

ESTABILIZACION Y RESOLUCION DE ALGUNOS PROBLEMAS
COMBINATORIOS EN GRAFOS SIMETRICOS

X. BERENGUER L.H. HARPER

Cuando en grafo de un cierto problema combinatorio presenta simetrías puede ser factible reducir su complejidad al aplicar sobre aquél la denominada -- "estabilización", que en esencia persigue la disminución del tamaño de los conjuntos a enumerar. En la primera parte de este artículo se resume la teoría de la estabilización en grafos simétricos y se presenta una algorítmica diseñada para la obtención de los conjuntos estables. En la segunda se consideran los problemas "suma de arcos" e "isoperimétrico", se formula su resolución sobre los conjuntos estables, y se presentan los programas desarrollados al efecto. De esta manera queda disponible en la práctica la teoría de estabilización y se muestran sus posibilidades al abordar problemas complejos.

0. INTRODUCCION

Cuando un problema combinatorio ha recibido el anatema de NP-Completo conviene abordar -- su resolución en base a algoritmos generales aproximativos o heurísticos, o bien observar los casos particulares que en la práctica deben resolverse y sacar partido de las propiedades de su estructura.

En la realidad es frecuente manejar grafos -- con cierto grado de simetría. Harper, en /3/ sentó las bases de la denominada "estabilización" que al considerar las simetrías de un grafo puede reducir notablemente el tamaño -- de una enumeración. La idea básica es que, -- por acción de estas simetrías todos los subconjuntos a enumerar "colapsan" sobre unos -- subconjuntos particulares denominados "estables", invariantes por aplicación de las simetrías. Si la función a optimizar sobre -- aquellos subconjuntos queda preservada sobre los subconjuntos estables correspondientes, entonces sólo será preciso considerar a éstos en la enumeración. Cuanto mayor es el número de simetrías del grafo, menor es el número de conjuntos estables y por lo tanto, -- la enumeración de éstos es menos costosa.

Este trabajo se compone de dos partes. En -- primer lugar se presentan las ideas esenciales de la teoría de la estabilización así como un sistema informático el cual, a partir

de los datos de un grafo, genera sus conjuntos estables. Con esta parte, queda utilizable en la práctica una herramienta eficiente para la obtención de estos conjuntos estables.

En la segunda parte se ilustra una aplicación de los conjuntos estables en la resolución -- de dos problemas NP-Completo: el problema -- de la "suma de arcos" y el problema "isoperimétrico". Se exponen las ideas para su resolución, los algoritmos diseñados a tal efecto, así como los resultados para diversos casos prácticos sobre los que se ha aplicado -- todo el aparato informático, desde la enumeración de los conjuntos estables hasta la resolución de los dos problemas. Finalmente, -- un apéndice ilustra, mediante un ejemplo sencillo, todo lo expuesto en este trabajo.

1. ESTABILIZACION

1.1 Simetrías y estabilización

Dado un grafo euclídeo $G=(V,E)$ en E^d se dice que R de G es una simetría estabilizadora -- cuando actúa como un automorfismo lineal de G , transformando vértices en vértices y aristas en aristas, de forma que si $u,v \in V$ se hallan en diferentes lados del plano de R y están conectados por una arista, entonces -- $R(u)=v$ (y también $R(v)=u$).

- X. Berenguer del CCUPB. Av. Dr. Gregorio Marañón, s/n. Barcelona 28. Aquest treball el va realitzar en visita postdoctoral al Dep. de Matemàtiques de la Univ. de California a Riverside, sota el patrocini del Ministeri d'Educació i la Universitat Politècnica de Barcelona.
- L.H. Harper del Dept. de Matemàtiques de la Univ. de California a Riverside. Riverside CA 92521, USA.
- Article rebut l'Agost del 1979.

Dado un punto $p \in E^d$ tal que $R(p) \neq p$, se define la estabilización u operador estabilizador $\text{Stab}_{R,p}(S)$ de un conjunto $S \subseteq V$ para la simetría R y el punto p de la siguiente manera:

Sea $S = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, con

$$\Sigma_0 = \{u \in S, R(u) \in S\}$$

$$\Sigma_1 = \{u \in S, R(u) \notin S, \|u-p\| < \|R(u)-p\|\}$$

$$\Sigma_2 = \{u \in S, R(u) \notin S, \|u-p\| > \|R(u)-p\|\}$$

Pues bien, $\text{Stab}_{R,p}(S) = \Sigma_0 \cup \Sigma_1 \cup R(\Sigma_2)$ (/3/)

Sea un conjunto $R_0, R_1 \dots R_{k-1}$ de simetrías -estabilizadoras de un grafo G . Definida la -secuencia $T_0, T_1, T_2 \dots$ con $T_0 = S$ y $T_{j+1} = \text{Stab}_{R_j(\text{mod } k), p}(T_j)$ en /2/, se demuestra que para todo $S \subseteq V$, la secuencia $T_0, T_1, T_2 \dots$ se hace constante a partir de un cierto instante, es decir, que para algún j , $T_{j+1} = T_j$.

Se trata de ver ahora cómo obtener todos los conjuntos distintos T_j ("estables").

1.2 El retículo de conjuntos estables

Dado un conjunto de simetrías estabilizadoras $R_0, R_1 \dots R_{k-1}$ se define la relación binaria siguiente:

Si $u, v \in V$ se tendrá $u < v$ si $R_i(u) = v$ y $\|u-p\| < \|v-p\|$.

Sea

$$S(G; R_0, R_1 \dots R_{k-1}; p)$$

el orden parcial inducido por esta relación de la siguiente manera:

$$P_1 = \{(u, v) : u < v \text{ para alguna } R_i\}$$

$$P_2 = P_1 \circ P_1$$

$$P_3 = P_2 \circ P_1$$

·
·

$$P_{i+1} = P_i \circ P_1 \quad (i > 1)$$

en donde \circ denota composición de relaciones

$S = \bigcup P_i$ y se comprueba que es un orden parcial

(cierre transitivo de P_1) que satisface las propiedades reflexiva, antisimétrica y transitiva.

