

## ALGORITMOS LINEALES PARA TESTEAR CONEXION, BICONEXION Y PLANARIDAD DE GRAFOS

**Gustavo Rossi (\*)**

Centro de Técnicas Analógico-Digitales  
Facultad de Ingeniería, Univ. Nac. de la Plata  
La Plata, Argentina

### INTRODUCCION:

Son bastantes conocidas las relaciones entre la teoría de grafos y otras disciplinas (la Química y la Electrónica entre otras) y la importancia que tiene en los problemas reales decidir si un grafo dado es planar o sea si es posible una representación gráfica del mismo en el plano sin cruces entre aristas, y encontrar dicha representación.

Este problema cobra fundamental importancia en el diseño de circuitos electrónicos impresos y máscaras de híbridos e integrados, puesto que representa un notable ahorro al no necesitar una doble faz de aluminizado en circuitos relativamente sencillos (ver Ref. 3). De la misma manera es importante decidir si un grafo es conexo o biconexo. (Para mayores detalles respecto a estas definiciones ver la sección siguiente).

Estos problemas justifican ampliamente la búsqueda de métodos para el testeo de planaridad, conexión y biconexión entre otros, y más aún justifican el empleo de métodos eficientes en lo que a tiempo de cómputo se refiere, puesto que en las aplicaciones reales los grafos a ser analizados son en general de gran magnitud en número de vértices y aristas.

El análisis que se presenta a continuación ha sido realizado en el Centro de Técnicas Analógico-Digitales de la Facultad de Ingeniería de la Universidad Nacional de La Plata, como base de un conjunto de programas de ayuda al diseño automático de circuitos y componentes electrónicos.

(\*) Trabajo realizado bajo la dirección del Ing. Antonio A. Quijano.

El autor agradece la colaboración del Sr. Carlos Meza.

## BREVE DESCRIPCION DE LOS ALGORITMOS USADOS:

Por razones de claridad se presenta un análisis separado de la resolución de los tres problemas planteados (conexión, biconexión y planaridad) ordenados por grado de complejidad.

Para mayor comprensión de ciertos tópicos teóricos, ver definiciones en el Apéndice 2.

### a) Testeo de Conexión:

Se dice que un grafo  $G$  es conexo si para todo par de vértices  $x$  e  $y$  de  $G$  existe un camino de  $x$  a  $y$ .

Para resolver este problema se ha utilizado una técnica de búsqueda que consiste en reenumerar los vértices del grafo según la siguiente regla: el vértice 1 recibe el número 1; dado un vértice numerado como  $I$  se numera como  $I+1$  al primer vértice adyacente a él, aún no numerado y de no existir tal vértice se procede igual con el vértice  $I-1$ .

Es sencillo ver que si un grafo es conexo, todos sus vértices son numerados durante el proceso, mientras que si  $G$  es no conexo existe al menos un vértice  $x$  tal que  $x$  no será numerado en la búsqueda.

Se podrá ver a continuación como esta técnica sumamente sencilla permite simplificar la resolución del testeo de biconexión de un grafo.

### b) Testeo de Biconexión:

Se dice que un grafo  $G$  es biconexo si al eliminar un vértice cualquiera  $x$  de  $G$ , y todas las aristas que inciden él, se obtiene un subgrafo de  $G$  conexo o equivalentemente, si dados 3 vértices cualesquiera,  $v$ ,  $w$ ,  $z$  existe un camino de  $v$  a  $w$  que no contiene a  $z$ . Si para alguna terna  $v, w, z$ , de vértices, todo camino de  $v$  a  $w$  contiene a  $z$ ,  $z$  se denomina punto de articulación. Esta última definición permite dar la siguiente caracterización: Un grafo es biconexo sí y sólo sí no contiene puntos de articulación.

Es evidente que este problema tiene una solución trivial aplicando el algoritmo de conexión, descrito en la sección anterior, a cada uno de los subgrafos de  $G$  obtenidos eliminando uno a uno todos los vértices del grafo. Sin embargo esta solución dista en mucho de ser la óptima, lo que podrá apreciarse claramente al presentarse el análisis estadístico de estos métodos.

Para analizar la biconexión de un grafo se ha utilizado un método propuesto por Tarjan (Ref. 1) basado en la técnica de búsqueda explicada en la sección anterior y que consiste en convertir el grafo  $G$  en un grafo orientado compuesto por un árbol generador  $T$  y un conjunto de aristas llama-

das frondas; definiendo durante el proceso de numeración una función ( $LOWPT(v)$ ) de los vértices, que se obtiene intuitivamente de la siguiente manera:  $LOWPT(v)$  es el menor de los vértices que se pueden obtener siguiendo un camino en  $T$  con inicio en  $v$ , teniendo como última arista una fronda.

