

UN ALGORITMO PARALELO PARA EL PROBLEMA DEL CONJUNTO INDEPENDIENTE

RAFAEL LÓPEZ BRACHO* – MARÍA PAULA ORTUÑO SÁNCHEZ**

Recibido: 12 Junio 2000

Resumen

Un conjunto S de vértices de una gráfica G es independiente si no existen dos vértices de S que sean adyacentes, esto es, la subgráfica de G inducida por S no tiene aristas. En este trabajo presentaremos un algoritmo paralelo que permite la obtención de todos los conjuntos independientes maximales de una gráfica. Presentaremos los fundamentos del algoritmo y algunas propiedades derivadas de éstos.

Palabras clave: Gráfica, Conjunto Independiente, Número de Independencia, Número de Estabilidad, Algoritmo Paralelo.

Abstract

A set S of vertices in a graph G is independent if there does not exist two adjacent vertices in S , that is, the subgraph of G induced by S does not have edges. In this work we present a parallel algorithm that permits to obtain all maximal independent sets in a graph. We present the foundations of the algorithm and some properties.

Keywords: Graph, Independent Set, Independence Number, Stability Number, Parallel Algorithm.

Mathematics Subject Classification: 90C27, 68R10

1. El problema del conjunto independiente

Considérese una gráfica $G = (V, A)$. Un conjunto S de vértices de G es independiente si no existen dos vértices de S que sean adyacentes, esto es, la subgráfica de G inducida

*Universidad Autónoma Metropolitana-Azcapotzalco, Av. San Pablo 180, Colonia Reynosa Tamaulipas, 02200 México, D.F. México. E-Mail: rlb@correo.azc.uam.mx

**Misma dirección; E-Mail: pos@correo.azc.uam.mx

por S no tiene aristas. El máximo número de vértices en un conjunto independiente de G se denota por $\alpha(G)$ y se llama número de independencia o número de estabilidad de G . Un conjunto independiente S es maximal, si para todo $v \in V - S$, $S \cup \{v\}$ no es independiente. Claramente $|S| \leq \alpha(G)$ y la igualdad es alcanzada sólo por algunos conjuntos independientes. El problema de determinar el número de independencia de una gráfica es un problema NP-Completo, de hecho es tan difícil determinar el número de independencia de una gráfica, como encontrar todos los conjuntos independientes maximales de la misma. Los algoritmos exactos que resuelven el primer problema resuelven en sí el segundo. Por lo anterior, vamos a considerar que El problema del Conjunto Independiente consiste en determinar todos los conjuntos independientes maximales de una gráfica.

2. Algunas definiciones y propiedades

Vamos a considerar gráficas simples. Para una gráfica $G = (V, A)$ denotaremos por $V(G)$ al conjunto de vértices de G y por $A(G)$ al conjunto de aristas. $\Delta(G)$ va a representar el grado máximo de la gráfica. La subgráfica de $G = (V, A)$ inducida por un conjunto $V - V'$ la denotaremos por $G - V'$. Si V' contiene sólo un elemento v , la subgráfica inducida correspondiente la denotaremos $G - v$. $\Gamma(v)$ va a representar al conjunto de vértices adyacentes al vértice v en una gráfica dada.

Una gráfica G es una gráfica perfecta si $\alpha(G_A) = \theta(G_A)$, para toda subgráfica inducida G_A de G , donde $\theta(G_A)$ es el número mínimo de clanes que particionan $V(G_A)$.

Sean $\mathbf{S}(G)$ la familia de conjuntos independientes maximales de G y $\mathbf{S}'(G - v)$ la familia de conjuntos independientes maximales de G que contienen algún elemento de $\Gamma(v)$, esto es: $S \in \mathbf{S}'(G - v) \Leftrightarrow S \in \mathbf{S}(G)$ y $\Gamma(v) \cap S \neq \emptyset$.

Sean $\sigma(G) = |\mathbf{S}(G)|$ y $\sigma'(G) = |\mathbf{S}'(G - v)|$.

