

NOTAS DE MATEMATICA

Nº 125

ESTRUCTURA TIB OPTIMA DE LAS BASES EN UNA CLASE
DE PROBLEMAS DE P.L.

POR

LINA GARCIA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMATICA
MERIDA - VENEZUELA

1993

En ocasión de la visita de la
Dra. Lina García de la Academia
de Ciencias de Cuba.

Mérida, Enero de 1993.

ESTRUCTURA TIB OPTIMA DE LAS BASES EN UNA CLASE DE PROBLEMAS DE P.L.

LINA GARCIA

INST. CIBERNETICA, MATEMATICA Y FISICA (ICIMAF)
ACADEMIA DE CIENCIAS DE CUBA

RESUMEN

Se demuestra que en los problemas de programación lineal en los que la matriz de coeficientes tiene a lo más dos elementos no nulos por columna (fila), la forma triangular inferior por bloques (TIB) óptima es tal que cada bloque de dimensión mayor que dos tiene exactamente una espiga. Además, en ambos casos se da un algoritmo para hallar una estructura TIB más sencillo y eficiente que el clásico método general en dos etapas : búsqueda de una transversal y cálculo de las componentes fuertemente conexas.

1. Introducción

En la implementación del Simplex para problemas con muy baja densidad de coeficientes no nulos en las restricciones, o, más concretamente, en la factorización LU de matrices cuadradas, ralas, el preprocesamiento con vistas a reducir el llenado durante la factorización juega un papel fundamental.

Es conocido - ver Duff (1972, 1977), Howell (1976) -, que mediante permutación de filas y columnas es posible llevar la base a una forma triangular inferior por bloques (TIB) óptima en el sentido que en cualquier otra forma TIB que pueda obtenerse por permutaciones de filas y columnas, los bloques diagonales los forman las filas y columnas de uno o más bloques de ella, por lo que en la forma TIB óptima simultáneamente se maximiza el número de bloques y se minimizan sus dimensiones. Esta estructura es esencialmente única excepto por permutaciones de los bloques diagonales o permutaciones de filas y columnas dentro de cada bloque. De hecho, se obtiene una forma TIB óptima mediante una permutación de columnas que deje la diagonal sin ceros y luego una

permutación simétrica que lleve la matriz resultante a una forma TIB irreducible.

Si consideramos la matriz binaria que resulta de poner 1 en cada posición correspondiente a un coeficiente no nulo, los dos procesos anteriores serían simplemente, en términos de grafos, hallar una transversal - es decir, un acoplamiento máximo - en el bigrafo asociado a la matriz binaria original, y luego, hallar las componentes fuertemente conexas del digrafo asociado a la matriz binaria después de permutadas las columnas -ver por ejemplo, Gustavson(1976), Bartels(1976), y Duff y otros(1989)-. Una forma muy eficiente de implementar todo esto es usando el algoritmo de Hall(1956) para el cálculo de la transversal, y el algoritmo de Tarjan - ver Duff y Reid(1978)- para las componentes fuertemente conexas, implementándolos con una búsqueda primero en profundidad - Tarjan (1972) -.

La matriz resultante está en dependencia de la transversal hallada y del ordenamiento de las filas y columnas dentro de cada bloque, por lo que el número de espigas total resultante - es decir, columnas con coeficientes no nulos encima de la diagonal - puede variar significativamente de una forma TIB óptima a otra. Puesto que el llenado durante la factorización ocurre precisamente sobre las espigas, un problema interesante es la cuestión de reducir el número de espigas. En términos de grafos, considerando permutaciones simétricas, éste sería el problema de encontrar el conjunto mínimo de vértices de retroceso - *minimum feedback vertex problem* - que es un problema NP-complejo. También se ha abordado este problema con técnicas heurísticas, como por ejemplo los algoritmos propuestos en Hellerman y Rarick (1971,1972) y Erisman y otros (1985).