Esta función permite aplicar muy fácilmente la caracterización de grafo biconexo dada anteriormente, basándose en la siguiente propiedad: Si la arista  $(a,v) \in T$  y  $LOWPT(v) \geq a$ , entonces  $a$  es un punto de articulación.

(Para una definición más rigurosa de la función  $LOWPT$  ver referencia 1).

### c) Testeo de Planaridad:

Se dijo en la INTRODUCCION que un grafo es planar si es posible encontrar una representación en el plano tal que no existan cruces entre aristas.

Para resolver este problema existen dos corrientes perfectamente definidas: la analítica, que basa su análisis en un criterio de Kuratowsky, y la constructiva que busca encontrar una representación plana del grafo. En esta corriente se encuentran los métodos propuestos por Demoucron, Malgrange y Pertuiset (ver Ref. 2 y 3), y Hopcroft y Tarjan (ver Ref. 4) El análisis que se presenta en este trabajo se basa en la implementación realizada del algoritmo propuesto por Hopcroft y Tarjan, aplicable a grafos biconexos, y que básicamente se puede resumir en los siguientes pasos:

- 1) Numeración de los vértices mediante la técnica de búsqueda explicada anteriormente y definición de dos funciones de los vértices ( $LOWPT$  y  $LOWPT2$ ).
- 2) Reordenamiento de las listas de adyacencia.
- 3) Búsqueda de caminos en el grafo con las nuevas listas de adyacencia.
- 4) Trazado de los caminos en el plano.

Cuando el paso 4) finaliza exitosamente se ha encontrado la representación plana del grafo, en caso contrario el grafo es no planar.

La implementación de los pasos 3) y 4) (con varias diferencias respecto al método propuesto por Hopcroft y Tarjan) hace que el algoritmo de testeo sea sumamente eficiente en tiempo de C.P.U., sobre todo para grafos de gran magnitud.

### ANALISIS DE LA EFICIENCIA DE LOS METODOS PRESENTADOS:

En todo problema relacionado con la teoría de grafos la eficiencia de un algoritmo se mide con la relación entre el tiem

po de cómputo y el número de vértices y aristas.

Los algoritmos presentados en este trabajo tienen una cota lineal respecto al tiempo de cómputo; para ser más precisos existen constantes A, B y C tal que  $T \leq AN + BM + C$ , para cualquier grafo con N vértices y M aristas, lo que para este tipo de problemas representa el máximo grado de eficiencia posible. (De aquí en más T representará el tiempo en segundos).

Para realizar este análisis se han implementado diversas rutinas de construcción de grafos, las que fueron utilizadas posteriormente como datos para los programas de inspección de conexión, biconexión y planaridad. (Una descripción completa de estos programas se presenta en el Apéndice 1).

Nuevamente, para mayor cohesión, es conveniente analizar los resultados obtenidos en la resolución de cada uno de los problemas por separado: (Los algoritmos aquí descriptos han sido implementados en lenguaje PL1 y ejecutados en la IBM 360/50 de la UNLP).

a) CONEXION: En el análisis de este problema puede verse claramente la importancia que tiene elegir las estructuras más adecuadas para la representación de grafos en computadora. Para ejemplificar se analizó la conexión de grafos de grado 3, 4 y 5 utilizando el método presentado anteriormente con dos estructuras de información: matriz de incidencia y listas de adyacencia, (para una descripción de estas estructuras ver Apéndice 2). Los resultados pueden verse en la Figura 1. Se aprecia allí claramente el comportamiento lineal del método descripto, respondiendo a ecuaciones  $T = 0.034 N + 1,2$ ,  $T = 0.035 N + 1,4$  y  $T = 0.036 N + 1,5$  para los grafos de grado 3, 4 y 5 respectivamente, y la notable diferencia existente respecto a los tiempos del método que utiliza la matriz de incidencia.

