gorithm 431

A Computer Routine for Quadratic and Linear Programming Problems [H]

Arunachalam Ravindran [Recd. 24 Aug. 1970, 11 June 1971, and 1 Nov. 1971]
School of Industrial Engineering, Purdue University, Lafayette, IN 47907

Abstract. A computer program based on Lemke's complementary pivot algorithm is presented. This can be used to solve linear and quadratic programming problems. The program has been extensively tested on a wide range of problems and the results have been extremely satisfactory.

Key Words and Phrases: linear program, quadratic program, complementary problem, Lemke's algorithm, simplex method
CR Categories: 5.41
Language: Fortran

Description

Introduction. The computer routine given below is based on Lemke's complementary pivot algorithm [2] to solve the complementary problem of the form:

$$\text{Find } w, z \geqq 0$$
$$\text{such that } w = Mz + q \tag{1}$$
$$w'z = 0$$

where M is an $(N \times N)$ square matrix; w, z and q are $(N \times 1)$ column vectors. ("Prime" denotes the transpose of a vector or matrix.)

A solution to the above problem will be called a complementary solution, and Lemke's algorithm is guaranteed to find a complementary solution to system (1) only if the matrix M satisfies one of the following:

1. M has all positive elements.
2. M is a positive semidefinite matrix or $x'Mx \geqq 0$ for all x.
3. M has positive principal determinants.

Applications. The two important applications of the complementary problem (1) are to solve linear and quadratic programming problems by converting them to an equivalent complementary problem.

Quadratic Programming. Consider the quadratic program:

$$\text{Minimize } Z = c'x + x'Qx$$
$$\text{subject to } Ax \geqq b$$
$$x \geqq 0$$

where A is an $(m \times n)$ matrix, Q is an $(n \times n)$ matrix of the quadratic form, c and x are $(n \times 1)$ column vectors, and b is an $(m \times 1)$ column vector.

An optimum solution to the above problem may be obtained by solving a complementary problem of the form:

$$\binom{v}{u} = \begin{pmatrix} Q+Q' & -A' \\ A & 0 \end{pmatrix}\binom{x}{y} + \binom{c}{-b} \tag{2}$$
$$u, v, x, y \geqq 0$$
$$v'x + u'y = 0$$

where u denotes the slack variables of the given quadratic program and (y, v) denotes the variables of the dual problem. Comparing the above system (2) with the original complementary problem (1), we note that

$$w = \binom{v}{u}, z = \binom{x}{y}, M = \begin{pmatrix} Q+Q' & A' \\ A & 0 \end{pmatrix} \text{ and } q = \binom{c}{-b}.$$

System (2) can be solved by the given computer routine and then an optimum solution to the given quadratic program may be obtained by reading off the values of $(z_1, z_2, \ldots, z_n, w_{n+1}, \ldots, w_{n+m})$ from the complementary solution. It should be remarked here that the matrix M in this case is positive semidefinite if and only if the matrix Q is positive semidefinite. Hence, the computer routine is guaranteed to find an optimum solution to the given quadratic program only if the objective function Z is a convex function.

Linear Programming. Consider the linear program:

$$\text{Minimize } Z = c'x$$
$$\text{subject to } Ax \geqq b$$
$$x \geqq 0.$$

The only difference between a linear program and a quadratic program is in the objective function. Hence, by setting $Q = 0$ in system (2), we get the equivalent complementary problem for a linear program.

Program. A detailed description of Lemke's algorithm to solve the complementary problem, on which the computer routine is based, is given in [3]. The program consists of six subroutines and a main program which calls these subroutines in proper order. The various input data to the program are the number of problems to be solved in succession, the size of the problem and the elements of matrix M and vector q. The original Lemke's algorithm [2] was modified by the author along the lines of the revised simplex method [1] for a linear program to take advantage of the fact that for solving linear and quadratic programs, the M matrix in system (1) has many zero entries. This led to a greater efficiency of the computer routine.

In an experimental study conducted by the author [4], this computer routine was extensively used to compare the relative efficiencies of the simplex method [1] and Lemke's algorithm to solve linear programs. The study revealed the superiority of Lemke's algorithm over the simplex method in a number of problems both with regard to the number of iterations and computation time. Also in [3], another modification of Lemke's algorithm for solving linear programs has been proposed which may save a considerable storage and computation time.

