ARAB REPUBLIC OF EGYPT

THE INSTITUTE OF NATIONAL PLANNING

Memo No. 1291

A Programming Method For Quasi and Explicitly Quasi-Concave Minimum Programming Problem

By

Dr. Amani Omar

Mars 1981

CAIRO
SALAH SALEM St.-NASR CITY

Introduction

In the fifties, the theory of nonlinear programming was almost limited to problems in which the constraints and the objective functions were convex. In the sixties, it was gradually recognized that for most of the theorems to hold and for many of the methods to work, only some weaker property of convex, or even linear functions, is required, property that share with wider classes of functions. The inspiration came from mathematical economics where the significance of the quasiconcavity concept was recognized before its emergence in the nonlinear programming theory. In fact, most business is based upon the notion of quasi-concave (and related) functions. For objective functions of this kind have played an important role in the real practical problems. For example, the fundamental property of the utility function in the theory of consumer demand is that the indifference curves define convex sets or a diminishing marginal rate of substitution. Thus, the minimal property of all utility functions is quasi-concavity. In addition, the theory of efficient production can now be extended to include production functions that are quasi-concave but not concave, that is to those cases in which there are increasing returns to scale but a diminishing marginal rate of substitution. For if a firm's production function takes the form $y= k^{\alpha} L^{B}, \alpha > o$, $B > o$, then y will be quasi-concave but not concave when $\alpha + B > 1$. Then the problem of determining the efficient combination of inputs given any specified output, is a quasi-concave minimum problem. That is, the problem of minimizing rK + WL, where r and w are the cost of a unit of K and L respectively, subject to the constraints $y-y^{o} \geqslant o$, $L \geqslant o$, and $K \geqslant o$, is a quasi-concave minimum problem.

In this paper, we are interested in developing an enumartive method for solving the following problem:

$\text{Min} \left\{ f(x) : x \in X \right\}$, where either the assumption

(I) f(x) is a quasi concave in X and X is bounded
or the assumption.
(II) f(x) is explicity quasi concave in X,
applies
X is the convex polyhedron described by the linear inequalities.

(III)... $a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n \leqslant d_i, \quad i = 1,2, \dots, m.$
and $x_1, x_{21} \dots, x_n \geqslant o.$

The characteristic of the developed method is that it produces all vertices and infinite rays (if any) of X in order to check them for optimality one by one. This class of methods, which they give a full description of the feasible region in terms of its vertices and infinite rays, is not very popular since the number of the vertices may be very large even for a moderately sized problem, On the other hand, this kind of methods are very power in the sense that the spectrum of problems that they can solve is very wide. In addition, they are applicable to a class of problems, e.g., quasi concave minimization, which are not amenable to any other method. However, when evaluating a programming method the most important feature to take into account is its power. A method which for example can solve a problem of quasi convex function is more powerful than another which requires convex function, and the latter is superior to a third in which the objective function must be linear. The second aspect is the computational efficiency, i.e., the number of arithmetical operations to be executed and the amount of data to be stored. These two aspects are, unfortunately, usually contradictory,i.e., more powerful methods tend to be less efficient and vice versa.

A FORTRAN VI program is coded for the proposed method and presented with all necessary comments in the last section. The program has been run on the INTERDATA 7/32 computer with some selected test problems. The outputs of some problems are given.

Some Basic Properties of (Explicitly) Quasi-Concave Function:

Capital letters are used to denote sets, lower case letters are applied for vectors (also called points) and Greek letters for scalars. All the sets are considered to be subsets of the Euclidian n-space E^n. In the sequel the set $[x,y]$ denotes a closed straight line segment connecting the points x and y, and (x,y) denotes an openOne.

For the reader's convenience, we recall a few well-known definitions.

Convex set: The set S is convex if $x_1, x_2 \in S$ implies $[x_1, x_2] \subseteq S$

Polyhedron: A polyhedron X is the set $\{x : AX \leq d, \; x \geqslant o\}$, where A is a matrix of order m x n and d is a constant vector of order m.

Polytope: A bounded polyhedron, marked as X^{Δ}.

Vertex: The point x is a vertex of X if $x \in X$ and $X - \{x\}$ is convex.

Adjacent Vertices: Two different vertices $x_1, x_2 \in X$ are adjacent if $X - [x_1, x_2]$ is convex.

Concave function: A function f is concave if $f(\theta\, x_1 + (1-\theta)\, x_2) \geqslant \theta\, f(x_1) + (1-\theta)\, f(x_2)$, $o \leq \theta \leqslant 1$, for all x_1, x_2 in the domain of $f(x)$.

