AN ALGORITHM FOR
GENERALISED CONVEX QUADRATIC PROGRAMMING

A Thesis

Presented in Partial Fulfillment

for the Requirements for the Degree of

Master of Agricultural Science

D. A. Evans

March 1965

ACKNOWLEDGEMENTS

The candidate wishes to record his appreciation to Professor W. V. Candler, whose advice, guidance, and stimulating discussion at all times made this study possible.

Further thanks are due to Professor A. L. Rae of the Sheep Husbandry Department in which the candidate was employed during the period of this study.

Grateful thanks are also extended to Miss M. Campbell and her staff for assistance in obtaining the necessary references.

INDEX

INTRODUCTION

The purpose of this thesis is to review work carried out by Professor W. V. Candler of the Department of Agricultural Economics and Farm Management at Massey University, leading to the development of a Generalised Convex Quadratic Programming Algorithm. However the responsibility for the following manner of presenting the material and forming the arguments rests with the candidate.

The first chapter gives a brief summary of the algebra of quadratic functions which will form a background for future developments. At the end of Chapter 1 is a bibliography for further and more detailed reading.

Chapter 2 classifies the problem in the title within the frame-work of the more general mathematical programming problem.

Chapter 3 describes and developes the mathematical conditions which any successful algorithm must satisfy, and Chapter 4 developes the algorithm, in the form of three separate algorithms, as a form of presentation. The last chapter provides a brief discussion.

CHAPTER 1

NOTATION AND DEFINITIONS

I.I. Quadratic Functions

I.I.I. A Quadratic Function

$$n_i = a_i + \underline{b_i}\underline{x}' + \underline{x}B_i\underline{x}' \quad (I)$$

defines a quadratic function where,

$\underline{x}$ is a Ixn vector of variables

a_i is a constant

$\underline{b_i}$ is a Ixn vector of constants, and

B_i is a nxn symmetric matrix of constants.

I.I.2. A Linear Function

If $B_i = 0$ then (I) reduces to a linear function of $\underline{x}$.

I.I.3. Partial Derivatives of a Quadratic Function

$$dn_{ij} = b_{ij} + \underline{2b_{ij}}\underline{x}' \quad (2)$$

is the partial derivative of n_i with respect to x_j where,

b_{ij} is the j^{th} element of $\underline{b_i}$, and

$\underline{b_{ij}}$ is the j^{th} row of B_i

Also defined is,

$$\underline{dn_i} = \underline{b_i'} + 2B_i\underline{x}' \quad (3)$$

where $\underline{dn_i}$ is a Ixn vector whose j^{th} element is dn_{ij}.

I.I.4. The Stationary Point of a Quadratic Function

$$\underline{dn}_i' = \underline{o}' \quad (4)$$

defines a set of simultaneous equations in $\underline{x}$, the solution of which yields the stationary point of n_i. If B_i is of rank n, then the stationary point of n_i is unique. If B_i is of rank n-r then the stationary point is an r+I dimensional hyperplane in n-space.

I.2. Properties of Quadratic Functions

I.2.I. The Quadratic Form

The Quadratic form is defined as $\underline{x}B_i\underline{x}'$. The following terminology applies to the quadratic form.

If for all $\underline{x}$; $\underline{x} \neq \underline{o}$,

$\underline{x}B_i\underline{x}' > o$ the quadratic form is Positive Definite

$\underline{x}B_i\underline{x}' \geq o$ the quadratic form is Positive Semidefinite

$\underline{x}B_i\underline{x}' < o$ the quadratic form is Negative Definite

$\underline{x}B_i\underline{x}' \leq o$ the quadratic form is Negative Semidefinite

$-oo \leq \underline{x}B_i\underline{x}' \leq \infty$ the quadratic form is Indefinite.

I.2.2. Latent Roots

$$\left| B_i - Iz_{ij} \right| = o \quad (5)$$

is the characteristic equation of B_i where,

I is the nxn identity matrix, and

z_{ij} is the j^{th} root of the polynomial (5).

z_{ij} is termed the j^{th} latent root of B_i, j = I.......n . As B_i is symmetric the latent roots are real.

I.2.3. Principle Minor Determinants

$$d_{ij} = \begin{vmatrix} B_{I2}B_{I2} & \cdots\cdots\cdots\cdots & B_{Ij} \\ B_{2I}B_{22} & \cdots\cdots\cdots\cdots & B_{2j} \\ \cdot & & \\ \cdot & & \\ B_{jI}B_{j2} & \cdots\cdots\cdots\cdots & B_{jj} \end{vmatrix}$$

is the j^{th} principle minor determinant of B_i, where the B_{pq} are the elements in the p^{th} row and q^{th} coloumn of B_i.

I.2.4. Determining Quadratic Form

The quadratic form of B_i is positive definite if,

(i) $z_{ij} > o$, $j = I.....n$, or alternatively,

(ii) $d_{ij} > o$, $j = I.....n$,

The quadratic form of B_i is negative definite if,

(i) $z_{ij} < o$, $j = I.....n$, or alternatively,

(ii) $d_{ij} = k$, $j = I.....n$, where,

$k < o$ if j is odd, and,

$k > o$ if j is even.

Semidefinite forms are indicated as above, with the strict inequalitiesreplaced by the weaker $\geq$ or $\leq$ conditions. Any other situation indicates an indefinite quadratic form.

I.2.5. Nature of the Stationary Point

If we define $\underline{x} = \underline{x}^{**} + \underline{x}^{*}$, where,

$\underline{x}^{**}$ is the solution to (4), and,

$\underline{x}^{*}$ is a vector $\underline{x}$ measured from $\underline{x}^{**}$,

then (I) becomes,

$$n_i = a_i + \underline{b_i}(\underline{x}^{**} + \underline{x}^*)' + (\underline{x}^{**} + \underline{x}^*)\, B_i\, (\underline{x}^{**} + x^*)' \quad (7)$$

$$= a_i^* + \underline{dn_i^*}\underline{x}^{*\prime} + \underline{x}^* B_i \underline{x}^{*\prime} \quad (8)$$

$$= a_i^* + \underline{x}^* B_i \underline{x}^{*\prime} \quad (9)$$

where, a_i^* is the value of n_i at the stationary point,

and,

$\underline{dn_i^*}$ is the value of $\underline{dn_i}$ evaluate at $\underline{x}^{**}$.

From section I.2.I. and equation (9) it follows that,

(i) when the quadratic form is positive definite (or semidefinite) the stationary point is a unique minumum point (or hyperplane).

(ii) when the quadratic form is negative definite (or semidefinite) the stationary point is a unique maximum point (or hyperplane).

(iii) when the quadratic form is indefinite the stationary point is a saddle-point.

I.2.6. The Differential

The linear approximation to n_i at the point $\underline{x}^*$ is measured by,

$$dn_i = \underline{dn_i^*}\underline{dx}' \quad (10)$$

where,

$\underline{dn_i^*}$ is $\underline{dn_i}$ evaluated at the point $\underline{x}^*$, and,

$\underline{dx}$ is a Ixn vector of small differential changes in $\underline{x}$ measured away from $\underline{x}^*$.

In geometric terms, dn_i is the tangent to n_i at the point $\underline{x}^*$.