Definiendo $H = P_1 - \bigcup_{i > 1} P_i$ se tiene que $H(G; R_0, R_1 \dots R_{k-1}; p)$ es el mínimo conjunto de P_1 cuyo cierre transitivo sigue siendo S . El diagrama de Hasse de S es el grafo orientado -- con vértices V y arcos H .

En un conjunto parcialmente ordenado, un conjunto inferior L es un conjunto tal que si $x \in L$ y $y \leq x$ entonces debe ser $y \in L$.

Harper en /3/ demuestra el siguiente teorema:

Un conjunto S es estable con respecto a $R_0, R_1 \dots R_{k-1}$ y p si y sólo si es un conjunto inferior en $S(G; R_0, R_1 \dots R_{k-1}; p)$.

Por lo tanto, para la obtención de los conjuntos estables bastará con enumerar los conjuntos inferiores de H .

1.3 Sistema informático para la obtención de los conjuntos estables

El sistema desarrollado para la obtención de los conjuntos estables de un grafo bajo determinadas simetrías estabilizadoras se compone de tres procesos: REFLEX, HASSE y LOWSET. Con REFLEX se genera la relación P_1 inducida por las simetrías; con HASSE se obtiene el diagrama de Hasse de dicho orden parcial y con LOWSET se enumeran sus conjuntos inferiores.

1.3.1 REFLEX

Introducidos los vértices y las aristas del grafo, se trata de generar las simetrías estabilizadoras que alberga (si no son entra--das directamente) y a partir de ellas establecer el conjunto $P_1 = \{(u, v) : u, v \in V \mid u < v\}$ respecto a un punto dado p que no se halla sobre ningún (hiper)plano de simetría.

Si e es un vector unitario perpendicular al (hiper)plano de simetría, es fácil ver que un vértice x se reflejará en otro w por acción de dicha simetría e cuando

$$w = -(xe)e + (x - (xe)e) = x - 2(xe)e$$

Si, dentro de una tolerancia preestablecida, se cumple esta igualdad basta entonces reconocer qué vértice está más cerca de p para incluirlos finalmente en P_1 .

La comprobación descrita se realiza entre todos los pares posibles de vértices y para todas las simetrías.

1.3.2 HASSE

Para la obtención del diagrama de Hasse se trata de eliminar de P_1 todas aquellas relaciones que puedan obtenerse por transitividad, o sea por la aplicación sucesiva de dos, tres, cuatro, etc... simetrías.

Se debe calcular pues $P_2, P_3 \dots$ con $P_{j+1} = P_j \circ P_1$.

Un primer procedimiento iterativo para ello es:

$$PP_1 = P_1$$

$$H_1 = P_1$$

$$PP_2 = P_1 \circ P_1 = PP_1 \circ H_1$$

$$H_2 = H_1 - PP_2$$

$$PP_3 = PP_2 \circ H_2$$

$$H_3 = H_2 - PP_3$$

·
·

en general

$$PP_{j+1} = PP_j \circ H_j$$

$$H_{j+1} = H_j - PP_{j+1}$$

terminándose cuando $PP_{j+1} = \emptyset$, con lo que $H = H_1$.

Sin embargo este procedimiento exige en cada paso j , el cálculo de todas las composiciones posibles PP_j para todos los vértices, lo que es costoso en memoria. Para su simplificación basta hacer toda la iteración anterior vértice a vértice, sacando ventaja progresiva de la reducción de P_1 obtenida para los vértices precedentes:

Si $n = |V|$ e identificando los vértices según

una numeración correlativa $1, 2, \dots, n$:

Para $i=1$ hasta n

Sea $P_1^i = \{v \in V \mid i < v\}$

(es evidente que $P_1 = \bigcup_{i=1}^n P_1^i$)

Hágase

$$PP_1^i = P_1^i$$

$$H_1^i = P_1^i$$

En general

$$PP_{j+1}^i = PP_j^i \circ P_1$$

$$H_{j+1}^i = H_j^i - PP_{j+1}^i$$

Cuando $PP_{j+1}^i = \emptyset$ se hace $P_1^i = H_j^i$.

De esta manera en P_1 va almacenándose progresivamente el diagrama de Hasse. En la operación $PP_{j+1}^i = PP_j^i \circ P_1$, en P_1 se tiene propiamente el orden H para todo vértice v de índice inferior a i , y el orden original P_1 para el resto de vértices. Terminada la iteración en P_1 se tiene pues H .

1.3.3 LOWSET

Los conjuntos inferiores, o sea los conjuntos estables, se obtienen por enumeración progresiva.

Para todo vértice $v \in V$, sea $d(v)$ el conjunto de vértices relacionados con v en el diagrama de Hasse:

$$d(v) = \{w \in V \mid v < w \text{ con } (v, w) \in H\}$$

Se construye también la relación inversa:

$$D(v) = \{w \in V \mid v \in d(w)\}$$

Se define también $d(0)$, en donde 0 es un vértice imaginario que se hallaría por debajo de cualquier otro vértice en la ordenación H

$$d(0) = \{w \in V \mid \text{si } z \in V \text{ y } z < w \Rightarrow z = w\}$$

Para los vértices $w \in d(0)$ es obvio que $D(w) = \{0\}$.

La enumeración se desarrolla en forma de -- "búsqueda prioritaria en profundidad". Aunque en el árbol correspondiente podría hablar se de niveles y de nodos dentro de cada nivel, en la práctica su desarrollo puede contemplarse como puramente lineal y cada nodo se identifica con un solo índice. Para el objetivo aquí presente se verá que esta técnica supone ventajas considerables en lo que a memoria se refiere.