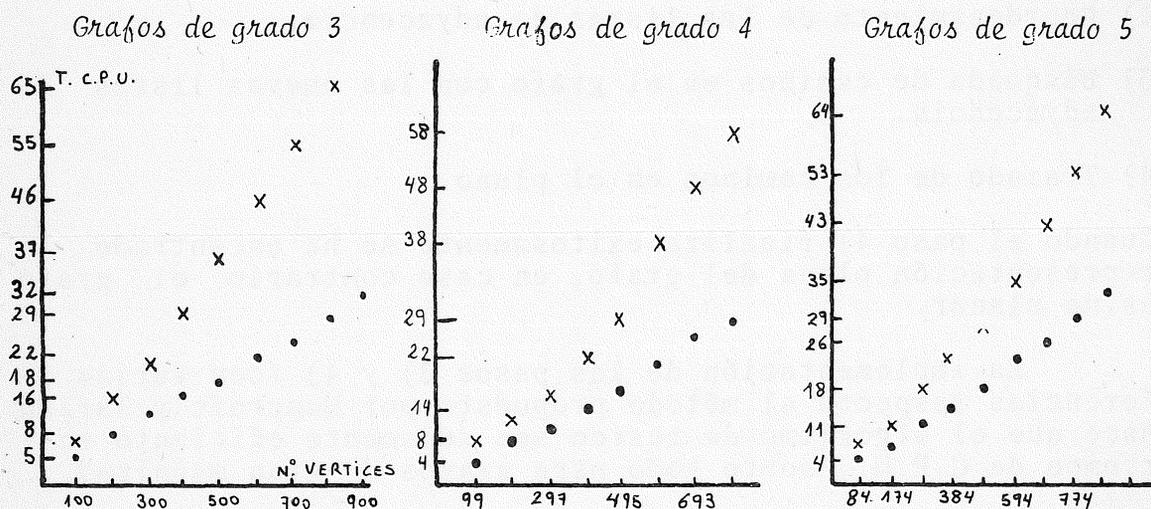


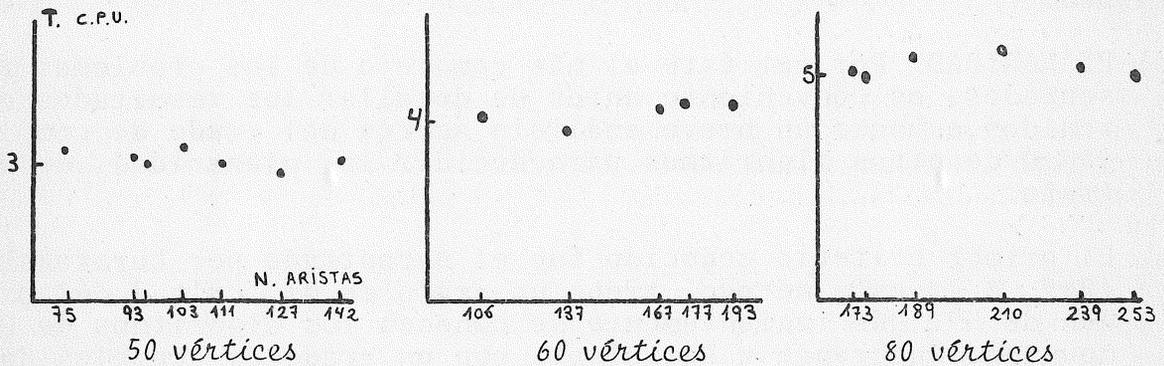
FIGURA 1: Test de Conexión.

(. usando listas de adyacencia · x usando matriz de incidencia)

Finalmente, en la Figura 2 se puede ver algunos de los resultados obtenidos mediante el empleo de grafos aleatorios generados automáticamente, y con los cuales, mediante el empleo de técnicas estadísticas, se obtuvo la siguiente ecuación:

$T = 0.036N + 0.0005M + 1,8$ , que representa la relación aproximada entre tiempo de cómputo y número de vértices y aristas para testear la conexión de grafos arbitrarios. Con dicha ecuación se confirma analíticamente la independencia del tiempo de cómputo respecto al número de aristas, lo que también puede apreciarse en la figura

