

Sistemas homogéneos de ecuaciones diofánticas. Resolución y aplicación al ajuste automático de reacciones químicas

ALFONSO RECUERO, Dr. Ing. Caminos, Canales y Puertos
 J. PEDRO GUTIERREZ, Ing. Caminos, Canales y Puertos
 (IETCC/CSIC)

RESUMEN

Se presenta un algoritmo de resolución de sistemas de ecuaciones diofánticas lineales y homogéneas. Seguidamente y como ejemplo de aplicación se describe un método de búsqueda heurística de una solución óptima que cumpla unas ciertas condiciones dentro de un espacio infinito de soluciones de varias dimensiones. Por último se describe la aplicación del algoritmo y del procedimiento de búsqueda heurística para el caso de ajuste de reacciones químicas. Se incluye el listado en BASIC del programa para el ordenador HP-75

SUMMARY

An algorithm to solve systems of linear homogeneous diophantine equations is given here. Subsequently, as an example of application, a method of heuristic search for the best solution is described; this fulfils certain conditions within an infinite space of solutions of various dimensions. Finally, an application of the algorithm and of the process of heuristic search for the best solution in adjusting chemical reactions is described. The listing in BASIC of the program of application mentioned is included, for the HP-75 portable computer.

INTRODUCCION

Se denominan diofánticas a aquellas ecuaciones de coeficientes enteros que admiten infinitas soluciones de las que nos interesan solamente las que corresponden a valores enteros de las variables. Tendremos un sistema de tales ecuaciones cuando deban verificarse simultáneamente varias ecuaciones de este tipo.

Esta clase de sistemas de ecuaciones se presentan, por ejemplo, en aquellos casos en los que se desea obtener un cierto número de conjuntos a partir de otro grupo de conjuntos de forma que todos y cada uno de los componentes de los conjuntos de entrada estén en los de salida. El ejemplo más típico es el ajuste de reacciones químicas, en las que a partir de unas ciertas moléculas de entrada se obtienen otras de salida de modo que todos los átomos de las moléculas de entrada estén en las de salida.

En el presente trabajo se describe un método general de resolución de sistemas homogéneos de ecuaciones diofánticas, incluyendo un procedimiento heurístico para la búsqueda de una solución óptima cuando el conjunto de soluciones tiene varios grados de libertad, así como la implementación del mismo en un programa escrito en BASIC para la calculadora HP-75 C.

Como ejemplo de aplicación se utiliza el ya citado de ajuste automático de reacciones químicas incluyéndose un programa en BASIC que realiza dicho ajuste.

RESOLUCION DE UN SISTEMA HOMOGENEO DE ECUACIONES DIOFANTICAS

Se considera el sistema de E ecuaciones con V variables dado por

$$K X = 0$$

donde K es una matriz de coeficientes enteros, y se desean obtener las soluciones del mismo en las que las componentes de X sean así mismo enteras.

El sistema debe ser indeterminado, pues en caso contrario solo admitiría la solución trivial $X = 0$.

Si denominamos R al rango de la matriz K , el conjunto de soluciones del sistema formará un espacio vectorial de $V-R$ dimensiones como máximo. El algoritmo que se describe permite la determinación de una base de este espacio vectorial de soluciones formadas por vectores cuyas componentes son todas números enteros, de modo que el *MCD* de las componentes de cada solución es 1. Cualquier solución entera del sistema se podrá expresar como una combinación lineal de las soluciones básicas multiplicadas por escalares enteros.

El algoritmo descrito está inspirado en el método de eliminación de Gauss, en el que se han introducido las modificaciones adecuadas con objeto de trabajar siempre con coeficientes enteros.

Se utilizan dos vectores auxiliares de direccionamiento L y C , de E y V componentes respectivamente, los cuales son inicialmente puestos a cero.

La primera etapa del proceso es la triangularización. En él se trata de encontrar una ecuación I que sirva de base para la determinación de la variable $X (J)$, de modo que en dicha ecuación intervengan exclusivamente las variables $X (L)$, $L = J, \dots V$. Para ello se procede ordenadamente, variable por variable buscando entre las ecuaciones que no hayan servido de base para una variable anterior, la primera en la que el coeficiente $K (I, J)$ sea distinto de cero. Una vez encontrada se utiliza esta ecuación para anular todos los coeficientes $K (L, J)$ en aquellas ecuaciones L que no hayan servido de base para una variable anterior. En este caso, se marcarán los vectores de direccionamiento con $L (I) = J$ y $C (J) = I$, indicando así que la ecuación I es la base para determinar la componente J , y que la componente J se determina con la ecuación I .