Cada nodo i , contiene un conjunto inferior - generado S_i (que se almacenará en el archivo de salida). Asimismo, cada nodo lleva asociado un subconjunto N_i de "vértices capturados". Si un nodo i fué generado por un nodo anterior j , entonces $N_i = S_i - S_j$.

ALGORITMO:

Paso 0)

Inicialización y generación del nudo inicial:

$i = 1$
 $S_1 = \emptyset$
 $N_1 = \emptyset$
 Ir a 1)

Paso 1)

Definición de subconjuntos válidos para generar nuevos nodos a partir de un nodo i :

Se construye el conjunto C_i de todos los vértices que pueden entrar en nuevos nodos sucesores:

$$C_i = \{w \in V \mid \forall v \in N_i, w \in d(v)\}$$

Los subconjuntos posibles de C_i que interesan vienen dados por todas las particiones - de C_i , con excepción del conjunto vacío:

$$P_i = P(C_i) - \emptyset$$

Sobre estos subconjuntos se forma el conjunto V_i de los subconjuntos válidos para generar nuevos nodos:

$$V_i = P_i - \{\Pi \in P_i \mid \exists z \in \Pi \text{ tal que } D(z) \not\subseteq S_i \cup \Pi\}$$

Sea $|V_i| = k$. Si $k=0$ ir a 3). En caso contrario ir a 2).

Paso 2)

Generación de un nuevo nodo a partir de un - nodo i :

Tomar uno de los k subconjuntos válidos $\Pi \in V_i$ y eliminarlo de V_i . El nuevo nodo generado - se caracteriza por

$$S_{i+1} = S_i \cup \Pi$$

$$N_{i+1} = \Pi$$

Hacer $i := i+1$ e ir a 1)

Paso 3)

Retroceso al nodo previo.

Hacer $i := i-1$. Si $i=0$ es final del proceso. Si todavía quedan subconjuntos válidos en V_i con los que generar nuevos nodos, ir a 2). - En caso contrario volver a 3).

Este algoritmo genera, en efecto, todos los conjuntos inferiores de S . En efecto, dado - un cierto nodo i en C_i se tienen todos los - vértices enlazados en el diagrama de Hasse - con los vértices capturados en i :

$$C_i = \{w \in V \mid \forall v \in N_i, v < w \text{ con } (v, w) \in H\}$$

Al construir P_i se forman todas las agrupaciones posibles de los vértices de C_i , de -- las que se eliminan las no válidas, o sea -- aquellos subconjuntos $\Pi \in P_i$ en los que se incluye algún vértice $z \in \Pi$ para el que hay algún vértice u tal que $u < z$ y que no está incluido en $S_i \cup \Pi$, lo que es en definitiva la exigencia de todo conjunto inferior.

Por último, en la medida en que todos los S_i producidos son distintos, queda garantizada la no repetición.

Desde el punto de vista informático sólo se precisa memoria para tantos nodos como niveles tiene el árbol. De cada nodo i se guarda básicamente N_i y V_i . Un nodo agotado ($V_i = \emptyset$ - bien como resultado del paso 1), bien porque se han explorado todos los V_i que tenía) se elimina de la memoria. Además el nodo "padre" de un nodo siempre es el de índice justamente inferior, lo que facilita notablemente la programación (paso 3). Por último, se consigue otro ahorro notable de memoria al organizar V_i en forma de lista para todos los nodos, y al representar los subconjuntos Π de V_i como enteros potencias de 2, en los que -

su representación en bits se corresponde directamente con la inclusión (o no) en Π de un vértice individual de C_i .

El programa LOWSET incluye además una subrutina general FUNC que se llama en tiempo del paso 2) de la iteración, para calcular una función asociada al conjunto inferior generado. El archivo de salida producido por LOWSET contiene todos los conjuntos inferiores, su cardinalidad y valor respectivos de la función FUNC.

2. RESOLUCION DE LOS PROBLEMAS SUMA DE ARCOS E ISOPERIMETRICO

2.1 Los problemas "suma de arcos" e "isoperimétrico"

Sea un grafo no orientado $G(V,E)$. Una numeración en G es una función

$$\psi : V \rightarrow \{1, 2, \dots, |V|\}$$

Para toda numeración ψ de G y una arista $e \in E$, se define la función

$$\Delta_e(\psi) = |\psi(u) - \psi(v)| \quad (u,v) \in e$$

El problema "suma de arcos" consiste en

$$\text{MIN}_{e \in E} \sum \Delta_e(\psi)$$

sobre todas las numeraciones posibles ψ_i de G .

Dado un conjunto $S \subseteq V$, el operador θ se define así:

$$\theta(S) = |\{e \in E \mid (u,v) \in e, u \in S \text{ y } v \notin S\}|$$

Dado un entero l , $0 \leq l \leq |V|$, el problema isoperimétrico consiste en

$$\text{MIN}_S \theta(S), \quad |S|=l$$

Los problemas suma de arcos e isoperimétrico se hallan estrechamente relacionados. Así, para todas las numeraciones ψ de G se tiene que

$$\sum_{e \in E} \Delta_e(\psi) = \sum_{i=0}^{|V|} \theta(S_i(\psi)) \quad (\text{véase /4/})$$

y si se puede hallar una familia de conjuntos

$$\emptyset = S_0 \subset S_1 \subset S_2 \subset \dots \subset S_{|V|} = V$$

con $|S_1|=1$ y cada S_1 solución del problema isoperimétrico, entonces la numeración definida por

$$\psi(v) = 1 \quad \text{si } v \in S_1 - S_{1-1}, \quad 1 \leq |v|$$

es una solución del problema suma de arcos.

