .$$

Now x_2 is the only candidate for the entering basic variable. We see that x_2 can be increased to 8 before the partial vanishes but can only be increased to 3 before encountering another constraint. So x_2 must be made basic with a value of 3. Furthermore the next iteration must remove the free variable introduced in the first step. This is the same problem that the original method encountered when it was used to solve the previous example, and is the exact problem that the modification was proposed to solve.

It should be pointed out that this example problem has a convex objective function (i.e., the Q matrix is positive definite), therefore convexity does not guarantee that the modification will work as proposed.

III. COMPUTATIONAL STUDY

Methods Used for Study

As a test of the usefulness and effectiveness of the modification to Beale's method, a comparative study was done that also includes Beale's original method, Lemke's, Wolfe's and a method called the Symmetric Quadratic method (Sym. Q) which is a method developed by Van De Panne and Whinston [12] and is based on Dantzig's method.

Beale's method and the other methods used are available on a computer package by Northwestern University, called the Multi-Purpose Optimization System (MPOS) [7]. The modified Beale's method was tested by taking Beale's method as coded on MPOS and making the appropriate change for selecting the entering basic variable.[2] Since all 5 algorithms used for the comparison are based on the similarly coded package by Northwestern Univ., it seems reasonable to be able to analyze the results by comparing both calculation times and iteration counts.

[2] With permission from Northwestern University for changes to be made for the purpose of this study only.

Problems Used in Study

The problems used for the comparison were randomly generated using a computer code provided by A. Ravindran at Purdue University and is based on a method by Rosen and Suzuki [10]. The code used for this algorithm is provided in Appendix I.

This method of producing problems starts with a randomly generated solution, then proceeds to generate a specified number of constraints and an objective function corresponding to the original solution. By generating problems in this manner the problem always has a bounded feasible region with a specific optimal value, because the technique used for generating the constraints and objective function are based on the Kuhn-Tucker conditions.

The problems used for the comparison were of six different dimensions, all of the form, minimize the objective function, subject to a set of less than or equal to constraints. Ten problems of each dimension were randomly generated so as to give a better estimate of their relative efficiencies. One dimension of problems used was with 5 variables and 5 constraints (5 x 5). Problems of 10 variables were generated with 5, (10 x 5) and 10, (10 x 10) constraints. Finally problems of 15 variables and 5 (15 x 5), 10 (15 x 10), and 15 (15 x 15) constraints were used for a total of 60 problems in all.

One difficulty with generating the problems by the method described is that the quadratic (Q) matrix may not be (and in fact is usually not) a positive definite matrix which can cause some problems with the algorithms, as we shall see later.

The results for each problem of each dimension are given for each method. The results given are the number of iterations required and the time taken for execution of those iterations. Table I contains the actual data from each problem and Table II is a summary of the average results on each method for each problem set.

Initial Results

In evaluating the data given in Table II we find that Sym. Q is almost always the fastest in terms of number of iteration and terms of the time required for those iterations. Generally the second fastest is Beale's original method, in both time and number of iterations. The modified Beale's method is usually third in number of iterations but the time required per iteration is consistantly higher than for the original Beale's method. This, the author feels, is due to the programming of the change and not due to the theoretical method involved. This point is made more apparent when both the original and modified methods follow the exact same solution paths but the modified method requires more time in doing so than does the original. The modification to Beale's was done on the same code in MPOS as the original, the code was just

Table I. Results of computational study.