I.2.7. Roots of the Simple Quadratic Function

If the vector $\underline{x}$ in (I) is replaced by the single variable x, then a simple quadratic function is obtained as

$$n_i = a_i + b_i x + B_i x^2 \qquad \text{(II)}$$

where, a_i, b_i, and B_i are all constants. The roots of (II) are the solutions to $n_i = o$, and are given by,

$$x_o = \frac{-b_i \pm \sqrt{b_i^2 - 4B_i a_i}}{2B_i} \qquad \text{(I2)}$$

Note that roots will be real only when,

$$b_i^2 \geqslant 4B_i a_i \qquad \text{(I3)}$$

I.3. Convexity

I.3.I. Convex Sets

S is a convex set if, for any two points $s_1 \in S$ and $s_2 \in S$,

$$(\phi s_1 + (1-\phi)s_2) \in S; \qquad 0 \leqslant \phi \leqslant 1 \qquad \text{(i4)}$$

I.3.2. Convex Functions

The function $f(\underline{x}) \in$ set Q is convex if for any two values of the function $f(\underline{x}_1) \in Q$ and $f(\underline{x}_2) \in Q$,

$$f(\phi \underline{x}_1 + (1-\phi)\underline{x}_2) \in Q; \qquad 0 \leqslant \phi \leq 1 \qquad \text{(15)}$$

In the following discussion Q will usually be the set $\geqslant$ or $\leqslant$ k, where k is an arbitrary constant.

I.3.3. Quadratic Convexity

Let Q be the set of real numbers $\leqslant k$, then if B_i has a positive definite (semidefinite) form, $n_i \in Q$ is a convex function, and the set of all $\underline{x}$ such that $n_i \in Q$ is a convex set.

Let Q be the set of real numbers $\geqslant$ k, then if B_i has a negative definite (semidefinite) form, $n_i \in Q$ is a convex function, and the set of all $\underline{x}$ such that $n_i \in Q$ is a convex set.

If B_i has an indefinite form, $n_i \in Q$ is a non-convex set for both of the above definitions of Q.

As the negative of a positive definite (semidefinite) quadratic form is negative definite (semidefinite) and vice versa, it follows that if $n_i \leq (\geqslant) k$ is a convex function, then $-n_i \geqslant (\leqslant) -k$, is also a convex function.

I.4. Bibliography

The above brief summaries may be supplemented by reading from the following sources.

(i) Hadley. G. - "Linear Programming," Addison-Wesley Pub. U.S.A. 1962.

(ii) Hadley, G. - "Nonlinear and Dynamic Programming," Addison-Wesley Pub. U.S.A. 1964.

(iii) Saaty.T.L. - "Mathematical Methods of Operations Research," McGraw-Hill Book Company, U.S.A. 1959.

(iv) Vajda. S. - "Mathematical Programming," Addison-Wesley Pub. U.S.A. 1961.

(v) Wolfe. P. - "Recent Developments in Nonlinear Programming," Rand Publication R-401-PR, 1962.

CHAPTER 2

CLASSIFICATION OF QUADRATIC PROGRAMMING PROBLEMS

2.1. Introduction

It is the intention in this section to use the previous nomenclature to give a concise classification of quadratic programming, within the framework of the General Mathematical Programming Problem.

The mathematical programming problem can be stated as (Wolfe, P., 1962), find a 1xn vector $\underline{x}$ such that,

$$f(\underline{x}) \quad \text{a max.} \tag{16}$$

subject to,

$$g_i\ (\underline{x}) \leqslant 0 \qquad ; \quad i = 1......m \tag{17}$$

Here $f(\underline{x})$ is the objective function and the $g_i(\underline{x})$ are the m constaint functions. The m constraint functions jointly determine the constraint set.

The non-negativity assumption ($\underline{x} \geqslant 0$) often made in the above statement, is a special case of (17).

The following terminology will be used. A particular value of $\underline{x}$, $\underline{x}^*$, will be a solution. A value of $\underline{x}^*$ satisfying (17) will be feasible solution. A value of $\underline{x}^*$ satisfying (16) and (17) will be a maximum feasible solution. A value of $\underline{x}^*$ satisfying (17), such that there exists no other feasible solution in the immediate neighbourhood of $\underline{x}^*$ with a higher value of $f(\underline{x})$ than $f(\underline{x}^*)$, will be a local optimum feasible solution, or more simply a local optimum.

If,

$$f(\underline{x}) \geq k, \text{ and,}$$

$$g_i(\underline{x}) \leq 0 \quad ; \quad i = 1.......m$$

are all convex functions, then problem (16) and (17) is a convex programming problem. This definition is made quite general for the minimization problem by noting that minimizing $f(\underline{x})$ is the same as maximizing $-f(\underline{x})$. In view of this generality, only the maximization case will be discussed.

Problem (16) and (17) will be termed a non-convex programming problem if (16) and (17) are not all convex functions.

2.2. The Convexity Assumption

In the past it has been convex programming problems that have received most attention, (Wolfe.P., 1962; Saaty.T.L., 1959 Hadley.G., 1964). The reasons for this are two-fold.

(i) For convex problems, any local optimum feasible solution is also the optimum feasible solution. That is, if $\underline{x}^*$ is a local optimum, then for any other constrained solution $\underline{x}^{**}$ we have,

$$g_i\ (\phi\underline{x}^* + (1-\phi)\underline{x}^{**}) \leq 0 \quad ; \quad i = 1.......m, \text{ and,}$$

$$f(\phi\underline{x}^* + (1-\phi)\underline{x}^{**}) \geq f(\underline{x}^{**})$$

This situation does not necessarily hold for non-convex problems.

(ii) Providing it is possible to make successive changes in $\underline{x}$ such that $f(\underline{x})$ is always being increased, while still remaining within the constraint set, then the optimum feasible solution will always be reached; where we preclude the possibility of converging on a point as a limit which is not the optimum feasible solution. Again this situation does not necessarily hold with non-convex problems.

2.3. Quadratic Programming Problems

Within the frame-work of the above definition of the general programming problem, we will discuss the case where $f(\underline{x})$ and $g_i(\underline{x})$; $i = 1 \ldots\ldots m$, are all quadratic functions. For purposes of generality, linear functions will be treated as the special case of a quadratic function with $B_i = 0$.

2.3.1. Convex Quadratic Problems

That a quadratic programming problem be convex implies,

(i) $f(\underline{x})$ is a negative definite (semidefinite) quadratic function, and,

(ii) $g_i(\underline{x})$; $i = 1 \ldots\ldots m$, are positive definite (semidefinite) quadratic functions. Linear functions may be considered as either negative or positive definite (semidefinite) quadratic functions.

Such a mathematical programming problem will be termed the Generalized Convex Quadratic Programming Problem, and will be the specific subject of this thesis.

The following special cases of the generalised convex quadratic programming problem may be recognized.

(a) If (16) and (17) are all linear the problem is one of Classical Linear Programming.

(b) If (17) are all linear and (16) is strictly negative definite (semidefinite), the problem is one of Classical Convex Quadratic Programming.

2.3.2. Non-convex Quadratic Programming

If $f(\underline{x})$ is not of negative definite form and/or at least one $g_i(\underline{x})$ is not of positive definite form, then the problem is one of Non-convex Quadratic Programming. For the special case where (16) is either of Indefinite or positive definite form and (17) are all linear functions, we have Classical Non-convex Quadratic Programming; (Candler.W. and Townsley.R.J., 1964). A more general attack on the non-cnnvex problem does not appear to exist at the present.

CHAPTER 3

CONDITIONS FOR A MAXIMUM FEASIBLE SOLUTION

3.1. Introduction

It is the intention in this section to briefly review the necessary and sufficient conditions for a boundary point $\underline{x}^*$ to be the maximum feasible solution to the general mathematical programming problem (16) and (17). We shall only be concerned with boundary point solutions, i.e. feasible solution such that $g_i(\underline{x}) = 0$ for at least one i. In the alternative case the maximum feasible solution is simply the unconstrained maximum of $f(\underline{x})$.

3.2. The Kuhn-Tucker Theorem

The follawing statement (Saaty.T.L., 1959) is a form of the Kuhn-Tucker theorem (Kuhn.H.W. and Tucker.A.W., 1951).