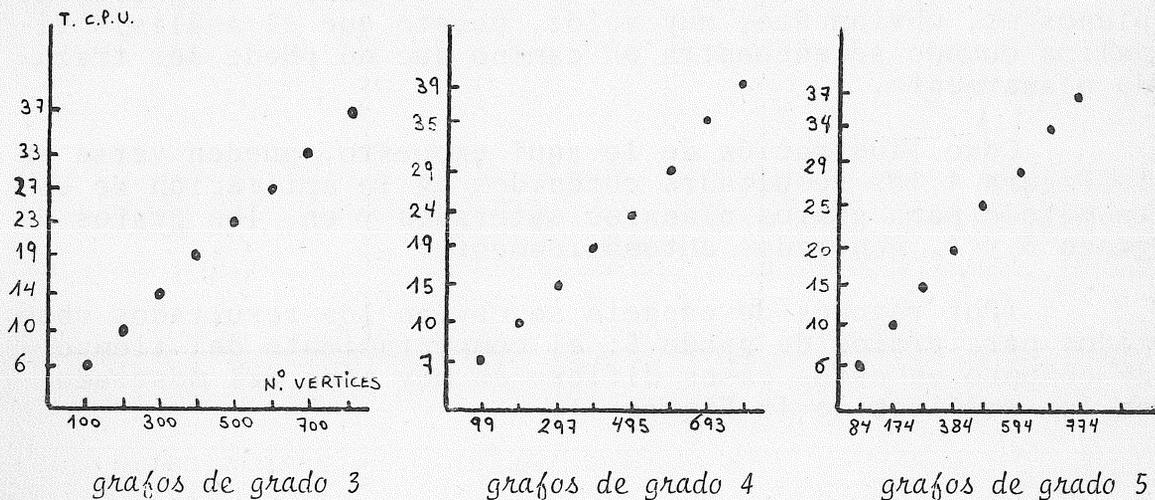
FIGURA 2: Test de Conexión (Grafos aleatorios)



b) BICONEXION: Respecto a este problema es importante destacar la gran diferencia existente entre el análisis de la biconexión de un grafo utilizando la definición de grafo biconexo y el criterio enunciado anteriormente.

Sea G un grafo con N vértices; en el primero de los casos es necesario analizar la conexión de N grafos con N-1 vértices cada uno, lo que implica, un tiempo proporcional a  $N^2$  mientras que el algoritmo presentado aquí presenta un grado de complejidad lineal en el número de vértices, lo que se puede apreciar en los gráficos de la Figura 3.

FIGURA 3: Test de Biconexión



Estas rectas corresponden a las ecuaciones  $T = 0.043 N + 1,5$  ,  
 $T = 0.045 N + 1,75$  ,  $T = 0.047 N + 1,6$  .

Por último y mediante las rutinas de generación de grafos al azar se obtuvo la siguiente ecuación de regresión :  
 $T = 0.037 N + 0.004 M + 1.68$  que ajusta estadísticamente la recta que relaciona el tiempo de cómputo y el número de vértices y aristas.

c) PLANARIDAD: Por ser éste el más complejo de los problemas presentados, es conveniente antes de detallar los resultados obtenidos, esbozar un breve análisis acerca del grado de complejidad de otros algoritmos para decidir la planaridad de un grafo.

El primer criterio conocido fue el encontrado por Kuratowsky (Ref. 5 ); este método tiene un grado de dificultad del orden de  $N^6$ . Más recientemente se conocen los algoritmos de Demoucron, Malgrange y Pertuiset, con un grado de complejidad del orden de  $N^3$  y de Tarjan (Ref.6) con una relación tiempo de cómputo  $T = N \log N$ .

El método aquí presentado, que tiene un comportamiento estadístico cercano a la linealidad, ha resultado sumamente eficiente, más aún teniendo en cuenta que el lenguaje PL1 (en el cual se realizó la implementación) no dispone de estructuras adecuadas para el manejo de grafos y árboles.

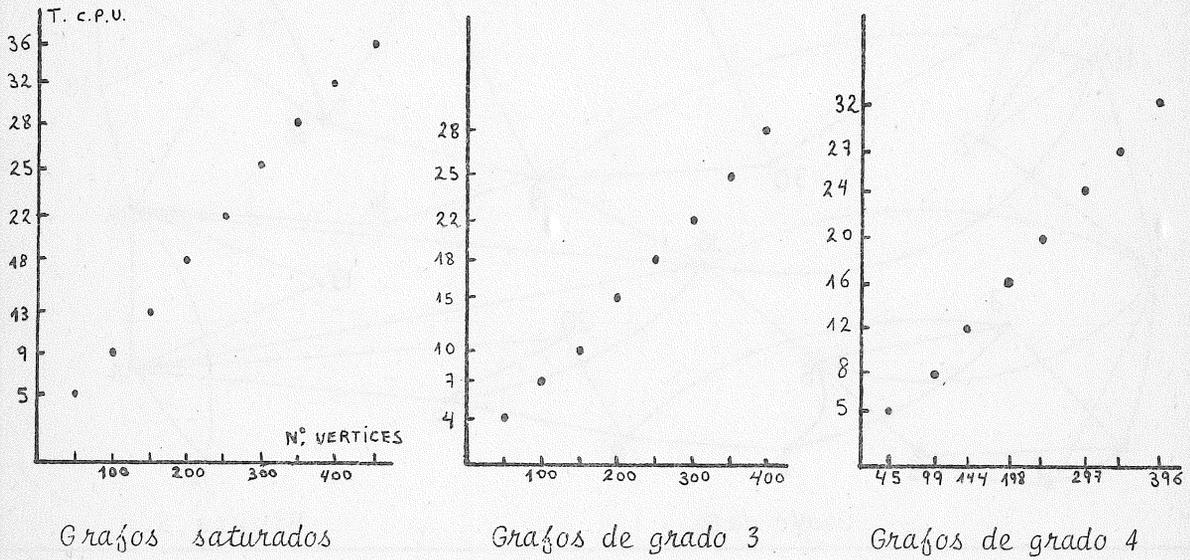
El programa ha sido probado con los grafos que representan el peor caso para este algoritmo, los grafos saturados, puesto que el número de aristas en este caso es máximo. Los resultados obtenidos son excelentes; se analizó un grafo de 500 vértices en 39 s de C.P.U.