Problem Number	Method: Modified Beale	Regular Beale	Lemke	Wolfe	Sym. Q
5 variables and 5 constraints#					
1	.047/3	.050/5	.068/9	*	.049/3
2	.170/5	.132/15	.172.23	*	.151/9
3	.127/10	.121.14	.137/19	*	.149/9
4	.096/7	.067/7	.054/17	.136/18	.115/7
5	.099/7	.073/8	.069/9	*	.085/5
6	.124/10	.107/12	.100/13	*	.114/7
7	.168/14	.138/15	.097/13	*	.119/7
8	.137/11	.108/12	.083/11	*	.118/7
9	.118/9	.050/5	.040/5	.083/9	.048/3
10	.121/9	.080/9	.114/15	*	.116/7
$\bar{x}$	.121/8.5	.093/10.2	.093/12.4	.110/13.5	.106/6.4
10 variables and 5 constraints					
1	.245/11	.063/4	.219/19	*	.151/7
2	.353/16	.277/20	.198/16	*	.196/9
3	.481/22	.075/5	.152/13	.203/13	.071/4
4	.231/16	.219.10	.222/18	*	.191/9
5	.239/11	.113/8	.204/16	*	.118/6
6	.512/24	.212/20	.284/25	*	.197/9
7	.331/16	.221/16	.230/19	*	.202/9
8	.228/10	.076/5	.063/5	*	.063/3
9	.304/14	.195/14	.354/29	*	.199/9
10	.464/22	.103/7	.166/12	.230/15	.120/6
$\bar{x}$	.339/16.2	.154/11.5	.209/17.2	.217/14	.151/7.1
10 variables and 10 constraints					
1	.910/35	.210/13	.417/24	*	.213/8
2	.146/5	.105/6	.150/8	.290/13	.122/5
3	.753/29	.416/27	.632/33	*	.458/16
4	.893/35	.658/41	.591/37	*	.557/20
5	.716/28	.401/26	.370/21	*	.437/15
6	.845/33	.395/25	.647/37	*	.403/14
7	.431/16	.228/14	.398/22	*	.243/9
8	.686/27	.451/27	.502/28	*	.428/15
9	1.087/43	.682/43	.675/39	*	.733/25
10	.790/30	.286/18	.470/28	*	.273/10
$\bar{x}$	.725/28.1	.383/24	.485/27.7	.290/13	.387/13.7

#Data given as: CP seconds of execution time/number of iterations.
*Solution not found because Q not positive definite.

Table I. Continued.

Problem Number	Method: Modified Beale	Regular Beale	Lemke	Wolfe	Sym. Q
15 variables and 5 constraints					
1	.963/29	.515/26	.480/29	*	.392/14
2	.575/17	.437/23	.330/19	*	.375/13
3	.888/27	.634/31	.762/44	*	.470/17
4	.700/21	.144/7	.419/24	*	.122/5
5	.320/9	.221/11	.560/30	*	.179/7
6	.722/22	.612/30	.396/23	*	.453/16
7	.958/29	.611/32	.436/25	*	.529/19
8	1.240/37	.658/33	.805/47	*	.424/15
9	.566/17	.157/8	.246/13	.510/21	.164/6
10	.466/14	.323/16	.390/21	.820/34	.267/10
$\bar{x}$	.740/22.2	.431/21.7	.482/27.5	.665/27.5	.338/12.2
15 variables and 10 constraints					
1	1.281/33	.375/16	.598/25	*	.375/10
2	2.004/53	.655/28	1.661/72	*	.480/13
3	1.170/28	.960/42	.560/24	*	.885/24
4	1.361/35	.788/35	1.592/64	*	.753/21
5	1.537/41	.920/41	.535/23	*	.804/22
6	1.525/39	.855/35	1.206/49	*	.809/22
7	2.046/52	1.027/46	.725/31	*	.982/27
8	.751/19	.434/19	.699/30	*	.421/12
9	1.136/29	.548/24	1.011/43	*	.512/14
10	1.783/46	.781/35	1.551/72	*	.727/20
$\bar{x}$	1.460/37.5	.734/32.1	1.013/43.3	--	.675/18.5
15 variables and 15 constraints					
1	2.069/46	1.131/45	.577/17	*	1.220/26
2	2.459/53	1.232/49	3.192/101	*	1.394/29
3	1.750/32	.947/37	1.507/50	*	1.021/22
4	3.192/71	1.574/62	1.659/56	*	1.600/34
5	2.143/46	1.169/46	1.781/60	*	1.257/27
6	3.151/70	1.738/69	2.110/73	*	1.710/36
7	1.729/36	1.318/52	2.532/86	*	1.485/31
8	1.153/24	.700/27	1.345/42	*	.867/18
9	2.216/48	1.276/50	1.169/39	*	1.182/26
10	2.521/55	1.435/55	2.060/72	*	1.286/27
$\bar{x}$	2.238/48.1	1.252/49.2	1.793/59.6	--	1.302/27.6