Definition 1: Quasi-Concavity. The function f(x) is quasi-concave in set S, if for all $x_1,x_2 \in S$ and for all $x_0 \in (x_1,x_2)$,

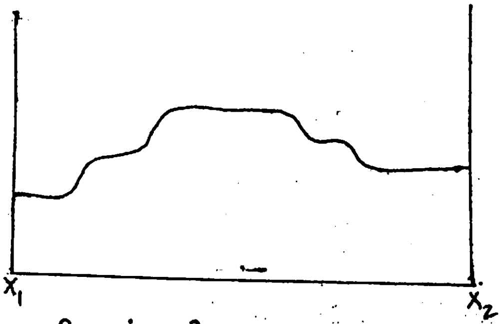
$$f(x_0) \geqslant \min\left[f(x_1),\ f(x_2)\right].$$

Equivalently , $f(x_1) \geqslant f(x_2)$ implies $f(x_0) \geqslant f(x_2)$

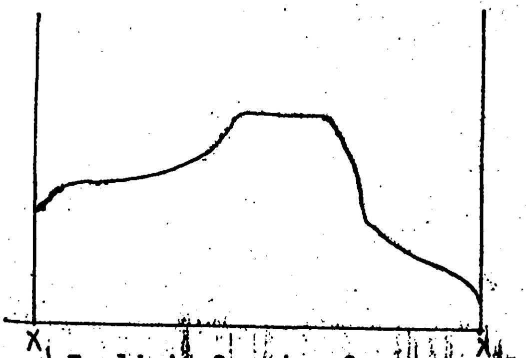
Definition 2: Explicit Quasi-Concavity. The function f(x) is called explicit quasi-concave in a set S, if for all $x_1,x_2 \in S$, $f(x_1) \neq f(x_2)$,

$$f(x_0) > \min\left[\ f(x_1),\ \ f(x_2)\right] \quad ,\ x_0 \in (x_1,x_2).$$

It is evident that explicit quasi concavity in volves quasi-concavity. The concept of quasi-concave functions was introduced by Arrow and Enthoven (1), and the concept of explicit quasi-concavity was that of Martos (3). The following figures illustrate two graphs for quasi-concave and explicity quasi-concave functions for the single variable case.



Quasi - Concave



Explicit Quasi - Concave

The quasi-concave function may have horizontal straight segment any where but the explicity quasi-concave can have it only at the top.

Definition 3: Global Minimum point. The point $\hat{x} \in S$ is a global point of the scalar function f(x) in S, if $f(\hat{x}) \leqslant f(x)$ for each $x \in S$.

Definition 4: Global Vertex-Minimum. The vertex x_o is a global vertex-minimum of f(x) in the polyhedron X , if $f(x_o) \leqslant f(x)$ for each vertex x of X .

Definition 5 : Infinite Ray.

Let x be a vertex of the polyhedron X and x_1 a point in E^n. Let X_N be the subvector of x containing the nonbasic variables and X_{N1} the subvector of x_1 whose components correspond to X_N.

The set $R = \left\{ r: \quad r = x + \lambda(x_1 - x), \lambda \geqslant o \right\}$

is called an infinite ray of X emanating from x if (i) R is a subset of X

and

(ii) Exactly n-1 components of X_{N_1} are zeros.

(x_1-x) is called the direction of R.

Since we are concerned with the minimum value of the (explicitly) quasi-concave function defined on X, we present the following two theorems (for the proof see Martos (4)).

Theorem 1

The function f (x) is quasiconcave in the convex set $S \subseteq E^n$ if and only if for each polytope $X^\Delta \subseteq S$ any global vertex - minimum point of f(X) in X^Δ is a global minimum point in X^Δ.

If the feasible region X is unbounded, then a quasiconcave function, of course, need not assume a minimum. But theorem (1) shows that a quasiconcave function assumes its minimum value at one of the vertices of the feasible region if that region is bounded. If the region is unbounded it need not assume a minimum value.-

and, even it has a minimum over the region X, it need not occur at a vertex of X. For example, let E_+^2 be the positive orthant of the plane which is a polyhedron whose only vertex is 0, then the function

$$f\ (X) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x > 0, \end{cases}$$

is quasi concave and assumes its minimum f (x) = 0 every where in E_+^2 except at the vertex 0. For the function which is explictly quasi concave, the necessary part of theorem (1) holds in the following sense.

Theorem 2

If f (X) is explicitly quasi concave in unbounded X and assumes its minimum in X, then the minimum is attained at a vertex of X. In other words, any global vertex-min. of f (X) in X is a global minimum point.

The following three theorems characterize the set of the points that are global minimum points of (explicitly) quasi concave function. These theorems may be useful when looking for all the optimum solutions . Let

$$S^* = \{x^* \in X : \text{ if } f\ (X^*) \leqslant f\ (X), \text{ for all } x \in X\},$$

$$\text{and } \bar{S} = X - S^* = \{\bar{x} \in x: \ f(X) < f\ (\bar{X}) \text{ for some } X \in X\} .$$

Theorem 3

If f (X) is quasiconvex in the convex set X, then S* is convex.

Theorem 4

If f (X) is quasiconcave in X, then $\bar{S}$ is convex.

