1965

Linear and quadratic programming with more than one objective function

William John Lodholz

LINEAR AND QUADRATIC PROGRAMMING WITH MORE THAN ONE OBJECTIVE FUNCTION

BY

WILLIAM JOHN LODHOLZ , 1927

A

THESIS

submitted to the faculty of the

UNIVERSITY OF MISSOURI AT ROLLA

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

Rolla, Missouri

1965

Approved by

Charles E. Antle (advisor) Billy E Gillett

ABSTRACT

A computational procedure is presented for determining optimal solutions to the linear and quadratic programming problem when there is more than one objective function subject to linear constraints. In general a unique solution does not exist and a set of "best" or "efficient" points is determined and presented in graphical or tabular form. To solve the mathematical programming problems the simplex method is used for linear objective functions and Wolfe's method is used for quadratic objective functions.

ACKNOWLEDGEMENT

The author would like to express his sincere appreciation to Dr. Charles E. Antle, Associate Professor of Mathematics, for his help in selecting the topic of this thesis and for his ever present guidance and encouragement.

In addition, the author would like to thank the University of Missouri at Rolla and Professor Ralph E. Lee, Director of the Computer Center, for the facilities and computing time made available for this study.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

FIGURE PAGE

CHAPTER I

INTRODUCTION

When attempting to optimize a process or system there is frequently more than a single property that should be maximized or minimized. If the various properties can readily be expressed in the same units, then the problem can be reduced to that of optimizing one composite property. However, it is often impractical or undesirable to express the various properties in common units, even though all are functions of the same independent conditions or quantities. For example: A manufacturer of electrical wire would like the wire to have maximum strength, minimum weight and maximum conductivity. All of these properties are functions of the raw materials used and the manufacturing conditions. Possibly all three properties can be expressed in terms of dollars, but it would be convenient to measure each in terms of its own units. A chemist would like a process to have maximum yield and maximum purity, both of which may be functions of the operating temperature, pressure, and other controllable variables.

To further complicate the problem there are usually practical limitations on some or all of the various factors which influence these properties. The chemist is not allowed to operate at unlimited temperatures or negative pressures.

It is unlikely that each of these properties will be optimum for the same set of operating conditions and a compromise must be selected. Some compromises will be better than others and the purpose of this thesis is to provide a computational procedure for selecting a group or set of optimum or best compromises. A "manufacturer" could then select one of these compromises and be assured that no other compromise would improve all of the properties under consideration.

Techniques developed in the area of mathematical programming will be used to examine cases where the restrictions may be expressed as linear relationships and the properties to be optimized are expressed as either linear or quadratic functions of the various independent variables.

CHAPTER II

THE GENERAL PROBLEM

The general mathematical programming problem can be stated as one of determining a set of values for n independent variables that will optimize a single given function of these variables while satisfying certain restrictions. This thesis is concerned with a similar problem involving r functions to be optimized. The functions to be optimized are referred to as the objective functions and correspond to the properties which are to be maximized or minimized. The "physical" limitations or restrictions are called constraints and we shall consider only those problems where the constraints can be expressed as linear equations or linear inequalities.

If the objective function is also linear the problem is a linear programming problem. Likewise a quadratic objective function with linear constraints represents a quadratic programming problem. The quadratic programming problem is a special case of the more general problem of non-linear mathematical programming. Only linear and quadratic objective functions will be considered in this thesis and the term "objective function" will imply one of these two types unless stated otherwise.

In a geometric sense the objective function can be considered as a surface in n+1 space where the n dimensions correspond to the n independent variables. In n space, contours for various values of the objective function can be represented as lines or curves. The linear constraints are represented as closed half-spaces in n dimensions and their intersection will form a closed convex set called the set of feasible points. The optimum solution is that point in the feasible set which optimizes the objective function.

Let x represent a vector with n components x_1, $x_2 \cdots x_n$ and z(x) be a linear or quadratic function of the n components of x. The problem of minimizing z(x) can be solved by maximizing the function -z(x). Thus, if the maximum value of -z(x) is z^0 and occurs when $x = x^0$ then the minimum value of z(x) is $-z^0$ and also occurs when $x = x^0$. Consequently, in the remainder of this thesis problems will be formulated in terms of maximization only without loss of generality.

In order to use the simplex method for maximizing the objective functions it will be necessary to impose nonnegative restrictions on the components of x. Most practical problems will fit this situation and problems where some of the variables are actually unrestricted in sign may be easily reduced to this case. For an example of the treatment of unrestricted variables see page 168 of Hadley[1].

Definition: A point x is a feasible point if it satisfies the constraint equations (inequalities) and the nonnegative restrictions.

In the absence of constraints all nonnegative points would be feasible points.

If there are m constraints in addition to the nonnegative restrictions the programming problem may be stated:

Maximize: z(x)

$$\text{Subject to:} \quad \sum_{i=1}^{n} a_{ij}x_i\{\leq \; = \; \geq\}b_j, \; j = 1,2\cdots m$$

$$x_i \geq 0, \; i = 1,2\cdots n \; ,$$

where the a's and b's are known constants, and only one of the symbols $\{\leq \; = \; \geq\}$ applies for each value of j.

The theory and techniques of the simplex method for linear programming are well developed [1] and will be used when z(x) is a linear function. The method developed by Wolfe[2], [3] can be applied to certain forms of quadratic objective functions. A survey of other methods available to solve non-linear programming problems is presented by Wolfe[3].

In either the linear or quadratic case a solution is usually a unique point where the objective function is optimum. When there is more than one objective function the concept of what constitutes a solution must be changed. It is unlikely, but possible, that a single point will simultaneously optimize all of the objective functions under consideration. The solution will usually consist of a set of points which are called "efficient points."

Definition: A point x^* is an efficient point for the r objective functions $z_1(x)$, $z_2(x) \cdots$ $z_r(x)$ (to be maximized) if there exist no feasible x such that:

$$z_i(x) \geq z_i(x^*) \text{ for all } i$$

$$\text{and } z_k(x) > z_k(x^*) \text{ for at least one } k.$$

The r objective function problem is considered to be solved when all efficient points for the given objective functions have been determined [4]. When attempting to solve a practical r objective function problem it is advisable to consider the reason the problem is to be solved and how the results will be used. In many cases it may not be necessary to determine all of the efficient points in order to obtain the desired information and/or results.

The manner in which the results are to be presented or displayed is another important practical consideration. When there are only two objective functions $z_1(x)$ and $z_2(x)$ they may be evaluated at the efficient points and the results displayed as a graph of $z_1(x)$ versus $z_2(x)$. For an example see Figure 1 on page 14. A person may then choose the point at which he wishes to operate. This thesis will be mainly concerned with the two objective function problem. In most cases the techniques can be easily extended to r objective functions with the major difficulty being that of presenting and interpreting the results. When there are more than two objective functions it seems that the results can best be presented in the form of a table. As the number of objective functions increases it becomes increasingly difficult to choose one efficient point over another. Still, it is useful to know the efficient points as there are no other points where the values of the objective functions are better.

The methods which will be used in the remainder of this thesis to determine efficient points for r linear or quadratic objective functions are based on the following theorems.

Theorem 1: If a convex combination of r objective functions, $\sum_{i=1}^{r} \alpha_i z_i(x)$ where $\alpha_i > 0$ for all i and $\sum_{i=1}^{r} \alpha_i = 1$, is a maximum at $x = x^*$, then x^* is an efficient point for the r objective functions $z_1(x)$, $z_2(x) \cdots z_r(x)$.

Proof: a.) let $z(x) = \sum_{i=1}^{r} \alpha_i z_i(x)$, $\alpha_i > 0$, $\sum_{i=1}^{r} \alpha_i = 1$

b.) given any x, $z(x^*) \geq z(x)$ since z(x) is a maximum at x^*.

c.) assume x^* is not an efficient point, then there exists at least one $x = x^1$ such that

$z_i(x^1) \geq z_i(x^*)$ for all i and

$z_k(x^1) > z_k(x^*)$ for at least one k.

d.) since $\alpha_i > 0$ for all i

$\alpha_i z_i(x^1) \geq \alpha_i z_i(x^*)$ for all i and

$\alpha_k z_k(x^1) > \alpha_k z_k(x^*)$ for at least one k.

e.) summing respective sides of the inequalities,

$$\sum_{i=1}^{r} \alpha_i z_i(x^1) > \sum_{i=1}^{r} \alpha_i z_i(x^*) \text{ or}$$

$$z(x^1) > z(x^*)$$

f.) this contradicts the fact that $z(x^*)$ is a maximum. Therefore, x^* is an efficient point for the functions $z_1(x), z_2(x) \cdots z_r(x)$.

Theorem 1 allows us to use techniques for maximizing a single objective function and thereby determine an efficient point for the r objective function problem. The following theorem will assure us that for a certain class of objective functions all of the efficient points may be found by applying Theorem 1.

Theorem 2: If x^* is an efficient point for r concave objective functions $z_1(x), z_2(x) \cdots z_r(x)$ then there exists some set of r constants α_1, $\alpha_2 \cdots \alpha_r$, with the properties $\alpha_i \geq 0$ and

$$\sum_{i=1}^{r} \alpha_i = 1, \text{ such that the function}$$

$$z(x) = \sum_{i=1}^{r} \alpha_i z_i(x) \text{ is a maximum at } x = x^*.$$

For the proof of this theorem see Lemma 7.4.1, page 217, of Karlin[4].

In the following two chapters the ideas presented in these two theorems will be used to solve programming problems with r objective functions subject to linear constraints.

CHAPTER III

LINEAR OBJECTIVE FUNCTIONS

A. GENERAL DISCUSSION

Many practical problems may be formulated by expressing or approximating functions to be optimized as linear functions of independent variables. If there are n independent variables the r objective functions each have the form $z_k(x) = c_k + \sum_{j=1}^{n} c_{kj}x_j$, where the c's are known constants. Applying Theorem 1, an efficient point may be determined by maximizing $Z(x) = \sum_{i=1}^{r} \sum_{j=1}^{n} \alpha_i c_{ij} x_j$. This is also a linear function and the simplex method or other techniques may be used to determine the point at which the maximum occurs. The constant term does not appear since it affects only the value of the objective function and not where the maximum occurs. A suggested procedure when there are r linear objective functions is to first choose a set of alphas with $\sum_{i=1}^{r} \alpha_i = 1$ and $\alpha_i \geq 0$. The function $Z(x) = \sum_{i=1}^{r} \alpha_i z_i(x)$ is then maximized subject to the constraints. Another set of alphas may yield a different efficient point. It should be noted that by choosing the alphas such that $\sum_{i \neq k}^{r} \alpha_i = 0$ and $\alpha_k = 1$ the kth objective function will be optimized. A systematic method for incrementing the alphas should be established. It is suggested that first large increments of alpha be used to obtain an overall picture of the efficient points. Areas of particular interest may then be studied in detail by sub-dividing the original increments. As previously stated, for more than two objective functions the interpretation of results becomes difficult.

A Fortran II computer program was written to determine efficient points for r linear objective functions with n independent variables. The simplex method is used to maximize linear combinations of the r objective functions. The program inputs are the r objective functions and the constraints in the form of the initial tableau required by the simplex method. Values of alpha are specified and the corresponding efficient points and values of the objective functions are computed. If uniform increments of alpha are specified the program computes efficient points for all of the implied alphas. The program is presented in detail in Appendix 1.

B. TWO OBJECTIVE FUNCTIONS.

When there are only two objective functions the techniques of parametric linear programming may be used to advantage. The function of Theorem 1 which is to be maximized now has the form, $Z(x) = \alpha_1 z_1(x) + \alpha_2 z_2(x)$ where $\alpha_1 + \alpha_2 = 1$ and $\alpha_1, \alpha_2 \geq 0$. By letting $\alpha_1 = \alpha$, then $\alpha_2 = 1 - \alpha$ and $Z(x)$ may be written in the form

$$\begin{aligned} Z(x) &= \alpha z_1(x) + (1 - \alpha) z_2(x) \\ &= z_2(x) + \alpha[z_1(x) - z_2(x)]. \end{aligned}$$

Let $z_1(x) = C_1 x$ and $z_2(x) = C_2 x$ where x is an n component column vector and C_1 and C_2 are n component row vectors. Now $Z(x)$ may be written

$$\begin{aligned} Z(x) &= C_2x + \alpha[C_1 - C_2]x \\ &= C_2x + \alpha Fx \end{aligned}$$

where $F = C_1 - C_2$. The function $Z(x)$ is now in standard form for the application of parametric linear programming with α as the parameter. First the maximum value of $Z(x)$ is found for $\alpha = 0$, which maximizes $z_2(x)$ only. The range of alpha for which this solution is optimum is then determined. The maximum alpha for which the above solution is valid is called a critical alpha. An optimum solution is then computed for the next value of alpha greater than the critical alpha. The process is repeated until an alpha equal to or greater than one is reached. When $\alpha = 1$ the single objective function $z_1(x)$ is optimized.

The graph of $z_1(x)$ versus $z_2(x)$ will be composed of connected straight line segments. The corners represent a range of alpha between the critical alphas represented by the line segments. Values of the n independent variables are known at the critical alphas and linear interpolation will give the values at any particular point between critical alphas. The problem is now solved as all efficient points have been determined. An example will serve to further illustrate the above ideas. If so instructed the computer program previously mentioned will also determine critical alphas and the associated information when there are only two objective functions.

C. A TWO OBJECTIVE FUNCTION EXAMPLE.

A two dimensional example is presented so the problem may be represented graphically. The process is identical for an n dimensional problem. The problem is:

Maximize: $z_1(x) = 30 - 2x_1 + x_2$

$z_2(x) = 2x_1 + x_2$

Subject to the constraints:

(a) $3x_1 - x_2 \geq 0$

(b) $-x_1 + x_2 \leq 4$

(c) $x_1 + 5x_2 \leq 50$

(d) $x_1 + x_2 \leq 18$

(e) $x_1 - x_2 \leq 10$

This example may be solved graphically as shown in Figure 1. The boundaries of the constraints are shown, enclosing the set of all feasible points. The efficient points for $z_1(x)$ and $z_2(x)$ are any points on the heavy shaded lines connecting the points labeled A, B, C, D. The line segments labeled with values of alpha represent the maximum of the Z(x) corresponding to that value of alpha. The values of the objective functions are not shown but may be found by direct evaluation. Figure 1 illustrates that at the corners of the constraint set there is a range of alpha for which the same point is optimum. For example, point C is optimum for all alphas in the interval $\frac{1}{4} \leq \alpha \leq \frac{9}{20}$. Along the line segments joining two corners a single critical alpha