"A necessary condition that the boundary point $\underline{x}^*$ yield a maximum feasible solution to the general mathematical programming problem (16) and (17), with differentiable functions, is that there exist,

$$h_i \geqslant 0 \quad ; \quad i = 1.......m \tag{18}$$

$$y_j \geqslant 0 \quad ; \quad j = 1.......n \tag{19}$$

such that,

$$df(\underline{x}^*)' = \sum_i h_i \underline{d}g_i(\underline{x}^*)' - \underline{y}' \tag{20}$$

where,

$$\text{if } x_j^* > 0 \quad \text{then} \quad y_j = 0, \text{ and,} \tag{21}$$

$$\text{if } g_i(\underline{x}^*) < 0 \quad \text{then} \quad h_i = 0 \tag{22}$$

where,

$df(\underline{x}^*)$ is a 1xn vector whose j^{th} element is $df(\underline{x})/dx_j$ evaluated at $\underline{x}^*$, and,

$\underline{dg}_i(\underline{x}^*)$ is a 1xm vector whose j^{th} element is $dg_i(\underline{x})/dx_j$ evaluated at $\underline{x}^*$."

A proof of this theorem is given in Appendix I. The sufficient condition for this theorem is that the problem be convex.

3.3. The Generalised Convex Quadratic Programming Case

If (16) and (17) are all quadratic functions, and in particular if (16) has a negative definite quadratic form and (17) have all positive definite forms, then we saw the problem was one of generalised convex quadratic programming. This restricted problem can be formally stated as; find a 1xn vector $\underline{x}$ such that,

$$n_o = a_o + \underline{b}_o\underline{x}' + \underline{x}B_o\underline{x}' \quad \text{a max.} \tag{23}$$

subject to,

$$a_i + \underline{b}_i\underline{x} + \underline{x}B_i\underline{x}' \leqslant 0 \quad ; \quad 1 = 1.......m \tag{24}$$

where,

B_o is a negative definite (semidefinite) matrix, and,

B_i $i = 1.......m$, are all positive definite (semidefinite) matrices.

The non-negativity assumption, $\underline{x} \geqslant \underline{0}$, is taken to be included in the constraint set (24).

By introducting the 1xm non-negative vector of slack variables $\underline{n}$, the constraints (24) take the form,

$$a_i + \underline{b}_i\underline{x}' + \underline{x}B_i\underline{x}' + n_i = 0 \quad i = 1......m \tag{25}$$

where,

n_i is the i^{th} element of $\underline{n}$.

The general convex quadratic programming problem can now be restated as; find a lxn vector $\underline{x}$ such that,

$$n_o = a_o + \underline{b}_o\underline{x}' + \underline{x}B_o\underline{x}' \quad \text{a max.} \tag{26}$$

subject to,

$$n_i = a_i + \underline{b}_i\underline{x}' + \underline{x}B_i\underline{x}' \geqslant 0 \quad ; \quad i = 1.......m, \tag{27}$$

where,

B_1 $i = 0.......m$, are now negative definite (semi-definite) matrices.

Note that the statements of section 1.3.3. ensure that the constraints (27) are still convex.

As (26) and (27) is a convex problem with differentiable functions, the conditions for a maximum feasible solution on a boundary will be given by the Kuhn-Tucker theorem in the following form.

"A necessary and sufficient condition that the boundary point $\underline{x}^*$ yield a maximum feasible solution to the generalised convex quadratic programming problem (26) and (27), is that there exist,

$$\underline{h} \geqslant 0 \tag{28}$$

$$\underline{y} \geqslant 0 \tag{29}$$

such that,

$$\underline{dn}_o^{*\prime} = \sum_i h_i \underline{dn}_i^{*\prime} - \underline{y} \tag{30}$$

where,

$$\underline{x}^*\underline{y}' = 0, \text{ and,} \tag{31}$$

$$\underline{hn}' = 0 \tag{32}$$

where,

$\underline{n}$ is a 1xm vector of variables whose i^{th} element is h_i

$\underline{y}$ is a 1xn vector of variables, and,

$\underline{dn}_i^*$ is $\underline{dn}_i$ evaluated at $\underline{x}^*$; i = 0.......m."

3.4. Lagrangian Approach to a Boundary Point Solution

The following more intuitive development of the conditions for a boundary point maximum to be generalised convex quadratic programming problem by the classical Lagrange Multiplier method, leads essentially to the same results as the Kuhn-Tucker theorem, but gives a better insight into the interpretation of the quantities (28) to (32). Such an approach will also be useful in obtaining alternative statements of the problem.

As n_i, i = 0.......m, are differentiable functions, problem (26) and (27) may be expressed in classical lagrangian form as; find a 1xn vector $\underline{x}$ such that,

$$Q(\underline{x}) = n_o + \sum_i h_i \ (a_i + \underline{b}_i\underline{x}' + \underline{x}B_i\underline{x}' - n_i) \ \text{ a max} \tag{33}$$

where,

h_i is the lagrange multiplier associated with the constraint $n_i \geq 0$

Writing $\underline{x}^*$, $\underline{h}^*$, and $\underline{n}^*$ for the vectors which maximize (33) we note that,

(i) the maximization condition implies $\underline{h}^* \geqslant 0$, and,

(ii) the addition of the constraints $\underline{n}^* \geqslant 0$ implies

$$\underline{h}^*\underline{n}^{*\prime} = 0. \tag{34}$$

Taking the partial derivatives of (33) with respect to $\underline{x}$ at the point $\underline{x}^*$, we obtain,

$$\underline{dQ}(\underline{x}^*)' = \underline{dn}_0^{*\prime} + \sum_i h_i^* \underline{dn}_i^{*\prime} \tag{35}$$

The required condition for a maximum with respect to the partial derivatives (35) is that the j^{th} element of $\underline{dQ}(\underline{x}^*)$ should satisfy,

$$dQ(\underline{x}^*)_j \begin{cases} = 0 \text{ if } x_j^* > 0 \\ < 0 \text{ if } x_j^* = 0 \end{cases} \tag{36}$$

By introducing the lxn non-negative vector $\underline{y}$, (36) may be written as,

$$\underline{dQ}(\underline{x}^*)' = \underline{dn}_0' + \sum_i h_i^* \underline{dn}_i^{*\prime} + \underline{y}^{*\prime} = \underline{0} \tag{37}$$

$$\underline{x}^*\underline{y}^* = 0 \tag{38}$$

Taking the partial derivative of (33) with respect to the lxm vector of lagrange multipliers, and setting them to zero we obtain the set of conditions,

$$a_i + \underline{b}_i\underline{x}^{*\prime} + \underline{x}^*B_i\underline{x}^{*\prime} - n_i^* = 0 \;;\quad i = 1.......m \tag{39}$$

We have already seen that for the optimum plan we can add the conditions (34).

Gathering the above conditions, we obtain the following requirements for the solution $\underline{x}^*$ to represent a maximum feasible solution to the generalised convex quadratic programming problem.

Find 1xn vectors $\underline{x}$ and $\underline{y}$, and 1xm vectors $\underline{h}$ and $\underline{n}$, such that,

$$\underline{h} \geqslant 0 \qquad (40)$$

$$\underline{y} \geqslant 0 \qquad (41)$$

$$\underline{dn}_o + \sum_i h_i \underline{dn}_i' + \underline{y} = 0 \qquad (42)$$

$$\underline{xy}' = 0 \qquad (43)$$

$$\underline{hn}' = 0 \qquad (44)$$

$$a_i + \underline{b_i x}' + \underline{x} B_i \underline{x}' - n_i = 0 \quad ; \quad i = 1.......m \qquad (45)$$

$$\underline{n} \geqslant \underline{0} \qquad (46)$$

Conditions (40) to (44) are identical to the Kuhn-Tucker conditions (28) to (32), that a boundary point solution yield a maximum feasible solution. Conditions (45) and (46) above ensure that the solution is in fact feasible, this being assumed for any boundary point being considered by the Kuhn-Tucker theorem.