Theorem 5

If f (X) is explicitly quasi concave in X, then for any closed segment $[x_1, x_2] \subset X$ either $[x_1, x_2]$ S^* or $(x_1, x_2) \not\subset \bar{S}$.

The following theorem characterizes the directions of infinite rays of X. It will help in constructing the infinite says (if there is any) of a polyhedron X.

Theorem 6

The set $R = \{ r : r = x + \lambda(x_1 - x), \lambda \geqslant 0 \}$ is an infinite ray of x if and only if

(i) $A (x_1 - x) = 0$

(ii) $x_1 - x \geqslant 0$

and (III) n-1 components of x_{N1} are zeros.

A Method For Locating Global Minimum Points:

The set of all vertices $V = \{ \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k \}$ and the set of all inifinite rays (if x is unbounded) $R = \{ R_1, R_2, \ldots, R_t \}$ have to be determined. Briefly, these two sets can be generated as follows: we start by any vertex $\hat{x}_s$ of X and save its index in a set S. We identify all adjacent vertices of $\hat{x}_s$ and keep their indices in a set W. We choose any element of W to calculate its corresponding vertex $\hat{x}_r$ and all new adjacent vertices of $\hat{x}_r$. The set W is updated to contain the indices of the uncalculated vertices and the set S is updated to contain the indices of $\hat{x}_s$ and $\hat{x}_r$. We continue in this manner till the set W be empty*. All vertices will have been found when $W = \phi$ (For the proof, see (6)). For any —

---

* For the detailed algorithme of this method, see (OMAR, 6).

vertex, say, $\hat{x}_q$, if the K^{th} nonbasic column $y^q_k = B^{-1}_q a_k$ in the simplex tableau associated with $\hat{x}_q$, B_q is the basic matrix corresponding to $\hat{x}_q$, has nonpositive values for all its components, then the infinite ray R_q emanating from x is constructed as follows:

$$R_q(\lambda) = \begin{bmatrix} \hat{x}_{B1} \\ \hat{x}_{B2} \\ \\ \hat{x}_{Bm} \\ \\ 0 \end{bmatrix} - \lambda \begin{bmatrix} y^q_{1k} \\ y^q_{2k} \\ \vdots \\ y^q_{mk} \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \leftarrow \text{the k-th position} \\ \\ \\ \\ \end{matrix} , \quad \lambda \geqslant 0,$$

where $\hat{x}_{B1}$, $\hat{x}_{B2}$, ..., $\hat{x}_{By}$ are the basic variables of $\hat{x}_q$. The set R will be empty if x is bounded. This will be recognized during the process if no columns such as y^q_k, i.e,nonbasic columns of nonpaositive elements, is found.

Now to solve the problem min$\{ f(x): x \in X \}$ under assumption (I), we have $R = \varphi$: Then by theorem (1), all the vertices $\hat{x}_j$, i = 1, 2, ... s, satisfying $f(\hat{x}_j) \leqslant f(\hat{x}_q)$ for any vertex $\hat{x}_q$, are global minimum solutions to the problem.

To solve the problem under assumption (II), we follow the following steps:

Step 1: Compute the value $f^* = \min \{ f(\hat{x}_i), i=1, 2, \dots, k \}$,
to determine the set $S^* = \{ \hat{x}_i : f(\hat{X}_i) = f^* \}$.

Step 2: (i) If R is empty, go to step 4: (ii) If R is not empty go to step 3.

Step 3: Compute

$$I_i = \lim_{\lambda \to \infty} f(R_i(\lambda)), \text{ for each } i = 1, 2, \dots, t.$$

(i) If $I_i < f^*$ for any i, then the problem has no solution.

(ii) If $I_i \geqslant f^*$ for all i = 1, 2, ..., t, go to step 4.

Step 4 : Any vertex $\hat{x}_i \in S^*$ is a global minimum solution to the problem with optimum value f* for the objective function.

Practically, the limit in step 3 is calculated for sufficiently large value of λ. In the following FORTRAN program, λ is taken to be 0.1×10^{60} . The largest positive value holded by the word of the INTER DATA 7/32 Computer which has been used to run the program, is 7.2×10^{70}. It remains to prove that:

Theorem 7:

The previous procedure solves the problem under assumption (II) finitely.

Proof :

The finiteness of the procedure follows from the fact that:

(i) The number of vertices of X is finite, an upper bound is $(n+m)! / n!\ m!$.

(ii) The number of inifinite rays is finite, an upper bound is

$$n \times \frac{(n+m)!}{n!\ m!}$$

(iii) The method used to genrate the sets V and R is finite (see, omar (6)).

Now, in case of step 2 (i), i.e, $R = \phi$, any point of S^* is an optimum solution to the problem by theorem (1).

To prove the statement of 3 (i), we have to show first that $f(R_i(\lambda))$ is explicitly quasiconcave function of λ for any $\lambda > 0$ and i = 1, 2, ..., t.