*Solution not found because Q not positive definite.

Table II. Average completion summary.

Method	Average Execution Time	Average Number of Iterations	Average Execution Time per Iteration
5 variables and 5 constraints			
Modified Beale	.121	8.5	.014
Beale	.093	10.2	.009
Lemke	.093	12.4	.008
Wolfe	.110	13.5	.008
Sym. Q	.106	6.4	.017
10 variables and 5 constraints			
Modified Beale	.338	16.2	.021
Beale	.154	11.5	.013
Lemke	.209	17.2	.012
Wolfe	.217	14	.016
Sym. Q	.151	7.1	.021
10 variables and 10 constraints			
Modified Beale	.725	28.1	.026
Beale	.383	24.0	.016
Lemke	.485	27.7	.018
Wolfe	.290	13	.022
Sym. Q	.387	13.7	.028
15 variables and 5 constraints			
Modified Beale	.740	22.2	.033
Beale	.431	21.7	.020
Lemke	.482	27.5	.018
Wolfe	.665	27.5	.024
Sym. Q	.338	12.2	.028
15 variables and 10 constraints			
Modified Beale	1.460	37.5	.039
Beale	.734	32.1	.023
Lemke	1.013	43.3	.023
Wolfe	*	*	*
Sym. Q	.675	18.5	.036
15 variables and 15 constraints			
Modified Beale	2.238	48.1	.047
Beale	1.252	49.2	.025
Lemke	1.793	59.6	.030
Wolfe	*	*	*
Sym. Q	1.302	27.6	.047

*No problems had positive definite Q.

modified to meet the criteria of the modification and not completely rewritten, and was compiled at different times at possibly different levels. So it does not seem unreasonable to assume that the modified method should have only slightly higher time per iteration than the original method.

A summary of the average results where the calculation time for the modified Beale's method is estimated by multiplying the average number of iteration it requires by the average time per iteration that the original method requires, is given in Table III. Evaluation of this data gives a more consistent result. With some exceptions, it appears that the order of efficiency of the methods would be; Sym. Q, Original Beale's, Modified Beale's, Lemke's and finally Wolfe's. It should be said here that since Wolfe's method required a positive definite Q matrix, which was obtained on only seven problems out of 60 used, no real comparison can be made about its relative efficiency although it is generally accepted to be the least efficient of the five methods used.

Since the modification to Beale's method was for an improvement on the original Beale's, some comparison is needed here. Since the calculation time per iteration should be approximately the same, the methods will be compared strictly on number of iterations. Table III shows us that the modified method has fewer average iterations than the original method does for problems of 5 variables and 5

Table III. Estimated average execution times for modified Beale's method.

Method	Average Execution Time per Iteration	Average Number of Iteration	Estimated Average Execution Time
5 variables and 5 constraints			
Modified Beale	.009	8.5	.077
Beale	.009	10.2	.092
10 variables and 5 constraints			
Modified Beale	.013	16.2	.211
Beale	.013	11.5	.150
10 variables and 10 constraints			
Modified Beale	.016	28.1	.450
Beale	.016	24.0	.384
15 variables and 5 constraints			
Modified Beale	.020	22.2	.444
Beale	.020	21.7	.434
15 variables and 10 constraints			
Modified Beale	.023	37.5	.863
Beale	.023	32.1	.738
15 variables and 15 constraints			
Modified Beale	.025	48.1	1.203
Beale	.025	49.2	1.230

