View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by idUS. Depósito de Investigación Universidad de Sevilla

CAPÍTULO V

Resolución del problema lineal múltiple con ADBASE

M. A. Hinojosa   A. Mármol

El presente capítulo contiene las características básicas del software ADBASE, desarrollado por el Prof. Ralph Steuer para el tratamiento y resolución del problemas lineales multiobjetivo. Es apropiado, por tanto, para la obtención de soluciones de juegos matriciales multicriterio, como se ha expuesto en los capítulos precedentes. En Steuer (1986) [88] y en el manual de ADBASE (Steuer, 1995 [89]) puede verse información más extensa sobre este programa.

1. Objetivo del programa ADBASE

ADBASE es un programa Fortran, creado para la enumeración de los puntos extremos eficientes y las aristas no acotadas eficientes de problemas lineales multiobjetivo. Básicamente resuelve problemas de la forma

$$\max\{(Cx + a) = z, x \in S\},$$

donde C es la matriz de criterios, $z \in \mathbb{R}^n$ el vector de objetivos y S la región factible

$$S = \{x \geq 0 : A_k x \leq b_k,\ A_e x = b_e,\ A_s x \geq b_s\}$$

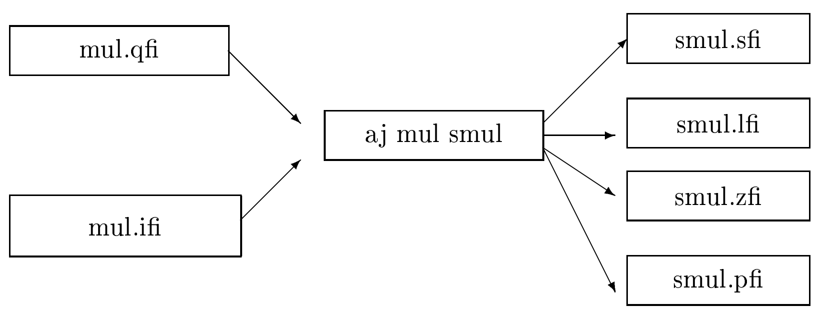
con los coeficientes del lado derecho de las restricciones b_k, b_e, b_s no negativos.

2. Estructura

Para la utilización del ADBASE en microordenador es necesario un editor para la preparación de ficheros. No hacemos referencia a un editor específico pues pueden servir la mayoría de los editores disponibles, así como los programas de tratamiento de texto, teniendo precaución en este caso de archivarlos como ficheros de texto para evitar los códigos internos del programa.

Para ejecutar ADBASE es necesario crear un fichero QFILE (con la extensión .qfi), y en la mayoría de los casos un fichero IFILE (extensión .ifi), en los que se recoge la información necesaria para resolver el problema. La salida aparece en un SFILE (.sfi) y algunas veces en un LFILE (.lfi), ZFILE (.zfi), y PFILE (.pfi).

Para ejecutarlo se da la orden AJ 'NOMBRE1' 'NOMBRE2', donde 'NOMBRE1' 'NOMBRE2' representan respectivamente los nombres asignados a los ficheros de entrada y de salida. Si se ejecuta el comando AJ MUL SMUL, la operación que realiza ADBASE se representa como sigue:



ADBASE trabaja en dos modos distintos:

MODE = 1: Modo Normal.

En MODE=1 el problema a resolver se manda al ADBASE desde un IFILE. En este modo, pueden resolverse los siguientes tipos de problemas:

a) Problemas lineales multiobjetivo:

$$\max\{(Cx + a) = z, x \in S\}$$

Utilizando el cono de criterios original, se obtienen todos los puntos extremos eficientes y todas las aristas no acotadas eficientes. Se realiza con IFASE0=0, IFASE2=1,2,3,4, o 5, y IFASE=2.