Es importante destacar que la detección de grafos no planos es, obviamente, muy veloz, puesto que el análisis finaliza cuando se encuentra un camino que no puede ser trazado planarmente.

Como ilustración de lo aquí expuesto, pueden verse en la Figura 4 los resultados obtenidos en la aplicación de este método para grafos planares saturados y en los grafos de grado 3 y 4, generados automáticamente.

(Por razones de espacio se obvian los resultados obtenidos para grafos de grado 5; el comportamiento del tiempo de cómputo en estos casos difiere en muy poco del mostrado en los gráficos de la Figura 4).

FIGURA 4: Test de Planaridad

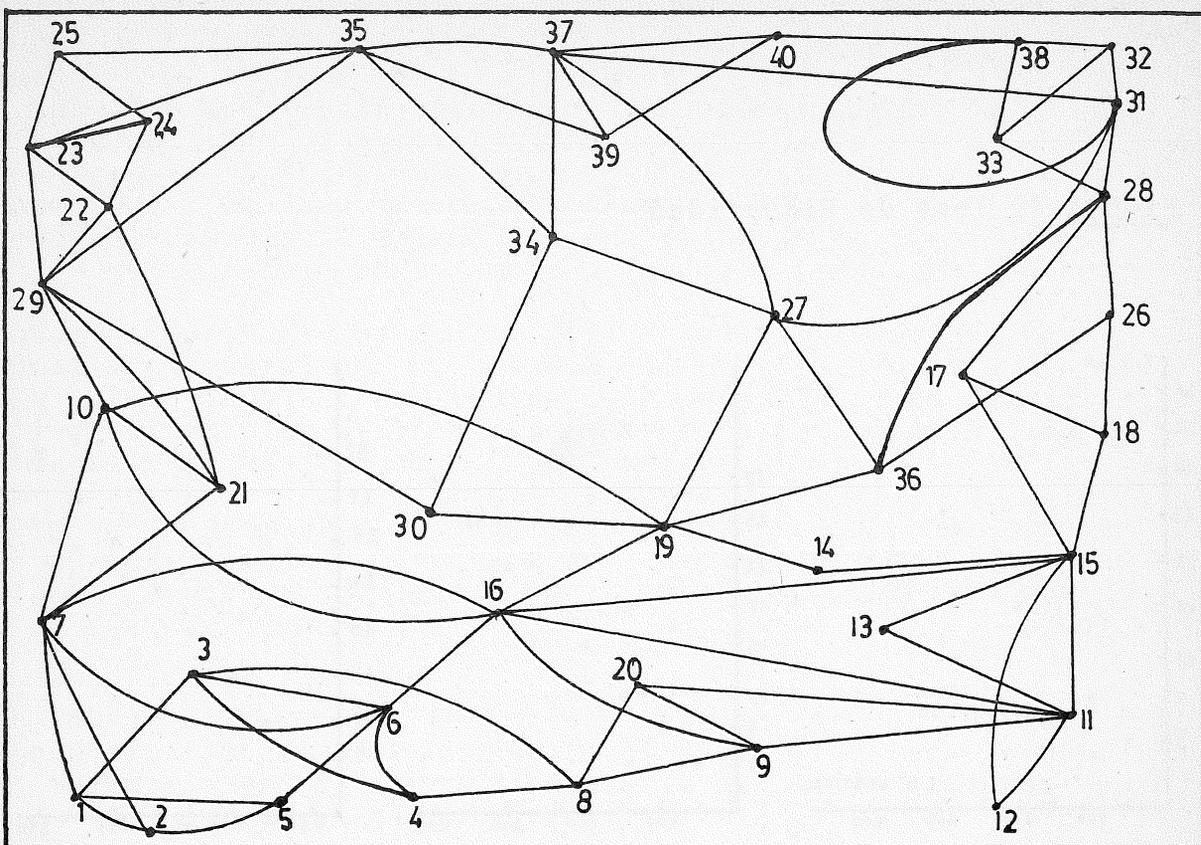


Puede observarse el comportamiento lineal del tiempo de cómputo para el análisis de estos grafos fuertemente conectados (más precisamente sin vértices de grado 2).