Let $\hat{x}_i$ be a vertex from which an infinite ray emanates let λ_1, λ_2 be $\geqslant 0$ such that

$$f(\hat{x}_i + \lambda_1 \hat{y}_i) > f(\hat{x}_i + \lambda_2 y_i).$$

If $(\hat{x}_i + \lambda_0 y_i) \in (\hat{x}_i + \lambda_1 y_i, \hat{x}_i + \lambda_2 y_i)$, for some $\lambda_0 > 0$, then since f is explictly quasiconcave in X and $\hat{x}_i + \lambda_0 y_i$, $\hat{x}_i + \lambda_1 y_i$, & $\hat{x}_i + \lambda_2 y_i$ are elements of X,

$$f(\hat{x}_i + \lambda_0 y_i) > f(\hat{x}_i + \lambda_2 y_i)$$

Thus f $(R_i(\lambda))$ is explicitly quasiconcave for $\lambda \geqslant 0$ and for any i = 1, 2, ..., t.

Since f is explicitly quasiconcave in X, then f $(R_i(\lambda))$ tends either to a finite limit, or to $+\infty$ or $-\infty$ as $\lambda \to \infty$ (Theorem).

Now, if $I_i < f^*$ for some i, then there exists $0 < \lambda_\rho < \infty$ such that

$$f(R_i(\lambda_\rho)) = f(\hat{x}_i + \lambda_\rho y_i) < f^*$$

Because $\hat{x}_i + \lambda_\rho y_i$ is an element of X the problem has no optimum solution since by theorem (2) the optimum solution must be a vertex. It remains to prove 3 (ii).

Since f is quasiconcave along the infinite rays of X, then f $(\hat{x}_i + \lambda y_i) \geqslant \min \left\{ f(R_i(0)), \lim_{\lambda + \infty} f(R_i(\lambda)) \right\}$, $\lambda \geqslant 0$, i=1, 2, ...,t.

Since $R_i(0) = \hat{x}_i$ is element of V and $I_i \geqslant f^*$ for all i, then min $\left\{ f(R_i(0)), I_i \right\} \geqslant f^*$.

Since the unbounded polyhedron X is the convex hull

$$X = \sum_{i=1}^{k} \mu_i \hat{x}_i + \sum_{j=1}^{k} \theta_j R_j, \text{ where}$$

$\sum \mu_i = 1$ and $\theta_j \geqslant 0$, hence, by the quasiconcavity of f in X we have

$$f(x) \geqslant f^* \quad \text{for any } x \in X.$$

Thus the statement of step 4 holds in case 3 (ii).

A FORTRAN Program for the proposed Method.

In this section we present a computer program for the method described in the previous section. The method is coded in FORTRAN VII and a number of examples have been chosen to test the program on the INTERDATA 7/32 computer. The following is an explanation of the basic symbols used in the program.

M - Number of constraints in system (III).

N - Number of nonbasic variables.

NV- Upper bound for the number of vertices.

A - Real Mx(N+1) - array for the nonbasic columns and the constant column d.

JP- Pivot column

IP- Pivot row

PE- Pivot element

INR- Integer M - array for the current indices of the basic variables.

INC- Integer N - array for the current indices of the nonbasic variables.

MS - Integer M x NV - array for the indices of the vertices.

KGL- Number of alternate global optima

X - Real (N+M) - array for the current solution

GLV- Real array for the global value (s) of the objective function.

VIR- Real array for the values $-f(R_i(\lambda))$ in case of unbounded X.

MCL- Pointer Pointing to the most right element in the left section of MS.

MCR- Pointer pointing to the most left element in the right section of MS.

LL- Number of basic rows that can be inter changed with the currently investigated column. LL $>$ 1 in case of ties and degener ate solutions.

DIF- Specifier assigned the value 0/1 if two examined vectors of basic indices are identical/different.

NXX - Number of infinite rays..