Para grafos arbitrarios, ambos problemas son NP-Completos /1/. Por la "fuerza bruta" una manera de resolver el problema suma de arcos sería construir las $n!$ numeraciones posibles ($n=|V|$) y tomar aquella que minimiza $\sum \Delta_e$. -- Por lo tanto la complejidad de este procedimiento sería $\theta(n^n)$.

2.2 Morfismo sobre el problema del camino mínimo

Una primera reducción de la complejidad de este problema "suma de arcos" se consigue estableciendo un morfismo entre este problema y el problema del camino mínimo entre dos vértices de una red (véase /5/).

Sea $G(V,E)$. Se construye la red $N(V',E',s,t)$ en donde $V'=P(V)$ (o sea todas las $2^{|V|}$ particiones posibles de V). A cada vértice $A \in V'$ se le asigna un peso que sea $\theta(A)$. Los arcos E' de la red se construyen así: entre dos vértices $A, B \in V'$ se establece un arco $A+B$ cuando $A \subseteq B$ y $|A|=|B|-1$, o sea A, B son conjuntos contenidos y de cardinalidad consecutiva. Por último, $s=\emptyset$ y $t=V$ (s : origen de todos los caminos y t final de los mismos).

La cadena de subconjuntos

$$\emptyset = S_0 \subset S_1 \subset S_2 \subset \dots \subset S_{|V|} = V$$

que antes se veía podía resolver el problema, es precisamente un camino que va desde $s=S_0$ a $t=S_{|V|}$ en la red N . Por lo tanto un camino mínimo, entendido aquí como que minimiza la suma de los θ de los vértices V' que atraviesa, definirá una numeración que será una solución óptima del problema suma de arcos.

Como que $|V|=2^n$ y el cálculo de un camino mínimo tiene una complejidad aproximada $\theta(n^2)$ con n en este caso el tamaño del conjunto de vértices del grafo, resulta que la complejidad del problema suma de arcos resuelto mediante este morfismo es $\theta(2^{2n})$.

2.3 Aplicación de la estabilización

La estabilización nos va a permitir, a su vez, reducir la dimensión de la red N .

En /2/ se demuestra que

$$\forall S \subseteq V, |\text{Stab}_{R,p}(S)| = |S|$$

y que,

$$\theta(\text{Stab}_{R,p}(S)) \leq \theta(S)$$

Así que la optimización del operador θ sobre todos los conjuntos posibles de V queda preservada si se consideran solamente los conjuntos estables. Es inmediato advertir que mediante la estabilización aparece otro morfismo sobre el problema del camino mínimo: desde la red N de todos los conjuntos posibles nos trasladamos a una red N cuyos vértices sean sólo los conjuntos estables (arcos, s y t de construcción idéntica que en N). El problema del camino mínimo se resuelve sobre N y queda así resuelto el problema de suma de arcos.

2.4 Sistema informático

El sistema desarrollado incluye por una parte la implementación de la función FUNC (del programa LOWSET), y un proceso SHPATH para el cálculo del camino mínimo en N .

2.4.1 Función FUNC asociada a LOWSET

Para un nodo i , el cálculo del valor de θ_i puede formularse en función del nodo "padre" k , según el algoritmo siguiente:

$$\text{Hacer } W = S_i - N_i = S_k$$

$$\text{Sea } N_i = \{v_1, v_2 \dots v_m\}$$

$$\text{Hacer } t = 0$$

Desde $j=1$ hasta m , hacer:

$$\text{Sea } A = \{w \in V \mid (v_j, w) \in E\}$$

A cada $w \in A$ asígnese

$$f_w = \begin{cases} 1 & \text{si } w \in W \\ 0 & \text{si } w \notin W \end{cases}$$

$$t := t + |A| - 2 \sum_{w \in A} f_w$$

$$W = S_i \cup \{v_i\}$$

Al final del bucle, t contiene el incremento marginal de θ , y entonces

$$\theta_i = \theta_k + t$$

2.4.2 SHPATH

El archivo de conjuntos estables producido por LOWSET se clasifica en orden creciente de cardinalidad y valor de θ . En tiempo de su lectura en SHPATH se extraen los conjuntos de menor θ para cada cardinalidad y queda así resuelto el problema isoperimétrico.

Para resolver el problema suma de arcos debe calcularse el camino más corto entre s y t . Para ello SHPATH utiliza el algoritmo de Dijkstra (véase por ejemplo /6/). La función a minimizar es la suma de los θ de los vértices que se atraviesan. A efectos de dicho algoritmo, basta asociar a cada arco de la red N el valor de θ del vértice al que el arco es incidente.

La estructura de N permite obtener diversas ventajas en lo que a memoria y cálculo se refiere. Nótese que sólo deben mantenerse en memoria los nodos adyacentes al último nodo que se ha "etiquetado definitivamente" (según la terminología habitual del algoritmo de Dijkstra). La disposición ordenada, según cardinalidad creciente, de los conjuntos estables que entra en SHPATH permite de inmediato la definición de los arcos y el cálculo de la función a optimizar: sólo se precisan leer (si no están en memoria por algún paso previo) los nodos (conjuntos estables) de cardinalidad una unidad superior a la del nodo etiquetado definitivamente, probar si -

Tabla 1
Casos tratados en la experiencia

figura	espacio	n	número de simetrías estabilizadoras	número de conjuntos estables	Σ mínimos θ	valor de θ acumulado para el camino mínimo
1) De Brujin	E^2	16	2	2916	74	76
2) Icosaedro	E^3	12	15	14	97	97
3) Dodecaedro	E^3	20	15	33	104	104
4) 24-celdas	E^4	24	12 ¹	111	556	556
5) 120 vértices	E^4	120	60	883	12616	12620
6) 600 vértices	E^4	600	60	$2^{20} \sim 2^{25}$ ²	---	---

Notas

¹Faltan simetrías por considerar. En los demás casos se han tratado todas las posibles.