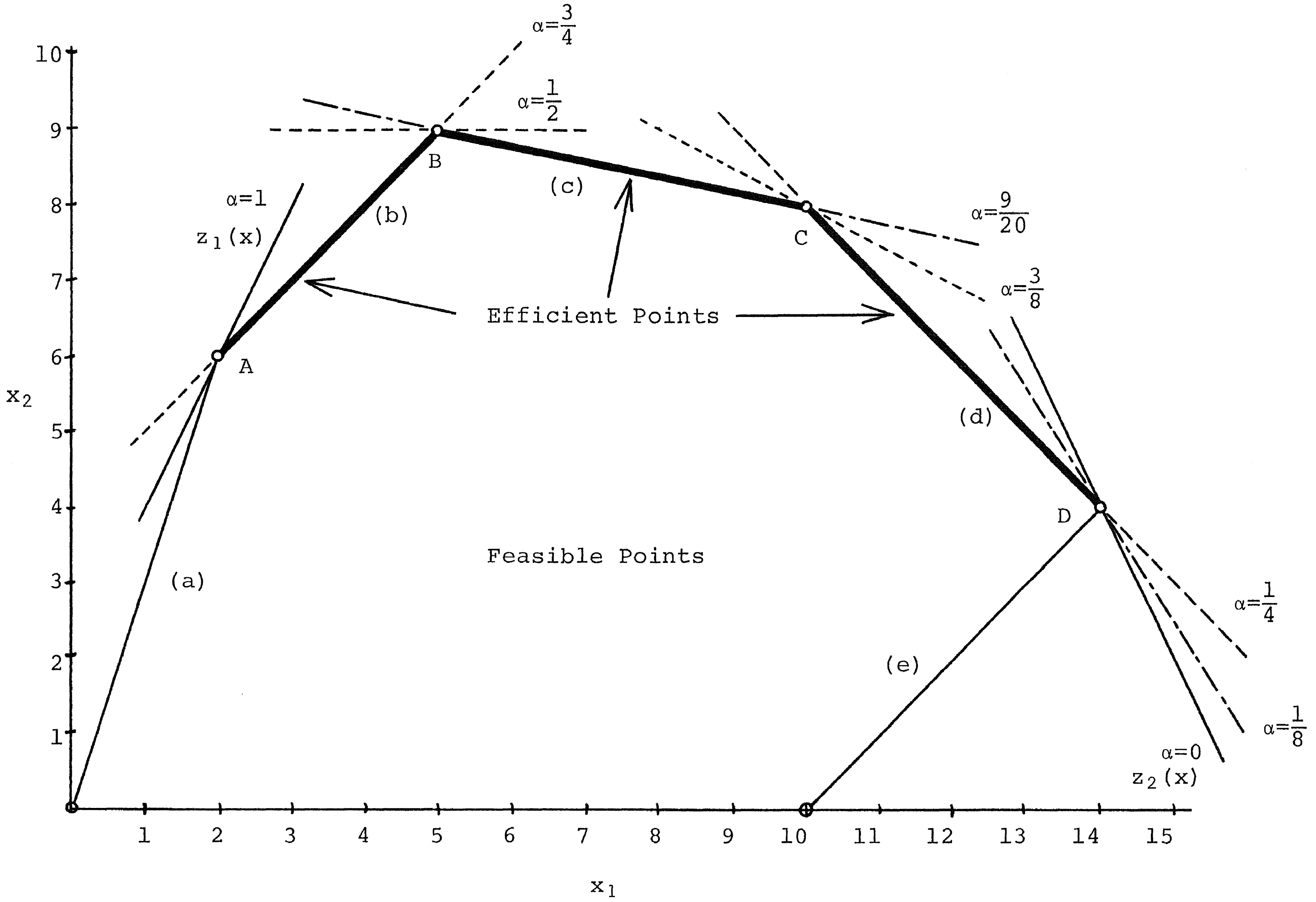


FIGURE I. Feasible points, efficient points and objective functions for Example 1.

corresponds to all of the efficient points.

Figure 2 is a graph of $z_1(x)$ versus $z_2(x)$ with alpha as a parameter. From this graph a desired operating point would be selected. Assume point P is selected as the desired operating point. Point P is located between the points A and B and the following properties are known:

Point	x_1	x_2	$z_1(x)$	$z_2(x)$
A	2	6	32	2
B	5	9	29	11
P	?	?	31	5

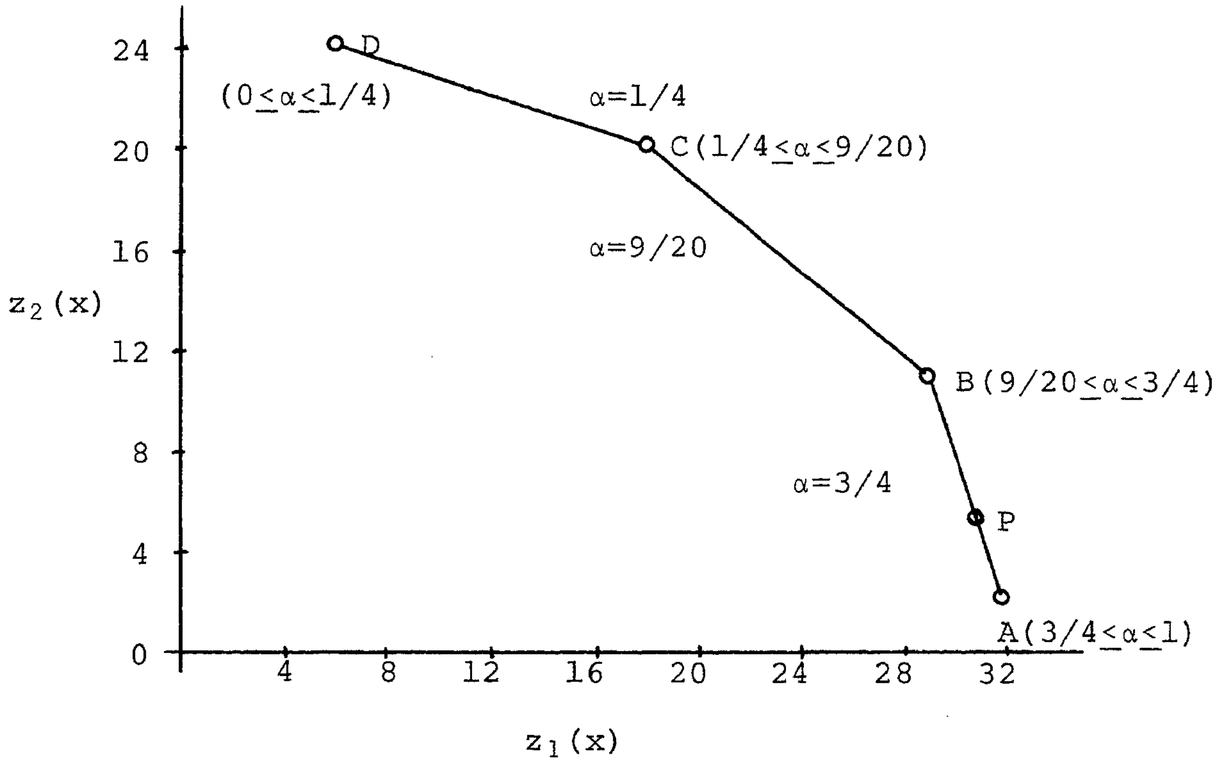


FIGURE 2. Values of Objective Functions at Efficient Points.

The values of x_1 and x_2 at the desired point P may be determined by using the following linear interpolation equation. At the point P

$$x_k = x_k(@A) + \frac{z_1(P) - z_2(A)}{z_1(B) - z_2(A)}[x_k(@B) - x_k(@A)$$

If so desired, the values of $z_2(x)$ could be used instead of the $z_1(x)$ values. For our example the coordinates at P are:

$$x_1 = 2 + \frac{31 - 32}{29 - 32}(5 - 2) = 3 \text{ , } x_2 = 6 + \frac{31 - 32}{29 - 32}(9 - 6) = 7$$

If the values of $z_2(x)$ are used the same coordinates are found as follows:

$$x_1 = 2 + \frac{5 - 2}{11 - 2}(5 - 2) = 3 \text{ , } x_2 = 6 + \frac{5 - 2}{11 - 2}(9 - 6) = 7$$

With the exception of Figure 1 an n dimensional problem would be solved in the same manner. This example as solved using the computer program is presented in Appendix 1.

CHAPTER IV

QUADRATIC OBJECTIVE FUNCTIONS

A. GENERAL DISCUSSION

Quadratic functions of independent variables often occur when developing a mathematical model to describe or approximate an actual process or operation. The simultaneous optimization of r quadratic functions subject to linear constraints will now be considered. Quadratic functions may be written in the form

$$(1) \quad z_k(x) = c_k + x'Q_kx + P_kx,$$

where x is an n component column vector, x' is the transpose of x, Q_k is an n×n symmetric matrix, P_k is an n component row vector, and c_k is a constant.

We shall restrict our considerations to quadratic functions in which the Q_k matrix of equation (1) is negative definite or negative semidefinite for all k. The concavity required by Theorem 2 is then assured. According to Theorem 1 efficient points may be determined by maximizing

$$\begin{aligned} Z(x) &= \sum_{k=1}^{r} \alpha_k(x'Q_kx + P_kx) \\ &= x'(\sum_{k=1}^{r} \alpha_kQ_k)x + (\sum_{k=1}^{r} \alpha_kP_k)x \\ (2) \qquad &= x'Qx + Px \ , \end{aligned}$$

subject to the linear constraints.

One approach to solving the r objective function problem is suggested by Antle[5]. A set of points which would be efficient points in the absence of constraints is determined. Points determined in this way will be called unconstrained efficient points. After a set of unconstrained efficient points is determined an operating point is selected from this set and checked for feasibility in the presence of constraints. If the selected point is feasible the problem is solved. When the selected point is not feasible, a set of feasible efficient points should be determined and an operating point chosen from this new set.

A second approach to the overall problem of locating an operating point would be to first determine a set of feasible efficient points and then select a desired operating point. The disadvantage of this approach is that it is a longer and therefore more expensive process to determine efficient points in the presence of constraints than in the absence of constraints.

The procedure of first determining a set of unconstrained efficient points is suggested by the author. If the selected point is feasible the constraints are of little concern. In the event that the selected point is not feasible the information obtained may still be of value by providing management with information about the effect of the constraints.

Whether determining constrained or unconstrained efficient points, the methods used are based on Theorems 1 and 2. However, the techniques used to actually locate the efficient points differ and will now be considered. It should be noted that the concept of unconstrained efficient points has no meaning when the objective functions are linear.

B. UNCONSTRAINED EFFICIENT POINTS

A method of determining unconstrained efficient points is presented by Antle[5] and will be stated here, without proof, in a slightly modified form.

Unconstrained efficient points for r objective functions of the form, $z_k(x) = c_k + x'Q_kx + Q_kx$, with Q_k being negative definite or negative semidefinite, are given by

$$x^* = .5[\sum_{k=1}^{r} \alpha_k Q_k]^{-1}[\sum_{k=1}^{r} \alpha_k P_k]$$

where $0 \leq \alpha_k < 1$ and $\sum_{k=1}^{r} \alpha_k = 1$.

Unconstrained efficient points are then obtained by using different sets of alphas. When there are only two objective functions the above equation may be reduced to

$$(3) \quad x^* = .5[\alpha Q_1 + (1 - \alpha)Q_2]^{-1}[\alpha P_1 + (1 - \alpha)P_2]; \quad 0 < \alpha < 1.$$

The above method will be referred to as Method A. The points where $\alpha = 0$ and $\alpha = 1$ must be given special attention and are discussed by Antle[5].

A special computer program was written to determine unconstrained efficient points of two objective functions using Method A and is presented in Appendix 2.

A different method, not based on Theorems 1 and 2, for determining unconstrained efficient points for two objective functions is presented by Umland and Smith[6]. Their approach is to treat one of the objective functions as a restricting relationship and use LaGrange multipliers to determine extermal values of the other function. The resulting LaGrange multiplier problem is then solved using different values of the restricting function. Each different value results in an unconstrained efficient point. This method will in general involve the solution of n+1 simultaneous non-linear equations for a problem with n independent variables. The above method of determining unconstrained efficient points will be referred to as Method B.

An example from the field of chemical research and originally presented by Umland and Smith[6] will now be given. One of the objective functions represents the yield from a chemical process and the other represents purity. The two independent variables could represent control factors such as temperature and pressure. Their objective functions were:

Predicted purity,

$$z_1(x) = 85.72 - 9.20x_1^2 - 5.18x_2^2 + 21.85x_1 + 8.59x_2 - 6.26x_1x_2$$

Predicted yield,

$$z_2(x) = 55.84 - 3.03x_1^2 - 6.96x_2^2 + 7.31x_1 + 26.65x_2 + 2.69x_1x_2$$

Purity is treated as the restricting function and the results obtained by Umland and Smith[6] for three selected values of purity are given in Table I, and plotted in Figure 3.

Table I

$z_1(x)$	$z_2(x)$	x_1	x_2
94.87	83.66	.965	1.088
92.47	86.73	1.005	1.316
89.995	88.68	1.075	1.479

The same problem was solved using the computer program written for Method A and the results are also plotted in Figure 3. These points are also listed in Appendix 2 as an example.

Figure 3 demonstrates the agreement of the two methods and by subdividing intervals of alpha, Method A yielded exactly those points (see Appendix 2) determined by Umland and Smith[6].

Although no comparable times are available the author believes that Method A, which involves only a summation and the inverse of a matrix, will be considerably faster than Method B, which requires the solution of systems of non-linear equations. Some sample computer times for Method A are included with the examples in Appendix 2.

C. FEASIBLE EFFICIENT POINTS.

When there are restrictions on the independent variables the methods of the previous section can no longer be used to maximize the function of Theorem 1 to determine efficient points. A technique was developed by Wolfe[2] to maximize quadratic functions of the form $Z(x) = x'Qx + Px$ subject to linear constraints. This method is becoming well known and is discussed in a recent book on non-linear programming by Hadley[7], and a book on operations research by Saaty[8]. A necessary restriction is that the matrix Q be negative definite or negative semidefinite. This agrees with our previous restrictions imposed by Theorem 2 and also

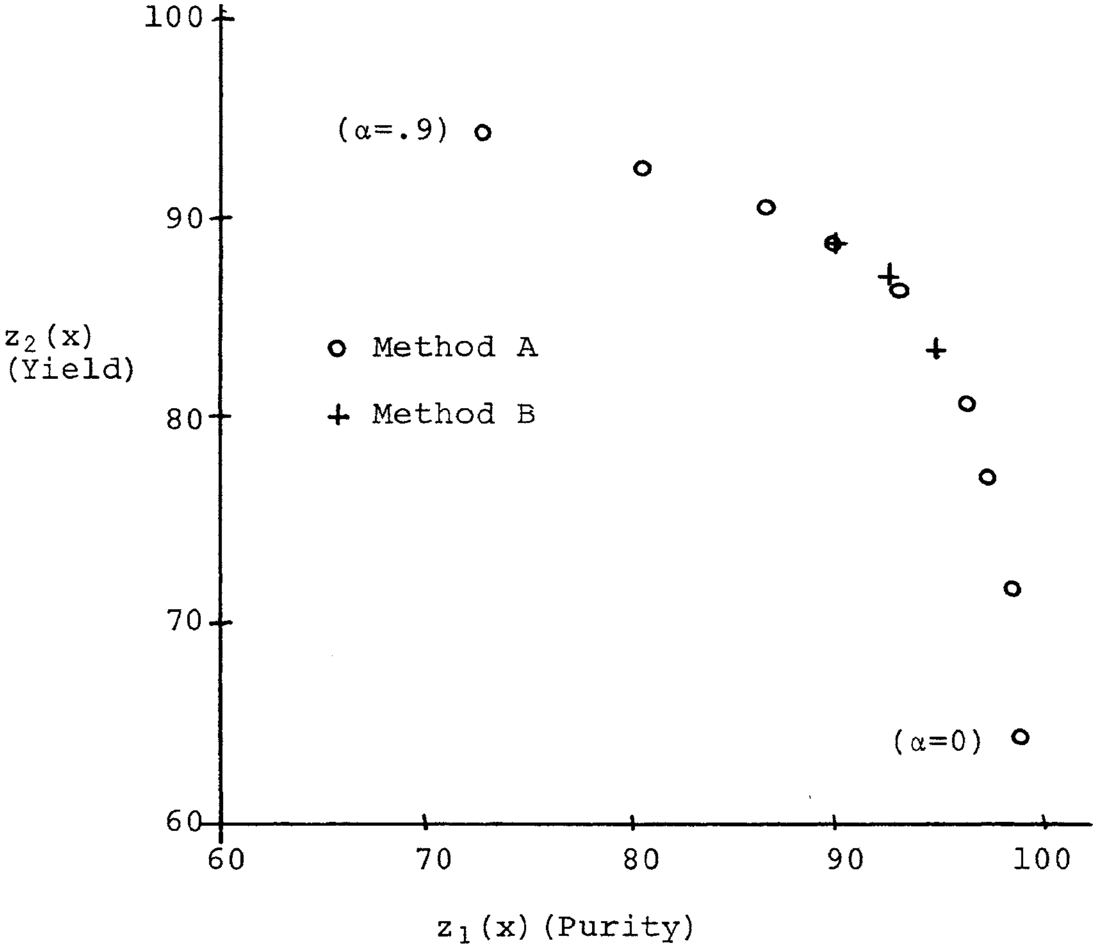


FIGURE 3. Values of objective functions at unconstrained efficient points.

assures us that any local maximum located will also be a global maximum[7].