constraints (5 x 5). As we move toward larger problems this advantage is lost. On the problems of dimension 10 x 5 and 10 x 10, the original method appears significantly better, but only slightly so on problems of 15 x 5. For problems of 15 x 10 the original method has the advantage but for problems of 15 x 15 the modified method does better. One explanation for ineffectiveness of the modification could be that the Q matrix was not positive definite. Examining the solution paths of the modified method for problems with a positive definite Q we find the method works as described in Chapter II, but in light of the counterexample, further study of problems with positive definite Q's will be examined. Problems without a positive definite Q do not always work as the proposed modification is described. In these problems regular iterations are made until a free variable must be made a nonbasic variable, but regular iterations again occur before the optimum is reached. It appears that in this case a regular variable may always have positive partial derivatives with respect to Q before a free variable is brought in, and then have a negative partial after, indicating it can be made basic forcing out the free variable.

Another possible explanation for the results is as follows. On some problems the original and modified method followed the same solution path to a point where the original method made nonbasic a free variable and the optimum was reached. At that point the modified method was forced to perform regular iterations until a free variable could be made nonbasic and the optimum reached. A geometric interpretation of this could be that when the present solution is at an extreme point where the optimal solution can be reached without encountering another original constraint, the original method may find the free variable as the best improvement in the objective function. However, the modified method is forced to find the "best" extreme point from which we can reach the optimum without encountering another original constraint, thus taking extra iterations.

One significant problem encountered when doing this computational study was a difference in the values of the variables at the optimum solution. The different methods were generally consistent in their solutions but these were almost always different from those given by the problem generator. The optimum value of the objective function, though, is agreed upon by all methods and the problem generator with differences of .001 to .5, but for problems of 15 variables and 20 to 60 iterations, this does not seem unreasonable.

There are two possible causes for this difference in the solution values. One explanation is round-off error at each iteration that causes the methods to continue past the optimum point with iterations that cause only minute changes in the objective function value, forcing variables that should be basic according to the problem generator to be made nonbasic. This is recognizable in many results based on Beale's method. In both the original and modified methods, 1/2 to 3/4 of the iterations seem to be of this type. The other methods do not allow an easy calculation of their solution paths so as to see when the optimum solution given by the problem generator is reached. Since the methods based on Beale do not stop at this "optimum" solution and because only the objective function value is given at each iteration (not the value of the variables) it cannot be definitely stated when or if this optimum solution is ever reached.

Another possibility for this difference is that there could exist multiple optima. A problem with multiple optima is one that has more than one set of solutions for the variables with all giving the same optimal value of the objective function. The method used by the problem generator could allow this to happen.

Still despite this problem, since the solutions are almost always the same among the methods and the objective functions value agrees with the problem generator, the comparison and analysis of the methods would be valid.

Additional Study

The results of the initial computational study indicate that the proposed modification is not, in general, an improvement. The counterexample shows some of the problems the modified method can encounter even with a positive definite Q matrix. However, the initial study contained only seven problems with a positive definite Q, all of which were solved as proposed by the modified method. These results indicated the need for additional study to investigate how well the modified method works on problems with a positive definite Q.

The problems used in this additional study were of 2 variables, 2 constraints and 5 variables, 5 constraints. The same problem generator was used but in order to increase the possibility of generating a positive definite Q, the Q was formed to have on the average, 50% positive values on the diagonal, 25% positive values off the diagonal and the remaining values zero. This compares with the 100% positive Q's used in the initial study.

Fifty problems of each dimension were generated for this part of the study. Of the 50 problems generated for the 2 by 2's, 45 had positive definite Q's. All 45 of these problems were solved by the original and modified methods using the same solution path. In other words, the two methods solved the problems identically except for the time required. The modified method encountered the trouble illustrated by the counterexample on 6 of the 45 problems.