A continuación probaremos algunos resultados importantes en los que se apoyará el algoritmo:

Lema 1 Sea $G = (V, A)$ una gráfica simple y v un vértice de G . Si S es un conjunto de vértices independiente maximal de $G - v$, entonces es también un conjunto independiente maximal de G , si y sólo si, $\Gamma(v) \cap S \neq \emptyset$.

PRUEBA: Claramente S es un conjunto independiente de G . Si $\Gamma(v) \cap S = \emptyset$, entonces el conjunto $S \cup \{v\}$ es también independiente y en consecuencia S no es independiente maximal de G . Si $\Gamma(v) \cap S \neq \emptyset$, entonces el conjunto $S \cup \{v\}$ no es independiente. Por lo anterior, S es independiente maximal de G . ■

Teorema 2 Sea $G = (V, A)$ una gráfica simple. $\mathbf{S}(G) = \mathbf{S}(G - \Gamma(v)) \cup \mathbf{S}'(G - v)$ para cada vértice v de G .

PRUEBA: Considérese un vértice v de G . De acuerdo con ese vértice, definimos $\mathbf{S}_1 = \{S \in \mathbf{S}(G) : v \in S\}$ y $\mathbf{S}_2 = \{S \in \mathbf{S}(G) : v \notin S\}$. Claramente $\mathbf{S}_1 \cap \mathbf{S}_2 = \emptyset$ y $\mathbf{S}_1 \cup \mathbf{S}_2 =$

$\mathbf{S}(G)$, de donde \mathbf{S}_1 y \mathbf{S}_2 constituyen una partición de $\mathbf{S}(G)$. Cada elemento de \mathbf{S}_1 no contiene elementos de $\Gamma(v)$, porque en caso contrario no sería independiente, de donde $\mathbf{S}_1 \subseteq \mathbf{S}(G - \Gamma(v))$, además, cada elemento de $\mathbf{S}(G - \Gamma(v))$ debe contener a v , porque si no, no sería maximal. Entonces $\mathbf{S}(G - \Gamma(v)) \subseteq \mathbf{S}_1$ y en consecuencia $\mathbf{S}_1 = \mathbf{S}(G - \Gamma(v))$. Por otra parte, $\mathbf{S}_2 = \mathbf{S}'(G - v)$ por Lema 1, de donde se obtiene el resultado. ■

Corolario 3 $\sigma(G) = \sigma(G - \Gamma(v)) + \sigma'(G - v)$ para cada vértice v de G .

3. Descripción del algoritmo propuesto

El algoritmo que presentamos es de enumeración implícita. Para efecto de la aplicación del algoritmo, diremos que una Gráfica es Básica si sus componentes conexas son alguna de las siguientes gráficas:

- a) Un solo vértice (Gráfica sin aristas).
- b) Una trayectoria de longitud 1.
- c) Una trayectoria de longitud 2.
- d) Una trayectoria de longitud 3.
- e) Una trayectoria de longitud 4.
- f) Un ciclo de longitud 3.

Si se denotan las gráficas anteriores por su secuencia de vértices como sigue: a) u_0 , b) u_0u_1 , c) $u_0u_1u_2$, d) $u_0u_1u_2u_3$, e) $u_0u_1u_2u_3u_4$, y f) $u_0u_1u_2u_0$, los conjuntos independientes maximales correspondientes serán: a) $\{u_0\}$, b) $\{u_0\}$ y $\{u_1\}$; c) $\{u_0, u_2\}$ y $\{u_1\}$; d) $\{u_0, u_2\}$, $\{u_0, u_3\}$ y $\{u_1, u_3\}$; e) $\{u_0, u_2, u_4\}$, $\{u_0, u_3\}$, $\{u_1, u_3\}$ y $\{u_1, u_4\}$; y f) $\{u_0\}$, $\{u_1\}$ y $\{u_2\}$.

Considerando que al unir dos conjuntos independientes maximales de distintas componentes conexas, se obtendrá un conjunto independiente maximal de la gráfica formada por dichas componentes, es posible obtener todos los conjuntos independientes maximales de una gráfica básica a través de la repetición de esta operación, incorporando una a una, todas las componentes conexas de la gráfica.