En este trabajo nos acercamos a esta problemática pero desde un ángulo diferente. El problema que nos preocupa es ¿cuándo podemos saber *a priori* que es posible encontrar una forma TIB óptima de una matriz en la que cada bloque no trivial - es decir, de dimensión mayor que dos - tenga exactamente una espiga? En este sentido demostramos mediante un proceso constructivo, que ésto es siempre posible cuando la base o su transpuesta tiene a lo más dos elementos no nulos por columna.

Cuando en lugar de dos son hasta tres los elementos no nulos por columna, es posible construir contraejemplos que demuestran que esa afirmación no es válida en general, aunque existen problemas con columnas hasta de cuatro elementos para las que también esto es cierto (García (1992)).

2. Forma TIB óptima

2.1 Caso por columnas

Comenzaremos considerando una matriz de restricciones de dimensión $M \times N$ y con a lo más dos elementos no nulos por columna, y una base B de dimensión M .

El Algoritmo 2.1 que describimos a continuación produce una permutación por filas y una permutación por columnas que llevan la base a una forma TIB.

Para describir el algoritmo usaremos las siguientes notaciones:

- M : dimensión de la base.

- $PFIL[i], PCOL[i] \quad I=1..M$:

En la salida son las permutaciones de filas y columnas que llevan la base a la estructura TIB deseada.

- $FSUBMAT, CSUBMAT$: Conjunto de índices de filas (respectivamente columnas) no asignadas. Inicialmente ninguna fila (respectivamente columna) está asignada.

- $RC[i], \quad i=1..M$:

Durante el algoritmo indica, para cada fila i no asignada, el número de elementos no nulos sobre las columnas no asignadas.

- $RCMIN$: Mínimo valor de $RC[i]$, para las filas no asignadas.

- $NASIG$: Durante el algoritmo indica el número de filas asignadas hasta el momento.

ALGORITMO 2.1 (Caso por columnas)

PASO 0 : Inicialización

$$FSUBMAT := \{1, \dots, M\}, \quad CSUBMAT := \{1, \dots, M\}.$$

Calcular $RC[j] \quad j=1..M$, y $RCMIN := \min_{j \in FSUBMAT} RC[j]$.

Hacer $NASIG = 0$.

Si $RCMIN=1$ ir al PASO 1, si no, ir al PASO 2.

PASO 1 : Columnas triangulares

Mientras existan filas no asignadas, y con un solo elemento,

repetir :

- a. Tomar una fila i_0 con $RC[i_0] = 1$.
- b. Determinar la columna j_0 de ese único coeficiente no nulo.
- c. Asignar j_0 a i_0 . Es decir:
 - Incrementar $NASIG$.
 - Hacer $PFIL[NASIG] := i_0$, $PCOL[NASIG] := j_0$.
 - Eliminar i_0 de $FSUBMAT$ y j_0 de $CSUBMAT$.
- d. Actualizar $RC[i]$ para las filas no asignadas.

Si $NASIG=M$, FIN :MATRIZ TRIANGULAR INFERIOR.

Si $NASIG < M$ ir al PASO 2.

PASO 2 : Bloques

Mientras $NASIG < M$, repetir crear bloque. Es decir :-

- a. Seleccionar una columna no asignada y poner el índice en la variable $ESPIGA$. Eliminar esa columna, y actualizar $RC[i]$ para las filas no asignadas.
- b. Mientras $RCMIN = 1$ y $NASIG < M-1$, proceder como en (a)-(d) del PASO 1.
- c. - Si $NASIG=M-1$, asignar la única fila no asignada a la columna $ESPIGA$, como en el PASO 1.c y FIN.
- Si $RCMIN=0$, asignar la fila nula a la columna $ESPIGA$ y volver a comenzar otro bloque.

Observaciones :

1. En el PASO 1 no puede suceder $RCMIN=0$ pues entonces $NASIG + 1$ filas tendrían sus elementos no nulos sobre $NASIG$ columnas y por lo tanto la matriz ser a estructuralmente singular, lo que contradice que es una base. Por lo tanto, cuando se llega al PASO 2, se tiene $RCMIN > 1$; más a n, como cada columna tiene a lo más dos elementos no nulos, necesariamente $RCMIN=2$ y cada fila tiene exactamente dos elementos no nulos sobre las columnas no asignadas.