Wolfe's method is used to maximize the composite objective function of equation (2), subject to the linear constraints, and thereby determine efficient points for the r objective functions $z_1(x)$, $z_2(x) \cdots z_r(x)$. Various values of alpha are chosen and the resulting efficient points are presented as in the linear case. A Fortran II computer program was written to determine efficient points using Wolfe's method to perform the maximization. This program and examples are presented in Appendix 3.

D. AN EXAMPLE.

Consider the problem:

Maximize:

$$z_1(x) = 370.0 - 5.3x_1^2 - 4.075x_2^2 + 19x_1 + 20x_2 - 4.2x_1x_2$$

$$z_2(x) = -176.0 - 9.0x_1^2 - 4.0x_2^2 + 144x_1 + 40x_2$$

Subject to:

$$4x_1 + 5x_2 \leq 36$$

$$x_2 \leq 2.25$$

$$x_1 - x_2 \geq 1$$

A two dimensional example was again chosen so that the methods and techniques may be more easily visualized. Figure 4 shows several contours for each of the objective functions and the set of feasible points bounded by the constraints.

Efficient points and unconstrained efficient points were both determined to illustrate the effect of the

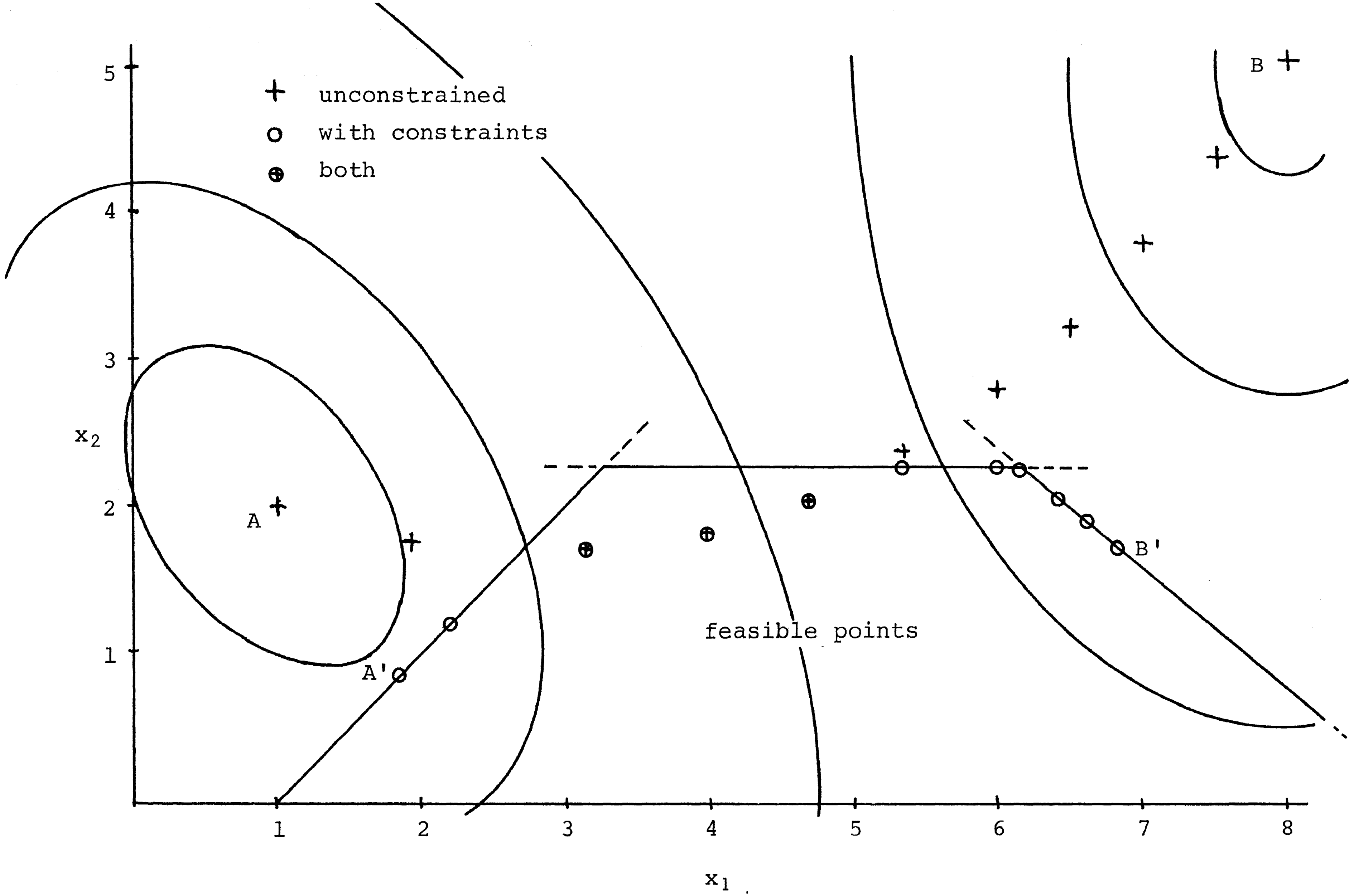


FIGURE 4. Objective function contours, feasible points, and efficient points. See example of section D for further explanation.

constraints. At the points labeled A and B in Figure 4, the maximum values of the objective functions $z_1(x)$ and $z_2(x)$, respectively, occur. In the presence of constraints the corresponding maximums occur at the points A' and B'. The feasible efficient points which are not unconstrained efficient points are on the boundary of the set of feasible points. The values of the objective functions at the efficient points are plotted in Figure 5. The four points labeled A, A', B, B' correspond to those with the same labels in Figure 4.

To determine the efficient points for a particular value of one of the objective functions a method of subdividing intervals of alpha is suggested. One such method is to repeatedly halve an interval of alpha, which is known to bracket the desired value, until the desired value is located within a predetermined limit. This technique is included in the computer program in Appendix 2. The approach used to solve this example would also be used to solve a two objective function problem with more than two independent variables. The results of such an example are presented in Appendix 3 as well as a problem with more than two objective functions.

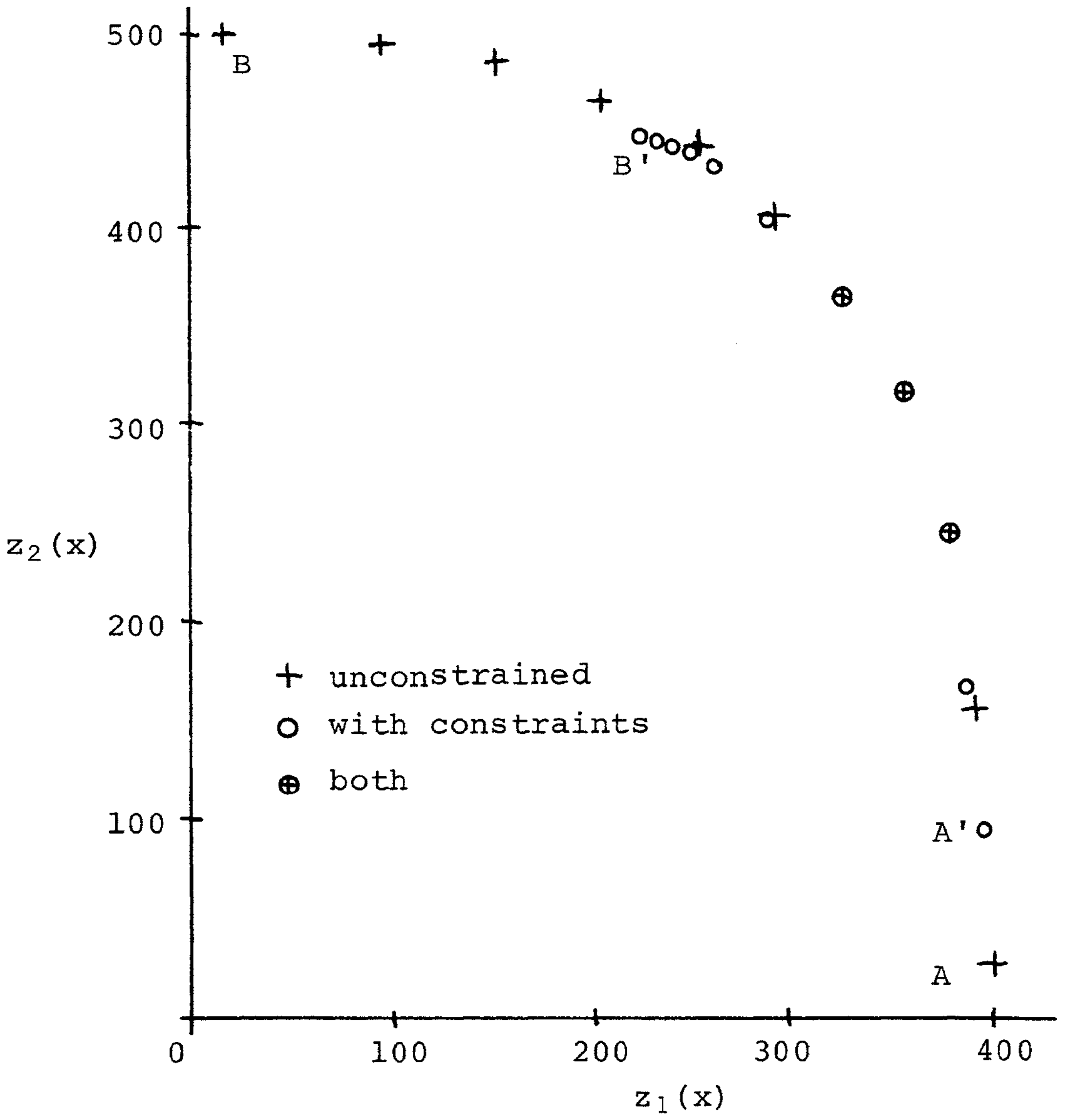


FIGURE 5. Objective functions evaluated at efficient points. See example of section D for further explanation.

CHAPTER V

CONCLUSIONS

A solution to the problem of simultaneously optimizing two objective functions which are subject to linear constraints may be obtained by using the computational methods presented. These methods are limited to either linear functions or concave quadratic functions. In general the solution consists of a set of efficient points with an associated parameter alpha. The number of independent variables and constraints is restricted only by the capabilities of the computer available. The results may be presented in a usable form and should provide management with a valuable tool.

If the number of objective functions is greater than two the usefulness of the results is decreased due to the difficulties of presentation and interpretation. Techniques of two parameter programming may provide a means of solving the three objective function problem.

The computational efficiency of the methods used could possibly be improved by using other variations of the simplex algorithm. If the idea of using a known efficient point as a starting place for finding another efficient point could be applied to quadratic objective functions, as it was to linear, a saving in computer time should be realized. The method presently used requires the solution of an entire quadratic programming problem for each efficient point.

As other computational techniques of non-linear programming become available they may be used together with the theorems presented to determine efficient points for the problem having more than one non-linear objective function.

APPENDIX 1

PROGRAM FOR DETERMINING EFFICIENT POINTS FOR LINEAR OBJECTIVE FUNCTIONS

I. PURPOSE.

The purpose of this program is to determine efficient points for r linear objective functions which are subject to linear constraints. If there are only two objective functions critical alphas are determined using the techniques of linear programming.

II. INPUT.

The input data consists of the constraints in the form of a simplex initial tableau, the variables in the initial basic solution, the objective function coefficients and control information. The six types of necessary data cards will now be discussed in detail.

TYPE 1. An identification card. Any information punched in columns 2 - 80 will be printed as the first line of output. Typical information would be problem identification, users name, date, etc. If no identification is desired a blank card should be used. FORMAT(79H)

TYPE 2. Control information. Punched as fixed point numbers, right justified, in the columns indicated.

columns

1 - 3 Number of independent variables, including slack, surplus, and artificial

variables.

4 - 6 Number of constraint equations.

7 - 9 Number of objective functions.

10 - 12 The subscript (number) of the first artificial (NOT slack or surplus) variable. If there are no artificial variables leave blank. FORMAT(24I3)

TYPE 3. Initial tableau. The coefficients of the constraint equations with slack, surplus, and artificial variables added. Punched as floating point numbers (with decimal point) in columns of 10 with up to 7 pieces of data per card. The constant term should be in columns 1 - 10 with the coefficients of x in the order x_1, $x_2 \cdots$. Each constraint equation must start a new card. FORMAT(7E10.2)

TYPE 4. Variables in basis. The subscript or column number of the variables in the original simplex tableau. Punched as fixed point numbers right justified in columns of three with up to 24 per card. The order should correspond to the rows, i.e., the subscript of the variable from the first row should be in columns 1 - 3, the subscript of the variable from the second row should be in columns 4 - 6, etc. (see example) FORMAT(24I3)

TYPE 5. Objective function coefficients. Punched as floating point numbers in columns of 10 with up to

7 pieces of data per card. The order of terms is constant, x_1, x_2 $\cdots$ x_n, including slack, surplus, and artificial variables. FORMAT(7E10.2)

TYPE 6. Alpha information. Punched as floating point numbers as indicated below. FORMAT(7E10.2)

columns

1 - 10 Leave this space blank to determine critical alphas. Punch any non-zero number to prohibit the finding of critical alphas.

11 - 20 Any number (non-zero) will result in the printing of the entire simplex tableau. If blank only efficient point information will be printed.

The following information is required only if critical alphas are NOT to be determined and there are only 2 objective functions.

21 - 30 starting alpha

31 - 40 stopping alpha

41 - 50 alpha increment

The following information must be furnished whenever there are more than 2 objective functions.

21 - 30 alpha (1)

31 - 40 alpha (2)

. .

. .

. .

61 - 70 alpha (5)

TYPE 7. Control card. To read new alpha information (a TYPE 4 card) this card should be left blank. To begin a new problem punch any positive number in columns 1 - 10. To terminate the program punch any negative number in columns 1 - 10. FORMAT(7E10.2)

III. OUTPUT.

The output consists of:

1. Identification from the TYPE 1 card.
2. objective function and constraint coefficients in the same order of the input.
3. efficient point information: alphas (or critical alphas), values of objective functions, and values of x.
4. the entire simplex tableau, if requested on the TYPE 4 input card. In the case of critical alphas this tableau will be the one corresponding to $\alpha = 0$, where $z_2(x)$ is optimum.

IV. EXAMPLE 1.

The example of Chapter III, section C, will now be given as solved on the computer.

A. The problem.

Maximize: $z_1(x) = 30 - 2x_1 + x_2$

$z_2(x) = -8 + 2x_1 + x_2$

Subject to: $3x_1 - x_2 \geq 0$

$-x_1 + x_2 \leq 4$

$$x_1 + 5x_2 \leq 18$$

$$x_1 - x_2 \leq 10$$

To illustrate the program use and options this problem will be solved 3 different times.

Determine efficient points:

1. at critical alphas.
2. at critical alphas and print final tableau.
3. for alpha from 0.0 to 0.8 in increments of 0.2.

After adding the necessary slack, surplus, and artificial variables the coefficient tableau is:

const	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	3	-1	-1	0	0	0	0	1
4	-1	1	0	1	0	0	0	0
50	1	5	0	0	1	0	0	0
18	1	1	0	0	0	1	0	0
10	1	-1	0	0	0	0	1	0

B. Program input data cards.

The data cards for the three desired outputs are the same except for the TYPE 5 control card and will be shown only once, but with the three different TYPE $\overset{6}{5}$ cards indicated.