Considérese una gráfica simple $G = (V, A)$ de orden n . El algoritmo va a construir un árbol binario con raíz en $G_0 = G$. Cada vértice de este árbol corresponderá a una gráfica inducida de G_0 , y en particular cada vértice terminal corresponderá a una gráfica básica. A cada vértice del árbol se puede llegar desde la raíz, a través de una sucesión de operaciones aplicadas a las gráficas asociadas a los vértices precedentes. Así, si G_i es la gráfica asociada a un vértice que se encuentra a una distancia k de la raíz y v_i^* es un vértice de grado máximo de G_i , si G_i no es básica, a partir de ésta se obtienen dos nuevas gráficas de acuerdo a las operaciones siguientes:

1. Inclusión. Se obtiene la subgráfica inducida resultado de eliminar de G_i los vértices adyacentes a v_i^* . Esta gráfica será denotada G_i nuevamente.

2. Exclusión. Se obtiene la subgráfica inducida resultado de eliminar de G_i el vértice v_i^* . Esta gráfica será denotada G_{i+2^k} .

De acuerdo con Lema 1, los conjuntos independientes que se obtengan en la gráfica asociada a un vértice terminal del árbol, serán maximales, si y sólo si, contienen algún vértice adyacente a los vértices eliminados en las operaciones precedentes. Esto significa que cada conjunto independiente será maximal si contiene al vértice v_i^* en cada operación de inclusión y algún vértice de $\Gamma(v_i^*)$ en cada operación de exclusión, llevadas a cabo en los vértices previos. Con base en lo anterior obtenemos el siguiente

ALGORITMO

Sea $G = (V, A)$ una gráfica simple de orden n y Sea $\mathbf{S}(G)$ la familia de conjuntos independientes maximales de G . k va a denotar la profundidad del árbol, $\Gamma(v_i^*)$ el conjunto de vértices vecinos a v_i^* en la gráfica G_i y T_i será el conjunto de índices de los vértices en la trayectoria de G_0 a G_i , a los que se llegó con una operación de exclusión.

Etapa Inicial:

Considérense $\mathbf{S}(G) = \emptyset$, $k = 0$, $T_0 = \emptyset$ y $G_0 = G$. Sea v_0^* un vértice de grado máximo de G_0 . Si G_0 no es básica, constrúyanse las gráficas $G_0 = G_0 - \Gamma(v_0^*)$ y $G_1 = G_0 - v_0^*$, considérese el conjunto $T_1 = T_0 \cup \{1\}$, tómese $k = k + 1$ y pase a la etapa intermedia. Si G_0 es básica, pase a la etapa final.

Etapa Intermedia (División):

Se analizan los vértices situados a distancia k del vértice raíz, desde $i = 0$ hasta $i = 2^k - 1$, proceda como sigue: considérese la gráfica G_i . Si G_i no es básica, sea v_i^* un vértice de grado máximo de G_i construya las gráficas $G_i = G_i - \Gamma_i(v_i^*)$ y $G_{i+s} = G_i - v_i^*$, donde $s = 2^k$, y considere el conjunto $T_{i+s} = T_i \cup \{i + 2^k\}$; mientras que si G_i es básica o no existe, espere la etapa final. Después de revisar todas las gráficas, si alguna fue no básica, tome $k = k + 1$ y repita esta etapa, si no pase a la etapa final.

Etapa Final (Construcción):

Todos los vértices terminales corresponden a gráficas básicas. Para $i = 0$ hasta $i = 2^k - 1$, proceda como sigue: considere la gráfica G_i y construya todos los conjuntos independientes maximales de ésta. Cada conjunto S que satisfaga $\Gamma_j(v_j^*) \cap S \neq \emptyset$, para cada $j \in T_i$ es un conjunto maximal de G , entonces tome $\mathbf{S}(G) = \mathbf{S}(G) \cup \{S\}$. Al finalizar este proceso, el conjunto $\mathbf{S}(G)$ contendrá todos los conjuntos independientes maximales de G . Fin.

