ALGORITMO PARA DETERMINAR SI UNA SUCESION DE NUMEROS ES BIPARTITA GRAFICA

Felicia Villarroel, Gladys Lárez, y Pedro J. Mago*

RESUMEN:

En este artículo se crea un algoritmo que permite, a partir de una sucesión de números determinar si es gráfica, construir una realización de la misma y verificar algunas propiedades que pueda tener una o todas las realizaciones de esa sucesión.

PALABRAS CLAVE: Sucesión, Grado, Bipartito.

ABSTRACT

In this paper we create an algorithm that allows us, starting from a given sequence of numbers to determine whether sequence is graphic, to construct a realization of this sequence, and to very some properties that one or all the realizations of this sequence may have.

KEY WORDS: Sequence, Degree, Bipartite.

INTRODUCCION

Los grafos considerados en este artículo son sin lazos ni lados múltiples. G(X,Y,E) denota un grafo bipartito balanceado de orden 2n, donde $X=\{x_1,x_2...,x_n\}$ y $Y=\{y_1,y_2,...,y_n\}$. En este artículo $a_i=d_G(x_i)$ y $b_i=d_G(y_i)$ para todo i=1,2,...,n, y $\Pi(G) = (\Pi_X, \Pi_Y)$ denota la sucesión con $\Pi_X=(a_1,...,a_n)$, $\Pi_Y=(b_1,...,b_n)$, $a_1\leq a_2 \leq...\leq a_n$ y $b_1 \leq b_2 \leq ... \leq b_n$.

Una **sucesión** $\Pi= (\Pi_X, \Pi_Y)$ es **bipartita gráfica** si existe un grafo bipartito simple G, para el cual $\Pi(G) = (\Pi_X, \Pi_Y)$. El grafo resultante G, el cual en general no es único, se llama una **realización** de Π.

Sea $\Pi=(\Pi_X, \Pi_Y)$ tal que $\max\{a_n,b_n\}\leq n$. La **sucesión reducida de Π en a_j**, es la sucesión $\Pi_{aj}= (a_1,...,a_{j-1}, 0,a_{j+1},...,a_n,b_1,...,b_{n-aj+1}-1,...,b_n-1)$ que se obtiene de Π reemplazando a_j en Π_X por cero y restando 1 a b_i para todo i : $n-a_j+1,...,n$ en Π_Y. Π_{aj} también se denota $\Pi^{(j)}$. La sucesión reducida de Π_{aj} en b_k, se denota Π_{ajbk}. La sucesión reducida Π_{bj} se obtiene de manera análoga que Π_{aj}.

Π tiene la **I^n-Propiedad** si los términos de las sucesiones reducidas $\Pi^{(1)}$, $\Pi^{(2)}$,..., $\Pi^{(n-1)}$ permanecen no negativos.

Sea P una propiedad definida en un grafo bipartito balanceado G. Una sucesión de grado bipartita gráfica, Π, es **fuerte respecto a la propiedad P** si toda realización de Π posee la propiedad P y es **débil respecto a la propiedad P** si existe al menos una realización de Π que posee la propiedad P. Las propiedades P, que se considerarán son: **conexidad, k-conexidad, hamilto-nicidad y bipanciclismo.** En el momento de analizar una propiedad P, se demostrará, basándose en teoremas conocidos, si la sucesión es débil respecto a P (débilmente P), es fuerte respecto a P (fuertemente P) o que no se puede concluir nada.

A continuación se mencionan algunos resultados que son condiciones suficientes para que una sucesión de grado sea bipartita gráfica y para que admita una determinada propiedad dada, los cuales serán utilizados en el algoritmo a desarrollar.

Teorema 1. Mago *et al.* (1996). Una sucesión de grados $\Pi = (\Pi_X, \Pi_Y)$ es bipartita gráfica sí y sólo sí Π_{aj} es bipartita gráfica.

Teorema 2. Mago *et al.* (1996). Sea $\Pi= (\Pi_X, \Pi_Y)$ una sucesión de grados tal que

i.- $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i$

ii.- $b_i \leq |X|$ y $a_i \leq |Y|$ para i: 1,..., n

Π es bipartita gráfica sí y sólo sí Π tiene la I^n-propiedad.

