Matchings

Martín D. Safe

Instituto de Cálculo, Universidad de Buenos Aires

Tópicos Fundamentales en Teoría de Grafos
2.º semestre 2018

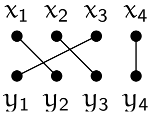
Matching

Matching. Matching perfecto.

Un matching en un grafo G es un conjunto de aristas (no bucles) que no comparten vértices.
Los vértices en los que inciden las aristas de un matching M se dicen saturados por M, o M-saturados.
Un matching perfecto es uno que satura todos los vértices.

Matchings perfectos en $K_{n,n}$

Consideremos $K_{n,n}$ con bipartición $X = \{x_1, \ldots, x_n\}$ e $Y = \{y_1, \ldots, y_n\}$. ¿Cuántos matchings perfectos distintos tiene $K_{n,n}$?



Un matching perfecto es una biyección entre X e Y. Entonces hay tantos matchings perfectos como la cantidad de permutaciones de $[n]$, es decir, $n!$. Estos matchings perfectos también pueden pensarse como matrices de ceros y unos con exactamente un uno por fila y por columna.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matchings perfectos en un grafo completo

Como K_{2n+1} tiene un número impar de vértices, no tiene matchings perfectos.

El número de matchings perfectos de K_{2n} es igual a la cantidad de formas de formar pares con $2n$ personas. ¿Cuál es el número f_n de matchings perfectos de K_{2n}?

- Una forma de determinarlo es la siguiente: Hay $2n-1$ opciones para la pareja del vértice v_{2n} y los demás vértices pueden ponerse formando pares de f_{n-1} formas. Como $f_0 = 1$, nos queda que $f_n = (2n-1)(2n-3)\cdots 1 = (2n-1)!!$.
- Otra forma es considerar que a cada ordenamiento secuencial de $[2n]$ le podemos hacer corresponder un matching: ponemos en un par el primero con el segundo, en otro par el tercero con el cuarto, y así siguiendo. Como cada matching es generado por $2^n n!$ ordenamientos distintos, la cantidad de matchings perfectos de K_{2n} es $(2n)!/(2^n n!)$.

Matchings máximos y maximales

Como un matching es un conjunto de aristas, su tamaño es la cantidad de aristas que lo forman. Podemos buscar un matching de tamaño grande seleccionando iterativamente una arista cuyos extremos no estén saturados por las aristas ya seleccionadas hasta que no se puedan agregar más aristas. De esta forma obtenemos un matching maximal pero no necesariamente un matching máximo.

Matchings maximales y máximos

Un matching maximal es un matching que no se puede extender agregando una arista.
Un matching máximo es un matching de tamaño máximo entre todos los matchings de un grafo.

Ejemplos de matchings maximales que no son máximos

Estos ejemplos nos sugieren una manera de buscar matchings más grandes.

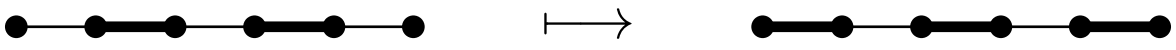
Caminos alternantes y caminos aumentantes

Caminos M-alternanes y M-aumentantes

Dado un matching M:

- Un camino M-alternante es un camino que alterna entre aristas de M y aristas que no son de M.
- Un camino M-aumentante es un camino M-alternante cuyos extremos no están saturados por M.

Dado un camino aumentante P podemos reemplazar las aristas de M en P por todas las restantes aristas de P para obtener un nuevo matching M' con una arista más:



Por lo tanto cuando M es un matching máximo, M no tiene caminos aumentantes.

De hecho, vamos a demostrar que los matchings máximos son exactamente aquellos que no tienen caminos aumentantes. Lo vamos a probar considerando dos matchings y examinando el conjunto de aristas que pertenecen a exactamente uno de ellos.

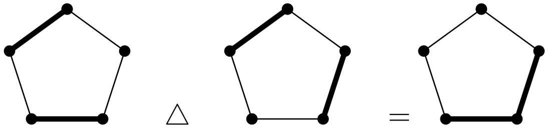
Diferencia simétrica

Diferencia simétrica

Si M y M' son dos conjuntos de aristas entonces su diferencia simétrica $M \triangle M'$ es el conjunto de aristas que pertenecen a exactamente a uno de los conjuntos M y M'. Esto es:

$$M \triangle M' = (M - M') \cup (M' - M).$$

Ejemplo de diferencia simétrica



Diferencia simétrica de dos matchings

Lema

Cada componente de la diferencia simétrica de dos matchings es un ciclo par o un camino.