²Estimación elaborada a partir de las dimensiones del diagrama de Hasse. Cálculo desechado -- por su excesivo coste.

contienen los vértices de éste, en caso afirmativo asignarles el nuevo valor de la función (caso de verse disminuida), y asignarles el nuevo nodo previo en el camino trazado.

Un nodo etiquetado definitivamente se graba sin más en un archivo de salida de SHPATH - (junto al valor de θ acumulado, su cardinalidad y el nodo previo al que se asocia en el camino) y se elimina de toda consideración posterior. La memoria ocupada por este nodo queda libre en la lista construida para nodos ulteriores.

Cuando se alcanza el nodo $t=v$ el proceso termina, resultando un archivo de salida con todos los conjuntos estables procesados en el cálculo del camino mínimo. Este archivo se clasifica en orden decreciente de cardinalidad y entra en un pequeño programa final que resigue el camino óptimo: se toma el nodo $t=v$ (primer registro del archivo leído) y el nodo previo que lleva asociado, a continuación se lee este nodo y el nodo previo al que se asocia, y así sucesivamente. El listado de dicha secuencia, en forma de vértice capturado para cada cardinalidad expone la numeración óptima que se perseguía.

3. RESULTADOS Y CONCLUSIONES

Los programas descritos² se han aplicado a la enumeración de los conjuntos estables y a la resolución de los problemas suma de arcos e isoperimétrico en diversos grafos simétricos

en E^2 , E^3 y E^4 . La tabla 1 expone los resultados más significativos de la experiencia. Obsérvese que para el sólido de 600 vértices en E^4 hubo de abandonarse la enumeración de conjuntos estables debido a su elevado número, a pesar de la notable reducción obtenida mediante la estabilización. Sigue en pie la resolución de aquellos problemas para este politopo regular. A pesar de este fracaso se han resuelto por primera vez aquellos problemas para el politopo regular de 120 vértices en E^4 , lo que habida cuenta de la complejidad inherente de este caso constituye un ejemplo de la aplicabilidad de la herramienta.

No es sencillo hacer previsiones para otros casos. Finalmente el éxito de la aplicación depende del número de conjuntos estables inducidos, lo que no sólo depende del número de vértices del grafo sino antes bien de su grado de simetría (nótese cómo el caso 1 de 16 vértices tiene 2.916 conjuntos estables, mientras que el caso 5 de 120 vértices tiene sólo 883, para el primero hay 2 simetrías, para el último 60).

Está claro que el espectro de aplicaciones queda abierto a otros problemas distintos de los aquí tratados, lo que confiere a la estabilización un interés general.

Con esta experiencia, los autores creen aportar una herramienta que puede ser muy útil para la resolución de problemas combinatorios NP-Completos sobre grafos simétricos y en los que las propiedades de la estabilización

preserven la optimización a realizar.

4. REFERENCIAS

- /1/ GAREY, M.R., GRAHAM, R.L., JOHNSON, D.J., KNUTH, D.E. "Complexity results for band with minimization". SIAM J. Appl. Math. v. 34, pp. 377-495.
- /2/ HARPER, L.H. "Chassis layout and Isoperimetric Problems". Jep Propulsion Laboratory SPS 37-66, vol. II, Sept. 1970.
- /3/ HARPER, L.H. "Stabilization and the edge sum problem". Ars Combinatoria, v. IV. - 1977. pp. 225-270.
- /4/ HARPER, L.H. "Optimal assignment of numbers to vertices". J. Soc. Indust. Appl. Math., v. 12, n° 1. 1964. pp. 131-135.
- /5/ HARPER, L.H. & BERENQUER, X. "On morphisms on the path problems". Próxima publicación.
- /6/ REINGOLD, E.M., NIEVERGELT, J. & DEO, N. "Combinatorial algorithms, theory and practice". Prentice-Hall, N.J. 1977.

5. NOTAS

¹En el apéndice se expone un ejemplo de esta secuencia. En general se ejemplifican allí la mayoría de los conceptos de este artículo; sirva esta nota para todos los demás -- puntos en los que el ejemplo del apéndice -- pueda ilustrar lo referido en el texto.

²Programas desarrollados en FORTRAN IV en un IBM 360/65 de la Universidad de California en Riverside. Recientemente se ha implementado una versión en el PDP 11/70 del Centro de Cálculo de la Universidad Politécnica de Barcelona.

6. APENDICE: Ejemplo

La figura 1 presenta un grafo ejemplo de 12 vértices en E^2 con 4 simetrías: R_0, R_1, R_2 y R_3 .

Tomemos $S = \{2,4,9,10\} = T_0$

Para la simetría $R_0, \Sigma_0 = \{4,10\}, \Sigma_1 = \{2\}, \Sigma_2 = \{9\}$.

Por lo tanto $\text{Stab}_{R_0,P}(S) = \{4,10,2,5\} = T_1$.

Apliquemos ahora R_1 sobre este último conjunto:

$\Sigma_0 = \{4,10,2,5\}, \Sigma_1 = \emptyset, \Sigma_2 = \emptyset$

Por lo tanto $\text{Stab}_{R_1,P}(T_1) = \{4,10,2,5\} = T_2$

Apliquemos ahora R_2 sobre T_2 :

$\Sigma_0 = \emptyset, \Sigma_1 = \{10\}, \Sigma_2 = \{4,2,5\}$

Por lo tanto $\text{Stab}_{R_2,P}(T_2) = \{10,7,8,6\} = T_3$

Y así sucesivamente se obtiene

$T_4 = \text{Stab}_{R_3,P}(T_3) = \{10,7,8,6\}$

$T_5 = \text{Stab}_{R_0,P}(T_4) = \{4,7,8,6\}$

$T_6 = \text{Stab}_{R_1,P}(T_5) = \{4,7,8,6\}$

$T_7 = \text{Stab}_{R_2,P}(T_6) = \{4,7,8,6\}$

$T_k = T_5$ para $k \geq 5$

por lo tanto $\{4,7,8,6\}$ es un conjunto estabilizado por efecto de las simetrías R_0, R_1, R_2, R_3 .