Note that in the case where $\underline{x}$ is not sign restricted, i.e. where the constraints (27) are not taken to include the non-negativity constraints $\underline{x} \geqslant \underline{0}$, (43) becomes,

$$x_j y_j = 0 \quad ; \quad j = 1.......n \qquad (47)$$

3.5. Comparison with Classical Convex Quadratic Programming

If the restraints (27) are all linear then the problem may be regarded as one of classical convex quadratic programming. In this case the equalities (45) are linear, and thus (42) and (45) form the constraints of a linear programming problem with no objective function. Thus the simplex method of linear programming may be used to solve the classical convex quadratic programming problem, where the iterative procedure is modified to take account of (43) and (44). This approach, originally formulated by Barankin.E.W. and Dorfman.R., 1955, has been the basis of several classical convex quadratic programming algorithms, (Frank.M. and Wolfe.P., 1956; Wolfe.P., 1959; Candler.W. and Evans.D.A., 1964), and a classical non-convex quadratic programming algorithm, (Candler.W. and Evans.D.A., 1964a).

The major difficulty then in the generalised case is the fact that equalities (45) are now quadratic. Fortunately, the differential of n_i is linear and it is to be expected that use will be made of this approximation in an attempt to obtain algorithms which fit largely into a simplex type of computation.

CHAPTER 4

GENERALISED CONVEX QUADRATIC PROGRAMMING

4.1. Introduction

We will now present different algebraic descriptions of the generalised convex quadratic programming problem (26) and (27). These descriptions will be orientated towards the different algorithms useful for the solution of this problem.

In the following description the non-negativity conditions $\underline{x} \geqslant \underline{0}$ will be stated explicitly in preference to including it in the constraints (27). Unless the constraint set is unbounded at $-\infty$ in some dimension, the problem can always be transformed to one in which $\underline{x} \geqslant \underline{0}$ holds.

4.2. Notation

The previous notation will be supplemented by the following.

$\underline{x}^*$ is a particular value of $\underline{x}$.

x_j^* is the j^{th} element of $\underline{x}^*$.

$\underline{dx}^*$ is a 1xn vector of differential changes in $\underline{x}$ measured from $\underline{x}^*$.

$\underline{dx}^{**}$ is a particular value of $\underline{dx}^*$.

$\underline{x}^{**} = \underline{x}^* + \underline{dx}^{**}$.

dn_{ij}^* is dn_{ij} evaluated at $\underline{x}^*$.

$\underline{dn}_i^*$ is $\underline{dn}_i$ evaluated at $\underline{x}^*$.

n_i^* is n_i evaluated at $\underline{x}^*$.

$\underline{x}^{*\phi} = \underline{x}^* + \phi\underline{dx}^{**}$; $\phi \geqslant 0$.

$n_i^{*\phi}$ is n_i evaluated at $\underline{x}^{*\phi}$.

4.3. Problem A

Let $\underline{x}^*$ be any solution, we may write,

$$\underline{x}^{**} = \underline{x}^* + \underline{dx}^* \tag{48}$$

Substituting (48) in (26) and (27) we obtain the new problem,

$$n_o = n_o^* + (\underline{b}_o + 2B_o\underline{x}^*)\ \underline{dx}^{*\prime} + \underline{dx}^*\ B_o\underline{dx}^{*\prime} \quad \text{a max.} \tag{49}$$

subject to,

$$n_i = n_i^* + (\underline{b}_i + 2B_i\underline{x}^{*\prime})\underline{dx}^{*\prime} + \underline{dx}^*B_i\underline{dx}^{*\prime} \geqslant 0;$$

$$i = 1....m \tag{50}$$

and,

$$\underline{dx}^* \geqslant -\underline{x}^* \tag{51}$$

The non-negativity constraints (51) ensure that if the solution $\underline{x}^*$ is feasible with respect to $\underline{x} \geqslant 0$, then $\underline{x}^{**} = \underline{x}^* + \underline{dx}^*$ will be non-negative also. Conseqeuntly it will be assumed that the initial solution $\underline{x}^*$ satisfies $\underline{x} \geqslant \underline{0}$. Note in particular that $\underline{x}^* = \underline{0}$ satisfies.

We now partition the subscripts i, into

i_+ if $n_i^* \geqslant 0$, and,

i_- if $n_i^* < 0$.

The intention now is to make successive changes in $\underline{dx}^*$ subject to (51), such that,

$$n_i > n_i^* \quad \text{if } i \in i_-, \text{ and,} \tag{52}$$

$$n_i \geqslant 0 \quad \text{if } i \in i_+, \text{ and,} \tag{53}$$

$$n_o > n_o^* \tag{54}$$

In general it will not be possible at every stage to make a choice of $\underline{dx}^*$ such that (52) to (54) always holds.

The constraints (50) may be written,

$$n_i = n_i^* + \underline{dn}_i\underline{dx}^{*\prime} + \underline{dx}^*B_i\underline{dx}^{*\prime} \geqslant 0 \quad ; \quad i = 1\ldots\ldots m \quad (55)$$

or,

$$\underline{dn}_i^*\underline{dx}^{*\prime} \geqslant -n_i^* - \underline{dx}^*B_i\underline{dx}^{*\prime} \quad ; \quad i = 1\ldots\ldots m \quad (56)$$

As $\underline{dn}_i^*\underline{dx}^{*\prime}$ has been defined as the differential of n_i at the point $\underline{x}^*$, the consequences of using the linear approximation to n_i as a constraint becomes apparent. This can be seen by letting $\underline{dx}^{**}$ be a solution to the problem, where the constraints (56) are replaced by their linear approximations,

$$\underline{dn}_i^*\underline{dx}^{*\prime} \geqslant - n_i^* \qquad i = 1\ldots\ldots m \quad (57)$$

The providing the constraints (56) are not inconsistent, $\underline{dx}^{**}$ can be chosen small enough such that (52) and (53) are satisfied. In practice we do not wish to restrict $\underline{dx}^{**}$ to be small differential values. Thus in general (52) and (53) will be satisfied by the choice $\underline{dx}^{**}$ if and only if,

$$\underline{dn}_i\underline{dx}^{**\prime} + \underline{dx}^{**}B_i\underline{dx}^{**\prime} > 0 \; ; \; i \in i_- \quad (58)$$

$$\underline{dn}_i\underline{dx}^{**\prime} + \underline{dx}^{**}B_i\underline{dx}^{**\prime} \geqslant -n_i^* \; ; \; i \in i_+ \quad (59)$$

It can be shown that conditions (58) and (59) may be inforced in the form of linear constraints, but this does not appear to be very useful.

Further, it is not necessary to observe (54) until $i \in i_+$ for $i = 1\ldots m$. However, <u>if</u> it is possible to increase the value of the objective function while attempting to attain feasibility then (54) should be observed. It will be shown that the problem can be formulated so that this is always the case.

As the constraints (57) have been shown to be insufficient to ensure that the conditions (52) and (53) are always met, it remains to show whether (57) is useful at all. We will now develope the property of the constraint set (57) upon which the algorithm depends.

Consider the vector $\underline{dx}^{**}$ to be the point on $n_i = 0$ which is also tangential to dn_i^* and proximal (by Euclidian distance) to any solution $\underline{x}^* \geqslant 0$. Then,

$$n_i^* + \underline{dn_i^*dx}^{**\prime} + \underline{dx}^{**}B_i\underline{dx}^{**\prime} = 0 \tag{60}$$

$$\therefore \quad \underline{dn_i^*dx}^{**\prime} = -n_i^* - \underline{dx}^{**}B_i\underline{dx}^{**\prime} \tag{61}$$

$$\therefore \quad \underline{dn_i^*dx}^{**\prime} \geqslant - n_i^* \tag{62}$$

The importance of (62) is that it validly allows the use of dn_i^* as a cutting plane constraint. If $\underline{dn_i^*dx}^{*\prime} = k$ is the hyperplane which passes through $\underline{dx}^{**}$, then (62) shows that $k \geqslant - n_i^*$. In particular we have,

$$k \geqslant - n_i^* \geqslant 0 \qquad i \in i_- \tag{63\_}$$

$$0 \geqslant k \geqslant - n_i^* \qquad i \in i_+ \tag{64}$$

The constraint,

$$\underline{dn_i^*dx}^{*\prime} \geqslant - n_i^* \tag{65}$$

then has the following properties.