References

1. Dantzig, G.B. *Linear Programming and Extensions*. Princeton U. Press, Princeton, N.J. 1963.
2. Lemke, C.E. Bimatrix equilibrium points and mathematical programming. *Management Sci. 11* (1965), 681–689.
3. Ravindran, A. Computational aspects of Lemke's complementary algorithm applied to linear programs. *Opsearch* 7 (1970), 241–262.

4. Ravindran, A. A comparison of the primal-simplex and complementary pivot methods for linear programming. Rep. No. 70-9 (July 1970), School of Industrial Engineering, Purdue U., Lafayette, Ind.

Algorithm

```
C REMARKS
C SINCE THIS PROGRAM IS COMPLETE IN ALL RESPECTS,IT CAN BE
C RUN AS IT IS WITHOUT ANY ADDITIONAL MØDIFICATION OR
C INSTRUCTION.IN SUCH CASE FOLLOW THE INPUT FORMAT AS GIVEN
C
C PROGRAM FOR SOLVING LINEAR AND QUADRATIC PROGRAMMING
C PROBLEMS IN THE FORM W=M*Z+Q, W.Z=0, W AND Z NONNEGATIVE
C BY LEMKE/S ALGORITHM.
C
C MAIN PROGRAM WHICH CALLS THE SIX SUBROUTINES-MATRIX,
C INITIA,NEWBAS,SORT,PIVØT AND PPRINT IN PRØPER ØRDER.
C
      CØMMON AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSION AM(50,50), Q(50), B(50,50), A(50)
      DIMENSION W(50), Z(50), MBASIS(100)
C DESCRIPTION ØF PARAMETERS IN COMMON
C   AM       A TWØ DIMENSIONAL ARRAY CONTAINING THE
C            ELEMENTS ØF MATRIX M.
C   Q        A SINGLY SUBSCRIPTED ARRAY CONTAINING THE
C            ELEMENTS ØF VECTOR Q.
C   L1       AN INTEGER VARIABLE INDICATING THE NUMBER ØF
C            ITERATIONS TAKEN FØR EACH PROBLEM.
C   B        A TWO DIMENSIØNAL ARRAY CØNTAINING THE
C            ELEMENTS OF THE INVERSE ØF THE CURRENT BASIS.
C   W        A SINGLY SUBSCRIPTED ARRAY CØNTAINING THE VALUES
C            ØF W VARIABLES IN EACH SOLUTION.
C   Z        A SINGLY SUBSCRIPTED ARRAY CØNTAINING THE VALUES
C            OF Z VARIABLES IN EACH SOLUTIØN.
C   NL1      AN INTEGER VARIABLE TAKING VALUE 1 OR 2 DEPEND-
C            ING ON WHETHER VARIABLE W ØR Z LEAVES THE BASIS
C   NE1      SIMILAR TO NL1 BUT INDICATES VARIABLE ENTERING
C   NL2      AN INTEGER VARIABLE INDICATING WHAT CØMPONENT
C            ØF W ØR Z VARIABLE LEAVES THE BASIS.
C   NE2      SIMILAR TO NL2 BUT INDICATES VARIABLE ENTERING
C   A        A SINGLY SUBSCRIPTED ARRAY CØNTAINING THE
C            ELEMENTS ØF THE TRANSFORMED CØLUMN THAT IS
C            ENTERING THE BASIS.
C   IR       AN INTEGER VARIABLE DENØTING THE PIVØT RØW AT
C            EACH ITERATIØN. ALSO USED TO INDICATE TERMINA-
C            TION ØF A PRØBLEM BY GIVING IT A VALUE ØF 1000.
C   MBASIS   A SINGLY SUBSCRIPTED ARRAY-INDICATOR FØR THE
C            BASIC VARIABLES. TWØ INDICATØRS ARE USED FØR
C            EACH BASIC VARIABLE-ØNE INDICATING WHETHER
C            IT IS A W ØR Z AND ANØTHER INDICATING WHAT
C            CØMPONENT ØF W ØR Z.
C
C READ IN THE VALUE ØF VARIABLE IP INDICATING THE