The input data cards for this example are as follows:

TYPE

```
 [1]          EXAMPLE OF CHAPTER III, SECTION C, FOR APPENDIX 1.
 [2]           8  5  2  8
 [3]          0.        3.        -1.       -1.       0.        0.        0.
 [3]          0.        1.
 [3]          4.        -1.       1.        0.        1.        0.        0.
 [3]          0.        0.
 [3]          50.       1.        5.        0.        0.        1.        0.
 [3]          0.        0.
 [3]          18.       1.        1.        0.        0.        0.        1.
 [3]          0.        0.
 [3]          10.       1.        -1.       0.        0.        0.        0.
 [3]          1.        0.
 [4]           8  4  5  6  7
 [5]          30.       -2.       1.        0.        0.        0.        0.
 [5]          0.        -1000.
 [5]          -8.       2.        1.        0.        0.        0.        0.
 [5]          0.        -1000.
 [6]           (**BLANK CARD**)
 [7]           (**BLANK CARD**)
 [6]                    1.
 [7]           (**BLANK CARD**)
 [6]           33.33              0.0       .8        0.2
 [7]           -99.99
```

C. Program output.

The first page of output is as follows:

```
EXAMPLE OF CHAPTER III, SECTION C, FOR APPENDIX 1.
CONSTRAINTS
 0.0000E-99  3.0000E+00 -1.0000E+00 -1.0000E+00  0.0000E-99  0.0000E-99
 0.0000E-99  0.0000E-99  1.0000E+00
 4.0000E+00 -1.0000E+00  1.0000E+00  0.0000E-99  1.0000E+00  0.0000E-99
 0.0000E-99  0.0000E-99  0.0000E-99
 5.0000E+01  1.0000E+00  5.0000E+00  0.0000E-99  0.0000E-99  1.0000E+00
 0.0000E-99  0.0000E-99  0.0000E-99
 1.8000E+01  1.0000E+00  1.0000E+00  0.0000E-99  0.0000E-99  0.0000E-99
 1.0000E+00  0.0000E-99  0.0000E-99
 1.0000E+01  1.0000E+00 -1.0000E+00  0.0000E-99  0.0000E-99  0.0000E-99
 0.0000E-99  1.0000E+00  0.0000E-99
Z1=    3.0000E+01 -2.0000E+00  1.0000E+00  0.0000E-99  0.0000E-99  0.0000E-99
       0.0000E-99  0.0000E-99 -1.0000E+03
Z2=   -8.0000E+00  2.0000E+00  1.0000E+00  0.0000E-99  0.0000E-99  0.0000E-99
       0.0000E-99  0.0000E-99 -1.0000E+03
  ALPHA RANGE       Z1(X)          Z2(X)                  X(I)
 0.0000  .2499  6.00000E+00  2.39999E+01    1.3999E+01  4.0000E+00  3.7999E+01
                                            1.3999E+01  1.5999E+01  0.0000E-99
                                            0.0000E-99  0.0000E-99
  .2499  .4500  1.79999E+01  1.99999E+01    1.0000E+01  7.9999E+00  2.2000E+01
                                            6.0000E+00  0.0000E-99  0.0000E-99
                                            7.9999E+00  0.0000E-99
  .4500  .7500  2.89999E+01  1.09999E+01    5.0000E+00  8.9999E+00  6.0000E+00
                                            0.0000E-99  0.0000E-99  4.0000E+00
                                            1.3999E+01  0.0000E-99
  .7500 1.0000  3.20000E+01  1.99999E+00    1.9999E+00  5.9999E+00  0.0000E-99
                                            0.0000E-99  1.8000E+01  1.0000E+01
                                            1.3999E+01  0.0000E-99
```

The second page of output is as follows:

```
 FINAL TABLEAU, NO. 12
VIB      XB                          A VECTORS
  1 13.99999E+00 10.00000E-01 00.00000E-99 00.00000E-99 00.00000E-99
                 00.00000E-99 49.99999E-02 49.99999E-02
  2 40.00004E-01 00.00000E-99 10.00000E-01 00.00000E-99 00.00000E-99
                 00.00000E-99 49.99998E-02-49.99997E-02
  3 37.99999E+00 00.00000E-99 00.00000E-99 10.00000E-01 00.00000E-99
                 00.00000E-99 10.00000E-01 19.99999E-01
  4 13.99999E+00 00.00000E-99 00.00000E-99 00.00000E-10 00.00000E-99
                 00.00000E-99 10.00000E-08 99.99999E-02
  5 15.99998E+00 00.00000E-99 00.00000E-99 00.00000E-99 00.00000E-99
                 10.00000E-01-29.99999E-01 19.99999E-01
  0 32.00000E+00 00.00000E-99 00.00000E-99 00.00000E-99 00.00000E-99
                 00.00000E-99 14.99999E-01 50.00001E-02
 ALPHA RANGE       Z1(X)          Z2(X)                  X(I)
0.0000  .2500  6.00000E+00  2.40000E+01   1.3999E+01  4.0000E+00  3.7999E+01
                                          1.3999E+01  1.5999E+01  0.0000E-99
                                          0.0000E-99  0.0000E-99
 .2500  .4500  1.79999E+01  2.00000E+01   1.0000E+01  7.9999E+00  2.2000E+01
                                          6.0000E+00  0.0000E-99  0.0000E-99
                                          7.9999E+00  0.0000E-99
 .4500  .7500  2.89999E+01  1.09999E+01   5.0000E+00  8.9999E+00  6.0000E+00
                                          0.0000E-99  0.0000E-99  4.0000E+00
                                          1.3999E+01  0.0000E-99
 .7500 1.0000  3.20000E+01  1.99999E+00   1.9999E+00  5.9999E+00  0.0000E-99
                                          0.0000E-99  1.8000E+01  1.0000E+01
                                          1.3999E+01  0.0000E-99
```

The third page of output is as follows:

```
ALPHA(I)       Z(I)
 0.0000    6.0000083E+00
 1.0000    2.4000000E+01
              X(I)
 1.3999E+01  4.0000E+00  3.7999E+01  1.3999E+01  1.5999E+01  0.0000E-99
 0.0000E-99  0.0000E-99
ALPHA(I)       Z(I)
  .2000    6.0000083E+00
  .8000    2.4000000E+01
              X(I)
 1.3999E+01  4.0000E+00  3.7999E+01  1.3999E+01  1.5999E+01  0.0000E-99
 0.0000E-99  0.0000E-99
ALPHA(I)       Z(I)
  .4000    1.7999997E+01
  .6000    2.0000001E+01
              X(I)
 1.0000E+01  7.9999E+00  2.2000E+01  6.0000E+00  0.0000E-99  0.0000E-99
 7.9999E+00  0.0000E-99
ALPHA(I)       Z(I)
  .6000    2.8999999E+01
  .4000    1.0999999E+01
              X(I)
 5.0000E+00  8.9999E+00  6.0000E+00  0.0000E-99  0.0000E-99  4.0000E+00
 1.3999E+01  0.0000E-99
ALPHA(I)       Z(I)
  .8000    3.1999996E+01
  .2000    2.0000009E+00
              X(I)
 2.0000E+00  5.9999E+00  0.0000E-99  0.0000E-99  1.8000E+01  1.0000E+01
 1.3999E+01  0.0000E-99
```

V. EXAMPLE 2.

A. Determine efficient points and critical alphas which will

maximize $z_1(x) = 2x_1 + x_2 + 4x_3 + 5x_4$ and

minimize $z_2(x) = x_1 + 2x_2 + 3x_3 + 4x_4$

Subject to: $x_1 + 3x_2 + 2x_3 + 5x_4 \leq 20$

$2x_1 + 16x_2 + x_3 + x_4 \geq 4$

$3x_1 - x_2 - 5x_3 + 10x_4 \leq -10$

B. Input data cards.

```
TYPE

[1]        EXAMPLE 2. FOR APPENDIX 1.              DETERMINE CRITICAL ALPHAS
[2]         9  3  2  8
[3]        20.        1.         3.         2.         5.         1.         0.
[3]        0.         0.         0.
[3]        4.         2.         16.        1.         1.         0.        -1.
[3]        0.         1.         0.
[3]        10.       -3.         1.         5.        -10.        0.         0.
[3]       -1.         0.         1.
[4]         5  8  9
[5]        0.         2.         1.         4.         5.         0.         0.
[5]        0.        -1000.     -1000.
[5]        0.0       -1.        -2.        -3.        -4.
[6]        0.        -1000.      -1000.
            (**BLANK CARD**)
```

C. Computer output.

```
 EXAMPLE 2. FOR APPENDIX 1.             DETERMINE CRITICAL ALPHAS
CONSTRAINTS
 2.0000E+01  1.0000E+00  3.0000E+00  2.0000E+00  5.0000E+00  1.0000E+00
 0.0000E-99  0.0000E-99  0.0000E-99  0.0000E-99
 4.0000E+00  2.0000E+00  1.6000E+01  1.0000E+00  1.0000E+00  0.0000E-99
-1.0000E+00  0.0000E-99  1.0000E+00  0.0000E-99
 1.0000E+01 -3.0000E+00  1.0000E+00  5.0000E+00 -1.0000E+01  0.0000E-99
 0.0000E-99 -1.0000E+00  0.0000E-99  1.0000E+00
Z1=     0.0000E-99  2.0000E+00  1.0000E+00  4.0000E+00  5.0000E+00  0.0000E-99
        0.0000E-99  0.0000E-99 -1.0000E+03 -1.0000E+03
Z2=     0.0000E-99 -1.0000E+00 -2.0000E+00 -3.0000E+00 -4.0000E+00  0.0000E-99
        0.0000E-99  0.0000E-99 -1.0000E+03 -1.0000E+03
  ALPHA RANGE       Z1(X)          Z2(X)                     X(I)
 0.0000  .3704  8.02531E+00 -6.17721E+00   0.0000E-99  1.2658E-01  1.9746E+00
                                           0.0000E-99  1.5670E+01  0.0000E-99
                                           1.3999E+01  0.0000E-99  0.0000E-99
  .3704  .3888  1.13846E+01 -8.15384E+00   7.6923E-01  0.0000E-99  2.4615E+00
                                           0.0000E-99  1.4307E+01  0.0000E-99
                                           1.3999E+01  0.0000E-99  0.0000E-99
  .3888 1.0000  4.00000E+01 -2.63636E+01   7.2727E+00  0.0000E-99  6.3636E+00
                                           0.0000E-99  0.0000E-99  1.6909E+01
                                           1.3999E+01  0.0000E-99  0.0000E-99
```

PROGRAM LISTING

```
*FANDK0805
C PROGRAM WJLL1
      DIMENSION Y(20,40),IV(40),C(40),IB(20),COB(5,40),
     1AL(5),R(5),F(40),X(40)
     1X(40)
    1 READ 999
      PRINT 970
      PRINT 999
      READ 900,NV,MC,NOBJ,JA
      M=MC+1
      N= NV+1
      ITAB=1
      M5=M*5
      ALS=0.0
      PRINT 9018
      DO  7000  I=1,MC
      READ 901,(Y(I,J),J=1,N)
 7000 PRINT 9023, (Y(I,J),J=1,N)
      DO7001 J=1,NV
 7001 IV(J)=0
    2 READ 900,(IB(I),I=1,MC)
      DO 3 I=1,MC
      K=IB(I)
    3 IV(K)=I
  302 DO 304 NO=1,NOBJ
      READ 901,(COB(NO,J),J=1,N)
  304 PRINT 9004, NO,(COB(NO,J),J=1,N)
  500 IF(NOBJ-2)505,515,510
  505 READ 901, CRIT,PRIT
      KRIT=0
      AL(1)=1.0
      GO TO 308
  510 KRIT=0
      READ 901,CRIT,PRIT,(AL(I),I=1,NOBJ)
      GO TO 308
  515 READ 901,CRIT,PRIT,AL(1),ALS,ALD
      IF(CRIT) 520,525,520
  520 KRIT=0
      GO TO 530
  525 KRIT=5
      AL(1)=0.0
  530 AL(2)=1.0-AL(1)
  308 DO 310 J=2,N
      C(J)=0.0
      DO 309 NO=1,NOBJ
  309 C(J)=C(J)+AL(NO)*COB(NO,J)
  310 CONTINUE
 1004 IB(M)=0
      NTIM =0
      C(1)=0.0
```

```
C     CALC ZJ-CJ
    5 DO11 J=1,N
      ZJ=0.0
      DO 10 I=1,MC
    7 K=IB(I) +1
   10 ZJ=ZJ+ C(K)*Y(I,J)
   11 Y(M,J)=ZJ-C(J)
      IF(KRIT) 13,20,20
   13 KRIT =1
C     SELECT IN VECTOR AND TEST FOR OPTIMUM
   20 ZCM=0.0
      JM=0
      DO 25 J=2,N
   22 IF(Y(M,J))23,25,25
   23 IF(ZCM-Y(M,J))25,25,24
   24 ZCM=Y(M,J)
      JM=J
   25 CONTINUE
      IF(ZCM)26,50,50
C     SELECT OUT VECTOR AND TEST FOR UNBOUNDED, JM GOING IN
   26 YM=1.E+45
      NTIM =0
      DO 35 I=1,MC
      IF(Y(I,JM))35,35,28
   28 IF(Y(I,1))36,36,30
   30 XY   =Y(I,1)/Y(I,JM)
      IF(YM-XY   )35,35,32
   32 YM=XY
      IM=I
   35 CONTINUE
      IF(YM-1.E+44)37,55,55
   36 IM=I
C     ROW OPERATIONS WITH JM REPLACING IM, PIVOT Y(IM,JM)
   37 PIV= Y(IM,JM)
      IF(M5-ITAB) 58,58,39
   39 ITAB=ITAB+1
      DO 38 J=1,N
   38 Y(IM,J)=Y(IM,J)/PIV
   40 DO 44 I=1,M
      IF(I-IM)41,44,41
   41 A=Y(I,JM)
      DO 43 J=1,N
      Y(I,J)=Y(I,J)-Y(IM,J)*A
   43 CONTINUE
   44 CONTINUE
      K=IB(IM)
      IV(K)=0
      IV(JM-1)=IM
      IB(IM)=JM -1
  400 IF(KRIT) 90,401,401
  401 IF(JA-1)20,45,45
```

```
C     TEST BASIS FOR ARTIFICIAL VARIABLES
   45 DO 46 I=1,MC
      IF(IB(I)-JA)46,20,20
   46 CONTINUE
      N=JA
      JA=0
      GO TO 5
C    OPTIMAL FEASIBLE SOLUTION
   50 IF(NTIM -1)51,52,52
   51 NTIM =1
      GO TO 5
   52 IF(JA-1)90,60,60
   53 PRINT 960,ITAB
      GO TO 93
   55 JUB=JM-1
      PRINT 961,JUB,ITAB
      GO TO 93
   58 PRINT 962
      GO TO 93
   60 DO 63 I=1,MC
      IF(IB(I)-JA)63,62,62
   62 IF(Y(I,1))53,63,53
   63 CONTINUE
C    OUTPUT
   90 CONTINUE
      DO 535 K=1,NV
  535 X(K)=0.0
      DO 540 K=1,NV
      KK=IB(K)
  540 X(KK)=Y(K,1)
      DO 545 NO=1,NOBJ
      R(NO)=COB(NO,1)
      DO 542 K=2,N
  542 R(NO)=R(NO)+COB(NO,K)*X(K-1)
  545 CONTINUE
  550 IF(KRIT)408,560,555
  555 IF(PRIT)404,404,330
  560 IF(PRIT)330,570,330
  570 PRINT9001,(AL(I),R(I), I=1,NOBJ)
      PRINT 9002, (X(I),I=1,N)
      GO TO 575
  330 CONTINUE
   93 PRINT 952,ITAB
   94 DO95 K=1,M
   95 PRINT 953,IB(K),(Y(K,J),J=1,N)
      IF(KRIT) 408,570,404
  575 IF(AL(1)-ALS)580,96,96
  580 AL(1)=AL(1)+ALD
      GO TO 530
   96 CONTINUE
   97 READ 901,CODE
      IF(CODE)  1001,500,1
 1001 CALL EXIT
```

```
C     DETERMINE CRITICAL VALUES OF ALPHA
C     COMPUTE F VECTOR
  404 KRIT =-1
      PRINT 9000
      DO 405 J=2,N
  405 F(J)= COB(1,J)-COB(2,J)
C     SELECT IN VECTOR  AND TEST  FOR  INFINITE ALPHA
  408 AC = 1.E+45
  410 DO  440  J=2,N
      FY=0.0
      IF(IV(J-1))440,417,440
  417 IF(Y(M,J)) 440,420,420
  420 DO 422 I=1,MC
      K=IB(I) +1
  422 FY = F(K)*Y(I,J) + FY
  425 FYF = FY - F(J)
      IF(FYF) 429,440,440
  429 AJ = -Y(M,J)/ FYF
      IF(AC -AJ) 440,440,430
  430 JM=J
      AC=AJ
  440 CONTINUE
      IF(AC-1.E+44) 441,445,445
  441 IF(AC-.99995)442,445,445
  442 PRINT 4900 , AL(1),AC,R(1),R(2),(X(I),I=1,NV)
      AL(1)= AC
      GO TO 26
  445 AC=1.
      PRINT 4900 , AL(1),AC,R(1),R(2),(X(I),I=1,NV)
      GO TO 97
  900 FORMAT(20I3)
  901 FORMAT(7E10.2)
  952 FORMAT(21H0  FINAL TABLEAU, NO.I3,/5H  VIB,6X,2HXB25X,
     19HA VECTORS )
  953 FORMAT(1X,I4,5E13.5,/(18X,4E13.5))
  960 FORMAT(23H1  NO FEASIBLE SOLUTION ,15X,11HTABLEAU NO.,I3)
  961 FORMAT(30H UNBOUNDED OBJECTIVE, VECTOR A,I2,10X,
     111HTABLEAU NO.,I3)
  962 FORMAT(18H1 POSSIBLE CYCLING    )
  970 FORMAT(1H1)
  999 FORMAT(1H1,79H
     1                                                 )
 4900 FORMAT(1X,2F7.4,2(1X,E12.5),1X,3(1X,E11.4)/(43X,E11.4,
     11X,E11.4,1X,E11.4))
 9000 FORMAT(2X,12H ALPHA RANGE  ,7X,5HZ1(X),10X,5HZ2(X),
     118X,4HX(I))
 9001 FORMAT(20H ALPHA(I)        Z(I) /(F8.4,3X,E14.7) )
 9002 FORMAT(15X,4HX(I),/6(1X,E11.4))
 9004 FORMAT(1X,1HZ,I1,1H=,2X,6(1X,E11.4),/(7X,E11.4,1X,
     1E11.4,1X,E11.4,1X,E11.4,1X,E11.4,1X,E11.4))
 9018 FORMAT( 12H CONSTRAINTS )
 9023 FORMAT(6(1X,E11.4))
      END
```