Teorema 3. Mago *et al.* (1996). Sea $\Pi = (\Pi_X, \Pi_Y)$ una sucesión de grados bipartita gráfica. Π es debilmente

---

*Departamento de Matemática, Escuela de Ciencias, Universidad de Oriente, Cumaná, Venezuela.
Recibido Enero 1997. Aprobado Diciembre 1997.

conexa sí y sólo sí

i.- $\min\{a_1, b_1\} \geq 1$

ii.- $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i \geq 2(2n-1)$

Teorema 4. Mago *et al.* (1996). Si $\Pi=(\Pi_X, \Pi_Y)$ es una sucesión de grados bipartita gráfica tal que $\min\{a_1,b_1\}\geq 2$ y

$m=\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i \geq 2(n-1)+\max\{a_n,b_n\}$, entonces Π es débilmente 2-conexa.

Teorema 5. Mago *et al.* (1996). Sea G = (X,Y,E) un grafo bipartito balanceado de orden 2n. Si G admite h vértices de grado n en X y h vértices de grado n en Y con $1 \leq h \leq n-1$ entonces G es h-conexo.

Teorema 6. Moon y Moser *et al.* (1963). Sea G=(X,Y,E). Un grafo bipartito balanceado con $2n \geq 4$ vértices. Si para todo par de vértices no adjacentes $x \in X$ y $y \in Y$, se cumple que $d(x)+d(y)>n$, entonces G es hamiltoniano.

Teorema 7. Amar *et al.* (1995). Sea G=(X,Y,E) un grafo bipartito balanceado de orden $2n \geq 4$. Si para todo par de vértices no adyacentes $x \in X$ y $y \in Y$, se cumple que $d(x)+d(y) \geq n+2$, entonces G es hamilton-conexo.

Teorema 8. Amar *et al.* (1995). Sea G = (X,Y,E) un grafo bipartito balanceado de orden $2n \geq 4$. Si para todo par de vértices no adyacentes $x \in X$ y $y \in Y$, se cumple que $d(x)+d(y) \geq n+1$, entonces G es bipancíclico.

El siguiente resultado es una consecuencia de los Teoremas 6, 7 y 8 .

Corolario 9. Mago *et al.* (1996). Sea $\Pi=(\Pi_X, \Pi_Y)$ una sucesión de grados bipartita gráfica, entonces

(a) Si $\min\{a_1,b_1\} > \frac{n}{2}$, entonces Π es fuertemente hamiltoniana.

(b) Si $\min\{a_1,b_1\} \geq \frac{n+2}{2}$, entonces Π es fuertemente hamilton-conexa.

(c) Si $\min\{a_1,b_1\} \geq \frac{n+1}{2}$, entonces Π es fuertemente bipancíclica.

Lema 10. Sea $\Pi=(\Pi_X, \Pi_Y)$ una sucesión de grado bipartita gráfica débilmente 2-conexa, entonces Π es débilmente hamiltoniana si y solo si Π^*: $(a_1-2 ,.., a_n -2, b_1-2,.., b_n-2)$ es gráfica.

Demostración. Por ser Π débilmente hamiltoniana, existe un grafo hamiltoniano G realización de Π. Sea C un ciclo hamiltoniano en G, entonces G-C es gráfica y es una realización de Π^*.

Claramente, si Π^* es gráfica y G es una realización de Π^*, se puede construir un ciclo hamiltoniano en G y obtener una realización hamiltoniana de Π.

El Lema 10 permite la demostración del siguiente teorema

Teorema 11. Mago *et al.* (1996). Sea $\Pi=(\Pi_X, \Pi_Y)$ una sucesión bipartita gráfica débilmente 2-conexa. Si $n-1 \leq a_n \leq n$ o $n-1 \leq b_n \leq n$, entonces Π es débilmente bipancíclico.

La sucesión Π= (2,2,2,2,3,2,2,2,2,3) muestra que el Teorema 11 es el mejor resultado posible ya que a_n=n-2, b_n=n-2 y Π es debilmente 2-conexa, pero no es débilmente bipancíclica.

ALGORITMO PRINCIPAL

Paso 1.- Lectura del número de elementos de la sucesión Π.

Paso 2.- Lectura de los elementos de la sucesión Π.

Paso 3.- Ordenamiento de la sucesión Π.

Paso 4.- Almacenamiento de la sucesión.

Paso 5.- Verificación si la sucesión es bipartita gráfica.

Paso 6.- Menú de opciones.

A continuación escribiremos el Menú de opciones y algunos de los procedimientos usados en el algorítmo principal.