C NUMBER ØF PRØBLEMS TØ BE SOLVED.
      READ(5,3) IP
C VARIABLE NØ INDICATES THE CURRENT PRØBLEM BEING SOLVED
      NØ=0
    1 NØ=NØ+1
      IF (NØ.GT.IP) GØ TØ 5
      WRITE(6,2) NØ
    2 FØRMAT (1H1,10X,11HPRØBLEM NØ.,I2)
C
C READ IN THE SIZE ØF THE MATRIX M
      READ(5,3) N
    3 FØRMAT (I2)
C PRØGRAM CALLING SEQUENCE
      CALL MATRIX (N)
C PARAMETER N INDICATES THE PRØBLEM SIZE
      CALL INITIA (N)
C SINCE FØR ANY PRØBLEM TERMINATIØN CAN ØCCUR IN INITIA,
C NEWBAS ØR SØRT SUBRØUTINE,THE VALUE ØF IR IS MATCHED WITH
C 1000 TØ CHECK WHETHER TØ CØNTINUE ØR GØ TØ NEXT PRØBLEM.
      IF (IR.EQ.1000) GØ TØ 1
    4 CALL NEWBAS (N)
      IF (IR.EQ.1000) GØ TØ 1
      CALL SØRT (N)
      IF (IR.EQ.1000) GØ TØ 1
      CALL PIVØT (N)
      GØ TØ 4
    5 STØP
      END
      SUBRØUTINE MATRIX (N)
C PURPØSE - TØ INITIALIZE AND READ IN THE VARIØUS INPUT DATA
C
      CØMMØN AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
C READ THE ELEMENTS ØF M MATRIX CØLUMN BY CØLUMN
      DØ 1 J=1,N
    1   READ(5,2) (AM(I,J),I=1,N)
    2 FØRMAT (7F10.5)
C READ THE ELEMENTS ØF Q VECTØR
      READ(5,2) (Q(I),I=1,N)
C IN ITERATIØN 1,BASIS INVERSE IS AN IDENTITY MATRIX.
      DØ 5 J=1,N
        DØ 4 I=1,N
          IF (I.EQ.J) GØ TØ 3
          B(I,J)=0.0
          GØ TØ 4
    3     B(I,J)=1.0
    4     CØNTINUE
    5   CØNTINUE
      RETURN
      END
      SUBRØUTINE INITIA (N)
C PURPØSE-TØ FIND THE INITIAL ALMØST CØMPLEMENTARY SØLUTIØN
C         BY ADDING AN ARTIFICIAL VARIABLE Z0.
C
      CØMMØN AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
C SET Z0 EQUAL TØ THE MØST NEGATIVE Q(I)
      I=1
      J=2
    1 IF (Q(I).LE.Q(J)) GØ TØ 2
      I=J
    2 J=J+1
      IF (J.LE.N) GØ TØ 1
C UPDATE Q VECTØR
      IR=I
      T1=-Q(IR)
      IF (T1.LE.0.0) GØ TØ 9
      DØ 3 I=1,N
        Q(I)=Q(I)+T1
    3   CØNTINUE
      Q(IR)=T1
C UPDATE BASIS INVERSE AND INDICATØR VECTØR
C ØF BASIC VARIABLES.
      DØ 4 J=1,N
        B(J,IR)=-1.0
        W(J)=Q(J)
        Z(J)=0.0
        MBASIS(J)=1
        L=N+J
        MBASIS(L)=J
    4   CØNTINUE
      NL1=1
      L=N+IR
      NL2=IR
      MBASIS(IR)=3
      MBASIS(L)=0
      W(IR)=0.0
      Z0=Q(IR)
      L1=1
C PRINT THE INITIAL ALMØST CØMPLEMENTARY SØLUTIØN
      WRITE(6,5)
    5 FØRMAT (3(/),5X,29HINITIAL ALMØST CØMPLEMENTARY ,
     1   8HSØLUTIØN)
      DØ 7 I=1,N
        WRITE(6,6) I,W(I)
    6   FØRMAT (10X,2HW(,I4,2H)=,F15.5)
    7   CØNTINUE
      WRITE(6,8) Z0
    8 FØRMAT (10X,3HZ0=,F15.5)
      RETURN
    9 WRITE (6,10)
   10 FØRMAT (5X,36HPRØBLEM HAS A TRIVIAL CØMPLEMENTARY ,
     1   23HSØLUTIØN WITH W=Q, Z=0.)
      IR=1000
      RETURN
      END
      SUBRØUTINE NEWBAS (N)
C PURPØSE - TØ FIND THE NEW BASIS CØLUMN TØ ENTER IN
C           TERMS ØF THE CURRENT BASIS.