Demostración.

Sean M y M' dos matchings y sea $F = M \triangle M'$. Como M y M' son matchings, cada vértice es incidente a lo sumo a una arista de cada uno de ellos. Luego, $F = M \triangle M'$ tiene a lo sumo dos aristas incidentes en cada vértice. Como $\Delta(F) \leqslant 2$, cada componente de F es un camino o un ciclo. Más aún, cada camino y ciclo de F alterna entre aristas de M y aristas de M'. Por lo tanto cada ciclo tiene longitud par, porque tiene igual número de aristas de M que de M'. $\square$

Teorema de Berge

Teorema (Berge, 1957)

Sea M un matching en un grafo G. El matching M es máximo en G si y sólo si G no tiene caminos M-aumentantes.

Demostración.

Si G tiene un camino M-aumentante entonces no es máximo porque ya vimos que podemos construir un matching más grande que M. Para la recíproca, vamos a probar que si G tiene un matching más grande que M entonces tiene un camino M-aumentante.

En efecto, sea M' un matching en G más grande que M. Por el lema anterior, $M \triangle M'$ consiste en caminos y ciclos pares y además los ciclos pares tienen tantas aristas de M como de M'. Como $|M'| > |M|$, F tiene que contener una componente con más aristas de M' que M. Tal componente tiene que ser un camino que empieza y termina con una arista de M'; por lo tanto es un camino M-aumentante de G. $\square$

Teorema de Hall

Supongamos que estamos en una situación en la cual contamos con un conjunto de ofertas laborales X y ciertos candidatos Y para cubrirlas. Para modelar este problema utilizamos un X, Y-bigrafo (un grafo bipartito con bipartición $\{X, Y\}$) donde la presencia de la arista xy indica que la oferta laboral $x \in X$ puede ser cubierta por el candidato $y \in Y$. Buscamos cubrir todas las ofertas laborales disponibles, es decir, buscamos un matching que sature X.

Si un matching M satura el conjunto X, entonces cada $S \subseteq X$ debe tener al menos $|S|$ vecinos en total, porque los vértices con los que los vértices de S forman pares en M pertenecen al conjunto de sus vecinos.

Notación

Recordemos que usamos $N_G(S)$, o simplemente $N(S)$, para indicar el conjunto de vecinos de S en G.

Sabemos que $|N(S)| \geqslant |S|$ es una condición necesaria y se conoce como la condición de Hall. Hall (1935) probó que esta condición es también suficiente.

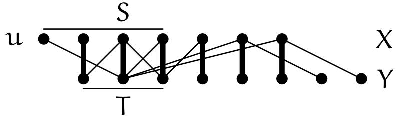
Teorema de Hall

Teorema (Hall, 1935)

Un X, Y-bigrafo tiene un matching que satura X si y sólo si $|N(S)| \geqslant |S|$ para todo $S \subseteq X$.

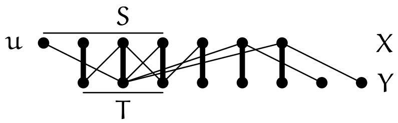
Demostración.

Ya vimos que es necesario que $|N(S)| \geqslant |S|$ porque $N(S)$ contiene todos los vértices con los que forman pares los vértices de S.
Para probar que la condición de Hall es suficiente supongamos que M es un matching máximo que no satura X entonces obtendremos un conjunto $S \subseteq X$ tal que $|N(S)| < |S|$.
Sea $u \in X$ un vértice no saturado por M. Sea $S \cup T$ el conjunto de los vértices alcanzables desde u por caminos M-alternantes en G donde $S \subseteq X$ y $T \subseteq Y$.



(...)

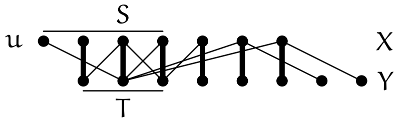
Teorema de Hall



Demostración (cont.)