APPENDIX 2

PROGRAM FOR DETERMINING UNCONSTRAINED EFFICIENT POINTS

I. PURPOSE.

The purpose of this program is to determine efficient points for two quadratic objective functions, each of the form $z(x) = C_0 - x'Qx + Bx$. The method presented by Antle [5] is used and is restricted to problems where the Q matrix is positive definite or positive semi-definite.

II. INPUT.

Five different types of input cards are required and a detailed description of each follows.

TYPE 1. An identification card. Any information punched in columns 2 - 80 will be printed as the first line of output on each page. Typical information would be problem identification, date, users name, etc. If no identification is needed the card may be left blank. FORMAT(79H)

TYPE 2. The number of independent variables. Punched as a fixed point number, right justified in columns 1 - 3. FORMAT(I3)

TYPE 3. The coefficients of the objective functions. Punched as floating point numbers (with decimal point) in columns of 10, with up to 7 pieces of data per card. Each objective function begins a new card. The order of the coefficients must be: constant, square terms, linear terms, cross product terms; i.e. c_0, x_1^2, x_2^2, $\cdots$ x_n^2, x_1, x_2, $\cdots$ x_1x_2, x_1x_3, $\cdots$ x_1x_n, x_2x_3, x_2x_4, $\cdots$ x_2x_n, $\cdots$ $x_{n-1}x_n$.

If any terms are missing either a zero should be punched or a blank left in the proper position.

FORMAT(7E10.2)

TYPE 4. Values and increments of alpha. Punched as floating point numbers in columns of 10 as follows:

columns	
1 - 10	starting alpha
11 - 20	stopping alpha
21 - 30	alpha increment

The starting alpha must be less than or equal to the stopping alpha and the increment must be positive. If only one value of alpha is to be used (instead of a sequence) that value should be punched as the starting alpha and the remainder of the card left blank. As many of this type card may be used as desired. FORMAT(7E10.2)

TYPE 5. Control card. To begin an entirely new problem punch any negative number in columns 1 - 10. The next data card should then be a TYPE 1 card. To terminate the program punch any negative number in columns 1 - 10 and any number in columns 31 - 40. FORMAT(7E10.2)

III. OUTPUT.

The normal output consists of:

1. Identification from the TYPE 1 card.
2. Coefficients of the objective functions in the same order as entered on the TYPE 2 cards.
3. Efficient point information: alpha, values of objective functions, and values of x.

IV. PROGRAM VARIATIONS.

A. The following output variations are possible:

1. The identification may be suppressed by punching any number in columns 41 - 50 of the TYPE 4 card. (The TYPE 1 card must still be included as an input card.)
2. The coefficients of the objective functions may be suppressed by punching any number in columns 51 - 60 of the TYPE 4 card.
3. A new page of output may be started by punching any number in columns 61 - 70 of the TYPE 4 card.

In all of the above "any number" is meant to imply any number except zero.

B. To locate the efficient point with a particular value of one of the objective functions the following card should be entered as a TYPE 4 card.

columns

1 - 10 Either a ±1.0 or ±2.0, denoting which objective function is being used. The (+) value should be used if the objective function is an increasing function of alpha and the (-) value if the objective function is a decreasing function of alpha.

11 - 20 The value of the objective function.

21 - 30 A value of alpha which is known to be lower than the alpha of the desired point. If not known, leave blank.

31 - 40 A value of alpha which is known to be higher than the alpha of the desired point. If not known, leave blank.

71 - 80 The allowable error in the value of the objective function. *This space MAY NOT be zero or blank*.

Columns 41 - 70 are used to offer the same output variations as discussed in the previous section. There is no limit on the number of these cards which may be used. If the specified value is not located after 20 steps of halving the interval of alpha the process is terminated, a message is printed and the next card is read. If an incorrect interval of alpha is specified the desired point will not be located.

VI. EXAMPLE 1.

A. The problem.

For the functions used in Chapter IV, section B,

$$z_1(x) = 85.72 - 9.20x_1^2 - 5.18x_2^2 + 21.85x_1 + 8.59x_2 - 6.26x_1x_2$$

$$z_2(x) = 55.84 - 3.03x_1^2 - 6.96x_2^2 + 7.31x_1 + 26.65x_2 + 2.69x_1x_2,$$

Determine efficient points:

(1) for alpha from 0 to 1 in increments of 0.1

(2) where, $z_1(x) = 94.87 \pm .001$

(3) where, $z_2(x) = 86.73 \pm .0005$

$z_2(x) = 88.68 \pm .0001$

and print the answers on a new page without the objective function coefficients.

(4) where, $z_1(x) = 30.00 \pm .5$

B. Input data.

```
TYPE
[1]    EXAMPLE FROM TECHNOMETRICS       PROGRAM WJLQ3      6/18/65
[2]     2
[3]    85.72     -9.20     -5.18     21.85     8.59      -6.26
[3]    55.84     -3.03     -6.96     7.31      26.65     2.69
[4]     0.0       1.0       0.1
[4]    1.0       94.87                                                    .001
[4]    -2.0      86.73                                    8         8     .0005
[4]    -2.0      88.68     0.3       0.4                                  .001
[4]    1.0       30.0                                                     0.5
[5]    -99.99                        -99.99
```

C. Program output.

```
      EXAMPLE FROM TECHNOMETRICS       PROGRAM WJLQ3      6/18/65
      Z1=    8.5720E+01 -9.2000E+00 -5.1800E+00  2.1850E+01  8.5900E+00 -6.2600E+00
      Z2=    5.5840E+01 -3.0300E+00 -6.9600E+00  7.3100E+00  2.6650E+01  2.6900E+00
        ALPHA        Z1              Z2                        X(I)
       0.0000   4.6846617E+01   9.5362406E+01   2.2490E+00  2.3491E+00
        .1000   6.9946571E+01   9.4308053E+01   1.7079E+00  2.0576E+00
        .2000   8.0556254E+01   9.2503227E+01   1.3922E+00  1.8391E+00
        .3000   8.6364882E+01   9.0600444E+01   1.1965E+00  1.6525E+00
        .4000   9.0016782E+01   8.8651685E+01   1.0740E+00  1.4780E+00
        .5000   9.2590480E+01   8.6554190E+01   1.0018E+00  1.3040E+00
        .6000   9.4570040E+01   8.4135175E+01   9.6755E-01  1.1219E+00
        .7000   9.6171670E+01   8.1151356E+01   9.6511E-01  9.2384E-01
        .8000   9.7462130E+01   7.7252533E+01   9.9210E-01  7.0152E-01
        .9000   9.8391780E+01   7.1908468E+01   1.0491E+00  4.4489E-01
       1.0000   9.8774657E+01   6.4272838E+01   1.1397E+00  1.4048E-01
        .6173   9.4870890E+01   8.3667250E+01   9.6496E-01  1.0890E+00
```

[New Page]

```
      EXAMPLE FROM TECHNOMETRICS       PROGRAM WJLQ3      6/18/65
        ALPHA        Z1              Z2                        X(I)
        .4920   9.2412160E+01   8.6729699E+01   1.0060E+00  1.3180E+00
        .3986   8.9975680E+01   8.8679007E+01   1.0753E+00  1.4804E+00
       CAN  NOT  LOCATE  Z1=   3.0000000E+01
```

VII. EXAMPLE 2.

A. The problem.

Determine unconstrained efficient points for alpha from 0 to 1 in increments of 0.1 for

$$z_1(x) = 370 - 5.3x_1^2 - 4.075x_2^2 + 19x_1 + 20x_2 - 4.2x_1x_2$$

$$z_2(x) = -176 - 9.0x_1^2 - 4.0x_2^2 + 144x_1 + 40x_2$$

B. Input cards.

```
TYPE
 [1]      UNCONSTRAINED EFFICIENT POINTS FOR THE EXAMPLE OF CHAPT. IV,SEC. D
 [2]       2
 [3]      370.        -5.3       -4.075      19.0        20.         -4.2
 [3]      -176.       -9.         -4.        144.        40.
 [4]      0.0         1.0         0.1
 [5]      -99.99                             -99.99
```

C. Program output.

```
UNCONSTRAINED EFFICIENT POINTS FOR THE EXAMPLE OF CHAPT. IV,SEC. D
Z1=     3.7000E+02 -5.3000E+00 -4.0750E+00  1.9000E+01  2.0000E+01 -4.2000E+00
Z2=    -1.7600E+02 -9.0000E+00 -4.0000E+00  1.4400E+02  4.0000E+01  0.0000E-99
   ALPHA          Z1             Z2                     X(I)
  0.0000    1.2925060E+01   5.0000004E+02    7.9999E+00  5.0000E+00
   .1000    8.6338690E+01   4.9616188E+02    7.5129E+00  4.3474E+00
   .2000    1.4985705E+02   4.8497378E+02    7.0127E+00  3.7496E+00
   .3000    2.0463179E+02   4.6672894E+02    6.4927E+00  3.2093E+00
   .4000    2.5167539E+02   4.4139832E+02    5.9449E+00  2.7310E+00
   .5000    2.9185447E+02   4.0850162E+02    5.3583E+00  2.3216E+00
   .6000    3.2585835E+02   3.6687254E+02    4.7184E+00  1.9912E+00
   .7000    3.5411903E+02   3.1422078E+02    4.0046E+00  1.7552E+00
   .8000    3.7662791E+02   2.4628097E+02    3.1870E+00  1.6368E+00
   .9000    3.9251326E+02   1.5506244E+02    2.2202E+00  1.6727E+00
  1.0000    3.9901927E+02   2.4956370E+01    1.0305E+00  1.9229E+00
```