The program includes a device indicating that incorrect estimate of NV has been used. This could be done by simply testing the equality of the pointers MCL and MCR. The two sections of MS will overlap, and hence NV is incorrecte estimate of the number of vertices, when MCL becomes $\geqslant$ MCR.

```
@BATCH
C ******************************************************************
C THIS IS A PROG. FOR CALCULATING THE GLOBAL MIN. VALUE OF A QUASI
C AND EXPLICITLY QUASI CONCAVE FUNCTION UNDER LINEAR CONSTRAINTS.
C ******************************************************************
      DIMENSION A(20,20),MS(21,100),MX(20),AB(20),AJ(20),X(20)
      DIMENSION AX(20),GLV(50),VIR(20)
      COMMON/C1/IDC,LL,LLX,M,N,IB(20)/C2/IDIF/C3/INR(20),INC(20)
      READ(1,11)M,N,NV,MIT
   11 FORMAT(15I5)
      XLANDA=0.1E+60
      N1=N+1
      READ(1,5) ((A(I,J),J=1,N1),I=1,M)
      READ(1,11)(INC(I),I=1,N),(INR(I),I=1,M)
    5 FORMAT (8F10.5)
      IGL=1
       NXX=0
      IST=0
      NSP=0
      MCL=0
      MCR=NV
      DO 100 I=1,M
  100 MS(I,MCR)=INR(I)
      MCR=MCR-1
      DO 6 I=1,M
    6 AB(I)=A(I,N1)
C ***********************************
C CALCULATE THE VALUE OF THE FUN. AT
C THE CURRENT VERTEX
C ***********************************
  200 DO 332 I=1,M
      NI=INR(I)
  332 X(NI)=AB(I)
      DO 300 I=1,N
      NI=INC(I)
  300 X(NI)=0.0
      XC=EVCONC(X,M+N,MIT)
C ***********************************
C TEST THE MINI. GLOBALITY OF THE
C CURRENT VALUE OF THE FUN.
C ***********************************
      IF(IGL.EQ.1)GO TO 299
      IF(XC.GT.GLV(1))GO TO 555
      IF(XC.EQ. GLV(1))GO TO 199
  299 KGL=1
  199 GLV(KGL)=XC
      KGL=KGL+1
  555 NK=1
      NX=0
C ***********************************
C TEST THE EXISTENCE OF A
C NONPOSITIVE NONBASIC COLUMN IN
C THE CURRENT TABLEAU
```

```
C ************************************
      DO 7 J=1,N
      DO 8 I=1,M
      IF(A(I,J))8,8,7
    8 CONTINUE
      NX=2
C ************************************
C CALCULATE THE VALUE OF THE FUNC.
C ALONG IN FINITE RAYS.
C ************************************
      DO 884 KI=1,M
      NI=INR(KI)
  884 X(NI)=AB(KI)-A(KI,J)*XLANDA
      DO 99 KI=1,N
      NI=INC(KI)
   99 X(NI)=0.0
      NI=INC(J)
      X(NI)=XLANDA
      XC=EVCONC(X,M+N,MIT)
      IF(NK.EQ. 1)GO TO 77
      IF(XC.GT.XM)GO TO 7
   77 XM=XC
      NK=2
    7 CONTINUE
      IF(NX.EQ.0)GO TO 14
       NXX=NXX+1
       VIR(NXX)=XM
   14 JP=0
C ************************************
C ANALYSIS OF THE CURRENT TABLEAU
C ************************************
      DO 15 JX=1,N
      DO 16 I=1,M
      IA=A(I,JX)*1.E+05
      AI=IA*1.E-05
      IF(AI.GT.1.E-05)GOTO17
   16 CONTINUE
C ************************************
C MOVE TO THE NEXT NONBASIC COLUMN
C IF THE CURRENT ONE IS NONPOSITIVE
C ************************************
      GO TO 15
   17 DO 18 I=1,M
   18 AX(I)=A(I,JX)
C ************************************
C STORE THE INDICES OF ALL VERTICES
C NEIGHBORING TO THE INITIAL VERTEX
C ************************************
      CALL MIN(AX,AB)
      NC=INC(JX)
      IF(IST .EQ. 1)GO TO 24
      DO 19 I=1,LL
      MCL=MCL+1
```

```
      LX=IB(I)
      DO 20 IX=1,M
      IF(IX-LX)21,22,21
   21 MS(IX,MCL)=INR(IX)
      GO TO 20
   22 MS(IX,MCL)=NC
   20 CONTINUE
   19 CONTINUE
      JP=JX
      IP=IB(LL)
      GO TO 15
C *********************************
C SEARCH FOR NEW VERTICES; IF THE
C INDICES CREATED FROM THE CURRENT
C COLUMN IS AMONG THE ELEMENT OF
C MS JUMP TO STAT.  15,OTHERWISE,
C JUMP TO STAT.  32 TO STORE THE NEW
C VERTICES.
C *********************************
   24 DO 88 I=1,LL
      LX=IB(I)
      DO 80 J=1,NV
      IF(J.LE.MCR.AND.J.GT.MCL)GO TO 80
      DO 28 IX=1,M
   28 MX(IX)=MS(IX,J)
      CALL LOOK (MX,NC,INR,LX,M)
      IF (IDIF.EQ.0)GO TO 15
   80 CONTINUE
      MCL=MCL+1
C *********************************
C TERMINATE THE PROG.  IF THE NUMBER
C OF VERTICES EXCEEDS THE SPACE OF
C MS.
C *********************************
   90 DO 85 IX=1,M
      IF(MCL-MCR)32,31,31
   31 WRITE(1,86)
   86 FORMAT(10X,7HOVERLAP)
      GO TO 1
   32 IF (IX-LX)233,234,233
  234 MS(IX,MCL)=NC
      GO TO 85
  233 MS(IX,MCL)=INR(IX)
   85 CONTINUE
   88 CONTINUE
      JP=JX
      IP=IB(LL)
   15 CONTINUE
C ************************************
C TRANSFER THE CURRENT TABLEAU TO THE
C ONE CORRESPONDING TO THE LAST VERTEX
C STORED IN MS.
C ************************************
```

```
      IST=1
      IF(JP.EQ.0)GO TO 41
      CALL TST(A,JP,IP,M,N1)