En caso de no encontrar ningún coeficiente $K (I, J)$ distinto de cero en el proceso anterior, la variable J se considerará como variable independiente.

El número máximo de variables dependientes será:

$$NVD = \min (E, V - 1)$$

Cada variable dependiente tendrá una ecuación base asociada. Las restantes variables serán consideradas independientes. El vector C permitirá diferenciar las mismas, ya que $C (J)$ será cero cuando $X (J)$ sea independiente, y distinto de cero en caso contrario.

El proceso de triangularización se sigue hasta que se hayan determinado las ecuaciones base de hasta NVD variables. En caso de que queden ecuaciones que no hayan servido como base de alguna variable, si todos sus coeficientes no fuesen nulos el sistema no sería indeterminado, y en consecuencia admitiría exclusivamente la solicitud trivial.

A continuación se pasará al proceso de sustitución hacia atrás. Se harán también tantos procesos de este tipo como variables independientes contenga el sistema, en cada uno de los cuales se asignará inicialmente el valor 1 a una de las variables independientes y cero a las restantes.

Se tomará entonces las ecuaciones base de las componentes $X(J)$ $J = V, \dots, 1$ y se determina una solución parcial haciendo:

$$X(J) = \sum_{L=J+1}^V K(I, L) X(L)$$
$$X(L) = X(L) \cdot K(I, J) \quad L = J + 1, \dots, V$$

y dividiendo posteriormente esta solución parcial por el *MCD* de las componentes no nulas del vector X , desde la J hasta la V .

Al final de cada uno de estos procesos de sustitución hacia atrás se obtendrá una solución del sistema que formará parte de la base de posibles soluciones.

BUSQUEDA HEURISTICA DE UNA SOLUCION OPTIMA

En un cierto número de problemas la solución buscada debe estar formada por enteros mayores que 0 y lo más pequeños posible. Cuando el espacio de solución es unidimensional, el algoritmo descrito proporciona directamente la solución. Sin embargo, cuando el espacio de soluciones es de más de una dimensión, el algoritmo proporciona exclusivamente una base de dicho espacio de soluciones. A continuación se propone un método heurístico, que se muestra eficaz en gran número de ocasiones, para determinar una solución de las características citadas a partir de las soluciones básicas proporcionadas por el algoritmo.

Al describir el algoritmo se clasificaron las variables en dependientes e independientes. En cada una de las soluciones básicas una y solo una de las variables independientes es distinta de 0. Teniendo en cuenta esta peculiaridad de las soluciones básicas generadas, el proceso propuesto es el siguiente:

- 1) En cada una de las soluciones básicas, detectar el valor de la variable independiente no nula de la misma. Si dicho valor es negativo, cambiar el signo a dicha solución. Determinar el *MCD* de los valores no nulos de las variables independientes. Sea L dicho *MCD*.
- 2) Generar todas las combinaciones lineales de las soluciones básicas así modificadas, multiplicando cada una de ellas por un coeficiente comprendido entre 1 y L . En cada combinación analizada la solución se generará componente a componente, comprobando al obtener cada uno de ellos que es mayor que 0 y que es divisible por L . Cuando alguna de las componentes no cumpla este requisito se desechará la combinación y se pasará a la siguiente.
- 3) La primera solución obtenida que pase por el filtro citado se reducirá, dividiéndola por el *MCD* de sus componentes el cual será L o un múltiplo de L . Esta solución se pondrá como la óptima.

EJEMPLO: AJUSTE DE REACCIONES QUIMICAS

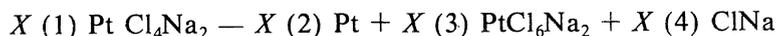
Un problema cuya resolución conduce al planteamiento de un sistema homogéneo de ecuaciones diofánticas es el de ajuste de reacciones químicas. En este caso se tiene un conjunto de compuestos que reaccionan para dar otro conjunto de compuestos, siendo conocida la composición atómica de todos ellos. El ajuste de la reacción se hace estableciendo que para cada tipo de átomo el balance en ambos grupos sea igual. En el caso particular de reacciones de oxidación reducción es frecuente que alguno de los compuestos sean iones, debiendo establecerse en este caso además el balance de cargas eléctricas.

Se plantea así un sistema de ecuaciones diofánticas, con tantas ecuaciones como tipos de átomos

intervienen, más uno en el caso de establecer el balance de cargas eléctricas, y con tantas incógnitas como tipos de compuestos intervienen.