PROGRAM LISTING

```
CPROGRAM WJLQ3
      DIMENSION A1(20,20),A2(20,20),B1(20),B2(20),C1(200),
     1C2(200),A(20,20),B(20),X(20)
     1A(20,20),B(20),X(20)
      PRINT 9000
    1 READ 9001
   10 AL1=2.0
      AL2=2.0
      ALS=0.0
      ALD=0.0
      FIND = 0.0
      LINE = 50
   15 READ   9002, N,LIST
      NN=  (N*(N+3))/2
      NNC=NN+1
      N1=N+1
   20 READ   9003, C1(NNC),(C1(J),J=1,NN)
      READ   9003, C2(NNC),(C2(J),J=1,NN)
      NM1=N-1
      L=2*N
C  SET UP A1,A2,B1,B2   MATRIX
      B1(N)=C1(L)
      B2(N)=C2(L)
   40 DO 65 I=1,NM1
      A1(I,I)=-C1(I)
      A2(I,I)=-C2(I)
   45 K=N+I
      B1(I)=C1(K)
      B2(I)=C2(K)
      J2=I+1
   50 DO 60 J1=J2,N
      L=L+1
      A1(I,J1)=-0.5*C1(L)
   55 A2(I,J1)=-.5*C2(L)
      A1(J1,I)=A1(I,J1)
   60 A2(J1,I)=A2(I,J1)
   65 CONTINUE
      A1(N,N)=-C1(N)
      A2(N,N)=-C2(N)
C   TEST ALPHAS
  100 IF(AL1-ALS)120,105,105
  105 READ  9008,AL1,ALS,ALD,STOP,COMT,COEF,PAGE,ERR
C ALTERNATELY=NOB,VOB,ALOW,ALHI,COMT,COEF,PAGE,ERR
      KONT=0
  106 IF(PAGE)107,108,107
  107 LINE= 50
  108 IF(ERR )300,109,300
  109 IF(AL1)110,125,125
  110 IF(STOP)115,1,115
  115 CALL EXIT
  120 AL1=AL1+ALD
  125 AL2=1.0-AL1
```

```
C  SET  UP  A  AND  B  MATRICES
  130 DO 140  I=1,N
      A(I,N1)=AL1*B1(I)+AL2*B2(I)
  135 DO 138  J=1,N
  138 A(I,J)=AL1*A1(I,J)+AL2*A2(I,J)
  140 CONTINUE
C GAUJOR IS A ROUTINE TO SOLVE N LINEAR EQUATIONS
      CALL GAUJOR(A,N,N1,20,20)
      DO  190  I=1,N
  190 X(I) = 0.5*A(I,N1)
C  EVALUATE RESPONSE FUNCTIONS
  200 L=2*N
      Y1=C1(NNC)+X(N)*(C1(L)+C1(N)*X(N))
      Y2=C2(NNC)+X(N)*(C2(L)+C2(N)*X(N))
  205 DO 225 I=1,NM1
      K=N+I
      Y1=Y1+X(I)*(C1(K)+C1(I)*X(I))
  210 Y2=Y2+X(I)*(C2(K)+C2(I)*X(I))
  215 DO 220 J=I,NM1
      L=L+1
      Y1=Y1+X(I)*X(J+1)*C1(L)
  220 Y2=Y2+X(I)*X(J+1)*C2(L)
  225 CONTINUE
C OUTPUT
  227 IF( ERR  )325,230,325
  230 IF(LINE - 40) 280,280,240
  240 PRINT  9000
      LINE = 0
      IF(COMT) 250,245,250
  245 PRINT 9001
      LINE = LINE + 1
  250 IF(COEF) 260,255,260
  255 K=1
      LINE = LINE + 2*((NNC+6)/7)
      PRINT 9004,K,C1(NNC),(C1(J),J=1,NN)
      K=2
      PRINT 9004,K,C2(NNC),(C2(J),J=1,NN)
  260 PRINT 9005
  280 PRINT 9006,AL1,Y1,Y2,(X(I),I=1,N)
      LINE = LINE +(N +3)/4
      GO TO 100
C  FIND  A  PATICULAR  VALUE  OF  OBJ.
  300 NOB=ABSF(AL1)
      VOB = ALS
      ALS=0.0
      FIND = AL1
      ALOW=ALD
      ALHI=STOP
      IF(  ALHI  )320,320,310
  310 AL1=ALOW
      ALIC = (ALHI -ALOW)
      GO  TO  125
  320 AL1=.5
```

```
      ALIC=.5
      GO  TO  125
  325 IF(NOB-1)330,330,340
  330 ZD = Y1- VOB
      GO TO 360
  340 ZD = Y2- VOB
  360 IF( ABSF(ZD)- ERR) 230,230,365
  365 ALIC= 0.5*ALIC
      ZD = FIND* ZD
  367 IF( ZD  )370,370,375
  370 AL1=  AL1 + ALIC
      GO  TO  380
  375 AL1=  AL1 - ALIC
  380 ZDP=ZD
      KONT = KONT +1
      IF(KONT  -  20)125,125,390
  390 PRINT 9007 , NOB,VOB
      GO TO 105
 9000 FORMAT(1H1)
 9001 FORMAT(1X,79H
     1                                  )
 9002 FORMAT(24I3)
 9003 FORMAT(7E10.2)
 9004 FORMAT(1X,1HZ,I1,1H=,2X,7(1X,E11.4),/(7X,E11.4,1X,
     1E11.4,1X,E11.4,1X,E11.4,1X,E11.4,1X,E11.4,1X,E11.4))
 9005 FORMAT(4X,5HALPHA,7X,2HZ1,14X,2HZ2,20X,4HX(I))
 9006 FORMAT(2X,F7.4,2(2X,E14.7),1X,4(1X,E11.4)/(43X,E11.4,
     11X,E11.4,1X,E11.4,1X,E11.4))
 9007 FORMAT(21H  CAN  NOT  LOCATE  Z,I1,1H=,2X,E14.7  )
 9008 FORMAT(8E10.2)
      END
```

APPENDIX 3

PROGRAM TO DETERMINE EFFICIENT POINTS FOR QUADRATIC OBJECTIVE FUNCTIONS SUBJECT TO LINEAR CONSTRAINTS

I. PURPOSE.

The purpose of this program is to utilize Wolfe's method for quadratic programming to determine efficient points for quadratic objective functions, $z_k(x) = x'Q_k x + P_k x$, $k = 1 \cdots r$, subject to linear constraints of the forms $A_{11}x \leq b_1$ and $A_{21}x \geq b_2$, with $b_1, b_2 \geq 0$. A modification of the form of Wolfe's method presented on page 397 of [2] is used. The program sets up the augmented coefficient tableau shown below, where the variables indicated correspond to those used by Wolfe.

$x\geq 0$	$y_1\geq 0$	$y_2\geq 0$	$v\geq 0$	$u_1\geq 0$	$u_2\geq 0$	$z\geq 0$	$w\geq 0$	
A_{11} $m\times n$	I	Φ					Φ	b_1 $m\times 1$
A_{21} $\ell\times n$	Φ	$-I$					I	b_2 $\ell\times p$
Q $n\times n$	Φ		$-I$	A_{11}' $n\times m$	$-A_{21}'$ $n\times\ell$	$\pm I$	Φ	P $n\times 1$

This tableau differs from that of Wolfe in that $-A_{21}'$ is used, instead of $+A_{21}'$, for the coefficient of u_2. This correction is necessary to handle constraints of the form $A_{21}x \geq b_2$ and is indicated by the Kuhn-Tucker theory as presented in Chapter 6 of Hadley[7]. The Q and P matrices are from the composite objective function and vary as the values of alpha change.

II. INPUT.

The input data consists of control information, the objective function coefficients and the constraint coefficients. A detailed description of each of the necessary input data cards follows.

TYPE 1. An identification card. Any information punched in columns 2 - 80 will be printed as the first line of output on each page. Typical information would be problem identification, date, users name, etc. If no identification is needed the card may be left blank. FORMAT(79H)

TYPE 2. Control information. Punched as fixed point numbers, right justified, in the columns indicated.

columns	
1 - 3	Number of independent variables
4 - 6	Number of constraints of the form $Ax \leq b$
7 - 9	Number of constraints of the form $Ax \geq b$
10 - 12	Number of objective functions
13 - 15	NPWR. The value $5\times10^{(NPWR-20)}$ is subtracted from the $z_j - c_j$ row to

avoid premature termination with w or z vectors in the basis. Normally these columns are blank.

FORMAT(24I3)

TYPE 3. The coefficients of the objective functions. Punched as floating point numbers (with decimal point) in columns of 10, with up to 7 pieces of data per card. Each objective function begins a new card. The order of the coefficients must be: constant, square terms, linear terms, cross product terms; i.e. c_0, x_1^2, x_2^2, $\cdots$ x_n^2, x_1, x_2, $\cdots$ x_1x_2, x_1x_3, $\cdots$ x_1x_n, x_2x_3, x_2x_4, $\cdots$ x_2x_n, $\cdots$ $x_{n-1}x_n$.
If any terms are missing either a zero should be punched or a blank left in the proper position.

FORMAT(7E10.2)

TYPE 4. Coefficients of the constraints. Punched as floating point numbers (with decimal point) in columns of 10, with up to 7 pieces of data per card. All constraints of the form $Ax \leq b$ must be first, followed by those of the form $Ax \geq b$. The order of the coefficients must be x_1, x_2 $\cdots$ x_n, b and each constraint equation begins a new card. If there are no constraints omit this type card. FORMAT(7E10.2)

TYPE 5. Alpha and control information. Punched as floating point numbers with the form depending on the number of objective functions. For one or two objective functions the form is:

columns	
1 - 10	starting alpha
11 - 20	stopping alpha
21 - 30	alpha increment

The starting alpha must be less than or equal to the stopping alpha and the increment must be positive. If only one value of alpha is to be used (instead of a sequence) that value should be punched as the starting alpha and columns 11 - 30 left blank. The remainder of this card is used for the control of output variations and is discussed later.

When there are more than two objective functions the form for this card is as follows:

columns		
1 - 10	alpha (1)	
11 - 20	alpha (2)	
21 - 30	alpha (3)	
31 - 40	alpha (4)	FORMAT(5E10.2,6E5.0)

TYPE 6. Control information. To begin an entirely new problem punch any negative number in columns 1 - 10. The next data card should then be a TYPE 1 card. To terminate the program punch any negative number in columns 1 - 10 and any number in columns 31 - 40. FORMAT(7E10.2)

III. OUTPUT.

A. The normal output consists of:

1. Identification from the TYPE 1 card.
2. Coefficients of the constraints and objective functions in the same order as entered on input cards.
3. Efficient point information: alpha, values of the objective functions, and values of x.

B. The following output variations are possible:

1. The identification may be surpressed by punching any number in columns 51 - 55 of the TYPE 5 card.
2. The coefficients of the constraints and objective functions may be surpressed by punching any number in columns 56 - 60 of the TYPE 5 card.
3. A new page of output may be started by punching any number in columns 61 - 65 of the TYPE 5 card.
4. To print the entire final tableau after each efficient point, punch any number in columns 66 - 70.

In each of the above "any number" is meant to imply a non-zero number.

IV. PROGRAM VARIATION.

In addition to locating efficient points for a specified value of alpha, or a sequence of values of alpha, the program will locate the efficient point with a specified value of one of the objective functions. This can only be done when there are two objective functions as the technique of halving the interval of alpha is used. If the desired value is not located after 20 steps of halving the interval the search is terminated and a message printed. To initiate a search of this type the following input card should be used as a TYPE 5 card.

columns

1 - 10 Either a ±1. or ±2., denoting which objective function is being used. The (+) value should be used if the objective function is an increasing function of alpha and the (-) if the objective function is a decreasing function of alpha.

11 - 20 The value of the objective function to be located.

21 - 30 A value of alpha which is known to be lower than the alpha of the desired point.

31 - 40 A value of alpha which is known to be higher than the alpha of the desired points.

Note: if <u>either</u> of the above values of alpha is not known, leave <u>both</u> values blank.

41 - 50 The allowable error in the value of the objective function to be located. This space may not be zero or blank.

51 - 70 Output variations as discussed in previous section.

V. EXAMPLE 1.

A. For the example of Chapter IV, section D,

Maximize:

$$z_1(x) = 370 - 5.3x_1^2 - 4.075x_2^2 + 19x_1 + 20x_2 - 4.2x_1x_2$$

$$z_2(x) = -176 - 9x_1^2 - 4.000x_2^2 + 144x_1 + 40x_2$$

Subject to:

$$\begin{aligned} 4x_1 + 5x_2 &\leq 36 \\ x_2 &\leq 2.25 \\ x_1 - x_2 &\geq 1 \end{aligned}$$

Determine:

1. efficient points for alpha from 0 to 1 in increments of 0.1.
2. the efficient point for alpha = 0.1234 and print the answers on a new page.
3. the efficient point for alpha = 0.432
4. efficient points for alpha from 0.75 to 0.85 in increments of 0.05.
5. the efficient point where $z_1(x) = 375 \pm .01$
6. the efficient point where $z_2(x) = 450 \pm .10$
7. the efficient point where $z_2(x) = 200 \pm 1.0$, which is known to be between $\alpha = .8$ and $\alpha = .9$.

8. the efficient point where $z_1(x) = 261.6 \pm .01$ and print the answer on a new page with the final tableau. Do not print coefficients.

9. efficient points for alpha from .002 to .003 in increments of 0.00025 and print the results on a new page without the indentification.

10. unconstrained efficient points for alpha from 0 to 1 in increments of 0.1.

Items 1 - 9 above may be worked as one problem with 9 different TYPE 5 cards. Item 10 is actually a new problem as the constraints are to be ignored.

B. Input data cards.

TYPE

```
[1]      EXAMPLE 1., CHAPTER IV, SECTION D               6/30/65
[2]       2  2  1  2
[3]       370.    -5.3      -4.075      19.       20.      -4.2
[3]      -176.    -9.       -4.000     144.       40.
[4]      4.        5.       36.
[4]                1.        2.25
[4]      1.       -1.        1.
[4]      0.        1.        .1
[5]      .1234                                              53   58   63
[5]      .432
[5]      .75       .85       .05
[5]      +1.       375.                           .01
[5]      -2.       450.                           .1
[5]      -2.       200.      .8         .9        1.
[5]      -2.       261.6                          .01            58   63      70
[5]      .002      .003      .00025                         53        63
[6]      -55.55
[1]      EXAMPLE 1., CHAPTER IV, SECTION D               6/30/65   UNCONSTRAINED POINTS
[2]       2  0  0  2
[3]       370.    -5.3      -4.075      19.       20.      -4.2
[3]      -176.    -9.       -4.000     144.       40.
[5]      0.0       1.0       0.1
[6]      -99.99                         -99.99
```

C. Program output.

```
EXAMPLE 1., CHAPTER IV, SECTION D                     6/30/65
CONSTRAINTS
 4.0000E+00  5.0000E+00  3.6000E+01
 0.0000E-99  1.0000E+00  2.2500E+00
 1.0000E+00 -1.0000E+00  1.0000E+00
Z1=     3.7000E+02 -5.3000E+00 -4.0750E+00  1.9000E+01  2.0000E+01 -4.2000E+00

Z2=    -1.7600E+02 -9.0000E+00 -4.0000E+00  1.4400E+02  4.0000E+01  0.0000E-99

   ALPHA         Z1              Z2                      X(I)
  0.0000   2.2485823E+02   4.4506575E+02   6.8373E+00  1.7301E+00
   .1000   2.3299672E+02   4.4462328E+02   6.6417E+00  1.8866E+00
   .2000   2.4183528E+02   4.4304308E+02   6.4190E+00  2.0647E+00
   .3000   2.5054963E+02   4.4018360E+02   6.1875E+00  2.2500E+00
   .4000   2.6094219E+02   4.3370211E+02   5.9986E+00  2.2500E+00
   .5000   2.9287126E+02   4.0744492E+02   5.3688E+00  2.2500E+00
   .6000   3.2585833E+02   3.6687253E+02   4.7184E+00  1.9912E+00
   .7000   3.5411901E+02   3.1422078E+02   4.0046E+00  1.7552E+00
   .8000   3.7662791E+02   2.4628099E+02   3.1870E+00  1.6368E+00
   .9000   3.9069653E+02   1.6846047E+02   2.4196E+00  1.4196E+00
  1.0000   3.9448528E+02   9.6634730E+01   1.8913E+00  8.9134E-01
```

[New Page]