4. Prueba del algoritmo

El algoritmo construye un árbol binario durante su aplicación, a través de las operaciones de inclusión de un vértice (operación 0) y exclusión del mismo (operación 1). En el vértice i del árbol, a profundidad k , se asocia la gráfica denotada G_i , donde i es el número que corresponde a una cadena binaria de ceros y unos de longitud k . La cadena binaria expresa las operaciones de inclusión o exclusión que se efectuaron en los vértices predecesores en el árbol, así al efectuar las dos operaciones en el vértice i se obtiene una bifurcación a dos nuevos vértices para los cuales se agrega un cero o un uno, en el extremo

izquierdo de la cadena binaria, según se haya aplicado la operación de inclusión o exclusión de un vértice, respectivamente. (Vea Figura 1).

De acuerdo con los resultados de la sección 2, $\mathbf{S}(G_i)$ denota la familia de conjuntos independientes maximales de G_i . Por el Teorema 2, para el vértice v_i^* seleccionado se tiene que $\mathbf{S}(G_i) = \mathbf{S}(G_i - \Gamma_i(v_i^*)) \cup \mathbf{S}'(G_i - v_i^*)$ con $\mathbf{S}(G_i - \Gamma_i(v_i^*)) \cap \mathbf{S}(G_i - v_i^*) = \emptyset$. Por consiguiente, cuando G_i corresponde a un vértice no terminal, para obtener los conjuntos independientes maximales de G_i , se parte de los conjuntos independientes maximales de $G_i - \Gamma_i(v_i^*)$ obtenidos a través de la operación de inclusión y los conjuntos independientes maximales de $G_i - v_i^*$ que intersectan con $\Gamma_i(v_i^*)$, obtenidos a través de la operación de exclusión. A partir de la obtención directa de los conjuntos independientes maximales de las gráficas básicas, correspondientes a los vértices terminales del árbol binario, el algoritmo obtiene de todos éstos la familia de conjuntos independientes maximales de G .

Considerando que en la operación de inclusión se eliminan $\Delta(G_i)$ vértices y en la operación de exclusión se elimina un solo vértice, se observa que, mientras el vértice terminal del árbol binario, obtenido con una secuencia de sólo operaciones de inclusión se encuentra a una distancia máxima $\alpha(G)$ de la raíz, el vértice terminal obtenido con una secuencia de sólo operaciones de exclusión se encuentra a una distancia máxima $|V(G)|$. Por lo anterior, será la secuencia de operaciones de exclusión la que determine la profundidad total del árbol. La profundidad máxima del mismo será entonces menor o igual al orden de la gráfica. Por otra parte, al tratarse de un árbol binario, el número máximo de gráficas terminales será entonces menor o igual a 2^n , donde $n = |V(G)|$. Cada vértice terminal del árbol corresponde a una subgráfica inducida de G básica. Lógicamente, el algoritmo, por ser enumerativo, es de tipo exponencial, pero para algunas gráficas particulares puede tener un buen desempeño. Además, a través de un proceso de paralelización el algoritmo puede mejorar ostensiblemente su desempeño. Este proceso lo presentamos en la sección siguiente.

5. Paralelización del algoritmo

En la paralelización del algoritmo, el procesador maestro recibirá la gráfica G y creará una réplica de ésta, denotándola G_0 , para enviarla a un procesador disponible. Posteriormente, el procesador maestro recibirá las nuevas gráficas generadas durante la aplicación del algoritmo, así como los conjuntos independientes obtenidos a partir de las gráficas básicas. Con las gráficas generadas que reciba, a través de un sistema de líneas de espera, procederá a enviarlas al primer procesador que se tenga disponible. Si el número de procesadores lo permitiera, sería posible realizar el análisis y construcción de los conjuntos independientes de las gráficas básicas obtenidas, en forma simultánea. El mecanismo de asignación de las gráficas a los procesadores se describe en el propio algoritmo.