b) Problemas lineales multiobjetivo con intervalos en los pesos:

$$\max\{\lambda^t(Cx+a), x \in S\}$$

con $\lambda \in \mathbb{R}^n$, $\quad \lambda_i \in (l_i, u_i), \quad 0 \leq l_i \leq u_i \leq 1, \forall i = 1, ..., n, \quad \sum_{i=1}^n \lambda_i = 1.$

Utilizando reducciones del cono de criterios, se generan subconjuntos del conjunto de puntos extremos eficientes. Se hace especificando los límites superiores e inferiores de los intervalos para los pesos, y con IFASE0=1 o 2, IFASE2=1,2,3,4, o 5, y IFASE3=2.

c) Problemas ponderados:

$$\max\{\lambda^t(Cx+a), x \in S\}$$

con λ fijo.

Se resuelve dando intervalos sobre los pesos de forma que $l_i = u_i, i = 1, ..., n$, con IFASE0=1, IFASE2=1,2,3,4 o 5, y IFASE3=0. Si se quieren todos los puntos extremos óptimos, IFASE3=1.

d) Problemas lineales uniobjetivo:

$$\max\{cx, x \in S\}$$

Se resuelve haciendo NOBJS=1, IFASE0=0, IFASE2=1,2,3,4 o 5, y IFASE3=0. Si se quieren obtener todos los puntos extremos y aristas no acotadas óptimas, IFASE3=2.

MODE = 2: Modo generador de problemas aleatorios.

En MODE = 2 los problemas no se mandan al ADBASE desde un IFILE. Se generan aleatoriamente en el ADBASE y se resuelven. Sólo pueden generarse aleatoriamente problemas con restricciones menor o igual. Este modo es útil para realizar experimentos computacionales.

ADBASE trabaja de acuerdo con las cuatro fases siguientes:

FASE 0: Utilización del cono de criterios original (cono convexo generado por las filas de C) o contracción a un subconjunto suyo.

FASE I: Obtención de un punto extremo inicial de la región factible. Si no existe ninguno, el problema acaba con el mensaje SISTEM OF CONSTRAINTS IS INCONSISTENT.

FASE II: Obtención de una base eficiente inicial. Si no encuentra ninguna y el número de objetivos es 1, el problema acaba con el mensaje PROBLEM HAS UNBOUNDED OBJECTIVE FUNCTION VALUE. Si no puede encontrar una base eficiente y el número de objetivos es mayor que 1, el problema termina con el mensaje UNABLE TO FIND AN EFFICIENT EXTREME POINT.

FASE III: Después de hallar una base eficiente inicial, pivota a partir de ella para obtener todos los demás extremos eficientes.

3. Ficheros de entrada y salida.

3.1. Informacion del QFILE(.qfi):

Cada ejecución del ADBASE requiere un QFILE. En este fichero se fijan las opciones y características que son comunes a diversas ejecuciones del programa. Un ejemplo de QFILE es el siguiente:

```
***__****1********______****3****4***** ADBASE MODE= 1 SECTION
    1. NUMB            1   (NUMBER OF PROBLEMS TO BE SOLVED)
----------------------------
    2. MODE            1   (REGULAR OR RANDOM PROBLEM MODE) 1,2
----------------------------
    3. IFASE2          2   (PHASE II OPTION) 1 TO 5
    4. IFASE3          2   (PHASE III OPTION) 0,1,2
    5. IZFMT           3   (EXPONENTIAL/FIXED FORMAT IN ZFILE) 0 TO 6
    6. IWEAK           0   (EFFICIENT OR WEAKLY-EFFICIENT) 0,1
    7. MLISTB       3000   (MAXIMUM NUMBER OF EFFICIENT BASES) <3000
    8. FV            3.0   (RE-INVERSION TIMES M3 FACTOR) >2
```

```
----------------------------
   9. IPRINT(1)        0   (OBVIOUS ERRORS) 0,1
  10. IPRINT(2)        1   (PROBLEM COEFFICIENTS) 0,1
  11. IPRINT(3)        4   (NOTHING/BASES/EXTR PTS) 0/1,2,3/4,5,6
  12. IPRINT(4)        1   (EFFICIENCY TOTALS) 0,1
  13. IPRINT(5)        0   (INDIVIDUAL PROBLEM DATA) 0,1
  14. IPRINT(6)        0   (CUMULATIVE DATA) 0,1
  15. IPRINT(7)        0   (CODE LISTS) 0,1
  16. IPRINT(8)        0   (ZFILE) 0,1
  17. IPRINT(9)        0   (REDUCED COSTS AND TABLEAUS) 0,1,2
  18.   IV9L           1   (BEGINNING TABLEAU VARIABLE)
  19.   IV9U         100   (ENDING TABLEAU VARIABLE)
  20.   I9L            0   (TABLEAUS BEGIN AT THIS BASIS)
  21.   I9U         9999   (TABLEAUS END AT THIS BASIS)
  22. IPRINT(10)       0   (LFILE) 0,1
  23.   I10L           0   (LFILE BEGINS ON WAY TO THIS BASIS)
  24.   I10U         100   (LFILE ENDS AT THIS BASIS)
  25. IPRINT(11)       0   (PREMULTIPLICATION T-MATRIX) 0,1
```