Se está trabajando actualmente en la elaboración de rutinas de generación de grafos biconexos (el algoritmo es aplicable sólo a este tipo de grafos) aleatorios para obtener una ecuación de regresión que ajuste correctamente el comportamiento del tiempo de cómputo de este método.

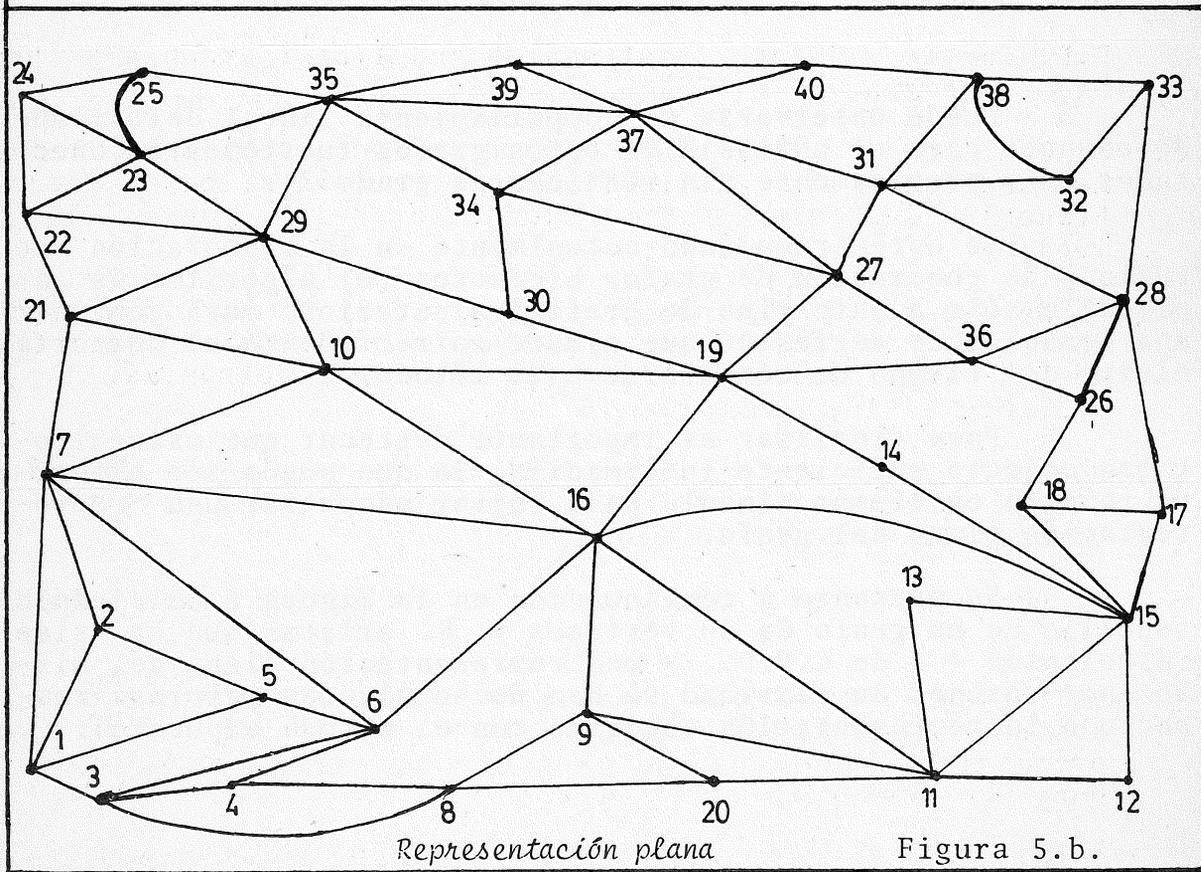
Para finalizar es importante destacar que este algoritmo colecta suficiente información, la que puede ser obtenida también en tiempo lineal, para lograr encontrar una representación plana del grafo.