ALGORITMO

Sea $G = (V, A)$ una gráfica simple y sea $\mathbf{S}(G)$ la familia de conjuntos independientes maximales de G . k va a denotar la profundidad del vértice asociado a la gráfica G_i , en el árbol binario generado por la aplicación del algoritmo, $\Gamma_i(v_i^*)$ el conjunto de vértices

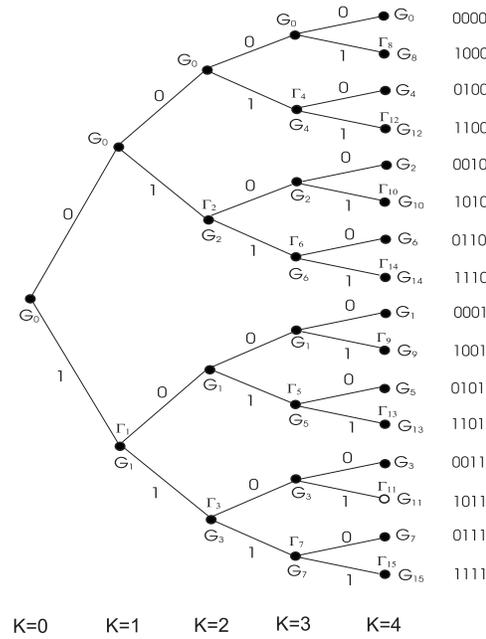


Figura 1: Árbol

vecinos a v_i^* en la gráfica G_i y T_i será el conjunto de índices de los vértices en la trayectoria de G_0 a G_i , a los que se llegó con una operación de exclusión. Mientras se tengan en el procesador maestro subgráficas de G sin analizar, éstas se asignaran a los procesadores que se vayan desocupando, de acuerdo a un proceso de líneas de espera, en el que se asigne primero la primera gráfica que se haya recibido.

Etapa Inicial:

Considere $\mathbf{S}(G) = \emptyset$, $k = 0$, $T_0 = \emptyset$ y $G_0 = G$. Asigne k , T_0 y G_0 al procesador 0 y continúe con la etapa general.

Etapa General:

Si al procesador p se le asigna la gráfica G_i , el conjunto T_i y el entero k , debe proceder en la forma siguiente:

Mientras G_i no sea básica, efectúe lo siguiente:

Considere v_i^* un vértice de grado máximo de G_i y construya las gráficas $G_i = G_i - \Gamma_i(v_i^*)$ y $G_{i+s} = G_i - v_i^*$, considere el conjunto $T_{i+s} = T_i \cup \{i+s\}$ donde $s = 2^k$. Envíe al procesador maestro la gráfica G_{i+s} junto con el conjunto T_{i+s} y el entero $k + 1$; mientras que en el procesador p conserve la gráfica G_i actualizada, junto con el conjunto T_i y tome $k = k + 1$ para continuar.

Cuando G_i sea básica, efectúe lo siguiente:

Construya todos los conjuntos independientes maximales de G_i . Cada conjunto S que satisfaga $\Gamma_j(v_j^*) \cap S \neq \emptyset$, para cada $j \in T_i$ es un conjunto maximal de G , entonces envíe todos los conjuntos así obtenidos al procesador maestro para que sean agregados al con-

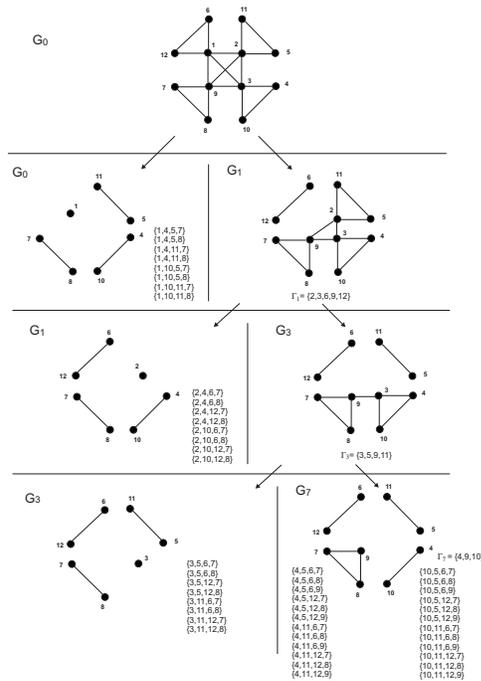


Figura 2: Ejemplo de aplicación

junto $S(G)$. Termine la actividad en el procesador p . El procesador maestro podrá enviar nuevamente una gráfica al procesador p .