La fig. 2 representa el orden parcial inducido por la relación P_1 (un arco $i \rightarrow j$ representa $i > j$, o sea $\|i-p\| > \|j-p\|$ por alguna simetría). El orden parcial completo no se ilustra, pero es fácil ver sobre la fig. 2 que contiene un total de 31 arcos.

La fig. 3 corresponde al listado de salida del programa REFLEX.

La fig. 4 representa el diagrama de Hasse de este orden parcial, mientras que la fig. 5 -- corresponde al listado de salida del programa HASSE, con inclusión de la relación inversa (que se prepara para LOWSET).

En la fig. 6 se representa el retículo de -- los conjuntos estables como producto. Se tienen pues $5 \times 12 = 60$ conjuntos estables. La fig. 7 representa el árbol de enumeración de dichos conjuntos estables según el desarrollo

producido por el algoritmo del programa LOW-SET. Los números pequeños identifican cada nudo i . Los otros números representan los --vértices capturados en cada nudo (N_i). El -- listado de salida de dicho programa, con indicación del número de conjuntos para cada -- cardinalidad, viene expuesto en la fig. 8.

Las figuras 9 y 10 corresponden a los listados de salida de SHPATH como sendas soluciones a los problemas isoperimétrico y suma de

arcos, respectivamente. Esta última solución se representa gráficamente, en trazo grueso, en la fig. 11, en la que se ha dibujado la red N de conjuntos estables. Aquí otra vez -- los números pequeños identifican un conjunto estable, el mismo de la fig. 7; los números en círculo indican vértice añadido (para no cargar molestamente la figura, sólo se exponen cuando aparecen por primera vez); sobre los arcos del camino mínimo se indican los -- incrementos marginales de θ .

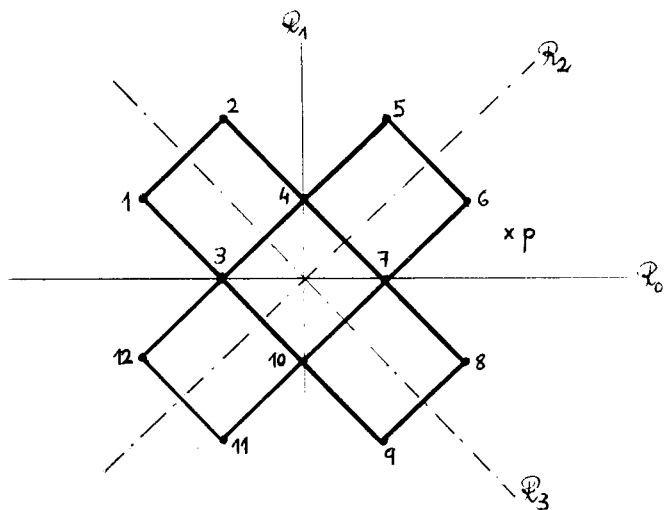


Fig. 1
Grafo ejemplo en E^2 $n=12$

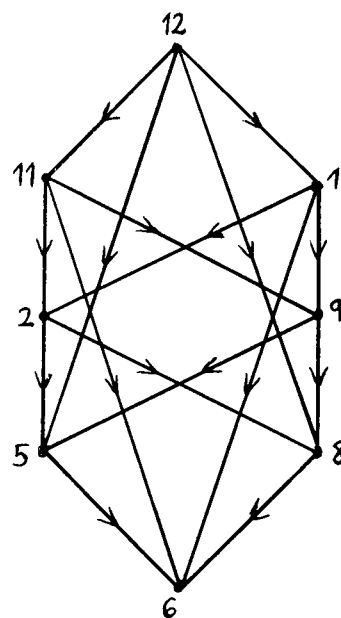


Fig. 2
Grafo de representación del P_1

*** 4 REFLECTIONS ***

REFLECTION	1	1.00000000	0.00000000
REFLECTION	2	0.00000000	1.00000000
REFLECTION	3	0.70709997	0.70709997
REFLECTION	4	-0.70709997	0.70709997

POINT P 2.50000000 0.50000000

*** SET P1 ***

VERTEX	1	IS RELATED BY <	WITH	1	VERTICES	12
VERTEX	2	IS RELATED BY <	WITH	2	VERTICES	1 11
VERTEX	3	IS RELATED BY <	WITH	0	VERTICES	0
VERTEX	4	IS RELATED BY <	WITH	2	VERTICES	3 10
VERTEX	5	IS RELATED BY <	WITH	3	VERTICES	2 9 12
VERTEX	6	IS RELATED BY <	WITH	4	VERTICES	1 5 8 11
VERTEX	7	IS RELATED BY <	WITH	3	VERTICES	3 4 10
VERTEX	8	IS RELATED BY <	WITH	3	VERTICES	2 12 9
VERTEX	9	IS RELATED BY <	WITH	2	VERTICES	1 11
VERTEX	10	IS RELATED BY <	WITH	1	VERTICES	3
VERTEX	11	IS RELATED BY <	WITH	1	VERTICES	12
VERTEX	12	IS RELATED BY <	WITH	0	VERTICES	0