The fifty 5 by 5 problems generated produced only 11 with a positive definite Q. Of these 11, five encountered the problem shown in the counterexample. Only one of the 11 was solved differently by the two methods. In this particular problem the original method required 13 iterations to find the solution where as the modified method required only 10 iterations for completion.

IV. CONCLUSIONS

The main purpose of this thesis was to propose a modification to the entering variable criterion in order to get an improvement in the number of iterations required to solve problems using Beale's method. It was proposed that the criterion of always choosing a regular iteration over a free variable iteration when possible would force the solution path to never hit a constraint after a free variable was introduced. It has been shown by the counterexample and the computational study that this proposed advantage of the new criterion does not always hold. It was shown by the initial study that it does not always hold for problems without a positive definite Q. The counterexample and the further study done, illustrate that the requirement of a positive definite Q is not always enough for the criterion to work as proposed.

It also appears that the modification may not always improve the number of iterations and can in fact increase the number even when the Q matrix is positive definite. This seems to occur when the present solution is an extreme point where the optimal solution can be reached without encountering another original constraint. The modification forces the solution path to the "best" extreme point from which we can reach the optimum without hitting another original constraint. The regular method could possibly move directly from the present solution toward the optimum using free variables.

The results of the computational studies seem to indicate that the modification is not a significant improvement on the original method. The average number of iterations and the increased time per iteration indicate that the modified method will usually require more time to solve problems than the original method. A possibility exists for further study on larger problems with the number of variables much larger than the number of constraints. This is because the regular method would be greatly affected by the partials and continually use free variables, then have to remove them when a constraint is hit; whereas the modified method might continually use the constraints until it was close to the optimum.

In conclusion, this study proposed a modification to Beale's method which attempted to avoid one of the problems encountered during the solution process. The results indicate that the modification does not always avoid this problem, even when the objective function is convex. It is felt, however that the modified method may be advantageous under certain circumstances. Further investigation is necessary to verify this.

BIBLIOGRAPHY

1. Beale, E.M.L., "On Minimizing a Convex Function Subject to Linear Inequalities," J. Roy. Statist. Soc. (B), Vol. 17, pp. 173-184, 1955.

2. Beale, E.M.L., "On Quadratic Programming," Naval Res. Logistics Quarterly, Vol. 6, pp. 227-243, 1959.

3. Beale, E.M.L., "Numerical Methods," in Nonlinear Programming, J. Abadie, Ed., North-Holland, Amsterdam, pp. 182-205, 1967.

4. Braitsch, R.J. Jr., "A Computer Comparison of Four Quadratic Programming Algorithms," Management Science, Vol. 18, No. 11, pp. 632-643, 1972.

5. Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, 1963.

6. Lemke, C.E., "Bi-Matrix Equilibrium Points and Mathematical Programming," Management Science, Vol. 45, pp. 309-317, 1963.

7. MPOS: The Multi-Purpose Optimization System, Version 3, Volgelback Computing Center, Northwestern University, Evanston, Ill., 1976.

8. Phillips, D.T., Ravindran, A., and Solberg, J.J., Operations Research: Principles and Practice, John Wiley, New York, 1976.

9. Ravindran, A., and Lee, H.K., "Computer Experiments on Quadratic Programming Algorithms," working paper, 1978.

10. Rosen, J.B., and Suzuki, S., "Construction of Nonlinear Programming Test Problems," Communications of the ACM, Vol. 8, No. 2, pp. 113, 1965.

11. Van de Panne, C., and Whinston, A., "A Comparison of Two Methods for Quadratic Programming," Operations Research, Vol. 14, No. 3, pp. 422-441, 1966.

12. Van de Panne, C., and Whinston, A., "The Symmetric Formulation of the Simplex Method for Quadratic Programming," Econometrics, Vol. 37, 1969.