Menú de opciones

Variable Booleana cierta
  Mientras Variable Booleana haga
    Variable Booleana falsa
    1.- Representación Gráfica del Grafo.
    2.- Propiedades y/o Realizaciones.
    3.- Salir {Fin Programa}
    Seleccionar Opción
      Si opción es 1 entonces
      Representación {Procedimiento}
    Sino
      Si opción es 3 entonces

Fin Programa
Parar
Sino
Mostrar SubMenú
1.- Conexidad
2.- K-Conexidad
3.- Hamiltonicidad
4.- Bipanciclismo
5.- Volver a Menú
6.- Salir {Fin Programa}
Seleccionar Opción1
Si Opción1 es 1 entonces
Conexidad {Procedimiento}
Si Opción1 es 2 entonces
K-Conexidad {Procedimiento}
Si Opción1 es 3 entonces
Hamiltonicidad {Procedimiento}
Si Opción1 es 4 entonces
Bipanciclismo {Procedimiento}
Si Opción1 es 5 entonces
Variable Booleanacierta
Si Opción1 es 6 entonces
Fin Programa
Parar
Finmientras

Procedimiento para verificar si la sucesión es bipartita gráfica.

Entrada: Elementos de la sucesiones Π_X y Π_Y ordenadas.

Salida: Mensaje si la sucesión es bipartita gráfica o no y lista de adyacencias en caso de ser bipartita gráfica.

Paso1.- Si (b(1) > p) o (a(1) > p) entonces

Paso 2.- Escribir mensaje
Paso 3.- Parar
Paso 4.- Sino
Paso 5.- i = 1
Paso 6.- Repetir
Paso 7.- Mayor = a(i)
Paso 8.- a(i) = 0
Paso 9.- j = 1
Paso 10.- Variable Booleana cierta
Paso 11.- Mientras j ≤ Mayor y Variable Booleana haga
Paso 12.- b(j) = b(j) - 1
Paso 13.- Si b(j) < 0 entonces
Paso 14.- Escribir mensaje
Paso 15.- Variable Booleanafalsa
Paso 16.- Finsi {Paso 13}
Paso 17.- Si Variable Booleana haga
Paso 18.- j = j+1
Paso 19.- Para z = 1 hasta Mayor haga
Paso 20.- Posa(i) es adyacente con Posb(z)
Paso 21.- Finpara {Paso 19}
Paso 22.- Ordenar Π_Y.
Paso 23.- i = i+1
Paso 24.- Finsi {Paso 17}
Paso 25.- Hasta (no Variable Booleana) o (i > p)
Paso 26.- Finsi {Paso 1}

Procedimiento para la propiedad de conexidad.

Entrada: Sucesión de números ordenada.

Salida: Mensaje si se cumple o no la propiedad y/o realización en caso de cumplirse.

Paso 1.- Si a(1) > b(1) entonces

Paso 2.- mínimo = b(1)

Paso 3.- Sino

Paso 4.- mínimo = a(1)

Paso 5.- Finsi {Paso 1}

Paso 6.- Si mínimo ≥ 1 y (Suma1+Suma2) ≥ 2(2n - 1) entonces

Paso 7.- Π es Débilmente Conexa

Paso 8.- Mostrar una realización

Paso 9.- Sino

Paso 10.- Escribir Mensaje

Paso 11.- Finsi {Paso 6}

Procedimiento para la propiedad de k-conexidad.

Entrada: Sucesión de números ordenada.

Salida: Mensaje si se cumple o no la propiedad y realización en caso de cumplirse.

Paso 1.- Si G admite h vértices de grado n en X y h vértices de grado n en Y con 1≤ h≤ n-1 entonces

Paso 2.- G es h-conexo

Paso 3.- Mostrar una realización

Paso 4.- Sino

Paso 5.- Escribir Mensaje

Paso 6.- Finsi

Procedimiento para la propiedad de hamiltonicidad.

Entrada: Sucesión de números ordenadas.

Salida: Mensaje si se cumple o no la propiedad y realización en caso de cumplirse.

Paso 1.- Si mínimo{a(1),b(1)} > p entonces

Paso 2.- Π es Forzosamente Hamiltoniano

Paso 3.- Mostrar una realización

Paso 4.- Sino

Paso 5.- Si mínimo{a(1),b(1)} ≥ 2 y numlados $\geq 2(n-1)$ + máximo{a(n),b(n)} y Π' es gráfica entonces

Paso 6.- Π es Débilmente Hamiltoniana

Paso 7.- Mostrar una realización

Paso 8.- Sino

Paso 9.- Escribir Mensaje

Paso 10.- Finsi {Paso 5}

Paso 11.- Finsi {Paso 1}

Procedimiento para la propiedad de bipanciclismo.

Entrada: Sucesión de números ordenadas.

Salida: Mensaje si se cumple o no la propiedad y realización en caso de cumplirse.