Fig. 3
Listado de REFLEX

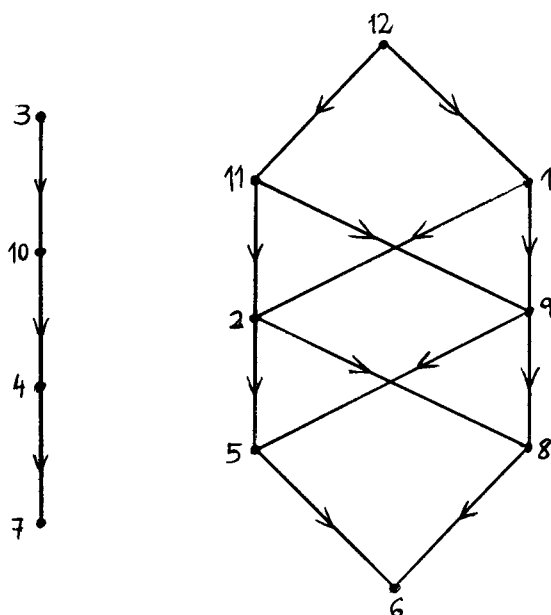


Fig. 4
Diagrama de Hasse

*** HASSE DIAGRAM ***

```

VERTEX 1 IS RELATED BY < WITH 1 VERTICES AND IS RELATED BY > WITH 2 VERTICES
  < TO VERTICES 12
  > TO VERTICES 2 9
VERTEX 2 IS RELATED BY < WITH 2 VERTICES AND IS RELATED BY > WITH 2 VERTICES
  < TO VERTICES 1 11
  > TO VERTICES 5 8
VERTEX 3 IS RELATED BY < WITH 0 VERTICES AND IS RELATED BY > WITH 1 VERTICES
  > TO VERTICES 10
VERTEX 4 IS RELATED BY < WITH 1 VERTICES AND IS RELATED BY > WITH 1 VERTICES
  < TO VERTICES 10
  > TO VERTICES 7
VERTEX 5 IS RELATED BY < WITH 2 VERTICES AND IS RELATED BY > WITH 1 VERTICES
  < TO VERTICES 2 9
  > TO VERTICES 6
VERTEX 6 IS RELATED BY < WITH 2 VERTICES AND IS RELATED BY > WITH 0 VERTICES
  < TO VERTICES 5 8
VERTEX 7 IS RELATED BY < WITH 1 VERTICES AND IS RELATED BY > WITH 0 VERTICES
  < TO VERTICES 4
VERTEX 8 IS RELATED BY < WITH 2 VERTICES AND IS RELATED BY > WITH 1 VERTICES
  < TO VERTICES 2 9
  > TO VERTICES 6
VERTEX 9 IS RELATED BY < WITH 2 VERTICES AND IS RELATED BY > WITH 2 VERTICES
  < TO VERTICES 1 11
  > TO VERTICES 5 8
VERTEX 10 IS RELATED BY < WITH 1 VERTICES AND IS RELATED BY > WITH 1 VERTICES
  < TO VERTICES 3
  > TO VERTICES 4
VERTEX 11 IS RELATED BY < WITH 1 VERTICES AND IS RELATED BY > WITH 2 VERTICES
  < TO VERTICES 12
  > TO VERTICES 2 9
VERTEX 12 IS RELATED BY < WITH 0 VERTICES AND IS RELATED BY > WITH 2 VERTICES
  > TO VERTICES 1 11

```

Fig. 5
Listado de HASSE

***** LOWER SETS FOR 2, 12 GRAPH *****

```

TOTAL OF RECORDED NODES      59

FOR CARDINALITY 1 THERE ARE 2 LOWER SETS
FOR CARDINALITY 2 THERE ARE 4 LOWER SETS
FOR CARDINALITY 3 THERE ARE 5 LOWER SETS
FOR CARDINALITY 4 THERE ARE 7 LOWER SETS
FOR CARDINALITY 5 THERE ARE 7 LOWER SETS
FOR CARDINALITY 6 THERE ARE 8 LOWER SETS
FOR CARDINALITY 7 THERE ARE 7 LOWER SETS
FOR CARDINALITY 8 THERE ARE 7 LOWER SETS
FOR CARDINALITY 9 THERE ARE 5 LOWER SETS
FOR CARDINALITY 10 THERE ARE 4 LOWER SETS
FOR CARDINALITY 11 THERE ARE 2 LOWER SETS
FOR CARDINALITY 12 THERE ARE 1 LOWER SETS

NUMBER OF LOWER SETS GENERATED 60
MAX LEVEL OF NODES 6
    
```

Fig. 8
Listado de LOWSET

```

BEST SUBSETS OF CARDINALITY 1, THETA= 2
SUBSET 1 6

BEST SUBSETS OF CARDINALITY 2, THETA= 2
SUBSET 1 5 6

BEST SUBSETS OF CARDINALITY 3, THETA= 4
SUBSET 1 5 6 8
SUBSET 2 6 7 8
SUBSET 3 5 6 7

BEST SUBSETS OF CARDINALITY 4, THETA= 4
SUBSET 1 5 6 8 9
SUBSET 2 5 6 7 8
SUBSET 3 4 5 6 7

BEST SUBSETS OF CARDINALITY 5, THETA= 4
SUBSET 1 5 6 7 8 9
SUBSET 2 4 5 6 7 8

BEST SUBSETS OF CARDINALITY 6, THETA= 4
SUBSET 1 2 4 5 6 7 8
SUBSET 2 4 5 6 7 8 9

BEST SUBSETS OF CARDINALITY 7, THETA= 4
SUBSET 1 2 4 5 6 7 8 9
SUBSET 2 4 5 6 7 8 9 10

BEST SUBSETS OF CARDINALITY 8, THETA= 4
SUBSET 1 3 4 5 6 7 8 9 10
SUBSET 2 1 2 4 5 6 7 8 9
SUBSET 3 2 4 5 6 7 8 9 10

BEST SUBSETS OF CARDINALITY 9, THETA= 4
SUBSET 1 1 2 4 5 6 7 8 9 10
SUBSET 2 2 3 4 5 6 7 8 9 10
SUBSET 3 2 4 5 6 7 8 9 10 11

BEST SUBSETS OF CARDINALITY 10, THETA= 2
SUBSET 1 1 2 3 4 5 6 7 8 9 10

BEST SUBSETS OF CARDINALITY 11, THETA= 2
SUBSET 1 1 2 3 4 5 6 7 8 9 10 11

READ NODES 59, RECORDED NODES 52

BOUND FOR OPTIMAL CHAIN 36
    
```