Etapas Final:

Cuando haya terminado la actividad en todos los procesadores, el conjunto $S(G)$ contendrá todos los conjuntos independientes maximales de G . Fin.

6. Ejemplo de aplicación

En Figura 2 se muestra la aplicación del algoritmo a una gráfica G . Cada subgráfica inducida que se va construyendo se enumera y se muestra de acuerdo al vértice que le corresponde en el árbol.

7. Otros resultados

Si se denotan por P_n a la trayectoria de orden n y por C_n al ciclo de orden n , de Teorema 2., Corolario 3. y la aplicación del algoritmo, se obtienen los siguientes resultados:

Teorema 4 $\sigma(P_n) = \sigma(P_{n-3}) + \sigma(P_{n-2})$, si $n \geq 4$.

PRUEBA: Supongamos que P_n es la secuencia de vértices $u_0u_1 \dots u_{n-3}u_{n-2}u_{n-1}$. Si tomamos el vértice u_{n-2} y consideramos las subgráficas inducidas $P_n - \Gamma(u_{n-2})$ y $P_n - u_{n-2}$, tenemos que $P_n - \Gamma(u_{n-2}) \sim P_{n-3} \cup \{u_{n-2}\}$ y $P_n - u_{n-2} \sim P_{n-2} \cup \{u_{n-1}\}$. Es claro que u_{n-2} y u_{n-1} se pueden incorporar a cada conjunto independiente maximal de P_{n-3} y P_{n-2} , respectivamente, para obtener los conjuntos independientes maximales de $P_{n-3} \cup \{u_{n-2}\}$ y de $P_{n-2} \cup \{u_{n-1}\}$, de donde se tiene $\sigma(P_n - \Gamma(u_{n-2})) = \sigma(P_{n-3} \cup \{u_{n-2}\})$ y $\sigma(P_n - u_{n-2}) = \sigma(P_{n-2} \cup \{u_{n-1}\})$. Por Corolario 3 obtenemos el resultado. ■

Lema 5 *Todos los conjuntos independientes maximales de P_n contienen al menos uno de los vértices terminales de la gráfica, si n es par.*

PRUEBA: Supongamos que S es un conjunto independiente maximal de P_n y consideremos los vértices de $S' = V - S$. Claramente, S' no puede contener una subtrayectoria $u_{i-1}u_iu_{i+1}$ de P_n . Si S' contiene un par de vértices adyacentes $u_{i-1}u_i$ de P_n , ninguno de estos puede ser terminal de P_n . Si S' no tiene vértices adyacentes, los vértices de P_n constituyen una secuencia alternada de pertenencia a S y S' ; Dado que n es par, en este caso uno de los dos extremos de P_n pertenece a S . ■

Corolario 6 *Sólo uno de los conjuntos independientes maximales de P_n no contiene a ninguno de los vértices terminales de la gráfica, si n es impar.*

PRUEBA: Si en la prueba del lema anterior suponemos que n es impar, cuando S' no tiene vértices adyacentes, los vértices de P_n constituyen una secuencia alternada de pertenencia a S y S' ; tanto S como S' son independientes maximales; Dado que n es impar, en este caso los dos extremos de P_n pertenecen a S o los dos extremos de P_n pertenecen a S' . ■

Teorema 7 $\sigma(C_n) = \sigma(P_{n-3}) + \sigma(P_{n-1})$, si $n \geq 4$ e impar y $\sigma(C_n) = \sigma(P_{n-3}) + \sigma(P_{n-1}) - 1$, si $n \geq 4$ y par.