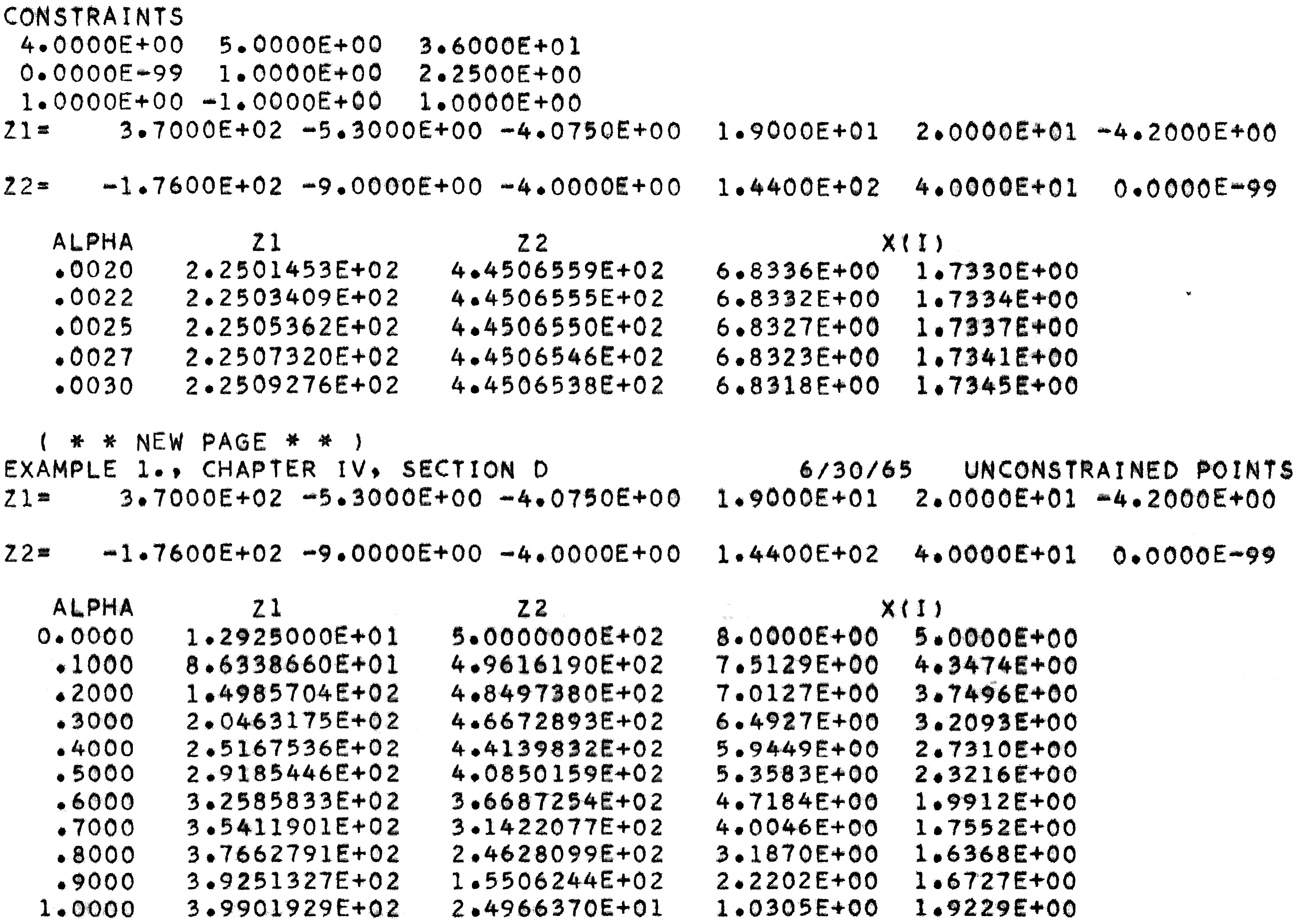
```
   ALPHA         Z1              Z2                      X(I)
   .1234   2.3500004E+02   4.4437115E+02   6.5922E+00  1.9262E+00
   .4320   2.7126114E+02   4.2634780E+02   5.8039E+00  2.2500E+00
   .7500   3.6611907E+02   2.8254365E+02   3.6112E+00  1.6794E+00
   .8000   3.7662791E+02   2.4628099E+02   3.1870E+00  1.6368E+00
   .8500   3.8550764E+02   2.0430830E+02   2.7260E+00  1.6324E+00
   .7917   3.7500100E+02   2.5262459E+02   3.2594E+00  1.6414E+00
CANNOT LOCATE Z2= 4.5000000E+02AL(1)= 0.000000  Z2=  4.4506574E+02
   .8546   3.8624892E+02   2.0002912E+02   2.6806E+00  1.6342E+00
```

Third page of computer output.

```
 EXAMPLE 1., CHAPTER IV, SECTION D                    6/30/65
   ALPHA        Z1               Z2                    X(I)
   .7797   3.7255300E+02   2.6160275E+02   3.3631E+00  1.6498E+00

  FINAL TABLEAU, NO.  3
 VIB      XB                              A VECTORS
   3    1.4298E+01   0.0000E-99   0.0000E-99   1.0000E+00   0.0000E-99
                   0.0000E-99   1.8175E-01   5.4266E-01  -3.4403E+00
                  -5.4266E-01  -3.6091E-01  -1.8175E-01  -5.4266E-01
                   1.4298E+01
   4    6.0015E-01   0.0000E-99   0.0000E-99   0.0000E-99   1.0000E+00
                   0.0000E-99  -3.6984E-02   1.3812E-01  -5.4266E-01
                  -1.3812E-01  -1.7510E-01   3.6984E-02  -1.3812E-01
                   6.0015E-01
   1    3.3631E+00   1.0000E+00   0.0000E-99   0.0000E-99   0.0000E-99
                   0.0000E-99  -9.1669E-02   3.6984E-02   1.8175E-01
                  -3.6984E-02  -1.2865E-01   9.1669E-02  -3.6984E-02
                   3.3631E+00
   5    7.1330E-01   0.0000E-99   0.0000E-99   0.0000E-99   0.0000E-99
                   1.0000E+00  -1.2865E-01   1.7510E-01  -3.6091E-01
                  -1.7510E-01  -3.0375E-01   1.2865E-01  -1.7510E-01
                   7.1330E-01
   2    1.6498E+00   0.0000E-99   1.0000E+00   0.0000E-99   0.0000E-99
                   0.0000E-99   3.6984E-02  -1.3812E-01   5.4266E-01
                   1.3812E-01   1.7510E-01  -3.6984E-02   1.3812E-01
                   1.6498E+00
   0    9.9999E-01   0.0000E-99   0.0000E-99   0.0000E-99   0.0000E-99
                   0.0000E-99   1.0000E-07  -1.0000E-07   1.0000E-07
                   1.0000E-07   1.0000E-07   9.9999E-01   1.0000E+00
                   9.9999E-01
```

Fourth and fifth pages of computer output.



```
CONSTRAINTS
 4.0000E+00  5.0000E+00  3.6000E+01
 0.0000E-99  1.0000E+00  2.2500E+00
 1.0000E+00 -1.0000E+00  1.0000E+00
Z1=     3.7000E+02 -5.3000E+00 -4.0750E+00  1.9000E+01  2.0000E+01 -4.2000E+00

Z2=    -1.7600E+02 -9.0000E+00 -4.0000E+00  1.4400E+02  4.0000E+01  0.0000E-99

   ALPHA       Z1              Z2                     X(I)
   .0020    2.2501453E+02   4.4506559E+02   6.8336E+00  1.7330E+00
   .0022    2.2503409E+02   4.4506555E+02   6.8332E+00  1.7334E+00
   .0025    2.2505362E+02   4.4506550E+02   6.8327E+00  1.7337E+00
   .0027    2.2507320E+02   4.4506546E+02   6.8323E+00  1.7341E+00
   .0030    2.2509276E+02   4.4506538E+02   6.8318E+00  1.7345E+00

  ( * * NEW PAGE * * )
EXAMPLE 1., CHAPTER IV, SECTION D                   6/30/65   UNCONSTRAINED POINTS
Z1=     3.7000E+02 -5.3000E+00 -4.0750E+00  1.9000E+01  2.0000E+01 -4.2000E+00

Z2=    -1.7600E+02 -9.0000E+00 -4.0000E+00  1.4400E+02  4.0000E+01  0.0000E-99

   ALPHA       Z1              Z2                     X(I)
  0.0000    1.2925000E+01   5.0000000E+02   8.0000E+00  5.0000E+00
   .1000    8.6338660E+01   4.9616190E+02   7.5129E+00  4.3474E+00
   .2000    1.4985704E+02   4.8497380E+02   7.0127E+00  3.7496E+00
   .3000    2.0463175E+02   4.6672893E+02   6.4927E+00  3.2093E+00
   .4000    2.5167536E+02   4.4139832E+02   5.9449E+00  2.7310E+00
   .5000    2.9185446E+02   4.0850159E+02   5.3583E+00  2.3216E+00
   .6000    3.2585833E+02   3.6687254E+02   4.7184E+00  1.9912E+00
   .7000    3.5411901E+02   3.1422077E+02   4.0046E+00  1.7552E+00
   .8000    3.7662791E+02   2.4628099E+02   3.1870E+00  1.6368E+00
   .9000    3.9251327E+02   1.5506244E+02   2.2202E+00  1.6727E+00
  1.0000    3.9901929E+02   2.4966370E+01   1.0305E+00  1.9229E+00
```

VI. EXAMPLE 2.

A. For the objective functions:

$$z_1(x) = 87 - x_1^2 - x_2^2 + 4x_1 + 6x_2$$

$$z_2(x) = 48 - x_1^2 - x_2^2 + 8x_1 + 12x_2$$

$$z_3(x) = 78 - x_1^2 - 2x_2^2 + 4x_1 + 12x_2$$

subject to:

$$x_1 + x_2 \leq 10$$

$$x_2 \leq 3.4$$

$$x_1 - x_2 \geq 0$$

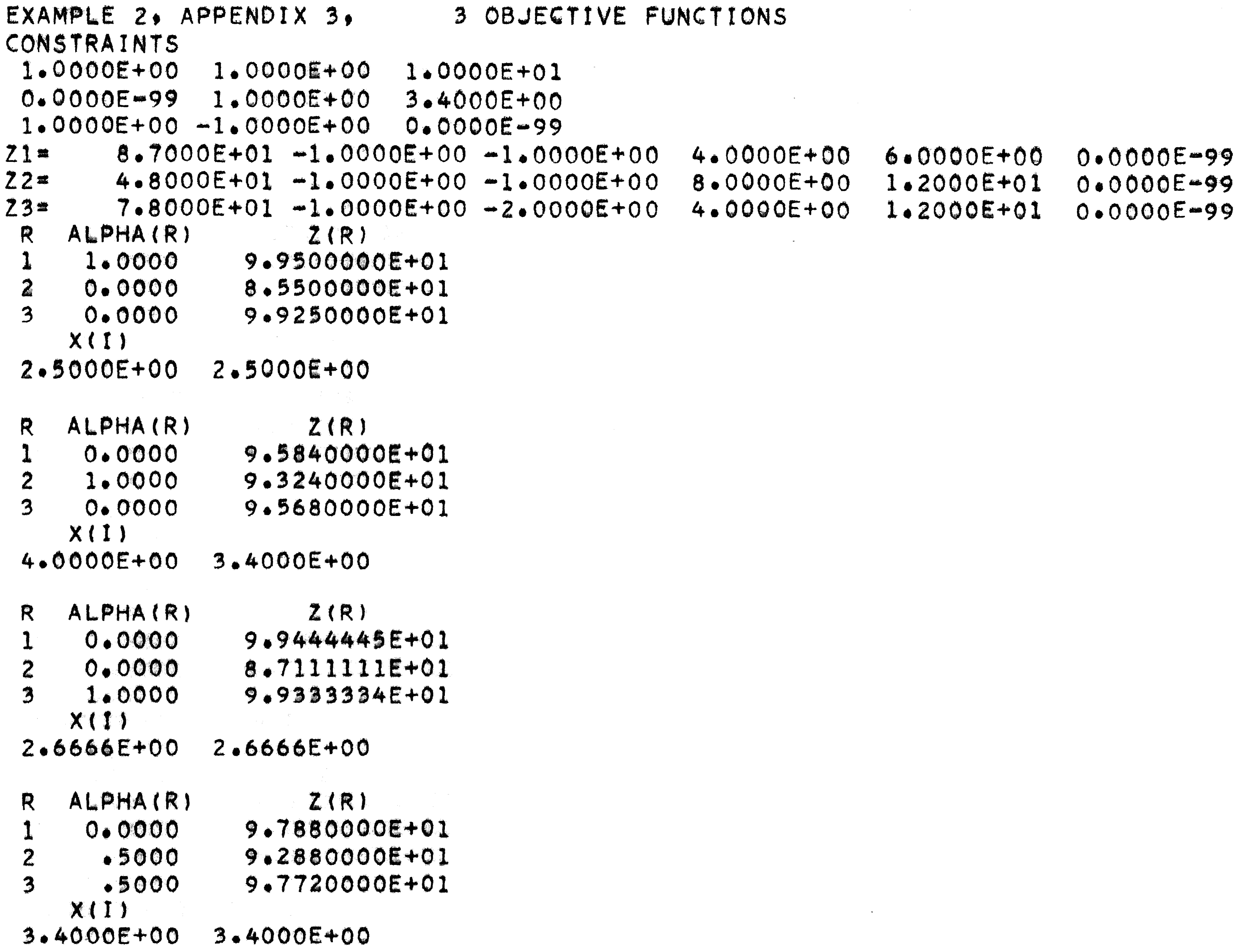
Determine efficient points for the following sets of alpha $[\alpha_1, \alpha_2, \alpha_3]$

[1, 0, 0]	[0, .5, .5]	[.6, .2, .2]
[0, 1, 0]	[.5, 0, .5]	[.2, .6, .2]
[0, 0, 1]	[.5, .5, 0]	[.2, .2, .6]

B. Computer input cards.

```
TYPE
[1]   EXAMPLE 2, APPENDIX 3,      3 OBJECTIVE FUNCTIONS
[2]    2  2  1  3
[3]   87.       -1.       -1.       4.        6.
[3]   48.       -1.       -1.       8.        12.
[3]   78.       -1.       -2.       4.        12.       0.
[4]   1.        1.        10.
[4]   0.        1.        3.4
[4]   1.0       -1.       0.
[5]    1.0      0.0       0.0
[5]    0.0      1.0       0.0
[5]    0.0      0.0       1.0
[5]    0.0      .5        .5
[5]    .5       0.0       .5
[5]    .5       .5        0.0
[5]    .6       .2        .2
[5]    .2       .6        .2
[5]    .2       .2        .6
[6]    -99.99                       -99.99
```

C. Computer output.



```
EXAMPLE 2, APPENDIX 3,      3 OBJECTIVE FUNCTIONS
CONSTRAINTS
 1.0000E+00  1.0000E+00  1.0000E+01
 0.0000E-99  1.0000E+00  3.4000E+00
 1.0000E+00 -1.0000E+00  0.0000E-99
Z1=     8.7000E+01 -1.0000E+00 -1.0000E+00  4.0000E+00  6.0000E+00  0.0000E-99
Z2=     4.8000E+01 -1.0000E+00 -1.0000E+00  8.0000E+00  1.2000E+01  0.0000E-99
Z3=     7.8000E+01 -1.0000E+00 -2.0000E+00  4.0000E+00  1.2000E+01  0.0000E-99
 R  ALPHA(R)        Z(R)
 1   1.0000     9.9500000E+01
 2   0.0000     8.5500000E+01
 3   0.0000     9.9250000E+01
    X(I)
 2.5000E+00  2.5000E+00

 R  ALPHA(R)        Z(R)
 1   0.0000     9.5840000E+01
 2   1.0000     9.3240000E+01
 3   0.0000     9.5680000E+01
    X(I)
 4.0000E+00  3.4000E+00

 R  ALPHA(R)        Z(R)
 1   0.0000     9.9444445E+01
 2   0.0000     8.7111111E+01
 3   1.0000     9.9333334E+01
    X(I)
 2.6666E+00  2.6666E+00

 R  ALPHA(R)        Z(R)
 1   0.0000     9.7880000E+01
 2    .5000     9.2880000E+01
 3    .5000     9.7720000E+01
    X(I)
 3.4000E+00  3.4000E+00
```

Computer output continued.

```
R   ALPHA(R)        Z(R)
1     .5000    9.9480000E+01
2    0.0000    8.6480000E+01
3     .5000    9.9320000E+01
    X(I)
2.6000E+00  2.6000E+00

R   ALPHA(R)        Z(R)
1     .5000    9.7880000E+01
2     .5000    9.2880000E+01
3    0.0000    9.7720000E+01
    X(I)
3.4000E+00  3.4000E+00

R   ALPHA(R)        Z(R)
1     .6000    9.9000000E+01
2     .2000    9.0000000E+01
3     .2000    9.9000000E+01
    X(I)
3.0000E+00  3.0000E+00

R   ALPHA(R)        Z(R)
1     .2000    9.7880000E+01
2     .6000    9.2880000E+01
3     .2000    9.7720000E+01
    X(I)
3.4000E+00  3.4000E+00

R   ALPHA(R)        Z(R)
1     .2000    9.9000000E+01
2     .2000    9.0000000E+01
3     .6000    9.9000000E+01
    X(I)
3.0000E+00  3.0000E+00
```