2. Nótese que durante la ejecución del PASO 2 y una vez vez seleccionada la $ESPIGA$, no puede tenerse en ningún momento $RCMIN=2$, pues $RCMIN$ es el mínimo valor de $RC[.]$ para las $r+1$ filas no asignadas, en tanto que $RC[i]$ se calcula como el número de coeficientes no nulos de la fila i en las r columnas no asignadas. La submatriz formada por las filas y columnas no asignadas (excluyendo la espiga), tiene exactamente $2r$ elementos no nulos que no permiten cubrir las $r+1$ filas con al menos 2

elementos cada una, como exigir a $RCMIN=2$.

3. Al definir un bloque en el PASO 2 las filas del bloque tienen sus dos coeficientes no nulos sobre las columnas previamente asignadas del bloque y la columna *ESPIGA*, con la que se cerrará el bloque, por lo que una vez definido el bloque nos encontramos que éste tiene dos coeficientes no nulos por fila y por lo tanto cada columna del bloque tiene sus dos únicos coeficientes no nulos sobre las propias filas del bloque, lo que hace que las filas que restan por asignar continúan con $RC[i]=2$ como al comenzar el PASO 2.

4. Nótese que una vez que se llega al PASO 2 se definen bloques (de dimensión dos por lo menos) hasta terminar. Es decir, las columnas asignadas en el PASO 1 serían las únicas columnas triangulares. Además, cada bloque definido en el Paso 2 tiene exactamente una espiga.

En conclusión, permutando las filas y columnas de B según *PERMUF* y *PERMUC*, se obtiene B_1 con la estructura siguiente :

$$\begin{pmatrix} T & O & O & \dots & O \\ H_1 & B_1 & O & \dots & O \\ H_2 & O & B_2 & & \cdot \\ \vdots & \vdots & & \cdot & \cdot \\ H_R & O & \cdot & \cdot & B_R \end{pmatrix} \quad (1)$$

donde

- T es triangular inferior,
- $H = (H_1, \dots, H_R)^T$ tiene a lo más un elemento no nulo en cada columna, y
- Cada B_i tiene la forma

$$\begin{pmatrix} x & o & \cdot & \cdot & \cdot & o & x \\ z & x & \cdot & \cdot & \cdot & \cdot & z \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ z & z & \cdot & \cdot & \cdot & o & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot \\ z & z & \cdot & \cdot & \cdot & x & z \end{pmatrix} \quad (2)$$

donde x representa los elementos necesariamente no nulos. En cada columna, el otro elemento no nulo está por debajo de la diagonal, y siempre de tal manera que hayan dos elementos no nulos por filas. En otras palabras, cada B_i es una matriz triangular

inferior excepto por la última columna : la única espiga del bloque .

Finalmente, obsérvese que el Paso 2 se puede implementar así: una vez escogida la columna espiga, con coeficientes no nulos en las filas r_0 y r_s , digamos, se toma directamente la fila r_0 que quedó con un solo elemento , y se le asigna la columna correspondiente. De nuevo se va directamente a la fila r_1 donde esta columna tiene su otro elemento y que acaba de quedar unitaria , y así sucesivamente hasta que la fila r_1 que introduce la última columna asignada, coincide con r_s y por lo tanto se toma directamente esta fila para cerrar el bloque. De esta manera los bloques quedarían con la siguiente estructura :

$$\begin{pmatrix} x & & & \circ & x \\ x & & & \circ & \circ \\ \vdots & & & \vdots & \vdots \\ \circ & & & x & \circ \\ \circ & \dots & & x & x \end{pmatrix} \quad (3)$$

2.2 Caso por filas

Consideremos ahora que la matriz de restricciones tiene a lo más dos elementos no nulos por fila y consideremos nuevamente B una base. El Algoritmo 2.2 que describimos a continuación lleva la base a una forma TIB.