Paso 1.- Si mínimo{a(1),b(1)} $\geq (n+1)/2$ entonces

Paso 2.- Π es Forzosamente Bipancíclica

Paso 3.- Mostrar una realización

Paso 4.- Sino

Paso 5.- Si mínimo{a(1),b(1)} ≥ 2 y numlados $\geq 2(n-1)$ + máximo{a(n),b(n)} y si $n-1 \leq a_n \leq n$ o $n-1 \leq b_n \leq n$ entonces

Paso 6.- Π es Débilmente Bipancíclica

7.- Mostrar una realización

Paso 8.- Sino

Paso 9.- Escribir Mensaje

Paso 10.- Finsi {Paso 5}

Paso 11.- Finsi {Paso 1}

ANALISIS ALGORITMICO Y DESCRIPCION DE LOS METODOS A UTILIZAR.

1.- Descripción del Lenguaje.

En los años 70, Nicklaus Wirth profesor del Instituto Tecnológico de Zurich, diseñó un lenguaje con el fin de poder enseñar mejor las técnicas de programación; había desarrollado un lenguaje que se llamaría Pascal, derivado del ALGOL.

Unos años después que Nicklaus Wirth desarrolló el lenguaje Pascal, Borland International INC creó (aproximadamente en 1.983) un entorno con el que se podía programar en Pascal estándar y que además añadía por su cuenta nuevas y potentes instrucciones al Pascal conocido hasta entonces. A este nuevo entorno, denominado así por contener el editor y el compilador integrados en el mismo programa se le dió el nombre de Turbo Pascal, saliendo al mercado desde ese momento diversidad de versiones; y es precisamente este lenguaje de Programación (en su versión 6.0) el que se utilizó para la programación de los algoritmos en este trabajo.

2.- Para la lectura de los elementos se utilizó un arreglo de Registro formado por tres campos: 1.- Posición, que indica la posición o la secuencia de los elementos de la sucesión Π. (donde "Posa" indica posición de la secuencia Π_X y "Posb" indica posición de la sucesión Π_Y). 2.- Grado, que indica el valor (grado) de los elementos de la sucesión Π. 3.- Adyacente, este campo está formado por un arreglo de apuntadores, que apuntan hacia la lista de los vértices y_i adyacentes a un vértice x_i. La lista de adyacencia está formada por: 1.- Identificador, que indica el valor de la posición (posb) para el cual y_i es adyacente con x_i. 2.- Siguiente, es un puntero que apunta al siguiente identificador.

Al principio se tendrá un apuntador que apuntará al campo Posicion de mayor valor (grado) del Registro, para así iniciar el ordenamiento en forma decreciente.

Esquema de la Lectura.

3.- El Ordenamiento de la sucesión se hace por el método de la Burbuja.

Descripción. El método de ordenación a utilizar es el ampliamente conocido Ordenamiento de Burbuja . Una de sus características es que es de fácil comprensión y programación.

La idea básica en el ordenamiento de burbuja es la de recorrer el archivo en forma secuencial varias veces. Cada paso consiste en comparar un elemento en el archivo con un sucesor (x[i] con x[i+1]) o predecesor (x[i] con x[i-1]) e intercambiar los dos elementos si no están en el orden apropiado (creciente o decreciente, siendo este último el orden a tratar en nuestro trabajo).

En general, x[n-i+1] (donde n es el número de elementos a ordenar) estará en su posición apropiada despues de i iteraciones. El método es denominado burbuja debido a que cada número va "burbujeando" hacia arriba hasta su posición apropiada.

Puesto que cada iteración coloca un nuevo elemento en su posición apropiada, un archivo de n elementos requiere únicamente n-1 iteraciones.

Hay que hacer notar que como todos aquellos elementos en posiciones mayores o iguales a n-i+1 están ya en la posición apropiada despues de la iteración i, no se requiere compararlos en las iteraciones posteriores. Es decir que en el primer paso se hacen n-1 comparaciones, en el segundo paso n-2 y en el (n-1) paso solamente se hace una comparación (entre (x[1] y x[2]). Es decir, que el proceso se acelera a medida que se va avanzando a través de los pasos sucesivos. Así que, n-1 pasos son suficientes para ordenar un archivo de tamaño n.

Hay que hacer notar, que para eliminar pasos innecesarios debemos detectar el hecho de que el archivo ya estaba ordenado. Esta es una tarea muy simple, puesto que en un archivo ordenado, no hay ningún intercambio en uno de los pasos. Manteniendo un registro de sí se realiza o no un intercambio en un paso determinado, se puede decidir si es necesario hacer otros pasos adicionales. En este método si el archivo puede ser ordenado en n-1 pasos, el paso final no hace ningún intercambio.

La única ventaja del método de burbuja es que se requiere poco espacio adicional (una posición de memoria para retener el valor temporal para intercambio), y en el caso que el archivo este completamente ordenado (o casi completamente ordenado) se requiere un paso de n-1 comparaciones (sin intercambio).