PRUEBA: Supongamos que C_n es la secuencia de vértices $u_0u_1 \dots u_{n-3}u_{n-2}u_{n-1}u_0$. Si tomamos el vértice u_0 y consideramos las subgráficas inducidas $C_n - \Gamma(u_0)$ y $C_n - u_0$, tenemos que $C_n - \Gamma(u_0) \sim P_{n-3} \cup \{u_0\}$ y $C_n - u_0 \sim P_{n-1}$. Es claro que u_0 se puede incorporar a cada conjunto independiente maximal de P_{n-3} , para obtener los conjuntos independientes maximales de $P_{n-3} \cup \{u_0\}$. Por otro lado, por Lema 5, cuando n es impar $S \cap \Gamma(u_0) \neq \emptyset$ para todo conjunto independiente maximal de $C_n - u_0$, mientras que, por Corolario 6, cuando n es par $S \cap \Gamma(u_0) = \emptyset$ sólo para uno de los conjuntos independientes maximales de $C_n - u_0$. Por Corolario 3. Obtenemos el resultado. ■

Observación 8 *Si G es una gráfica perfecta, cada una de las subgráficas inducidas que se vayan obteniendo en el algoritmo conservará la propiedad de ser perfecta. Así, $\alpha(G_i) = \theta(G_i)$ en cada vértice generado por el algoritmo, de donde $\alpha(G_i)$ puede indicar la profundidad a alcanzar por el árbol a partir del vértice i , mientras que $\theta(G_i)$ podrá indicar el ancho o número de vértices terminales del árbol. Así, si alguna gráfica G_i es isomorfa a P_n , no obstante que $\alpha(P_n) \cong n/2$, por lo que el árbol puede alcanzar mucha profundidad, en cada bifurcación del árbol se eliminan dos vértices de la trayectoria en cada rama y no*

tendrá un número muy grande de vértices terminales. Esta situación puede observarse en la gráfica de Figura 3, en la cual los conjuntos independientes marcados con X , no son maximales, porque no cumplen la condición de validez para la operación de exclusión.

8. Conclusiones

La obtención de cotas para el número de conjuntos independientes maximales de gráficas es un problema que ha sido abordado por Füredi [3] y Griggs, Grinstead y Guichard [6]. Este problema, junto con el de la búsqueda de algoritmos eficientes para gráficas con características particulares constituyen problemas interesantes para tratar. Consideramos que las ideas planteadas en el desarrollo del algoritmo que hemos presentado pueden ser de utilidad no sólo para tratar esos problemas, sino para el desarrollo de algoritmos paralelos eficientes.

Referencias

- [1] Berge, C. (1970) *Graphes et Hypergraphes*. Dunod, Paris.
- [2] Christofides, N. (1975) *Graph Theory: An Algorithmic Approach*. Academic Press, New York.
- [3] Füredi, Z. (1987) “The ‘number of maximal independent sets in connected graphs’”, *Journal of Graph Theory* **11**(4): 463–470.
- [4] Garey, M. R.; Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co. Publishers, San Francisco.
- [5] Golumbic, M. C. (1980) *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York.
- [6] Griggs, J. R.; Grinstead, C. M.; Guichard, D.R. (1988) “The number of maximal independent sets in a connected graph”, *Discrete Mathematics* **68**: 211–220.
- [7] Gutiérrez, M. A. (1991) *La Técnica de Recocido Simulado*. Tesis doctoral, UNAM, México.
- [8] Harary, F. (1972) *Graph Theory*. Addison Wesley, Reading MS.
- [9] Neumann-Lara, V. (1985) “Introducción a la teoría de gráficas”, *IV Coloquio del Departamento de Matemáticas*, CINVESTAV, México.
- [10] Papadimitriou, C. H.; Steiglitz, K. (1982) *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall Inc., Englewood Cliffs N.J.
- [11] Syslo, M.; Deo, N.; Kowalik, J. (1983) *Discrete Optimization Algorithms with Pascal Programs*. Prentice Hall Inc., Englewood Cliffs N.J.