Usaremos las mismas notaciones que para el Algoritmo 2.1; además necesitaremos :

- $CC[j]$, $j=1..M$:

Durante el algoritmo indica, para cada columna j no asignada, el número de elementos no nulos sobre las filas no asignadas.

- $CCMIN$: Mínimo valor de $CC[j]$, para las columnas no asignadas.

- IND : Índice de $PERMUF$, $PERMUC$, donde debe ponerse los próximos índices de fila y columna.

ALGORITMO 2.2 (Caso por filas)

PASO 0 : Inicialización

$$FSUBMAT := \{1, \dots, M\}, \quad CSUBMAT := \{1, \dots, M\}.$$

Calcular $CC[j]$ $j=1..M$, y $CCMIN := \min_{j \in CSUBMAT} CC[j]$.

Hacer $NASIG = 0$, $IND = M+1$.

Si $CCMIN=1$ ir al PASO 1, si no, ir al PASO 2.

PASO 1 : Columnas triangulares

Mientras existan columnas no asignadas, y con un solo elemento, repetir :

- a. Tomar una columna j_0 con $CC[j_0] = 1$.
- b. Determinar la fila i_0 de ese único coeficiente no nulo.
- c. Asignar j_0 a i_0 . Es decir:
 - Incrementar $NASIG$, decrementar IND .
 - Hacer $PFIL[IND]:=i_0$, $PCOL[IND]:=j_0$.
 - Eliminar i_0 de $FSUBMAT$ y j_0 de $CSUBMAT$.
- d. Actualizar los valores $CC[.]$ de las columnas no asignadas.

Si $NASIG=M$, FIN :MATRIZ TRIANGULAR INFERIOR.

Si $NASIG < M$, ir al PASO 2.

PASO 2 : Bloques

Calcular $RC[j]$ $j \in FSUBMAT$, y $RCMIN := \min_{j \in FSUBMAT} RC[j]$,

Hacer $IND = 0$.

Mientras $NASIG < M$, repetir crear bloque. Es decir :

- a. Seleccionar una columna no asignada y poner el índice en la variable $ESPIGA$. Eliminar esa columna, y actualizar $RC[i]$ para las filas no asignadas.
- b. Mientras $RCMIN = 1$ y $NASIG < M-1$, proceder como en (a)-(d) del PASO 1 del Algoritmo 2.1. La asignación de j_0 a i_0 consiste en :
 - Incrementar $NASIG$ e IND .
 - Hacer $PFIL[IND]:=i_0$, $PCOL[IND]:=j_0$.
 - Eliminar i_0 de $FSUBMAT$ y j_0 de $CSUBMAT$.
- c. - Si $NASIG=M-1$, asignar la única fila no asignada a la columna $ESPIGA$, como en el Paso 1.c del Algoritmo 2.1 y FIN.
 - Si $RCMIN=0$, asignar la fila nula a la columna $ESPIGA$ y volver a comenzar otro bloque.

La estructura que se obtiene mediante las permutaciones de filas y columnas calculadas mediante este algoritmo es la que aparece en (4), donde B_i , H_i y T son como en (1).

La observación (1) para el Algoritmo 2.1 es válida aquí poniendo $CCMIN$ en lugar de $RCMIN$. El resto de las observaciones se aplican exactamente en este caso, incluyendo la

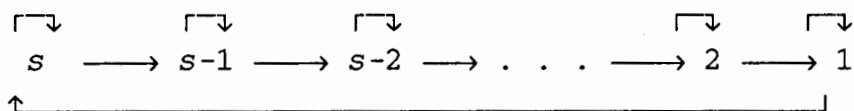
posibilidad de llevar los bloques diagonales a la forma (3).

$$\begin{pmatrix} B_1 & O & \dots & O & O \\ O & B_2 & & \cdot & \cdot \\ \vdots & & \cdot & \cdot & \cdot \\ O & \cdot & \cdot & B_R & O \\ H_1 & \cdot & \cdot & H_R & T \end{pmatrix} \quad (4)$$

PROPOSICION 2.1 : En todo problema de P.L. donde cada columna (respectivamente fila) tenga a lo más dos coeficientes no nulos, cualquier base puede llevarse , mediante permutación de filas y columnas, a una forma TIB óptima en la que cada bloque no unitario tiene exactamente una espiga. El ALGORITMO 2.1 (respectivamente el ALGORITMO 2.2) permite encontrar tales permutaciones.