Veamos algunos casos particulares:

Ejemplo 1



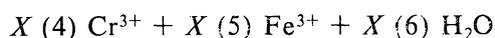
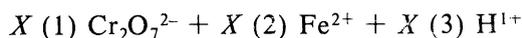
		X (1)	X (2)	X (3)	X (4)
BALANCE DE	Pt	1	-1	-1	0
	Cl	4	0	-6	-1
	Na	2	0	-2	-1

3 ecuaciones con 4 incógnitas

Número soluciones básicas: 1

Solución básica: $X(1) = 2$; $X(2) = 1$; $X(3) = 1$; $X(4) = 2$.

Ejemplo 2



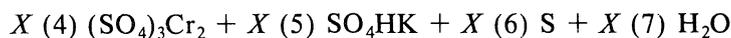
		X (1)	X (2)	X (3)	X (4)	X (5)	X (6)
BALANCE DE	Cr	2	0	0	-1	0	0
	O	7	0	0	0	0	-1
	Fe	0	1	0	0	-1	0
	H	0	0	1	0	0	-2
	Cargas	-2	+2	+1	-3	-3	0

5 ecuaciones 6 incógnitas

Número de soluciones básicas: 1

Solución básica: $X(1) = 1$; $X(2) = 6$; $X(3) = 14$; $X(4) = 2$; $X(5) = 6$; $X(6) = 7$.

Ejemplo 3



		X (1)	X (2)	X (3)	X (4)	X (5)	X (6)	X (7)
BALANCE DE	Cr	2	0	0	2	0	0	0
	O	7	4	0	12	4	0	1
	K	2	0	0	0	1	0	0
	S	0	0	1	3	1	1	0
	H	0	2	2	0	1	0	2

5 ecuaciones 7 incógnitas.

Número soluciones básicas: 2

		X (1)	X (2)	X (3)	X (4)	X (5)	X (6)	X (7)
Solución Básica	1	- 4	- 13	9	- 4	- 8	16	0
Solución Básica	2	4	17	3	4	8	0	16
Solución propuesta		1	5	3	1	2	3	7

Los coeficientes de ponderación probados en la búsqueda heurística para las soluciones básicas 1 y 2, han sido sucesivamente las parejas:

(1, 1), (1, 2), (2, 2), (1, 3), (2, 3), (3, 3), (1, 4), (2, 4), (3, 4), (4, 4), (1, 5), (2, 5), (3, 5), (4, 5), (5, 5), (1, 6), (2, 6), (3, 6), (4, 6), (5, 6), (6, 6), (1, 7), (2, 7), (3, 7),

La pareja (3, 7) ha proporcionado la solución propuesta.

PROGRAMACION DEL ALGORITMO

El algoritmo descrito ha sido programado en BASIC para una calculadora HP-75 C, adjuntándose un listado del programa.

El algoritmo se desarrolla en el programa DIOFAN, que está preparado para ser utilizado como subrutina. Recibe los datos en un archivo llamado DATDIO que contiene, en forma secuencial, el número de ecuaciones, el número de incógnitas, y los coeficientes del sistema ordenados por filas. Devuelve la solución o soluciones básicas en el archivo SOLDIO que contiene, de forma secuencial, el número de soluciones básicas y los componentes de cada una de las soluciones.

Como ejemplo de aplicación se incluye así mismo el programa AJUSTE, el cual pide como datos el número de moléculas de entrada y de salida, y a continuación la fórmula de cada una de las moléculas, que debe darse en la forma tradicional pero sin paréntesis, escribiendo los símbolos de los elementos, con mayúsculas y minúsculas, seguidos del número de átomos si son más de uno. En el caso de iones, las cargas eléctricas del mismo deberán introducirse precedidas de un símbolo de elemento ficticio, Xx, por ejemplo, seguido del número de cargas, sin signo si son positivas, y precedido del signo — si son negativas.

El programa genera automáticamente la matriz de coeficientes, y presenta la relación de átomos detectados, a efectos de comprobación. A continuación escribe los datos en el archivo DATDIO, pasa el control al programa DIOFAN, y lee los resultados del archivo SOLDIO. Si el número de soluciones básicas es mayor que 1, realiza la búsqueda heurística de la solución óptima. En caso de que la solución sea única, o que encuentre una solución óptima en la búsqueda, presenta los coeficientes de la solución, y a continuación escribe la ecuación química ya ajustada.