C
      CØMMØN AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
C IF NL1 IS NEITHER 1 NØR 2 THEN THE VARIABLE Z0 LEAVES THE
C BASIS INDICATING TERMINATIØN WITH A CØMPLEMENTARY SØLUTIØN
      IF (NL1.EQ.1) GØ TØ 2
      IF (NL1.EQ.2) GØ TØ 5
      WRITE(6,1)
    1 FØRMAT (5X,22HCØMPLEMENTARY SØLUTIØN)
      CALL PPRINT (N)
      IR=1000
      RETURN
    2 NE1=2
      NE2=NL2
C UPDATE NEW BASIC CØLUMN BY MULTIPLYING BY BASIS INVERSE.
      DØ 4 I=1,N
        T1=0.0
        DØ 3 J=1,N
    3     T1=T1-B(I,J)*AM(J,NE2)
        A(I)=T1
    4   CØNTINUE
      RETURN
    5 NE1=1
      NE2=NL2
      DØ 6 I=1,N
        A(I)=B(I,NE2)
    6   CØNTINUE
      RETURN
      END
      SUBRØUTINE SØRT (N)
C PURPØSE - TØ FIND THE PIVØT RØW FØR NEXT ITERATIØN BY THE
C           USE ØF (SIMPLEX-TYPE) MINIMUM RATIØ RULE.
C
      CØMMØN AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
      I=1
    1 IF (A(I).GT.0.0) GØ TØ 2
      I=I+1
      IF (I.GT.N) GØ TØ 6
      GØ TØ 1
    2 T1=Q(I)/A(I)
      IR=I
    3 I=I+1
      IF (I.GT.N) GØ TØ 5
      IF (A(I).GT.0.0) GØ TØ 4
      GØ TØ 3
    4 T2=Q(I)/A(I)
      IF (T2.GE.T1) GØ TØ 3
      IR=I
      T1=T2
      GØ TØ 3
    5 RETURN
C FAILURE ØF THE RATIØ RULE INDICATES TERMINATIØN WITH
C NØ CØMPLEMENTARY SØLUTIØN.
    6 WRITE(6,7)
    7 FØRMAT (5X,37HPRØBLEM HAS NØ CØMPLEMENTARY SØLUTIØN)
      WRITE(6,8) L1
    8 FØRMAT (10X,13HITERATIØN NØ.,I4)
      IR=1000
      RETURN
      END
      SUBRØUTINE PIVØT (N)
C PURPØSE - TØ PERFØRM THE PIVOT OPERATIØN BY UPDATING THE
C           INVERSE ØF THE BASIS AND Q VECTOR.
C
      CØMMON AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
      DØ 1 I=1,N
    1   B(IR,I)=B(IR,I)/A(IR)
      Q(IR)=Q(IR)/A(IR)
      DØ 3 I=1,N
        IF (I.EQ.IR) GØ TØ 3
        Q(I)=Q(I)-Q(IR)*A(I)
        DØ 2 J=1,N
          B(I,J)=B(I,J)-B(IR,J)*A(I)
    2     CØNTINUE
    3   CØNTINUE
C UPDATE THE INDICATØR VECTØR ØF BASIC VARIABLES
      NL1=MBASIS(IR)
      L=N+IR
      NL2=MBASIS(L)
      MBASIS(IR)=NE1
      MBASIS(L)=NE2
      L1=L1+1
      RETURN
      END
      SUBRØUTINE PPRINT (N)
C PURPØSE - TØ PRINT THE CURRENT SØLUTION TØ COMPLEMENTARY
C           PRØBLEM AND THE ITERATIØN NUMBER.
C
      CØMMØN AM,Q,L1,B,NL1,NL2,A,NE1,NE2,IR,MBASIS,W,Z
      DIMENSIØN AM(50,50), Q(50), B(50,50), A(50)
      DIMENSIØN W(50), Z(50), MBASIS(100)
      WRITE(6,1) L1
    1 FØRMAT (10X,13HITERATIØN NØ.,I4)
      I=N+1
      J=1
    2 K1=MBASIS(I)
      K2=MBASIS(J)
      IF (Q(J).GE.0.0) GØ TØ 3
      Q(J)=0.0
    3 IF (K2.EQ.1) GØ TØ 5
      WRITE(6,4) K1,Q(J)
    4 FØRMAT (10X,2HZ(,I4,2H)=,F15.5)
      GØ TØ 7
    5 WRITE(6,6) K1,Q(J)
    6 FØRMAT (10X,2HW(,I4,2H)=,F15.5)
    7 I=I+1
      J=J+1
      IF (J.LE.N) GO TO 2
      RETURN
      END
```