3.2. Información del IFILE (.ifi):

El IFILE contiene los datos concretos del problema a resolver. Si se quiere resolver más de un problema en una sola ejecución puede hacerse agrupando todos los ficheros de datos en un único IFILE.

Un fichero de datos IFILE contiene 5 tipos de registros.

1. Título del problema: Es el primer registro de cada fichero de datos de un problema, donde se especifica un título literal para el problema. Debe estar presente aunque esté vacío. Se sitúa en las columnas 2-72.

2. Cabecera: Es el segundo registro de cada fichero de datos. Tiene ocho campos donde se especifican las siguientes cantidades:

NOMB: Número del problema (no puede ser mayor que 9999).

NOBJS: Número de objetivos.

N1: Número de variables estructurales.

IK: Número de restricciones $\leq$.

IE: Número de restricciones $=$.

IS: Número de restricciones $\geq$.

IFASE0: Tipo de cono de criterios. Cono original IFASE0=0. Cono con intervalos en los pesos IFASE0=1. Cono envolvente reducido IFASE0=2.

NGRAYS: Con IFASE0=1, NGRAYS especifica el límite superior de generadores del cono de criterios antes de tomar la opción IFASE0=2. Si IFASE0 tiene otro valor, el valor de NGRAYS es ignorado.

Los datos deben estar situados en las siguientes columnas:

NOMB	NOBJS	N1	IK	IE	IS	IFASE0	NGRAYS
1-8	9-16	17-24	25-32	33-40	41-48	49-56	57-64

3. Contadores: Cada fichero de datos tiene ocho contadores correspondientes al número de coeficientes no nulos de $A_k, b_k, A_e, b_e, A_s, b_s$, C, y constantes a. Deben incluirse los 8 contadores incluso si no van seguidos de coeficientes no nulos. Se especifican en las columnas 1-8.

4. Registros de coeficientes no nulos: Especifican los valores de los coeficientes no nulos de $A_k, b_k, A_e, b_e, A_s, b_s, C$, y a. Cada registro tiene cuatro particiones compuestas de 3 subcampos. En los dos primeros se especifican los índices de la fila y la columna del coeficiente correspondiente, en el tercer subcampo se especifica el valor no nulo del coeficiente. Para los coeficientes de b_k, b_e, b_s , y a, el segundo subcampo se deja blanco, pues es innecesaria su designación.

Los datos de este registro se sitúan en las siguientes columnas:

Fila	Columna	Coef.no nulo	....	Fila	Columna	Coef.no nulo
1-3	4-6	7-18	....	55-57	58-60	61-72

5. Registro de intervalos de los pesos: (opcional) Este registro se incluye si y sólo si IFASE0 es 1 o 2 en la cabecera. Se especifican las cotas l_i, u_i de los intervalos de los pesos, con el formato:

i	l_i	u_i
1-8	11-22	23-34

De esta forma, el fichero de datos de un problema (.ifi) se configura como sigue:

Registro de título.