Fig. 9
Listado de SHPATH
(solución al prob. isoperimétrico)

CARDINALITY	1: POINT ADDED IS	6, THETA OF SUBSET IS	2, ACCUMULATED THETA IS	2
CARDINALITY	2: POINT ADDED IS	5, THETA OF SUBSET IS	2, ACCUMULATED THETA IS	4
CARDINALITY	3: POINT ADDED IS	7, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	8
CARDINALITY	4: POINT ADDED IS	4, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	12
CARDINALITY	5: POINT ADDED IS	8, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	16
CARDINALITY	6: POINT ADDED IS	2, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	20
CARDINALITY	7: POINT ADDED IS	9, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	24
CARDINALITY	8: POINT ADDED IS	10, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	28
CARDINALITY	9: POINT ADDED IS	1, THETA OF SUBSET IS	4, ACCUMULATED THETA IS	32
CARDINALITY	10: POINT ADDED IS	3, THETA OF SUBSET IS	2, ACCUMULATED THETA IS	34
CARDINALITY	11: POINT ADDED IS	11, THETA OF SUBSET IS	2, ACCUMULATED THETA IS	36
CARDINALITY	12: POINT ADDED IS	12, THETA OF SUBSET IS	0, ACCUMULATED THETA IS	36

Fig. 10
Listado de SHPATH
(solución al prob. suma de arcos)

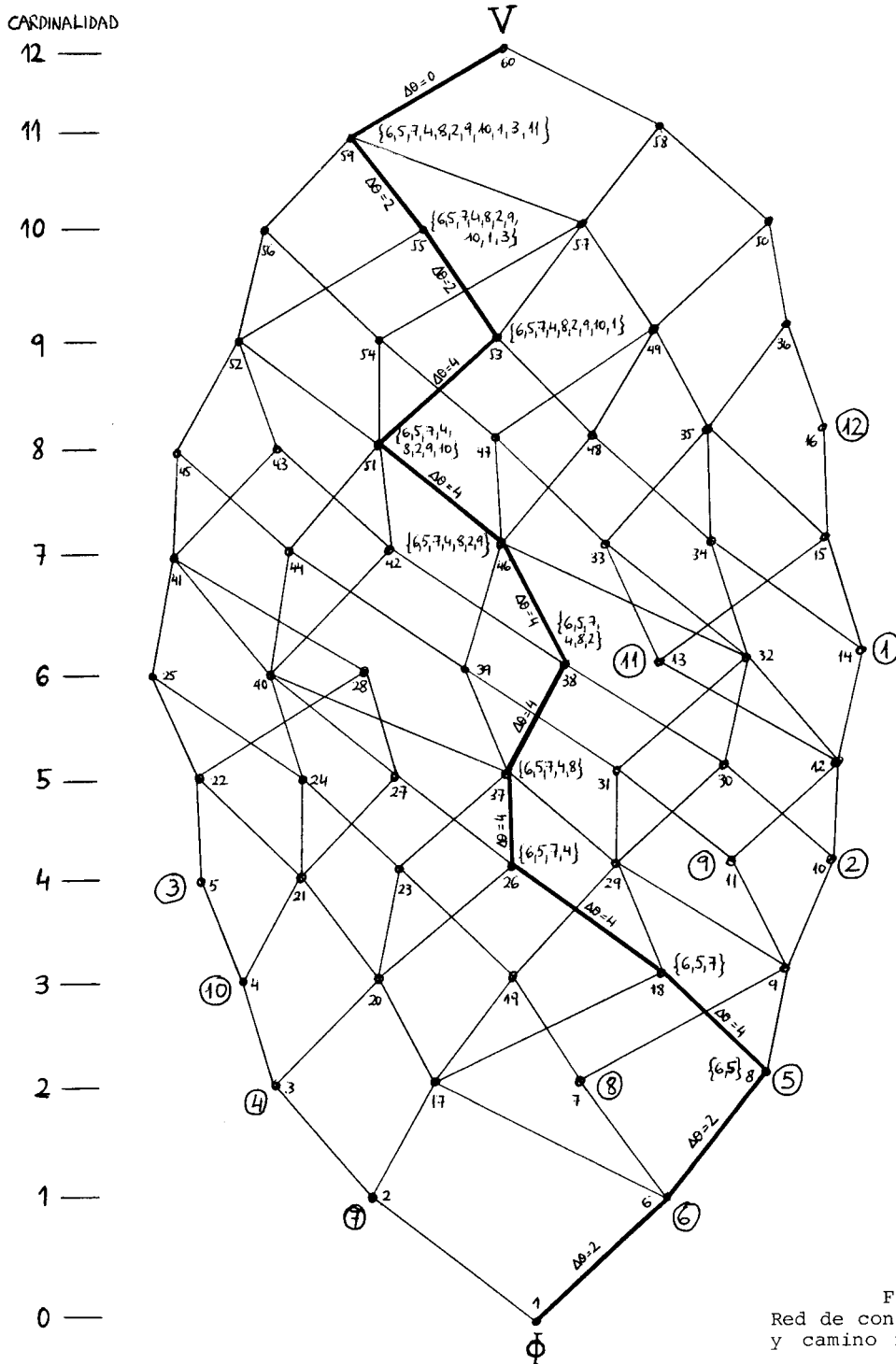


Fig. 11
Red de conjuntos estables
y camino mínimo de φ a V