(i) If $i \in i_-$ the right hand side of (65) is necessarily positive, and the constraint thus eliminates the infeasible point $\underline{x}^*$. However, the convexity of $n_i \geqslant 0$ and the inequality (63) ensures that (65) can eliminate no space for which $n_i \geqslant 0$.

(ii) If $i \in i_+$ the right hand side of (65) is necessarily negative or zero, and the constraint thus includes the feasible point $\underline{x}^*$. Again the convexity of $n_i \geqslant 0$ and the inequality (64) ensures that (65) can eliminate no space for which $n_i \geqslant 0$.

It is now easy to show that the joint constraint set (57) can exclude no mutually feasible region with respect to $n_i \geqslant 0$, $i = 1.....m$. That is, any feasible solution is by definition feasible with respect to $n_i \geqslant 0$, for all i. Consequently it can not be excluded by (65) for all i.

In summary then the constraints (57) will always exclude a solution $\underline{x}^*$ providing the set i_- is not empty, but will never exclude a feasible solution or the feasible set of solutions.

4.3.1. Algorithm A

Given any solution $\underline{x}^* \geqslant 0$, find a lxn vector $\underline{dx}^*$ such that,

$$n_o = n_o^* + \underline{dn_o^* dx^{*\prime}} + \underline{dx}^* B_o \underline{dx}^{*\prime} \quad \text{a max} \tag{66}$$

subject to,

$$dn_i^* \geqslant -n_i^* \qquad i = 1.......m \tag{67}$$

and,

$$\underline{dx}^* \geqslant - \underline{x}^* \tag{68}$$

Let the solution to this problem be the vector $\underline{dx}^{**}$. We then have the new solution $\underline{x}^{**} = \underline{x}^* + \underline{dx}^{**}$ to the original generalized convex quadratic programming problem (26) and (27). The quantities in (66) to (68) may then be re-written in terms of the new solution $\underline{x}^{**}$ and a new problem obtained. The constraints (67) of the new problem will be supplemented by the constraints (67) of the old problem which acted as effective restrictions.

The generalized convex quadratic programming problem will be solved when the solution to problem (66) to (68) is such that,

(i) $n_i \geqslant 0, \quad i = 1 \ldots\ldots m$ (69)

(ii) Every effective restriction is associated with a value of n_i for that restriction as close to zero as is a satisfactory approximation. (70)

4.3.2. Discussion on Algorithm A

The successive problems (66) to (68) are classical convex quadratic programming problems and as such algorithms are available for their solution. If a simplex-like algorithm such as that of Candler and Evans 1964a, is used for their solution, we are faced with a simplex matrix of dimension (n + 2m + k) x (n + m), where k is the number of effective restrictions from the previous problem. The size of this matrix may be reduced by rewriting (66) to (68) in the following way.

Let,

$$\underline{x} = \underline{x}^* + \underline{dx}^* \tag{71}$$

or,

$$\underline{dx}^* = \underline{x} - \underline{x}^* \tag{72}$$

Substituting (72) in (66) to (68) yields the new problems, find a 1xn vector $\underline{x}$ such that,

$$n_o = a_o + \underline{b}_o\underline{x}' + \underline{x}B_o\underline{x} \qquad \text{a max.} \tag{73}$$

subject to,

$$\underline{dn}_i^*\underline{x}' \geqslant \underline{dn}_i^*\underline{x}^{*\prime} - n_i^* \qquad i = 1 \ldots\ldots m \tag{74}$$

and,

$$\underline{x} \geqslant \underline{0} \tag{75}$$

These problems can each be solved by the above method where the dimension of the simplex matrix is now $(n + m + k) \times (n + m)$; $k \leq m$. Again the constraints which acted as effective restrictions in the previous problem are carried over into the new problem to supplement (74).

Convergence rests on the fact that as $\underline{dx}^{**}$ becomes small, then $\underline{dx}^{**} B_i \underline{dx}^{**'}$ becomes even smaller. Thus the linear approximations (67) more closely approximate the quadratic constraints. It is evident from the results of (62) that the accumulation of <u>all</u> of the generated constraints (67) into successive problems will ensure convergence. The impracticable nature of such a scheme is evident. By carrying over only those constraints which acted as effective restrictions in the previous problem we are essentially looking for successive sets of effective constraints, eliminating old solutions as we go, until a set is obtained which satisfies as a optimum feasible solution. The criteria (69) and (70) used to indicate optimality, is essentially the Kühn-Tucker conditions, $\underline{hn}' = 0$. The remaining conditions (28) to (31) are satisfied for <u>all</u> of the optimum solutions to the problems generated by (66) to (68).

An inconsistant set of constraints (67) either with or without the carried over constraints immediately indicates that the quadratic constraints (27) are inconsistent.

If the above-mentioned algorithm of Candler and Evans is used to solve the successive problems (66) to (68) the following may be noted.

(i) The optimum solution yields "shadow prices" of scarce resources and unused activities along with the solution.

(ii) In the special case where $B_i = 0$, $i = 1....m$, the algorithm reduces to one of classical quadratic programming.

(iii) In the special case where $B_i = 0$, $i = 0, 1....m$, the algorithm reduces to one of classical linear programming, with a slightly modified simplex format.

(iv) In the special case where $B_0 = 0$, the successive problems (66) to (68) may yield unbounded solutions, and hence arbitrary constraints of the form $\sum_i x_i = k$ may be required.

(v) The objective function may be of negative definite or of negative semidefinite form. In the latter case the possibility of unbounded solutions to (66) to (68) again arises and an arbitrary constraint may be carried.

4.4. Problem B

In this section we will start with a rather different approach than that taken in Problem A, but will show that it reduces to essentially the same algorithm.

Given any solution $\underline{x}^* \geqslant \underline{0}$, we start by writing the generalized convex quadratic programming problem in the form (49) to (51). For the objective function and each constraint we define a differential vector of maximum ascent,

$$\underline{dx}_i^* = \underline{dn}_i^* \quad ; \quad i = 0, 1,......m \tag{76}$$

We may now define each quadratic function in terms of the maximum ascent vector $\underline{dx}_k^*$.

$$n_i^{**} = n_i^* + \underline{dn}_i^* \underline{dx}_k^{*\prime} + \underline{dx}_k^* B_i \underline{dx}_k^{*\prime} \quad ; \quad i = 0, 1,.....m \tag{77}$$

As we do not wish to restrict the maximum ascent vectors to be differential changes in $\underline{x}$, we will redefine them as,

$$\underline{dx}^*_i = \underline{dn}^*_i\phi, \ \phi \geq 0 \quad ; i = 0, 1, \ldots m \tag{78}$$

For each quadratic function wirtten in terms of $\underline{dx}^*_k$ (77) becomes,

$$n^{*\phi}_i = n^*_i + \phi\underline{dn}^*_i\underline{dn}^{*\prime}_k + \phi^2\underline{dn}^*_k B_i \underline{dn}^{*\prime}_k \ ; i = 0, 1, \ldots m \tag{79}$$

or

$$n^{*\phi}_i = n^*_i + \phi b^*_{ik} + \phi^2 B^*_{ik} \quad ; i, k = 0, 1, \ldots m \tag{80}$$

where,

$$b^*_{ik} = \underline{dn}^*_i\underline{dn}^{*\prime}_k \quad ; i, k = 0, 1, \ldots m \tag{81}$$

and,

$$B^*_{ik} = \underline{dn}^*_k B \underline{dn}^{*\prime}_k \quad ; i, k - 0, 1, \ldots m \tag{82}$$

Again note that (80) gives the value of n_i when we move a distance ϕ in the direction of $\underline{dn}^*_k$ from the point $\underline{x}^*$.