13. Wolfe, P., "The Simplex Method for Quadratic Programming," Econometrica, Vol. 27, pp. 382-398, 1959.

APPENDICES

Appendix I: Problem Generator Code

The computer code for the problem generator was provided by A. Ravindran of Purdue University. Only some slight modifications were made to make the output compatible with the MPOS system.

The code is as follows:

```
      PROGRAM RANDOM (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)
      DIMENSION X(75),U(75),A(75,75),Q(75,75),B(75),C(75),D(75)
      NI=5
      NO=6
      NP=8
      EPS=10.0E-06
      READ (NI,201) NVAR,NCON,NPROB,XZERO,UZERO,ANEG,AZERO,QNEG,QZERO
      READ (NI,200) ISEED
      QSUM=QNEG+QZERO
      ASUM=ANEG+AZERO
      TEMP=RANF(ISEED)
      IP=1
    5 DO 10 I=1,NVAR
      X(I)=0.0
      IF (RANF(0) .LE. XZERO) GO TO 10
      X(I)=10.0*RANF(0)
   10 CONTINUE
      DO 20 I=1,NCON
      U(I)=0.0
      IF (RANF(0) .LE. UZERO) GO TO 20
      U(I)=10.0*RANF(0)
   20 CONTINUE
C     GENERATE THE CONSTRAINT MATRIX AND THE RIGHT HAND SIDE VECTOR.
      DO 40 I=1,NCON
      K=0
      DO 30 J=1,NVAR
      AY=RANF(0)
      IF (AY .LT. ANEG) GO TO 22
      IF (AY .LT. ASUM) GO TO 24
      A(I,J)=10.0*RANF(0)
      GO TO 30
   22 A(I,J)= -10.0*RANF(0)
      GO TO 30
   24 A(I,J)=0.0
   30 CONTINUE
   31 TEMP=0.0
      DO 35 J=1,NVAR
      TEMP=A(I,J)*X(J)+TEMP
```

```
   35 CONTINUE
      IF (TEMP .LE. 0.0) GO TO 38
   36 K=K+1
      IF (A(I,K) .LT. EPS) GO TO 36
      A(I,K) = -A(I,K)
      GO TO 31
   38 B(I)=TEMP
      IF (U(I) .GT. EPS) GO TO 40
      B(I) =B(I)-10.0*RANF(0)
   40 CONTINUE
C     GENERATE THE QUADRATIC FORM MATRIX.
      DO 50 I=1,NVAR
      DY=RANF(0)
      IF (DY .LT. QNEG) GO TO 42
      IF (DY .LT. QSUM) GO TO 44
      D(I)=RANF(0)
      GO TO 50
   42 D(I)= -RANF(0)
      GO TO 50
   44 D(I)=0.0
   50 CONTINUE
      DO 70 I=1,NVAR
      DO 60 J=1,NVAR
      Q(I,J)=D(I)*D(J)*(50.0)
   60 CONTINUE
   70 CONTINUE
      DO 90 I=1,NVAR
      TEMP=0.0
      DO 80 J=1,NVAR
      TEMP=TEMP -2.0*Q(I,J)*X(J)
   80 CONTINUE
      DO 85 J=1,NCON
      C(I)=TEMP+U(J)*A(J,I)
      IF (X(I) .GT. EPS) GO TO 85
      C(I)=C(I)+10.0*RANF(0)
   85 CONTINUE
   90 CONTINUE
      DO 94 I=1,NCON
      B(I)=-B(I)
      DO 93 J=1,NVAR
      A(I,J)=-A(I,J)
   93 CONTINUE
   94 CONTINUE
      WRITE (NP,212)
      WRITE (NO,209) IP
      WRITE (NO,203)
      DO 100 J=1,NVAR
      WRITE (NO,2025) (Q(I,J),I=1,NVAR)
      WRITE (NP,202) (Q(I,J),I=1,NVAR)
```

```
 100 CONTINUE