Demostración: Tenemos que demostrar que las formas TIB descritas en (1) y (4) , halladas por los Algoritmos 2.1 y 2.2 respectivamente, son óptimas. Para ello (ver por ejemplo Duff[1972, 1977], Howell[1976]) basta encontrar para cada bloque no unitario una permutación de filas y columnas que ponga elementos no nulos sobre la diagonal, y tal que el grafo asociado al bloque permutado sea conexo.

Consideremos entonces solamente un bloque, de dimensión s en general. Para simplificar las notaciones renumeremos los índices de filas y columnas relativamente al bloque. No perdemos generalidad si suponemos que cada bloque tiene la forma (3) , pues en caso contrario se puede llevar a ella mediante una permutación simétrica de filas y columnas. Pero eso corresponde al digrafo conexo siguiente



con lo que se completa la demostración . ■

3. Ejemplos

Consideremos una base B con el patrón de elementos no nulos que aparece en la Tabla 3.1.

		1	2	3	4	5	6	7	8	9
1	(x			x			
2				x						
3						x	x			
4		x								
5			x						x	
6					x					x
7				x					x	
8					x			x		
9		x						x		x

Tabla 3.1

Como resultado de aplicar el Algoritmo 2.1 se obtiene la estructura siguiente :

		3	1	6	5	8	2	9	7	4	
2	(x									
4			x								
1				x	x						
3					x	x					
5							x	x			
7							x	x			
6									x		x
9									x	x	
8		x								x	x

Tabla 3.2

Si aplicamos el Algoritmo 2.2 a B^T se obtiene la estructura que aparece en la Tabla 3.3 .

4. Resultados computacionales

Aunque teóricamente hemos visto que las bases de los modelos analizados pueden llevarse a las estructuras (1) y (4) mediante permutación de filas y de columnas, quisimos analizar qué incidencia tenían realmente los bloques dentro de esa estructura teórica.

		7	5	8	9	6	3	1	4	2
8	(x	x							
2		x	x							
4				x		x				
7				x	x					
9					x	x				
5							x			
6							x	x		
1									x	
3					x			x		x
)									

Tabla 3.3

Para la aplicación del Algoritmo 2.1 no interesan los valores concretos de los coeficientes de la base sino sólo la ubicación de los no nulos. Por ello se generaron aleatoriamente matrices binarias de dimensión $m \times m$, estructuralmente no singulares, con exactamente dos elementos no nulos en cada fila.

La Tabla 4.1 presenta los resultados de aplicar el Algoritmo 2.2 a bases generadas según esta subrutina. Cada línea corresponde al promedio de los resultados obtenidos en 10 bases .

% col.triang. : % columnas triangulares

No. bloques : Número promedio de bloques.

Dimens. prom : Promedio de las dimensiones de todos los bloques no unitarios de las diez bases consideradas.

<i>N</i>	<i>% col. triang.</i>	<i>No. bloques</i>	<i>Dimens. prom.</i>
50	85.60	1.7	4.23
100	98.88	2.0	5.6
200	92.55	2.2	6.77
300	94.93	2.0	7.6
400	93.75	2.5	10.0
500	93.66	3.2	9.90

Tabla 4.1

*Datos de las estructuras encontradas
mediante el Algoritmo 2.2*

Como se aprecia, se mantiene un alto porcentaje de columnas triangulares, y las dimensiones de los bloques no crecen

significativamente con m .