For a fixed value of ϕ we have a problem of choice as to which maximum ascent vector to use. Algebraically we may define the vector to be used as a weighted sum of all $m+1$ maximum ascent vectors.

$$\underline{dx}^*_w = \underline{w}D^*\phi \tag{83}$$

where,

$\underline{w}$ is a $1x\ (m + 1)$ non-negative vector whose i^{th} element is w_i.

and,

D^* is a $(m + 1)\ xn$ matrix whose i^{th} row is $\underline{dn}^*_i$; $i = 0, 1, \ldots m$.

Corrresponding to (79) and (80) we have,

$$n^{*\phi}_{iw} = n^{*}_{i} + \phi \underline{dn}^{*}_{i}D^{*\prime}\underline{w}' + \phi^{2}\underline{w}D^{*}B_{i}D^{*\prime}\underline{w}' ; \quad i = 0, 1, \ldots m \quad (84)$$

or,

$$n^{*\phi}_{iw} = n^{*}_{i} + \phi \underline{b}^{*}_{iw}\underline{w}' + \phi^{2}\underline{w}B^{*}_{iw}\underline{w}' \quad ; \quad i = 0, 1, \ldots m \quad (85)$$

where,

$$\underline{b}^{*}_{iw} = \underline{dn}^{*}_{i}D^{*\prime} \quad ; \quad i = 0, 1, \ldots m \quad (86)$$

and,

$$B^{*}_{iw} = D^{*}B_{i}D^{*\prime} \quad ; \quad i = 0, 1, \ldots m \quad (87)$$

We now propose solving the following problem. Find a 1x (m + 1) vector $\underline{w}$ such that,

$$n^{*\phi}_{ow} = n^{*}_{o} + \phi \underline{b}^{*}_{ow}\underline{w}' + \phi^{2}\underline{w}B^{*}_{ow}\underline{w}' \quad \text{a max.} \quad (88)$$

subject to,

$$\phi \, \underline{b}^{*}_{iw}\underline{w}' \geqslant - n^{*}_{i} \quad ; \quad i = 1 \ldots\ldots m \quad (89)$$

and,

$$\phi \underline{w}D^{*} \geqslant - \underline{x}^{*} \quad (90)$$

$$\underline{w} \geqslant \underline{0} \quad (91)$$

Suppose we restrict ϕ to the convenient value of $\phi = 1$, and then obtain an optimum feasible solution to problem (88) to (91). Corresponding to this solution $\underline{w}^{*}$ we have from (83) the weighted maximum ascent vector,

$$\underline{dx}^{**}_{w} = \underline{w}^{*}D^{*}\phi \quad ; \quad \phi = 1 \quad (92)$$

Substituting $\underline{w}^{*}$ in (85) we obtain,

$$n^{**\phi}_{iw} = n^{*}_{i} + \phi b^{**}_{iw} + \phi^{2}B^{**}_{iw} \quad ; \quad i = 0, 1, \ldots m \quad (93)$$

where,

$$b^{**}_{iw} = \underline{b}^{*}_{iw}\underline{w}^{*\prime} \quad (94)$$

and,

$$B^{**}_{iw} = B^{*}_{iw}\underline{w}^{*\prime} \tag{95}$$

Having evaluated (93) for all i, we can allow ϕ to vary and calculate the maximum value of ϕ consistant with retaining $\underline{x} \geqslant \underline{0}$, and $n_i \geqslant 0$, for the constraints for which this has been achieved. In particular we define,

$$\phi_o = \frac{-b^{**}_{ow}}{2B^{**}_{ow}} \tag{96}$$

$$\phi_i = \max_{\phi}(n^*_i = -b^{**}_{iw}\phi - B^{**}_{iw}\phi^2) \quad ; \; b^{**2}_{iw} - 4B^{**}_{iw}n^*_i \geqslant 0 \tag{97}$$

$$\phi_i = \frac{-b^{**}_{iw}}{2B^{**}_{iw}} \quad ; \; b^{**2}_{iw} - 4B^{**}_{iw}n^*_i < 0 \tag{98}$$

$$\phi_j = \frac{-x^*_j}{dx^{**}_j} \quad ; \; dx^{**}_j < 0 \tag{99}$$

$$\phi_j = \infty \quad ; \; dx^{**}_j \geqslant 0 \tag{100}$$

The optimum value of ϕ, ϕ^* is given by,

$$\phi^* = \min_{i,\, j}.\ (\phi_i, \phi_j) \tag{101}$$

The minimization is over i = 0, 1,.....m if $\underline{x}^*$ is feasible, and is over i = 1.......m if $\underline{x}^*$ is infeasible.

Note that constraints (89) ensure that $b^{**}_{iw} > 0$, if $n^*_i < 0$. Also by writing B^{**}_{iw} in the form $\underline{dx}^{**}_w B_i \underline{dx}^{**\prime}_w$, we see that $B^{**}_{iw} \leqslant 0$, i = 0, 1,...m. Conseqeuntly $\phi^* \geqslant 0$.

Having obtained ϕ^* we can insert the value in (92) to obtain the vector $\underline{dx}^{**}_w$. Letting $\underline{x}^{**} = \underline{x}^* + \underline{dx}^{**}_w$, we have a new solution to the generalized problem. Further, the constraints (90) and (99) ensure that the new solution $\underline{x}^{**}$ is non-negative. Associated with this new solution we have

$$n_i^{**} > n_i^* \qquad ; n_i^* < 0 \qquad (102)$$

$$n_i^{**} \geqslant 0 \qquad ; n_i^* \geqslant 0 \qquad (103)$$

The necessary quantities are recalculated in terms of the new solution and a new problem (88) to (91) obtained. It should be noted that the quadratic form in (88) may not be negative definite, in fact it will more often be indefinite. Consequently one of the non-convex quadratic programming algorithms previously mentioned may be used for the solution of the generated problems (88) to (91), although it will shortly be shown thatthis is not necessary.

If at any stage $\underline{x}^*$ is feasible and their exists no weights $\underline{w}$ other than $\underline{w}^* = 0$ as a solution to problem (88) to (91), then $\underline{x}^*$ is the optimum feasible solution to the problem (26) and (27). If $\underline{x}^*$ is infeasible and the constraints (89) to (91) have no consistant solution, then the constraints of the original problem are inconsistant.

We will now show that the solution $\underline{dx}_w^{**} = \underline{w}^* D^* \phi$; $\phi = 1$, obtained by solving (88) to (91) is same solution as that obtained from solving problem (66) to (68). If we substitute (83) in (84) then problem (88) to (91) becomes; find a lxn vector $\underline{dx}_w^*$ such that,

$$n_{ow}^{**} = n_o^* + \phi \underline{dn}_o^* \underline{dx}_w^{*\prime} + \phi^2 \underline{dx}_w^* B_o \underline{dx}_w^{*\prime} \quad \text{a max.} \qquad (104)$$

subject to,

$$\phi \underline{dn}_i^* \underline{dx}_w^{*\prime} \geqslant - n_i^* \qquad ; i = 1.......m \qquad (105)$$

and,

$$\phi \underline{dx}^*_w \geqslant - \underline{x}^* \tag{106}$$

If we set $\phi = 1$, then problem (104) to (106) is identical to problem (66) to (68). Also as $\underline{dx}^*_w$ lies in the convex cone generated by the columns of $D^{*'}$ it is evident that (91) is satisfied.