Se presenta a continuación en la Figura 5 un ejemplo sencillo de un grafo de 40 vértices y 82 aristas que ha sido resuelto en 4 s de C.P.U., y una representación plana del mismo (por razones de claridad se han hecho algunas reformas respecto a la representación obtenida con el método expuesto).



Grafo dato

Figura 5.a.



Representación plana

Figura 5.b.

## CONCLUSIONES:

Se ha presentado un conjunto de algoritmos de manejo de grafos con un alto grado de eficiencia, y se han detallado los resultados obtenidos en la aplicación de estos métodos sobre un conjunto de grafos generados automáticamente, quedando perfectamente demostrada la bondad de estos algoritmos respecto a otros que buscan resolver el mismo tipo de problema.

Convendría antes de finalizar este trabajo, hacer un breve comentario acerca de los tópicos que se están analizando actualmente, estrechamente relacionados con los aquí presentados. El primero de ellos es obtener a partir de la información colectada por el algoritmo de planaridad una representación planar - automática del grafo con métodos interactivos, y el segundo de ellos es obtener un algoritmo eficiente para testear si un grafo es planar-exterior o sea si es posible una representación plana del mismo de manera que todos sus vértices queden en la periferia.

Por último, es importante reiterar que el esfuerzo realizado en obtener algoritmos de este tipo con un alto grado de eficiencia está ampliamente justificado en la importancia que pueden tener los mismo en el diseño interactivo de circuitos electrónicos impresos, entre otras aplicaciones.

## REFERENCIAS:

- 1.- R.E. Tarjan - "Depth first search and linear graph algorithms"; SIAM J. comput, 1 (1972) - pag. 146-159.
- 2.- G. Demoucron, Y. Malgrange y R. Pertuiset - "Graphes planaires Reconnaissance et construction des representations". Revue Francaise de Recherche Operationnelle. Vol 8, pag. 33-47. Año 1964.
- 3.- A. De Giusti, C. Gutzner, G. Rossi y J. Diaz - "Planarización de grafos por computadora. Aplicación a circuitos electrónicos". - Publicación del Plan Nacional de Electrónica, Año I N°3, pag. 4-10, Noviembre 1979.
- 4.- Hopcroft y Tarjan, "Efficient planarity testing" - Journal of the association for-computing machinery, Vol. 21, N°4 October 1974 - pag. 549-568.
- 5.- C. Berge - "Theorie des graphes et ses applications". DUNOD 1963.
- 6.- R. Tarjan - "Planarity testing in  $V \log V$  seteps: Extended abstract" - Proc IFIP cong. 1971: Foundation of information processing, Ljubljana, Yugoslavia, Aug. 1971, North Holland Pub. Co. Amsterdam - Pag. 18-22.
- 7.- Meza, Rossi y Diaz: "Generación de grafos por computadora Informe Técnico CeTAD 101/80.

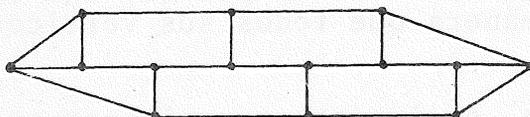
GENERACION DE GRAFOS POR COMPUTADORA:

Es sumamente importante disponer de una batería de programas de generación de grafos por computadora no sólo para un testeo intensivo de la corrección de los algoritmos implementados, sino también para realizar un análisis amplio y profundo de la eficiencia de dichos algoritmos.

Para la elaboración del análisis aquí presentado se confeccionaron rutinas de generación de grafos de los siguientes tipos:

- a) Grafos regulares de grado 3: Se generan grafos triconexos - planares que responden al esquema de la Figura 6.

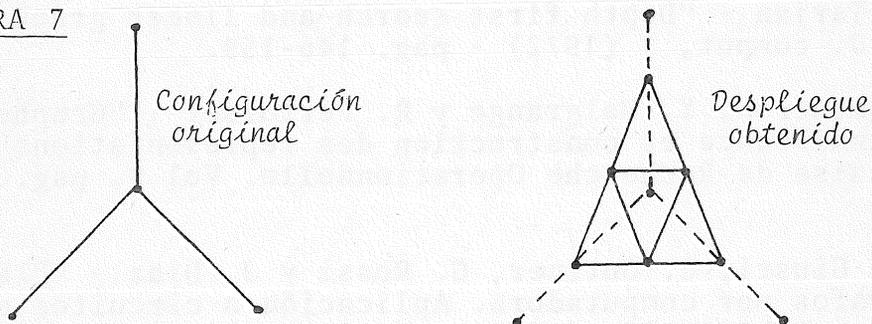
FIGURA 6



*Grafo regular de grado 3 con 14 vértices*

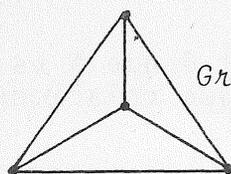
- b) Grafos regulares de grado 4: Para producir estos grafos planos se utilizó una técnica de despliegue que consiste en reemplazar cada vértice de un grafo generado por el método anterior, por seis vértices conectados entre sí (Figura 7).