Existen otras formas de mejorar el ordenamiento de burbuja, uno de estos es el de observar que el número de pasos necesarios, antes de que el archivo este ordenado, es la distancia mas larga mediante la cual un número se debe mover hacia el comienzo del arreglo. El ordenamiento de burbuja se puede acelerar teniendo pasos sucesivos en dirección opuesta, tal que el número de pasos es reducido.

4.- Las sucesiones originales se guardarán en otro arreglo, debido a que serán utilizadas más adelante, ya sea para la aplicación de las propiedades y/o para la representación del grafo.

5.- Por medio del Teorema 2, se verificará que la sucesión dada es bipartita gráfica, para que así, una vez que se compruebe las hipótesis y la I^n-propiedad se pueda construir una realización.

6.- El procedimiento para la representación del grafo se ejecuta una vez que se haya demostrado que la sucesión es bipartita gráfica, y al emitir el mensaje interactivo el usuario decida ver el gráfico.

Descripción. Para la gráfica del grafo, se trabajará con la Unidad Gráfica de Turbo Pascal. Para la ubicación de los vértices la Unidad Gráfica permite dibujar un pequeño punto en un lugar de la pantalla; dicho punto es la unidad más pequeña que se puede representar por pantalla, y esa Unidad se conoce como Pixel. Si se dibuja un sólo pixel en pantalla, es probable que se tarde en encontrar de lo pequeño que es. Por ello para obtener cada vértice se rellenaran "cuadritos" con una cantidad determinada de Pixeles para ser más visibles. Las adyacencias de cada x a los y vendrán dadas por líneas o trazos.

Para poder apreciar mejor el grafo en la pantalla, la cantidad de vértices no puede ser muy "grande".

7.- En el Procedimiento para la verificación de las propiedades al igual que en la representación del grafo se demostrará primero que la sucesión es gráfica y el usuario decidirá si desea verificar alguna. Entre estas propiedades se encuentran:

a.- Conexidad. Para la cual se utilizará:

Si $\min\{a_1, b_1\} \geq 1$ y $\sum_{i=1}^{p} a_i + \sum_{i=1}^{p} b_i \geq 2(2n-1)$ entonces Π es débilmente conexa. (Teorema 4)

b.- k- Conexidad. Para la cual se utilizará:

Si G admite h vértices de grado n en X y h vértices de grado n en Y con $1 \leq h \leq n-1$ entonces G es h-conexo. (Teorema 5)

c.- Débilmente 2-Conexo. Para el cual se utilizará:

Si $\min\{a_1,b_1\} \geq 2$ y $m \geq 2(n-1) + \max\{a_n,b_n\}$ entonces Π es debilmente 2-conexa. (Teorema 4)

Nota: Esta propiedad se verificará previamente si se desean ver las propiedades de débilmente Hamiltoniano y débilmente Bipancíclico.

d.- Hamiltonicidad. Para la cual se utilizará:

1.- Si $\min\{a_1,b_1\} > n/2$ entonces Π es forzosamente Hamiltoniano. (Teorema 10)

2.- Si $m(G) \geq (n-1)^2 + n+1$ entonces G es Hamiltoniano.

3.- Si Π es débilmente 2-conexa y Π^* es gráfica entonces Π es debilmente Hamiltoniano. (Corolario 9)

e.- Bipanciclismo. Para el cual se utilizará:

1.- Si $\min\{a_1,b_1\} \geq (n+1)/2$ entonces Π es forzosamente Bipancíclico. (Corolario 9)

2.- Si $m \geq (n-1)^2 + n+1$ entonces Π es forzosamente Bipanciclico.

3.- Si Π es debilmente 2-conexo y si $n-1 \leq a_n \leq n$ o $n-1 \leq b_n \leq n$ entonces Π es debilmente Bipancíclico. (Teorema 11)

BIBLIOGRAFIA

AMAR D.; FAVARON O.; MAGO P. AND ORDAZ O. 1995. Biclousure and bistability in a balanced bipartite graph, *J. Graph Theory* 4: 513-529.

HARARY F. 1969. *Graph Theory.* Addison-Wesley Publishing Company Massachusetts.

MITCHEM J. AND SCHMEICHE E. F. 1982 Bipartite graphs with cycles of all even lengths, *J. Graph Theory* 6: 429-439.

MOON J. AND MOSER L. 1963. On hamiltonian bipartite graphs. *Israel J. Math.* 1: 163-165.

MAGO P., LAREZ G., VILLARROEL F. Sucesión de grados de grafos bipartitos balanceados. Preprint.