PROGRAM LISTING

```
C PROGRAM WJLQ8
      DIMENSION X(40),COB(4,66),A(10,10),Y(21,41),AL(4),B(10),
     1CAB(66),IB(10),IV(40),Z(4),C(41)
    1 READ 9001
    5 READ 9002,NV,ML,MG,NOBJ,NPWR
   10 NCB=(NV*(NV+3))/2
      NCBC=NCB+1
      MC=ML+MG
      JX=1
      JXN=NV
      JY=JXN+1
      JYN=JXN+MC
      JV=JYN+1
      JVN=JYN+NV
      JU=JVN +1
      JUN=JVN+MC
      JZ=JUN+1
      JZN=JUN+NV
      JW=JZN+1
      JWN=JZN+MG
      NX=JWN+1
      NM=JWN
      MN=MC+NV
      MZ=MN+1
      IF( JZN - JWN )14,12,12
   12 IPHZ=2
      JB =JZ
      JE=JZN
      GO TO 15
   14 IPHZ=1
      JB=JW
      JE=JYN
   15 N2V=2*NV
      NVM1= NV-1
      FLM=MN
      YJV=JYN
      YJV=YJV+.5
      AL(1)=1.0
      ALS=0.0
      LINE = 50
      PRT=0.0
      PRRT = 5.*(10.**(NPWR-20))
      DO 20 J=1,JZN
   20 C(J)=0.0
   24 DO   25  J=JB,JWN
   25 C(J)=-1.0
      C(NX) =0.0
   28 DO 30 NO=1,NOBJ
   30 READ 9003,COB(NO,NCBC),(COB(NO,L),L=1,NCB)
   35 IF(MC)50,50,40
   40 CONTINUE
```

```
      DO 45  I=1,MC
   45 READ 9003,(A(I,J),J=1,NV),B(I)
   50 IF(NOBJ-2)60,55,85
   55 IF(AL(1)-ALS)90,60,60
   60 READ 9015, AL(1),ALS,ALD,STOP,ERR,COMT,COEF,PAGE,PRIT
      KONT=0
      IF(PAGE)65,68,65
   65 LINE=50
   68 IF( ERR )800,70,800
   70 IF(AL(1))  75,95,95
   75 IF(STOP)80,1,80
   80 CALL EXIT
   85 READ 9015,(AL(I),I=1,4)
      STOP=AL(4)
      IF(AL(1))75,99,99
   90 AL(1)=AL(1)+ALD
   95 AL(2)=1.0-AL(1)
   99 IF(LINE-40)100,703,703
  100 DO     102  J=1,NM
  102 IV(J)=0
      IPHZ=1
      NX=JWN+1
      JB=JW
      IF(MC)140,140,101
C  SET UP Y, I, AND NULL MATRICIES
  101 DO 130  I=1,MC
      DO    103  J=1,NV
  103 Y(I,J)= A(I,J)
      Y(I,NX)= B(I)
      DO  104  J=JY,JWN
  104 Y(I,J)=0.0
      J1= JXN+I
      IF(  I - ML) 110,110,120
  110 Y(I,J1)=1.0
      IB(I)=J1
      IV(J1)=I
      GO TO 130
  120 Y(I,J1)=-1.0
      J2=JZN+I-ML
      Y(I,J2)=1.0
      IB(I)=J2
      IV(J2)=I
  130 CONTINUE
C   CALCULATE OBJECTIVE FUNCTIONS FOR THIS LOOP
  140 DO 145 L=1,NCB
      CAB(L)=0.0
      DO 145 NO=1,NOBJ
  145 CAB(L)=CAB(L)+AL(NO)*COB(NO,L)
```

```
C  SET UP Q,U,Z, MATRICIES
  200 K=N2V
      ITAB = 0
  210 DO 235 I=1,NVM1
      I1=I+MC
      I2=I+1
  220 DO 225 J=I2,NV
      K=K+1
      Y(I1,J)=-CAB(K)
      J2=J+MC
  225 Y(J2,I)=-CAB(K)
  235 Y(I1,I)=-2.0*CAB(I)
  240 Y(MN,NV)=-2.0*CAB(NV)
  250 DO 300 I=1,NV
      I1=I+MC
      DO 265 J=JY,JVN
  265 Y(I1,J)=0.0
      DO  266   J=JZ,JWN
  266 Y(I1,J)=0.0
      K=NV+I
      J1=JYN+I
      Y(I1,J1)  =  -1.0
      J1=JUN+I
      Y(I1,J1)  =   1.0
      IF( CAB(K)) 280,285,285
  280 DO  282 J=1,JUN
  282 Y(I1,J)=-Y(I1,J)
      Y(I1,NX)=-CAB(K)
      GO TO 288
  285 Y(I1,NX)=CAB(K)
  288 K1=JUN+I
      IB(I1)=K1
  300 IV(K1)=I1
      IB(MZ)=0
      IF(MC)320,320,310
  310 DO  315  I=1,MC
      J1=JVN+I
      DO  315  J=1,NV
      I1= MC+J
      IF(I-ML) 311,311,313
  311 Y(I1,J1)=A(I,J)
      GO TO 315
  313 Y(I1,J1)=-A(I,J)
  315 CONTINUE
C     ARE Z(S) IN BASIS
  320 DO 335 J=JZ,JZN
      IF(IV(J))330,335,330
  330 K=IV(J)
      IF(Y(K,NX))338,335,338
  335 CONTINUE
      GO TO 600
```

```
C      CALC ZJ-CJ
  338 DO 345 J=1,NX
      ZJ=0.0
      DO 340 I=1,MN
      K=IB(I)
  340 ZJ=ZJ+C(K)*Y(I,J)
  345 Y(MZ,J)=ZJ-C(J)
      IF(SENSE SWITCH 2 ) 8347,8349
 8347 PRINT 9017 , ITAB
      DO 8348  I=1,MZ
 8348 PRINT 9020,IB(I),(Y(I,J),J=1,NX)
 8349 CONTINUE
C      SELECT IN VECTOR
  350 ZCM=0.0
      JIN=0
      DO 425 J=1,JE
      TEST = Y(MZ,J)-ERR
      IF ( TEST)360,425,425
  360 IF(IV(J))365,365,425
  365 TJ=J
      TJ=TJ-YJV
      IF(ABSF(TJ)-FLM)370,400,400
  370 IF(TJ)375,375,380
  375 JT=J+MN
      GO TO 385
  380 JT=J-MN
  385 IF(IV(JT))400,400,425
  390 K=IV(JT)
      IF(Y(K,NX))425,400,425
  400 IF(ZCM-TEST    )425,425,405
  405 ZCM=TEST
      JIN=J
  425 CONTINUE
      IF(JIN)550,550,430
C   SELECT OUT VECTOR
  430 YM=1.E+45
      PRT=0.0
      DO 460 I=1,MN
      IF(Y(I,JIN))460,460,440
  440 IF(Y(I,NX))460,465,445
  445 XY=Y(I,NX)/Y(I,JIN)
      IF(YM-XY)460,460,450
  450 YM=XY
      IOUT=I
  460 CONTINUE
      IF(YM-1.E+44)500,800,800
  465 IOUT=I
C   ROW OPERATIONS
  500 PIV=Y(IOUT,JIN)
      ITAB=ITAB+1
      DO 505 J=1,NX
  505 Y(IOUT,J)=Y(IOUT,J)/PIV
```

```
      DO 530 I=1,MZ
      IF(Y(I,JIN))510,530,510
  510 IF(I-IOUT)515,530,515
  515 H=Y(I,JIN)
  520 DO 525 J=1,NX
  525 Y(I,J)=Y(I,J)-Y(IOUT,J)*H
  530 CONTINUE
      K=IB(IOUT)
      IV(K)=0
      IV(JIN)=IOUT
      IB(IOUT)=JIN
      IF(IPHZ -1 )531,531,320
  531 DO  532  J=JW,JWN
      IF(IV(J)) 338,532,338
  532 CONTINUE
C  PHASE ONE ENDED
  533 IPHZ=2
      JB=JZ
      JE=JZN
      DO  535  I=1,MN
  535 Y(I,JW)=Y(I,NX)
      NX=JW
      DO  537  J=JZ,JZN
  537 C(J)=-1.0
      GO TO 320
C  OPTIMUM INDICATED WITH Z(S) IN BASIS
  550 IF( PRT )558,555,558
  555 PRT = PRRT
      PRINT 9021,PRT
      GO TO 350
  558 IF(  IPHZ-1  )570,570,560
  560 PRINT 9016, ITAB
      GO TO 775
C OPTIMUM INDICATED WITH W(S) IN BASIS
  570 PRINT 9022 ,ITAB
      GO TO 775
C  OPTIMUM INDICATED
  600 CONTINUE
  604 DO 615 J=1,NM
      IF(IV(J))605,605,610
  605 X(J)=0.0
      GO TO 615
  610 K=IV(J)
      X(J)=Y(K,NX)
  615 CONTINUE
C  EVALUATE OBJECTIVE FUNCTIONS
  620 DO 650 NO=1,NOBJ
      Z(NO)=COB(NO,NCBC)+COB(NO,NV)*X(NV)*X(NV)+COB(NO,N2V)*X(N
      KIJ=N2V
  630 DO 650 I=1,NVM1
      K=NV+I
      G=X(I)
      Z(NO)=Z(NO)+COB(NO,I)*G*G+COB(NO,K)*G
  640 DO 650 J=I,NVM1
      KIJ=KIJ+1
      Z(NO)=Z(NO)+COB(NO,KIJ)*X(I)*X(J+1)
  650 CONTINUE
      IF( ERR )825,750,825
C  OUTPUT
C    HEADINGS
  703 PRINT 9019
      LINE=0
      IF(COMT)710,705,710
  705 PRINT 9001
  710 IF(COEF)725,714,725
  714 IF(MC)715,719,715
  715 PRINT 9018
      DO 716  I=1,MC
  716 PRINT 9023, (A(I,J), J=1,NV) , B(I)
  719 DO  720 NO=1,NOBJ
  720 PRINT 9004,NO,COB(NO,NCBC),(COB(NO,I),I=1,NCB)
      LINE=2+2*NCBC/7 +MC+LINE
  725 IF(NOBJ-2)730,740,100
  730 PRINT 9007
      GO TO 100
  740 PRINT 9005
      GO TO 100
C   ANSWERS
  750 IF(NOBJ-2)755,770,760
  755 PRINT 9008,Z(1),(X(I),I=1,NV)
      LINE = LINE +2 + NV/4
      GO TO 775
  760 PRINT 9009
      DO  765 NO=1,NOBJ
  765 PRINT 9010,NO,AL(NO),Z(NO)
      PRINT 9011,(X(I),I=1,NV)
      LINE = LINE +NO + 1 + NV/6
      GO TO 775
  770 PRINT 9006,AL(1),Z(1),Z(2),(X(I),I=1,NV)
      LINE = LINE +2  + NV/4
  775 IF(PRIT)780,50,780
  780 PRINT 9012,ITAB
      DO 790 I=1,MZ
  790 PRINT 9013,IB(I),Y(I,NX),(Y(I,J),J=1,NM)
      GO TO 50
  795 PRINT 9014
      GO TO 775
C  LOCATE A PARTICULAR VALUE OF AN OBJECTIVE FUNCTION
  800 NOB=ABSF(AL(1))
      VOB=ALS
      FIND=AL(1)
      ALOW=ALD
      ALHI=STOP
      ALS=0.0
      IF(  ALHI  )820,820,810
  810 AL(1)=ALOW
      ALIC =ALHI-ALOW
      GO TO 95
  820 AL(1)=.5
      ALIC=.5
      GO TO 95
  825 ZD=Z(NOB)-VOB
  860 IF( ABSF(ZD) - ERR )770,770,863
  863 KONT = KONT+1
      IF(KONT-20) 865,865,864
  864 PRINT 9027, NOB,VOB,AL(1),NOB,Z(NOB)
      GO TO  60
  865 ALIC=.5*ALIC
       ZD=FIND*ZD
      IF(  ZD  )870,870,875
  870 AL(1)= AL(1) + ALIC
      GO TO 95
  875 AL(1)= AL(1) -ALIC
      GO TO 95
 9000 FORMAT(1H1)
 9001 FORMAT(1X,79H
     1                                                                  )
 9002 FORMAT(24I3)
 9003 FORMAT(7E10.2)
 9004 FORMAT(1X,1HZ,I1,1H=,2X,6(1X,E11.4),/(7X,E11.4,1X,E11.4,
     1 1X,E11.4,1X,E11.4,1X,E11.4,1X,E11.4))
 9005 FORMAT(4X,5HALPHA,7X,2HZ1,14X,2HZ2,20X,4HX(I))
 9006 FORMAT(2X,F7.4,2(2X,E14.7),1X,3(1X,E11.4)/(43X,E11.4,
     11X,E11.4,1X,E11.4))
 9007 FORMAT(8X,1HZ,20X,4HX(I))
 9008 FORMAT(2X,E14.7,1X,4(1X,E11.4)/(19X,E11.4,1X,E11.4,
     11X,E11.4,1X,E11.4))
 9009 FORMAT(13H0 R  ALPHA(R),7X,4HZ(R))
 9010 FORMAT(I3,F9.4,3X,E14.7)
 9011 FORMAT(5X,4HX(I) ,/6(1X,E11.4))
 9012 FORMAT(21H0  FINAL TABLEAU, NO.I3,/5H  VIB,6X,2HXB25X,
     19HA VECTORS )
 9013 FORMAT(1X,I4,5(2X,E11.4),/(18X,E11.4,2X,E11.4,2X,E11.4,
     12X,E11.4))
 9014 FORMAT(10H UNBOUNDED )
 9015 FORMAT(5E10.2,6E5.0)
 9016 FORMAT( 25H OPT WITH Z IN BASIS, TAB ,I3)
 9017 FORMAT( 8H TABLEAU ,I3)
 9018 FORMAT( 12H CONSTRAINTS )
 9019 FORMAT(6H1WJLQ8)
 9020 FORMAT(I3,14F7.2,/3X,14F7.2,/3X,14F7.2,/3X,14F7.2)
 9021 FORMAT(5H ERR=,1X,E14.7)
 9022 FORMAT(25H OPT WITH W IN BASIS, TAB  ,I3 )
 9023 FORMAT(6(1X,E11.4))
 9027 FORMAT(16H CANNOT LOCATE Z,I1,1H=,   E14.7,8H  AL(1)=,
     1F9.6,3H  Z,I1,1H=,1X,E14.7)
      END
```

BIBLIOGRAPHY

1. Hadley, G. (1962) Linear Programming. Addison-Wesley Publishing Company, Inc., Reading, Mass.

2. Wolfe, P. (1959) "The Simplex Method for Quadratic Programming." Econometrica, Vol. 27, No. 3, pp. 382-398.

3. Wolfe, P. and Graves, R. L. (editors) (1963) Recent Advances in Mathematical Programming. McGraw-Hill Book Company, Inc., New York, pp. 67-86.

4. Karlin, S. (1959) Mathematical Methods and Theory in Games, Programming and Economics. Addison-Wesley Publishing Company, Inc., Reading, Mass.

5. Antle, C. E. (1962) "The N Response Problem." Ph.D. dissertation, Oklahoma State University, 54 p.

6. Umland, A. W. and Smith, W. N. (1959) "The Use of LaGrange Multipliers with Response Surfaces." Technometrics, Vol. 1, No. 3, pp. 289-292.

7. Hadley, G. (1964) Nonlinear and Dynamic Programming. Addison-Wesley Publishing Company, Inc., Reading, Mass.

8. Saaty, T. L. (1959) Mathematical Methods of Operations Research. McGraw-Hill Book Company, Inc., New York.

VITA

The author was born on December 28, 1937, in Ferguson, Missouri. He received his primary and secondary education from the public schools in Ferguson, Missouri and graduated from Ferguson High School in June, 1956.

He enrolled at the Missouri School of Mines and Metallurgy in September, 1956, and received a Bachelor of Science degree in electrical engineering on July 30, 1960. In January, 1961, he enrolled in the graduate school of the Missouri School of Mines and Metallurgy.

He has spent four summers as an employee of Aro, Inc. at the Arnold Engineering and Development Center, Tullahoma, Tennessee.

Since September, 1962, he has been an instructor in the Department of Mathematics of the University of Missouri at Rolla. He is married and has one child.