Por otra parte, en la Tabla 4.2 se comparan los tiempos de ejecución (en centésimas de segundos) del Algoritmo 2.1 y una implementación muy eficiente de los algoritmos de Hall y de Tarjan basada en Gustavson(1976), para encontrar la forma TIB óptima. Ambos métodos fueron programados en Turbo Pascal v. 6.00 , al igual que la subrutina de generación, y se corrieron en una PC IBM AT de 10 MHz. Cada fila corresponde al promedio obtenido en la aplicación de ambos métodos a diez problemas generados aleatoriamente.

Como se observa, ambos métodos son muy rápidos, pero el Algoritmo 2.1 resulta, como era lógico esperar, notablemente más rápido. Cabe señalar además, que en la forma TIB óptima obtenida por el método clásico, los bloques aparecen alternando con las columnas triangulares, lo que al resolver los sistemas de ecuaciones resulta una desventaja con respecto a los algoritmos que proponemos.

N	K	I	NH	Tiempo de ejec.(cent simas seg.)	
				Alg. cl sico	Alg. 2.1
50	30	20	2	2.9	1.1
100	50	50	10	7.7	4.0
200	100	100	20	13.3	7.3
300	150	150	30	18.5	13.9
400	200	200	40	25.5	18.1
500	250	250	50	31.2	24.3

Tabla 4.2

Comparación de los tiempos de ejecución

REFERENCIAS

Bartels, Richard : *Large, Sparse Linear Programming.*(Notes for the Applied Matrix Methods Workshop). Johns Hopkins U., Baltimore, Maryland (1976)1-44.

Duff, I.S. : *On permutations to block triangular form* . J. Inst. Maths. Applics **19**(1977),339-342.

Duff, I.S. :D.Phil. thesis . University of Oxford, 1972.

- Duff, I.S.; A.M. Erisman y J.K. Reid : **Direct Methods for Sparse Matrices** . Monographs on Numerical Analysis, Oxford University Press .1989.
- Duff, I.S.; J.K. Reid : *An implementation of Tarjan's algorithm for the block triangularization of a matrix.* ACM Trans. Math. Software **4** (1978), 137-147 y 189-192.
- Erisman, A.M.; Grimes, R.G., Lewis, J.G., y Poole, W.G. Jr.: *A structurally stable modification of Hellerman-Rarick's P^4 algorithm for reordering unsymmetric sparse matrices.* SIAM J. Numer. Anal. **22** (1985) 369-385.
- Gustavson, Fred : *Finding the block lower triangular form of a sparse matrix* : En : Bunch, J.R.; Rose, D.J. (eds): **Sparse Matrix Computations**, A.P. (1976) 275-289.
- Hall, M. : *An algorithm for distinct representatives.* Amer. Math. Monthly **63** (1956), 716-717.
- Hellerman, E.; D. Rarick : *Reinversion with the preassigned pivot procedure.* Mathematical Programming **1** (1971) 195-216.
- Hellerman, E.; D.C. Rarick : *The partitioned preassigned pivot procedure (P^4).* En : Rose, D.J.; R.A. Willoughby (eds.): **Sparse Matrices and their Applications**. Plenum Press (1972).
- Howell, T.D. : *Partitioning using PAQ.* En : Bunch, J.R.; Rose, D.J. (eds): **Sparse Matrix Computations**, A.P. (1976) p.23-37
- García, L. : *Generalizaciones del problema de transporte : estructura de las bases y solución.* Presentado en : VI CLAIO, C.México, octubre 1992.
- Rose, D.J., R.E. Tarjan : *Algorithmic aspects of vertex elimination.* **Proc. Seventh Annual ACM Symp. on theory of computing**, 1975, 245-254.
- Rose, D.J., R.E. Tarjan : *Algorithmic aspects of vertex elimination on directed graphs.* SIAM J. Appl. Math. **34** (1978) 176-197.
- Tarjan, R.E. (1972) *Depth first search and linear graph algorithms* SIAM J. Comput **1** , 146-160.
- Tarjan, R.E. : *Graph Theory and Gaussian Elimination* . En: Bunch, J.R. y Rose, D.J. (Eds.) : **Sparse Matrix Computations**, Academic Press (1976) 3-22.