4.4.1. Algorithm B

Find a 1xn vector $\underline{x}$ such that,

$$n_o = a_o + \underline{b}_o\underline{x}' + \underline{x}B_o\underline{x}' \qquad \text{a max.} \tag{107}$$

subject to,

$$\underline{dn}^*_i\underline{x}' \geqslant \underline{dn}^*_i\underline{x}^{*'} - n^*_i \qquad ; i = 1.......m \tag{108}$$

and,

$$\underline{x} \geqslant \underline{0} \tag{109}$$

where $\underline{x}^* \geqslant \underline{0}$ is any solution. Denote the solution to this problem as $\underline{x}^{**}$ and evaluate,

$$\underline{dx}^{**} = \underline{x}^{**} - \underline{x}^* \tag{110}$$

$$b^{**}_i = \underline{dn}^*_i\underline{dx}^{**'} \qquad ; i = 0, 1,.....m \tag{111}$$

$$B^{**}_i = \underline{dn}^{**}B_i\underline{dx}^{**'} \qquad ; i = 0, 1, ...m \tag{112}$$

$$\phi_o = \frac{-b^{**}_o}{2B^{**}_o} \tag{113}$$

$$\phi_i = \max_{\phi}(n^*_i = -\phi b^{**}_i - \phi^2 B^{**}_i) \; ; b^{**2}_i - 4B^{**}_i n^*_i \geqslant 0 \tag{114}$$

$$\phi = \frac{-b^{**}_i}{2B^{**}_i} \qquad ; b^{**2}_i - 4B^{**}_i n^*_i < 0 \tag{115}$$

$$\phi_j = \frac{-x^*_j}{dx^{**}_j} \qquad ; \; dx^{**}_j < 0 \qquad (116)$$

$$\phi_j = \infty \qquad ; \; dx^{**}_j \geqslant 0 \qquad (117)$$

$$\phi^* = \min_{i,\, j} (\phi_i, \phi_j) \qquad ; \qquad (118)$$

where the minimization is over i = 0, 1,....m if $\underline{x}^*$ is feasible and over i = 1......m if $\underline{x}^*$ is infeasible.

$$\underline{x}^{**}_{\phi} = \underline{x}^* + \phi^* \underline{dx}^{**} \qquad (119)$$

The solution $\underline{x}^{**}_{\phi}$ is a new and better solution to the original generalized convex quadratic programming problem (26) and (27). The quantities in (108) are reobtained in terms of this new solution and a new problem is obtained. If at any stage it is not possible to obtain a feasible solution to the current problem (107) to (109) then the constraints (27) are inconsistent. The original problem (26) and (27) is solved when $\underline{x}^{**} = \underline{x}^*$. Note that $\underline{x}^{**} \neq \underline{x}^*$ if $\underline{x}^*$ is not feasible with respect to (27) as the constraints (108) will exclude $\underline{x}^*$ in such a case.

4.4.2. Discussion in Algorithm B

The strength of this algorithm is that if a feasible solution to the constraints of the original problem exists, then it is always possible to make successive changes in $\underline{x}$ such that n^*_i is increased if it is negative but is maintained $\geqslant 0$ if it becomes $\geqslant 0$. No such statement can be made for algorithm A where the constraints (108) are used to <u>eventually</u> converge on boundary point optimum feasible solution. Also a proof of convergence for algorithm A is lacking. In the case of algorithm B however, we have seen that any solution to (107) to (109) must satisfy $b_i^{**} > 0$, if $\underline{x}^*$ is infeasible with respect to $n_i \geqslant 0$. Consequently we can choose ϕ^* to make some improvement in all restrictions which have $n_i^* < 0$. As with algorithm A, an inconsistent set of constraints (108) immediately indicates that there exists no feasible solution to the original problem, as we saw in section 4.3. that the constraints (108) can never eliminate any mutually feasible area. Conversely, if a feasible solution to (27) exists, then a feasible solution to (108) exists which yields $b_i^{**} > 0$ for $n_i^* < 0$. Thus algorithm B may be developed with out reference to the vector $\underline{w}$, although it is instructive to obtain this particular interpretation.

The carrying over of effective constraints as suggested in algorithm A could also be used with algorithm B. Again, unbounded solutions to problem (107) to (109) can be contained with a parametric constraint of the form $\sum_j x_j = k$, which may occur if $\mathbf{B}_o$ is negative semidefinite.

4.5. Problem C

Given any solution $\underline{x}^* \geqslant 0$ to the generalised convex quadratic programming problem, we may partition $\underline{x}^*$ into,

$$\underline{x}_o^* = \underline{0}, \text{ the } j^{th} \text{ element of which is } x_{j_o}^* \qquad (120)$$

and,

$$\underline{x}_+^* > \underline{0}, \text{ the } j^{th} \text{ element of which is } x_{j_+}^* \qquad (121)$$

The Kuhn-Tucker conditions (41) and (43) reduce to the requirement that the following conditions must satisfy.

$$\underline{y}_o > \underline{0} \tag{122}$$

$$\underline{y}_+ = \underline{0} \tag{123}$$

where $\underline{y}_o$ and $\underline{y}_+$ are a partition of $\underline{y}$ conforming to $\underline{x}^*_o$ and $\underline{x}^*_+$.

Corresponding to $\underline{x}^*$ we may partition the subscripts i into,

$$i \in i_- \text{ if } n^*_i \leq 0 \quad ; i \neq 0 \tag{124}$$

$$i \in i_+ \text{ if } n^*_i > 0 \quad ; i \neq 0 \tag{125}$$

The Kuhn-Tucker conditions (40) and (44) reduce to the requirement that the following conditions must satisfy.

$$h_i > 0 \quad ; i \in i_- \tag{126}$$

$$h_i = 0 \quad ; i \in i_+ \tag{127}$$

where we ignore for the moment the fact that at an optimal feasible solution the set $i \in i_-$ will have $n^*_i = 0$.

From (42) we have,

$$\underline{y}' = - \underline{dn}^{*\prime}_o - \sum_{i_-} h_i \underline{dn}^{*\prime}_i \tag{128}$$

In an attempt to obtain (122) and (123) subject to (126) and (127) we propose solving the following problem. Find a 1xp vector $\underline{h}$, $p \leq m$, such that,

$$Z = \sum_{j_+} |y_j| - \sum_{j_o} y_j \qquad \text{a min.} \tag{129}$$

subject to,

$$\underline{y}' = - \underline{dn}^{*\prime}_o - \sum_{i_-}^{p} h_i \underline{dn}^{*\prime}_i \geq \underline{0}' \tag{130}$$

$$\underline{h} \geq \underline{0} \tag{131}$$

Substituting the solution $\underline{h}^*$ of this linear programming problem in (128) yields the corresponding minimized vector $\underline{y}^*$. As $\underline{y}$ is a vector whose elements are the negative of the partial derivatives of the constrained objective function we may set $\underline{dx}^{**} = -\underline{Y}^*$. The quantities (111) and (112) may then be obtained and ϕ^* estimated. Unfortunately, for any value of $n_i^* < 0$, we can not guarantee that we will have $b_i^{**} > 0$. This can be overcome by writing (111) in the form,

$$b_i^{**} = \underline{dn}_i^*(-\underline{y}^*)' \tag{132}$$

$$= \underline{dn}_i^*(\underline{dn}_o^{*\prime} + \sum_{i_-}^{p} h_i^*\underline{dn}^{*\prime}) \tag{133}$$

We can then ensure that,

$$b_i^{**} > 0 \qquad ; i \in i_- \tag{134}$$

by adding to problem (129) to (131) the constraints,

$$\underline{dn}_i^*(\underline{dn}_o^{*\prime} + \sum_{i_-}^{p} h_i^*\underline{dn}_i^{*\prime}) \geq - n_i^* \qquad ; i = 1.......m \tag{135}$$

Finally we may note that $\underline{h}^* \geq \underline{0}$ can be incompatible with $\underline{y}^* \geq \underline{0}$. Thus we ignore the constraints (130) and allow $\underline{y}$ to be unrestricted in sign. We can however add the non-negativity condition,

$$\underline{y} \geq \underline{x}^* \tag{136}$$

4.5.1. Algorithm C.

Find a 1xp ($p \leq m$) vector $\underline{h}$ such that,

$$Z = \sum_{j+} |y_j| - \sum_{j_o} y_j \qquad \text{a min} \tag{137}$$

subject to,

$$-\underline{dn}_i^*\underline{y} \geqslant - n_i^* \qquad ; i = 1.......m \tag{138}$$

$$\underline{y} \geqslant \underline{x}^* \tag{139}$$

$$\underline{h} \geqslant \underline{0} \tag{140}$$

If $\underline{h}^*$ is the solution to this problem we proceed to calculate $\underline{y}^*$ from (13), $\underline{dx}^{**}$, and ϕ^*. This allows us to obtain a solution which satisfied

$$n_i^{**} > n_i^* \qquad ; i \in i_- \tag{141}$$

$$n_i^{**} \geqslant 0 \qquad ; i \in i_+ \tag{142}$$

The necessary quantities can be obtained in terms of the new solution and a new problem (137) to (140) obtained.