En caso que el sistema no tenga solución lo indicará mediante un mensaje. También indicará con un mensaje el que no haya tenido éxito en la búsqueda heurística de la solución.

REFERENCIAS

- (1) BAREISS, E. H.: "Computational solutions of matrix problems over an integral domain". J. Inst. Maths. Applics. 10,68-104. 1972.
- (2) HILKINSON, J. H.: "Some recent advanced in numerical linear algebra". The state of the art in numerical analysis. Edited by D. Jacob. Academic Press 1977, London.

```

10 REM RESOLUCION DE ECUACIONES DIOFANTICAS HOMOGENEAS DE E ECUACIONES CON V VAR
TABLES
20 OPTION BASE 1
30 INTEGER K(10,10),L(10),C(10),X(10)
40 ASSIGN # 1 TO "DATDIO"
50 READ # 1 ; E,V
60 FOR I=1 TO E
70 FOR J=1 TO V
80 READ # 1 ; K(I,J)
90 NEXT J
100 NEXT I
110 FOR I=1 TO E @ L(I)=0 @ NEXT I
120 FOR J=1 TO V @ C(J)=0 @ NEXT J
130 FOR J=1 TO V
140 FOR I=1 TO E
150 IF L(I)=0 AND K(I,J)#0 THEN L(I)=J @ C(J)=1 @ GOTO 180
160 NEXT I
170 GOTO 290
180 FOR I1=1 TO E
190 IF L(I1)#0 OR K(I1,J)=0 THEN 280
200 P1=K(I,J) @ P2=K(I1,J) @ M=0
210 FOR J1=J TO V
220 K(I1,J1)=K(I1,J1)*P1-K(I,J1)*P2
230 M=FNM(M,K(I1,J1))
240 NEXT J1
250 FOR J1=J+1 TO V
260 IF M THEN K(I1,J1)=K(I1,J1)/M
270 NEXT J1
280 NEXT I1
290 NEXT J
300 S=V
310 FOR J=1 TO V
320 IF C(J) THEN S=S-1
330 NEXT J
340 ASSIGN # 1 TO #
350 ASSIGN # 1 TO "SOLDIO"
360 PRINT # 1 ; S
370 FOR S1=1 TO S
380 S2=0
390 FOR J=1 TO V
400 IF C(J)=0 THEN S2=S2+1
410 IF S2=S1 THEN X(J)=1 ELSE X(J)=0
420 NEXT J
430 FOR J=V TO 1 STEP -1
440 I=C(J)
450 IF I=0 THEN 590
460 S2=0
470 M=0
480 FOR J1=J+1 TO V
490 S2=S2+K(I,J1)*X(J1)
500 X(J1)=X(J1)*K(I,J)
510 M=FNM(M,X(J1))
520 NEXT J1
530 X(J)=-S2
540 M=FNM(M,X(J))
550 M=M*SGN(X(J))
560 FOR J1=J TO V
570 IF M THEN X(J1)=X(J1)/M
580 NEXT J1
590 NEXT J

```

```

600 FOR J=1 TO V
610 PRINT # 1 ; X(J)
620 NEXT J
630 NEXT S1
640 END
650 DEF FNM(A,B)
660 A1=ABS(A) @ B1=ABS(B)
670 IF A1<B1 THEN B1=ABS(A) @ A1=ABS(B)
680 IF B1=0 THEN FNM=A1 @ GOTO 720
690 C1=MOD(A1,B1)
700 IF C1 THEN A1=B1 @ B1=C1 @ GOTO 690
710 FNM=B1
720 END DEF