      DO 666 I=1,M
  666 AB(I)=A(I,N1)
C ***********************************
C UPDATE THE POINTERS MCL&MCR.
C ***********************************
      DO 140 I=1,M
  140 MS(I,MCR)=MS(I,MCL)
      MCL=MCL-1
      MCR=MCR-1
      IGL=2
      GO TO 200
   41 IF(MCL.EQ.0)GO TO 1000
C ***********************************
C THIS PART IS EXECUTED IF NO NEW
C VERTICES HAVE BEEN CREATED FROM
C THE CURRENT TABLEAU.  THE MOST
C RIGHT VERTEX OF THE LEFT SECTION
C OF MS IS SELECTED AND THE
C CURRENT TABLEAU IS TRANSFORMED
C INTO IT.
C ***********************************
      DO 144 I=1,M
      IN=INR(I)
      DO 145 J=1,M
      IF(IN-MS(J,MCL))145,144,145
  145 CONTINUE
      IP=I
      DO 148 IX=1,M
      MT=MS(IX,MCL)
      DO 146 J=1,N
      IF(MT-INC(J))146,147,146
  147 JP=J
      IA=A(IP,JP)*1.E+05
      AI=IA*1.E-05
      IF(AI .EQ. 1.E-05)GO TO 148
      CALL TST(A,JP,IP,M,N1)
      GO TO 144
  146 CONTINUE
  148 CONTINUE
  144 CONTINUE
      DO 677 I=1,M
  677 AB(I)=A(I,N1)
      DO 150 I=1,M
  150 MS(I,MCR)=MS(I,MCL)
      MCL=MCL-1
      MCR=MCR-1
      IGL=2
      GO TO 200
C ***********************************
C IF NO INIFINITE RAYS EXIST GO TO
```

```
C 1100 TO PRINT THE GLOBAL VALUE
C ********************************
 1000 IF(NXX.EQ.0)GO TO 1100
       XC=GLV(1)
       DO 111 J=1,NXX
      XXC=VIR(J)
       IF(XXC.LT.XC)GO TO 2000
  111  CONTINUE
 1100 KGL=KGL-1
      WRITE(2,2222)KGL,GLV(1)
 2222 FORMAT(2X,I3,'EQUAL GLOB. OPT. OF VALUE',E14.8,'FOUND')
      GO TO 1
 2000 WRITE(2,333)XC,XXC
  333 FORMAT(2X,'SMALLEST VALUE OF F ',E14.8,2X,'THE PROBLEM HAS NO OPT
     1. SOL. ONE OF IR=',E14.8)
    1 WRITE(2,3333)
 3333 FORMAT(2X,'END OF PROG.')
      END
```

```
C *************************************************************
C EVONCE FUNCT. CALCULATES THE VALUES OF THREE FUNCTION
C AT THE VERTCIES OF THE FEASIBLE REGION
C *************************************************************
      FUNCTION EVCONC(X,ML,MIT)
      DIMENSION X(20)
      IF(MIT)50,51,52
   51 EVCONC=X(1)*X(2)*(X(3)**2)*X(4)*X(5)**3
      GO TO 100
   52 IF(X(1))100,53,54
   53 IF(X(2))100,55,56
   55 EVCONC=-2.0+2.0*X(3)
      GO TO 100
   56 EVCONC=3.0+2.0*X(3)+(3.0*X(2)+2.0)/(X(2)+1.0)
      GO TO 100
   54 IF(X(2))100,57,58
   57 EVCONC=-3.0*X(1)+2.0*X(3)
      GO TO 100
   58 EVCONC=5.0-3.0*X(1)+2.0*X(3)+(3.0*X(2)+2.0)/(X(2)+1.0)
      GO TO 100
   50 EVCONC=(-(X(1)-2.0*X(2))**2+2.0*X(1)+X(2)+1.)/(X(1)+3.0*X(2)+1.0
  100 RETURN
      END
```

```
C ****************************************
C TST SUB.TRANSFORMS A SIMPLEX TABLEAU TO
C ANOTHER AND UPDATES INR & INC
C ****************************************
      SUBROUTINE TST(X,JP,IP,M,N1)
      DIMENSION X(20,20)
      COMMON /C3/INR(20),INC(20)
      PE=1.0/X(IP,JP)
      DO 15 I=1,M
      IF(I .EQ. IP) GO TO 15
      DO 7 J=1,N1
      IF(J .EQ. JP) GO TO 7
      X(I,J)=X(I,J)-X(I,JP)*X(IP,J)*PE
    7 CONTINUE
   15 CONTINUE
      DO 12 J=1,N1
   12 X(IP,J)=X(IP,J)*PE
      DO 13 I=1,M
   13 X(I,JP)=-X(I,JP)*PE
      X(IP,JP)=PE
      IN=INR(IP)
      INR(IP)=INC(JP)
      INC(JP)=IN
   27 RETURN
      END
```

```
C ***************************************
C MIN DETERMINES THE PIVOTAL ROW OR ROWS
C IN CASE OF TIE AND DEGENERATE VERTEX, OF
C NONBASIC COLUMN X.
C *************************************
      SUBROUTINE MIN(X, Y)
      DIMENSION X(20), Y(20)
      COMMON /C1/IDC, LL, LLX, M, N, IB(20)
      C=1. 0E+20
      IDC=0
      LL=0
      LLX=0
      DO 14 L2=1, M
      IF(Y(L2))12, 18, 14
   14 CONTINUE
      LLX=1
   23 DO 11 L2=1, M
      IF (X(L2))11, 11, 22
   22 IF(Y(L2))12, 11, 16
   16 IF (Y(L2)/X(L2)-C)17, 11, 11
   17 C=Y(L2)/X(L2)
      IIB=L2
   11 CONTINUE
      LL=LL+1
      IB(LL)=IIB
      GO TO 12
   18 DO 19 L2=1, M
      IF (Y(L2))12, 15, 19
   15 IF(X(L2))21, 19, 24
   24 IDC=1
   21 LL=LL+1
      IB(LL)=L2
   19 CONTINUE
      IF(IDC .EQ. 0)GO TO 23
   12 RETURN
      END
```

```
C ****************************************
C LOOK SUB. TESTS THE SIMILARITY OF THE
C INDICES OF TWO VERTECES, IT SETS DIF=0 IF
C THEY ARE THE SAME, OTHERWISE DIF=1.
C ****************************************
      SUBROUTINE LOOK(MX,NC,INR,LX,M)
      DIMENSION MX(20),INR(20)
      COMMON /C2/IDIF
      IDIF=0
   80 FORMAT(10I8)
      DO 5 I=1,M
      IF(I-LX)9,8,9
    8 N=NC
      GOTO6
    9 N=INR(I)
    6 DO 7 IX=1,M
      IF(N-MX(IX))7,5,7
    7 CONTINUE
      IDIF=1
      GOTO10
    5 CONTINUE
   10 RETURN
      END
```