The original problem is solved when $\underline{x}^*$ is feasible and $Z \geqslant 0$, only when $\underline{xy}' = 0$, $\underline{y} \geqslant \underline{0}$.

4.5.2. Discussion on Algorithm C

It is obvious that algorithm C is weaker than the previous two. If we set $\underline{w}^* = -\underline{h}^*$ and restrict $w_o^* = 1$, then the three algorithms given the same solution $\underline{dx}^{**}$. The difference is that with A and B the use of n_o as the objective function ensures $\underline{xy}' = 0$ in obtaining (134). With algorithm C we explicitly attempt to make $\underline{xy}' = 0$ in obtaining (134).

In terms of algorithm A we may see that (137) to (140) is simply,

$$Z = \sum_{j+} |\underline{dx}_j| - \sum_{j_o} dx_j \quad \text{a min}$$

subject to,

$$\underline{dn}_i^* \underline{dx}' \geqslant - n_i^*$$

$$\underline{dx} \geqslant - \underline{x}^*$$

It is evident that the objective function for this problem is "weaker" than that for algorithms A and B.

CHAPTER 5

DISCUSSION

In the previous sections we have attempted to develope a concise algorithm for the solution of the generalised convex quadratic programming problem. By concise is meant, an algorithm for which it can be explicitly shown, provides a series of improved solutions until the optimum is reached, or indicates that an optimum feasible solution does not exist. It is suggested that Algorithm B is successful as a concise algorithm. Algorithm A has been presented as its properties provide a proof that Algorithm B is concise. Algorithm C simply interprets algorithm B in the light of the Kuhn-Tucker theorem.

It is true that there exist other algorithms like A which after a series of iterations converge on the optimum feasible solution. However there is nothing "natural" about the intermediate solutions provided by these, and proofs of convergence are lacking.

The convexity assumption implies that given a solution which is infeasible, it is possible to make successive changes improving feasibility with respect to all infeasible constraints at each step until feasibility is attained. Given a feasible solution it is possible to make successive increases in the objective function until the optimum is reached. Algorithm B is defined to be concise in that it makes use of these properties of convexity.

The comments at the end of section 4.3.2. substantiate the generality of algorithm B. That is any or all of n_i, $i = 0, 1, \ldots m$, m may be linear, negative definite, or negative semidefinite. Subsidary dual information is provided with the optimum feasible solution.

APPENDIX 1

A PROOF OF THE KUHN-TUCKER THEOREM

Proof

It is clear that $\underline{x}^*$ may lie only on the boundary of some of the constraints $g_i(\underline{x}) \leq 0$. Let i_o denote the set of subscripts i such that $g_i\ (\underline{x}^*) = 0$.

Let the vector $(\underline{x} - \underline{x}^*)$ be directed from $\underline{x}^*$ to $\underline{x} \geq 0$. It follows that,

$$\underline{d}f(\underline{x}^*)\ (\underline{x} - \underline{x}^*)' \leq 0 \quad (1)$$

$$\underline{d}g_i\ (\underline{x}^*)\ (\underline{x} - \underline{x}^*)' \leq 0 \qquad ;\ i \in i_o \quad (2)$$

The inequalities (1) and (2) are linear in the non-negative vector $\underline{x}$. Thus the problem of finding $\underline{x}^*$ which maximizes $f(x)$ subject to the constraints $g_i(\underline{x}) = 0$, $i \in i_o$, becomes one of finding $\underline{x}^* \geq 0$ which maximizes $\underline{d}f(\underline{x}^*)\ \underline{x}'$ subject to the linear constraints,

$$\underline{d}g_i(\underline{x}^*)\underline{x} \leq \underline{d}g_i(\underline{x}^*)\underline{x}^{*\prime} \quad (3)$$

The dual of the linear programming problem requires that we find $h_i \geq 0$ which minimizes,

$$\sum_{i \in i_o} h_i(\underline{d}g_i(\underline{x}^*)\underline{x}^{*\prime}) \quad (4)$$

subject to the linear constraints,

$$\sum_{i \in i_o} h_i(\underline{d}g_i(\underline{x}^*)) \geq \underline{d}f(\underline{x}^*) \quad (5)$$

The remaining h_i for $i \notin i_o$ are all zero and correspond to those $g_i(\underline{x})$ such that $g_i(\underline{x}^*) \leq 0$. Thus from the set of inequalities (5) we may write,

$$\underline{d}f(\underline{x}^*)' = \underline{h}G - \underline{y}' \qquad (6)$$

where,

$\underline{h} \geq \underline{0}$ is a 1xm vector whose i^{th} element equals zero if $i \notin i_o$.

G is a mxn matrix whose i^{th} row is $\underline{d}g_i(\underline{x}^*)'$.

$\underline{y}$ is a 1xn vector of slack variables.

On taking the scalar product with $\underline{x}^*$,

$$\underline{x}^*\underline{y}' = \underline{h}G\underline{x}^{*\prime} - \underline{d}f(\underline{x}^*)\underline{x}^{*\prime} = 0 \qquad (7)$$

since the duality theorem requires that the two expressions in the middle should be equal.

From the fact that $\underline{x}^*\underline{y}' = 0$ and the fact that $\underline{y}$ and $\underline{x}^*$ are non-negative it follows that if $\underline{x}^*_j > 0$ then $y_j = 0$.

REFERENCES

Barankin.E.W. and Dorfman.R.: "Towards Quadratic Programming," Office of Naval Research Logistics Project at Columbia University and University of California, Berkeley, 1955.

Candler.W. and Evans.D.A.: "Classical Quadratic Maximization and Parametric Quadratic Programming," Technical Discussion Paper No. 2, Department of Agricultural Economics and Farm Management, Massey University of Manawatu, 1964.

Candler.W.aand Evans.D.A.: 1964a, to be published.

Candler.W. and Townsley.R.J.: "The Maximization of a Quadratic Function Subject to Linear Inequalities," Management Science Vol. 10, No. 3, April 1964.

Frank.M. and Wolfe.P.: "An Algorithm for Quadratic Programming," Naval Research Logistics Quarterly, 3, pages 79-85, 1956.

Hadley.G.: "Nonlinear and Dynamic Programming," Addison-Wesley Publishing Company, U.S.A., 1964.

Kuhn.H.W. and Tucker.A.W.: "Nonlinear Programming," Proceedings of the Second Berkley Symposium on Mathematical Statistics and Probability, J. Neyman ed., University of California Press, pages 481-492, 1951.

Saaty.T.L.: "Mathematical Methods of Operations Research," McGraw-Hill Book Company, U.S.A., 1959.

Wolfe.P.: "The Simplex Method for Quadratic Programming," Econometrica 27, pages 382-398, 1959.

Wolfe.P.: "Recent Developments in Nonlinear Programming," Rand Publication R-401-PR, 1962.