FIGURA 7



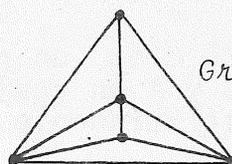
- c) Grafos regulares de grado 5: En este caso se reemplazó cada vértice de un grafo generado por el método anterior por una representación compleja obteniéndose un grafo planar ( Para mayores detalles ver Ref. 7 ).
- d) Grafos planares saturados: (Un grafo planar se dice saturado si no es posible agregarle ninguna arista sin afectar la planaridad). En este caso se utilizó para generar los grafos planos saturados el siguiente método: Dado como base el grafo de 4 vértices de la Figura 8, se obtiene un grafo de 5 vértices eligiendo al azar una de las 3 caras triangulares, colocando allí un vértice y uniéndolo con los 3 vértices que limitan la cara (Figura 9). En general dado un grafo planar saturado de  $I$  vértices de este tipo, se obtiene un grafo de

$I + 1$  vértices eligiendo al azar una de las  $2I - 5$  caras -- triangulares internas y aplicando el proceso explicado anteriormente.



*Grafo saturado  
de 4 vértices*

FIGURA 8



*Grafo saturado  
de 5 vértices*

FIGURA 9

- e) Grafos aleatorios de tipo 1: Se construyeron este tipo de grafos utilizando rutinas de generación de números al azar, decidiéndose en forma aleatoria cuántas y qué aristas formarían el grafo.
- f) Grafos aleatorios tipo 2: En este caso se generaron grafos completos (cada vértice está conectado con todos los demás) y se eliminaron un número aleatorio de aristas elegidas también mediante el proceso explicado en e). Es importante aclarar que este tipo de grafos aleatorios tienen una fuerte tendencia (debido al método utilizado) a resultar conexos y en menor medida biconexos, lo que permitió hacer un análisis de la eficiencia de los métodos de testeo en casos que la propiedad buscada se cumpliera y en casos en que no fuera así.

Un análisis completo de las propiedades de todos estos grafos generados automáticamente se presenta en la Referencia 7.

## APENDICE 2

### DEFINICIONES MATEMATICAS Y FORMAS DE REPRESENTAR UN GRAFO POR COMPUTADORA:

Un grafo  $G$  es un par ordenado  $(X, A)$  donde  $X$  es un conjunto de vértices y  $A$  un conjunto de aristas que relacionan estos vértices arbitrariamente.

Si cada arista está formada por un par  $(u, v)$  no ordenado de vértices, el grafo se dice no orientado, en caso contrario el grafo se dice orientado. Se dice que 2 vértices son adyacentes si existe una arista que los une.

Se denomina grado de un vértice al número de aristas que inciden en él; un grafo se dice regular si todos los vértices tienen el mismo grado.

Sean  $u$  y  $w$  dos vértices de  $G$ ; un camino de  $u$  a  $v$  es una sucesión de aristas  $(l_i)_{i=1, n}$  tal que  $l_i = (V_i, V_{i+1})$  con  $V_i = u$  y  $V_n = w$ .

Un subgrafo  $G'$ , de  $G$  es un grafo tal que  $G' = (X', A')$  con  $X' \subset X$  y  $A' \subset A$ ; Si  $G$  es orientado un subgrafo  $T$  se denomina árbol si  $T$  está formado por un vértice  $V$  (llamado raíz) en el que no incide ninguna arista y en todo vértice  $u \neq v$  incide una y sólo una arista. Si todo vértice de  $G$  es vértice de  $T$  el árbol se denomina generador; toda arista de  $G$  que no pertenece a  $T$  se denomina fronda.

Finalmente un grafo  $G$  se puede representar por una matriz  $M$  tal que  $m_{ij} = 1$  si y sólo si el vértice  $i$  está conectado al vértice  $j$  y en caso contrario  $m_{ij} = 0$ ;  $0$  por un conjunto de listas, donde la lista  $i$  contiene todos los vértices adyacentes al vértice  $i$ .