Editor's note: *Algorithm 432 described here is available on magnetic tape from the Department of Computer Science, University of Colorado, Boulder, CO 80302. The cost for the tape is $16.00 (U.S. and Canada) or $18.00 (elsewhere). If the user sends a small tape (wt. less than 1 lb.) the algorithm will be copied on it and returned to him at a charge of $10.00 (U.S. only). All orders are to be prepaid with checks payable to ACM Algorithms. The algorithm is recorded as one file of BCD 80 character card images at 556 B.P.I., even parity, on seven track tape. We will supply the algorithm at a density of 800 B.P.I. if requested. The cards for the algorithm are sequenced starting at 10 and incremented by 10. The sequence number is right justified in colums 80. Although we will make every attempt to insure that the algorithm conforms to the description printed here, we cannot guarantee it, nor can we guarantee that the algorithm is correct.—L.D.F.*

Algorithm 432

Solution of the Matrix Equation AX + XB = C [F4]

R.H. Bartels and G.W. Stewart [Recd. 21 Oct. 1970 and 7 March 1971]
Center for Numerical Analysis, The University of Texas at Austin, Austin, TX 78712

Key Words and Phrases: linear algebra, matrices, linear equations
CR Categories: 5.14
Language: Fortran

Description

The following programs are a collection of Fortran IV subroutines to solve the matrix equation

$$AX + XB = C \qquad (1)$$

where A, B, and C are real matrices of dimensions $m \times m$, $n \times n$, and $m \times n$, respectively. Additional subroutines permit the efficient solution of the equation

$$A^TX + XA = C, \qquad (2)$$

where C is symmetric. Equation (1) has applications to the direct solution of discrete Poisson equations [2].

It is well known that (1) has a unique solution if and only if the eigenvalues $\alpha_1, \alpha_2, \dots, \alpha_m$ of A and $\beta_1, \beta_2, \dots, \beta_n$ of B satisfy

$$\alpha_i + \beta_j \neq 0 \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, n).$$

One proof of the result amounts to constructing the solution from complete systems of eigenvalues and eigenvectors of A and B, when they exist. This technique has been proposed as a computational method (e.g. see [1]); however, it is unstable when the eigensystem is ill conditioned. The method proposed here is based on the Schur reduction to triangular form by orthogonal similarity transformations.

Equation (1) is solved as follows. The matrix A is reduced to lower real Schur form A' by an orthogonal similarity transformation U; that is A is reduced to the real, block lower triangular form.

$$A' = U^TAU = \begin{bmatrix} A'_{11} & & & \bigcirc \\ A'_{21} & A'_{22} & & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ A'_{p1} & A'_{p2} & \cdots & A'_{pp} \end{bmatrix},$$

where each matrix A'_{ii} is of order at most two. Similarly B is reduced to upper real Schur form by the orthogonal matrix V:

$$B' = V^TBV = \begin{bmatrix} B'_{11} & B'_{12} & \cdots & B'_{1q} \\ & B'_{22} & \cdots & B'_{2q} \\ & & \cdot & \cdot \\ & & \cdot & \cdot \\ & & \cdot & \cdot \\ & \bigcirc & & B'_{qq} \end{bmatrix},$$

This research was supported in part by Grant DA-ARO(D)-31-124-G721, Army Research Office, Durham, and by National Science Foundation Grant GP-5253 awarded to The University of Texas at Austin.