     WRITE (NO,204)
     WRITE (NP,202) (C(I),I=1,NVAR)
     DO 110 I=1,NCON
     WRITE (NO,2025) (A(I,J),J=1,NVAR)
     WRITE (NP,202) (A(I,J),J=1,NVAR)
 110 CONTINUE
     WRITE (NO,205)
     WRITE (NO,2025) (C(I),I=1,NVAR)
     WRITE (NO,206)
     WRITE (NO,2025) (B(I),I=1,NCON)
     WRITE (NP,202) (B(I),I=1,NCON)
     WRITE (NP,213)
     WRITE (NO,207)
     WRITE (NO,2025) (X(I),I=1,NVAR)
     WRITE (NO,208)
     WRITE (NO,2025) (U(I),I=1,NCON)
     TEMP=0.0
     DO 130 I=1,NVAR
     TEMP =TEMP+C(I)*X(I)
     DO 120 J=1,NVAR
     TEMP =TEMP+Q(I,J)*X(I)*X(J)
 120 CONTINUE
 130 CONTINUE
     WRITE (NO,211) TEMP
     WRITE (NO,210) IP
     IP=IP+1
     IF (IP .LE. 5) GO TO 5
     WRITE (NP,214)
     STOP
 200 FORMAT (10X,O20)
 201 FORMAT (3I5,6F5.3,O20)
 202 FORMAT (8F10.4)
2025 FORMAT (1H0,8(F10.4,2X))
 203 FORMAT (26H0THE QUADRATIC FORM MATRIX       )
 204 FORMAT (22H0THE CONSTRAINT MATRIX      )
 205 FORMAT (16H0THE COST VECTOR     )
 206 FORMAT (27H0THE RIGHT HAND SIDE VECTOR     )
 207 FORMAT (27H0THE PRIMAL SOLUTION VECTOR      )
 208 FORMAT (25H0THE DUAL SOLUTION VECTOR      )
 209 FORMAT (25H1DATA FOR PROBLEM NUMBER ,I3)
 210 FORMAT (32H0END OF DATA FOR PROBLEM NUMBER ,I3)
 211 FORMAT (41H0THE OPTIMAL OBJECTIVE FUNCTION VALUE IS   ,F10.6  )
 212 FORMAT(*BEALE*/*TITLE*/*TEST PROBLEMS*/*VARIABLES*/
    1*X1 TO X5 */*MATRIX*/*MINIMIZE*/*CONSTRAINTS  5*/
    1*++++++*/*FORMAT*/*(5F10.4)*/*READ*)
 213 FORMAT (*OPTIMIZE*)
 214 FORMAT (*STOP*)
     END
EOI ENCOUNTERED.
```

Appendix II: Example Problem Input

A feature of the MPOS package that was used was the alternate input format. The problems were input to MPOS in the matrix format that they were generated in by the problem generator. An example input is given as follows:

```
BEALE
TITLE
TEST PROBLEMS
VARIABLES
X1 TO X5
MATRIX
MINIMIZE
CONSTRAINTS  5
+++++
FORMAT
(5F10.4)
READ
   11.4758    0.0000    5.0903    0.0000    7.7717
    0.0000    0.0000    0.0000    0.0000    0.0000
    5.0903    0.0000    2.2579    0.0000    3.4473
    0.0000    0.0000    0.0000    0.0000    0.0000
    7.7717    0.0000    3.4473    0.0000    5.2632
 -413.4877  -22.1462 -219.6347   79.8471 -226.5278
    9.6663    9.5625    2.6964    3.0187   -2.2771
    7.7783    4.6520    4.8758    -.5346   -7.4025
    1.8853    7.9461    0.0000    3.5325   -4.8380
    1.3073    0.0000    7.9717    0.0000   -2.9753
    4.4793    2.7158    6.4290   -9.7917   -3.5269
  109.9115   29.8151   16.3481   33.6562   11.8874
QCHECK STOP
OPTIMIZE
?
```