The previous FORTRAN program has been used to run a number of problems. For some large size problems, for which the theoritical estimate of the number of vertices exceeds the available core storage, we use the value of NV which fits the data into the main memory, and in case of the overlapping of the two sections of MS we stop the program.

The computer results of the following test problems are shown below:

1) Minimize $f\,(x_1, x_2, x_3) = f_1\,(x_1) + f_2(x_2) + f_3\,(x_3)$,

$$f_1(x_1) = \begin{cases} 0 & \text{if } x_1 = 0 \\ 2-3x_1 & \text{if } x_1 > 0 \end{cases},$$

$$f_2(x_2) \begin{cases} -5 & \text{if } x_2 = 0 \\ \dfrac{3x_2+2}{x_2+1} & \text{if } x_2 > 0 \end{cases},$$

$$f_3\,(x_3) = 3 + 2x_3\ ,$$

subject to the constraints

$$2x_1 + x_2 - 2x_3 \leqslant 6\quad,$$

$$x_1 + 2x_2 - 2x_3 \leqslant 7\quad,$$

$$x_1 - x_2 \leqslant 1,$$

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

The feasible region x has the vertices :

$\hat{x}_1 = (0, 0, 0)^t$, $\hat{x}_2 = (1, 0, 0)^t$, $\hat{x}_3 = (7/3, 4/3, 0)^t$, $\hat{x}_4 = (5/3, 8/3, 0)^t$,

$\hat{x}_5 = (0, 7/2, 0)^t$, and the five infinite rays:

$R_1 = (0, 0, 0)^t + \lambda(0, 0, 1)$, $R_2 = (1, 0, 0)^t + \lambda(0, 0, 1)^t$,

$R_3 = (7/3, 4/3, 0)^t + \lambda(2, 2, 3)^3$, $R_4 = (5/3, 8/3, 0)^t + \lambda(2, 2, 3)^t$,

$R_5 = (0, 7/2, 0)^t + \lambda(0, 1, 1)^t$.

The values of F at the vertices are:

$f(\hat{x}_1) = -2$, $f(\hat{x}_2) = -3$, $f(\hat{x}_3) = 4/7$, $f(\hat{x}_4) = 28/11$,

$f(\hat{x}_5) = 5.7/9$.