```

```

10 ! =
20 REM AJUSTE AUTOMATICO DE UNA ECUACION QUIMICA
30 OPTION BASE 1
40 DIM A0$(2),A$(20),M0$(20),M$(200),R$(50)
50 INTEGER K(10,10),X(10),P(10)
60 E=0 @ A$=""
70 INPUT "MOLECULAS ENTRADA , SALIDA";M1,M2
80 M=M1+M2
90 FOR I=1 TO 10
100 FOR J=1 TO 10
110 K(I,J)=0
120 NEXT J
130 NEXT I
140 FOR J=1 TO M
150 IF J>M1 THEN DISP "MOLECULA SALIDA";J-M1;ELSE DISP "MOLECULA ENTRADA";J
160 INPUT M0$
170 M$(20*J-19,20*J)=M0$
180 L=LEN(M0$)
190 FOR N=1 TO L
200 A0%=M0$(N,N)
210 IF N=L THEN A0%=A0$% " " @ GOTO 230
220 IF M0$(N+1,N+1)<"a" THEN A0%=A0$% " " ELSE N=N+1 @ A0%=A0$%M0$(N,N)
230 I=(POS(A$,A0$)+1) DIV 2
240 IF I=0 THEN E=E+1 @ I=E @ A%=A$%A0$
250 IF N=L THEN K(I,J)=1 @ GOTO 320
260 IF M0$(N+1,N+1)="-" THEN N=N+1 @ IO=-1 ELSE IO=1
270 IF M0$(N+1,N+1)<"A" THEN N=N+1 @ K(I,J)=VAL(M0$(N,N)) ELSE K(I,J)=1
280 K(I,J)=K(I,J)*IO
290 IF N=L THEN 320
300 IF M0$(N+1,N+1)<"A" THEN N=N+1 @ K(I,J)=VAL(M0$(N-1,N))
310 K(I,J)=K(I,J)*IO
320 NEXT N
330 NEXT J
340 ASSIGN # 1 TO "DATDIO"
350 PRINT # 1 ; E,M
360 FOR I=1 TO E
370 FOR J=1 TO M
380 IF J>M1 THEN PRINT # 1 ; -K(I,J) ELSE PRINT # 1 ; K(I,J)
390 NEXT J
400 NEXT I
410 DISP A$
420 CALL "DIOFAN"
430 ASSIGN # 1 TO "SOLDIO"
440 READ # 1 ; S
450 IF S=0 THEN DISP "NO EXISTE SOLUCION " @ GOTO 640

```

```

460 IF S>1 THEN 650
470 FOR J=1 TO M
480 READ # 1 ; X(J)
490 NEXT J
500 R$=""
510 FOR J=1 TO M @ DISP X(J); @ NEXT J
520 DISP
530 FOR J=1 TO M
540 M0$=K$[20*J-19,20*J]
550 L=POS(M0$, " ")
560 M0$=M0$[1,L-1]
570 IF J#1 AND J#M+1 THEN R$=R$&" + "
580 IF J=M+1 THEN R$=R$&" (-) "
590 IF X(J)#1 THEN R$=R$&STR$(X(J))
600 R$=R$&M0$
610 NEXT J
620 DISP R$ @ STOP
630 STOP
640 END
650 FOR I=1 TO S
660 FOR J=1 TO M
670 READ # 1 ; K(I,J)
680 NEXT J
690 NEXT I
700 DISP "EXISTEN INFINITAS SOLUCIONES"
710 L=0
720 FOR J=1 TO M
730 L1=0
740 FOR I=1 TO S
750 IF K(I,J)#0 THEN L1=L1+1 @ L2=I
760 NEXT I
770 IF L1=1 THEN L3=SGN(K(L2,J)) @ L=FNH(L,L3*K(L2,J)) ELSE GOTO 790
780 FOR J1=1 TO M @ K(L2,J1)=K(L2,J1)*L3 @ NEXT J1
790 NEXT J
800 FOR P0=1 TO L
810 FOR I=1 TO S @ P(I)=0 @ NEXT I
820 I=1 @ P(S)=P0-1
830 P(I)=P(I)+1 @ IF P(I)<=P0 THEN 850
840 P(I)=0 @ I=I-1 @ IF I THEN 830 ELSE 950
850 IF I<S THEN I=I+1 @ GOTO 830
860 FOR J=1 TO M
870 X(J)=0
880 FOR I1=1 TO S @ X(J)=X(J)+P(I1)*K(I1,J) @ NEXT I1
890 IF X(J)<=0 OR MOD(X(J),L) THEN 830
900 NEXT J
910 L1=0
920 FOR J=1 TO M @ L1=FNH(L1,X(J)) @ NEXT J
930 FOR J=1 TO M @ X(J)=X(J)/L1 @ NEXT J
940 GOTO 500
950 NEXT P0
960 DISP "NO ENCUENTRO SOLUCION EN UN TIEMPO RAZONABLE" @ STOP
970 END
980 DEF FNH(A,B)
990 A1=ABS(A) @ B1=ABS(B)
1000 IF A1<B1 THEN B1=ABS(A) @ A1=-ABS(B)
1010 IF B1=0 THEN FNH=A1 @ GOTO 1050
1020 C1=MOD(A1,B1)
1030 IF C1 THEN A1=B1 @ B1=C1 @ GOTO 1020
1040 FNH=B1
1050 END DEF

```