Registro cabecera.

Contador para A_k .

} Registro de coeficientes (si existen) para A_k .

Contador para b_k .

} Registro de coeficientes (si existen) para b_k .

Contador para A_e .

} Registro de coeficientes (si existen) para A_e .

Contador para b_e .

} Registro de coeficientes (si existen) para b_e .

Contador para A_s .

} Registro de coeficientes (si existen) para A_s .

Contador para b_s .

} Registro de coeficientes (si existen) para b_s .

Contador para C.

} Registro de coeficientes (si existen) para C.

Contador para a.

} Registro de coeficientes (si existen) para a.

Registro de intervalos de pesos (Si existen).

Ejemplo 5.1 *Consideremos el siguiente problema de maximización vectorial:*

$$\begin{array}{ll} \max & \{x+y+z, 4x+2y+z, -2x-y+z\} \\ s.a. & 2x+5y+z \leq 1600 \\ & 2x+2y+3z \leq 1200 \\ & x,y,z \geq 0 \end{array}$$

El fichero IFILE para este problema lo hemos denominado mul.ifi, y se muestra a continuación. Obsérvese que el problema no tiene restricciones de de igualdad ni de mayor o igual, ni constantes a, por lo que aparece 0 *en los contadores correspondientes.*

```
PROBLEMA MULTIOBJETIVO
       1       3       3       2       0       0       0      40
       6
  1  1  2          1  2  5          1  3  1          2  1  2
  2  2  2          2  3  3
       2
  1     1600       2     1200
       0
       0
       0
       0
       9
  1  1  1          1  2  1          1  3  1          2  1  4
  2  2  2          2  3  1          3  1  -2         3  2  -1
  3  3  1
       0
```

Con el comando AJ MUL SMUL se ejecuta ADBASE generando el fichero smul.sfi y opcionalmente smul.lfi y smul.zfi.

3.3. Informacion del SFILE:(.sfi)

En este fichero ADBASE escribe los puntos extremos eficientes y las aristas no acotadas eficientes. Tiene la opción de escribir también los costes reducidos y las tablas correspondientes.

Ejemplo 5.2 *Para el problema anterior, un fichero de salida smul.sfi es:*

```
**1*******2*******3*******4*******5*******6*******7**

         1. NUMB..........      1
         2. MODE..........      1
         3. IFASE2........      2
         4. IFASE3........      2
         5. IZFMT.........      3
         6. IWEAK.........      0
         7. MLISTB........   3000
```

```
      8. FV...........      3.000
      9. IPRINT(1).....      0
     10. IPRINT(2).....      1
     11. IPRINT(3).....      4
     12. IPRINT(4).....      1
     13. IPRINT(5).....      0
     14. IPRINT(6).....      0
     15. IPRINT(7).....      0
     16. IPRINT(8).....      0
     17. IPRINT(9).....      0
     18.   IV9L........      1
     19.   IV9U........    100
     20.   I9L.........      0
     21.   I9U.........   9999
     22. IPRINT(10)....      0
     23.   I10L........      0
     24.   I10U........    100
     25. IPRINT(11)....      0

**1*******2*******3*******4*******5*******6*******7**

                          PROBLEM NO.    1

PROBLEMA MULTIOBJETIVO

NOBJS...   3
N1......   3
IK......   2
IE......   0
IS......   0
IFASE0..   0
NGRAYS..  40

A(  1,  1) =       2.000000
A(  1,  2) =       5.000000
A(  1,  3) =       1.000000
```

```
A(  2,  1) =       2.000000
A(  2,  2) =       2.000000
A(  2,  3) =       3.000000

B(  1) =    1600.000000
B(  2) =    1200.000000

C(  1,  1) =       1.000000
C(  1,  2) =       1.000000
C(  1,  3) =       1.000000
C(  2,  1) =       4.000000
C(  2,  2) =       2.000000
C(  2,  3) =       1.000000
C(  3,  1) =      -2.000000
C(  3,  2) =      -1.000000
C(  3,  3) =       1.000000


EXTREME                                        NONZERO STRUCTURAL
 POINT         CRITERION VALUES              BASIC VARIABLE VALUES

    1       Z( 1) =       600.000000        X(   1) =  600.000000
            Z( 2) =      2400.000000
            Z( 3) =     -1200.000000

    2       Z( 1) =       600.000000        X(   1) =  466.666667
            Z( 2) =      2133.333333        X(   2) =  133.333333
            Z( 3) =     -1066.666667

    3       Z( 1) =       400.000000        X(   3) =  400.000000
            Z( 2) =       400.000000
            Z( 3) =       400.000000

    4       Z( 1) =       492.307692        X(   2) =  276.923077
            Z( 2) =       769.230769        X(   3) =  215.384615
            Z( 3) =       -61.538462
```

```
NUMBER OF EFFICIENT BASES VISITED        =    4
NUMBER OF EFFICIENT EXTREME POINTS       =    4
NUMBER OF UNBOUNDED EFFICIENT EDGES      =    0

**1*******2*******3*******4*******5*******6*******7**
```