Since

$$\lim_{\lambda\to\infty} f(R_1(\lambda)) = \lim_{\lambda\to\infty} f(R_2(\lambda)) = \lim_{\lambda\to\infty} f(R_5(\lambda)) = +\infty,$$

$$\lim_{\lambda\to\infty} f(R_3(\lambda)) = 1,$$

and $\lim_{\lambda\to\infty}(f(R_4(\lambda))) = 3$, i. e, $f^* = -3 < I_i$, $i = 1, 2, \dots, 5$,

hence, $\hat{x}_2$ is the global minimum solution with global value -3.

2) If the function $f_3(x_3)$ is changed to $f_3(x_3) = 3 + x_3$ in the objective function, then $\lim_{\lambda\to+\infty} f(R_3(\lambda)) = -\infty$ and the problem will have no optimum solution.

3) Minimize $f(x_1, x_2) = -\dfrac{(x_1 - 2x_2)^2 + 2x_1 + x_2 + 1}{x_1 + 3x_2 + 1}$

subject to

$$\begin{aligned} x_1 - x_2 &\leq 2, \\ 2x_1 - 5x_2 &\leq 1, \\ -x_1 + 2x_2 &\leq 0, \\ -2x_1 + 3x_2 &\leq -1, \\ x_1 \text{ and } x_2 &\geq 0. \end{aligned}$$

The vertices are: $\hat{x}_1 = (\frac{1}{2}, 0)^t$, $\hat{x}_2 = (2, 1)^t$, $\hat{x}_3 = (3, 7)^t$, and $\hat{x}_4 = (4,2)^t$.

The values of f are; $f(\hat{x}_1) = 7/6$, $f(\hat{x}_2) = 1$, $f(\hat{x}_5) = 1$, $f(\hat{x}_4) = 1$. Hence $\hat{x}_2$, $\hat{x}_3$ and $\hat{x}_4$ are optimum solutions.

It worth to note that in case of quasiconcave minimization the set of optimum solutions need not be convex and there is no general procedure for calculating it. For example, in this problem the set of optimum solutions has the isolated point $\hat{x}_3$ and the segment line $[\hat{x}_2, \hat{x}_4]$.

4) The following problem has not been calculated by hand:

Minimize $f(x_1, x_2, x_3, x_4, x_5) = x_1 \cdot x_2 \cdot x_3^2 \cdot x_4 \cdot x_5^3$

subject to

$$\begin{aligned}
&x_1 + x_2 + x_3 + x_4 + x_5 \leqslant 1, \\
&x_1 - x_2 - 2x_3 + x_5 \leqslant \frac{1}{2} , \\
&x_2 + \frac{1}{2} x_3 + x_5 \leqslant 2 \\
&x_1 + x_3 - x_4 \leqslant 3, \\
&x_1 + \frac{1}{2} x_2 + x_3 + x_4 - x_5 \leqslant 1, \\
&x_4 + x_5 \leqslant \frac{1}{2} , \\
&-x_1 - 2x_2 \leqslant 4, \\
&zx_1 + 2x_2 + x_3 + x_4 \leqslant 1, \\
&-2x_1 - x_4 + x_5 \leqslant 5, \\
&x_1 + x_2 - 2x_3 - x_4 + x_5 \leqslant 1, \\
&+ x_2 = x_3 - x_4 + \frac{1}{2} x_5 \leqslant \frac{1}{2} , \\
&x_1, x_2, x_3, x_4, \text{ and } x_5 \geqslant 0.
\end{aligned}$$

```
    3EQUAL GLOB. OPT. OF VALUE0.10000000E+01FOUND
  END OF PROG.
   15EQUAL GLOB. OPT. OF VALUE0.00000000E+00FOUND
  END OF PROG.
    1EQUAL GLOB. OPT. OF VALUE-.30000000E+01FOUND
  END OF PROG.

SMALLEST VALUE OF F - 30000000E+01   THE PROBLEM HAS NO OPT . SOL. ONE OF IR=-.95780971E+53
END OF PROG.
```

References .

1. ARROW, K.J. and ENTHOVEN, A.D., "Quasi-concave Programming", Econometrica, Vol. 29, No. 4(october 1961), PP. 779-800.

2. ——— And ———, "Nonlinear Power of Adjacent Extreme Point Methods in Linear Programming", Econometrica, Vol. 25, No.1 (1956), PP. 132 - 153.

3. MARTOS, B., "The Direct Power of Adjacent vertex Programming Methods", Mang. Sci., Vol. 12, No. 3 (1965), PP. 241-252.

4. MARTOS, B., "Nonlinear Programming. Theory and Methods", AK ademiai Kiado, Budapest (1975), PP 81 - 93.

5. KOVACS, L.B., "Gradient Projection Method for Quasi-Concave Programming", Colloqu. On Appl. of Math. to Economics, Budapest (1963).

6. Omar A. Z., "Finding all Extreme Points and Extreme Rays of A Convex Polyhedron", Ph. D. Thesis, Prague (1976).