Por tanto el problema tiene cuatro soluciones eficientes extremas que son:

$$(600, 0, 0), (1400/3, 400/3, 0), (0, 0, 400), (0, 3600/13, 2800/13)$$

y los valores que toman los objetivos en estos puntos vienen dados en el vector z.

Con las opciones IPRINT(3)=3 y IPRINT(9) = 1 en el QFILE del problema aparecen en el fichero de salida SFILE, además de las soluciones eficientes extremas, los costes reducidos correspondientes. Esta información es útil para realizar análisis de sensibilidad. A continuación se muestra un fichero SFILE para este problema con los costes reducidos, en el que se ha eliminado la impresión de los coeficientes del problema con la opción IPRINT(2) = 0.

```
**1*******2*******3*******4*******5*******6*******7**

                            PROBLEM NO.    1

 PROBLEMA MULTIOBJETIVO


                                                       ALL
  BASIS          CRITERION VALUES              BASIC VARIABLE VALUES


     1       Z( 1) =       600.000000        X(   1) =   600.000000
             Z( 2) =      2400.000000        X(   4) =   400.000000
             Z( 3) =     -1200.000000

     C(J)-Z(J) REDUCED COSTS
```

```
      2            3            5
-----------------------------------------------------------
    .00000      -.50000      -.50000
  -2.00000     -5.00000     -2.00000
   1.00000      4.00000      1.00000
-----------------------------------------------------------


2       Z( 1) =      600.000000       X(  1) =  466.666667
        Z( 2) =     2133.333333       X(  2) =  133.333333
        Z( 3) =    -1066.666667

C(J)-Z(J) REDUCED COSTS
      3            4            5
-----------------------------------------------------------
   -.50000       .00000      -.50000
  -6.33333       .66667     -2.66667
   4.66667      -.33333      1.33333
-----------------------------------------------------------


3       Z( 1) =      400.000000       X(  3) =  400.000000
        Z( 2) =      400.000000       X(  4) = 1200.000000
        Z( 3) =      400.000000

C(J)-Z(J) REDUCED COSTS
      1            2            5
-----------------------------------------------------------
    .33333       .33333      -.33333
   3.33333      1.33333      -.33333
  -2.66667     -1.66667      -.33333
-----------------------------------------------------------


4       Z( 1) =      492.307692       X(  2) =  276.923077
        Z( 2) =      769.230769       X(  3) =  215.384615
```

```
        Z( 3) =       -61.538462

    C(J)-Z(J) REDUCED COSTS
          1             4             5
    ------------------------------------------------------------
        .23077       -.07692       -.30769
       2.92308       -.30769       -.23077
      -2.15385        .38462       -.46154
    ------------------------------------------------------------

 NUMBER OF EFFICIENT BASES VISITED         =     4
 NUMBER OF EFFICIENT EXTREME POINTS        =     4
 NUMBER OF UNBOUNDED EFFICIENT EDGES       =     0


**1*******2*******3*******4*******5*******6*******7**
```

3.4. Información del LFILE:(.lfi)

La creación de este fichero de salida es opcional. Si elegimos la opción IPRINT(10) = 1 en el QFILE correspondiente, se crea un LFILE que contiene la información sobre los pivoteos realizados para obtener los extremos y las aristas eficientes.

3.5. Información del ZFILE:(.zfi)

La creación de este fichero también es opcional. Con IPRINT(8) = 1, ADBASE escribe en un ZFILE los vectores de criterios de cada uno de los puntos extremos eficientes calculados.

3.6. Información del PFILE:(.pfi)

En MODE = 2 y con IPRINT(12)=1, ADBASE escribe los problemas generados aleatoriamente en un PFILE en formato de entrada ADBASE.

Ejemplo 5.3 *Usaremos el programa ADBASE para obtener las estrategias de seguridad Pareto-óptimas del juego vectorial del ejemplo 2.1, cuya matriz de* *pagos es*

$$\begin{pmatrix} (1,3) & (2,1) \\ (3,1) & (1,2) \\ (1,1) & (3,3) \end{pmatrix}$$

El conjunto de las estrategias de seguridad Pareto-óptimas para el jugador I, es el conjunto de soluciones eficientes del problema lineal bicriterio

$$\begin{array}{ll} \max & v_1, v_2 \\ s.a. & x_1 + 3x_2 + x_3 \geq v_1 \\ & 2x_1 + x_2 + 3x_3 \geq v_1 \\ & 3x_1 + x_2 + x_3 \geq v_2 \\ & x_1 + 2x_2 + 3x_3 \geq v_2 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \\ & v_1, v_2 \in \mathbb{R} \end{array}$$

El fichero de datos LFILE para este problema es

```
JUEGO VECTORIAL
       6         2         5         0         1         4         0        40
       0
       0
       3
  1   1   1           1   2   1           1   3   1
       1
  1       1
      16
  1   1   1           1   2   3           1   3   1           1   4  -1
  2   1   2           2   2   1           2   3   3           2   4  -1
  3   1   3           3   2   1           3   3   1           3   5  -1
  4   1   1           4   2   2           4   3   3           4   5  -1
       0
       2
  1   4   1           2   5   1
```

Un fichero SFILE para este problema es

```
**1*******2*******3*******4*******5*******6*******7**

                            PROBLEM NO.    6

 JUEGO VECTORIAL

 EXTREME                                              NONZERO STRUCTURAL
  POINT          CRITERION VALUES                   BASIC VARIABLE VALUES

     1       Z( 1) =          2.000000          X(   2) =      .500000
             Z( 2) =          1.000000          X(   3) =      .500000
                                                X(   4) =     2.000000
                                                X(   5) =     1.000000

     2       Z( 1) =          1.800000          X(   1) =      .400000
             Z( 2) =          1.800000          X(   2) =      .400000
                                                X(   3) =      .200000
                                                X(   4) =     1.800000
                                                X(   5) =     1.800000

     3       Z( 1) =          1.000000          X(   1) =      .500000
             Z( 2) =          2.000000          X(   3) =      .500000
                                                X(   4) =     1.000000
                                                X(   5) =     2.000000

 NUMBER OF EFFICIENT BASES VISITED         =     3
 NUMBER OF EFFICIENT EXTREME POINTS        =     3
 NUMBER OF UNBOUNDED EFFICIENT EDGES       =     0


**1*******2*******3*******4*******5*******6*******7**
```

Este fichero proporciona las estrategias de seguridad Pareto-óptimas extremas del juego bicriterio, así como los niveles de seguridad correspondientes:

$(v_1, x_1) = (2, 1; 0, 1/2, 1/2)$,

$(v_2, x_2) = (9/5, 9/5; 2/5, 2/5, 1/5)$,

$(v_3, x_3) = (1, 2; 1/2, 0, 1/2)$.