Afirmamos que M une cada vértice de T con un vértices de $S - \{u\}$ y viceversa. Los caminos M-alternantes que salen desde u llegan a Y a través de aristas que no están en M y vuelven a X a través de aristas de M. Como no hay caminos M-aumentantantes, cada vértice de T está saturado; por lo tanto un camino M-aumentante que llega a $y \in Y$ se extiende a través de M hasta un vértice de S. Y por construcción, cada vértice de $S - \{u\}$ se alcanza a través de una arista en M desde un vértice en T. Luego, las aristas de M establecen una biyección entre T y $S - \{u\}$ y concluimos que $|T| = |S - \{u\}|$. (...)

Teorema de Hall



Demostración (cont.)

Como cada vértice de T es adyacente a un vértice de $S - \{u\}$, $T \subseteq N(S)$. Vamos a probar que $T = N(S)$. Supóngase, por el absurdo, que existe $y \in Y - T$ tiene un vecino v en S. La arista vy no puede estar en M porque u es no saturado y los restantes vértices de S están unidos por M a vértices de T. Luego, agregando vy a un camino M-alternante de u a v da lugar a un camino M-alternante de u hasta y. Pero esto contradice el hecho que $y \notin T$. Esta contradicción prueba que ningún vértice $y \in Y - T$ puede ser un vecino de un vértice de S y concluimos que $N(S) = T$. Como $N(S) = T$ entonces $|N(S)| = |T| = |S| - 1 < |S|$, como queríamos. $\square$

Teorema de Hall

Observaciones

- El Teorema de Hall nos dice que si un X, Y-bigrafo no tiene un matching que satura todo X es posible demostrarlo exhibiendo un conjunto $S \subseteq X$ con demasiado pocos vecinos en Y.
- Cuando los conjuntos de la bipartición tienen el mismo tamaño, el Teorema de Hall se vuelve el Teorema de los matrimonios probado originalmente por Frobenius en 1917. El nombre surge del problema de encontrar como establecer n matrimonios entre n hombres y n mujeres si se conoce cuáles mujeres son compatibles con cuáles hombres.
  Resulta que si cada hombre es compatible con k mujeres y cada mujer es compatible con k hombres entonces tiene que existir una forma de formar n matrimonios.

Matchings perfectos en grafos bipartitos regulares

Corolario (Frobenius, 1917)

Para $k > 0$, todo grafo bipartito k-regular tiene un matching perfecto.

Demostración.

Sea G un X, Y-bigrafo k-regular. Contando las aristas por sus extremos en X y por sus extremos en Y tenemos que $k|X| = k|Y|$. Como $k > 0$, $|X| = |Y|$. Por lo tanto alcanza con verificar la condición de Hall porque un matching que satura X satura también Y, o sea, es un matching perfecto.

Sea $S \subseteq X$. Sea m el número de aristas uniendo S con $N(S)$. Como G es k-regular, $m = k|S|$. Estas m aristas son incidentes en $N(S)$, así que $m \leqslant k|N(S)|$. En particular, $k|S| \leqslant k|N(S)|$ donde $k > 0$, lo que implica $|S| \leqslant |N(S)|$. Como S es un subconjunto arbitrario de X, el Teorema de Hall garantiza la existencia del matching perfecto. $\square$

Conjunto de representantes distintos

El Teorema de Hall puede ser enunciado en términos de conjuntos de representantes distintos de una familia de conjuntos.

Conjunto de representantes distintos

Un conjunto de representantes distintos de una familia de conjuntos $\{A_1, \ldots, A_n\}$ es un conjunto $\{a_1, \ldots, a_n\}$ de n elementos distintos de a pares tales que $a_i \in A_i$ para cada $i \in \{1, \ldots, n\}$.

Teorema (Hall, 1935)

Si $\mathcal{F} = \{A_1, \ldots, A_n\}$ es una familia de subconjuntos de un conjunto finito tal que, para cada $k \in \{1, \ldots, n\}$, toda unión de k elementos de $\mathcal{F}$ tiene tamaño al menos k, entonces $\mathcal{F}$ admite un conjunto de representantes distintos.

Teoremas mín-máx

Podemos probar que un matching M es máximo mostrando que no tiene caminos M-aumentantes. Sin embargo, explorar todos los posibles caminos M-alternantes para ver que ninguno es M-aumentante puede tomar mucho tiempo.

Quisiéramos lograr tener una situación similar a aquella que observamos para los grafos bipartitos: Para demostrar que un grafo no es bipartito no hace falta explorar todas las particiones en dos conjuntos de vértices para concluir que ninguna es una bipartición del grafo. Alcanza con exhibir un ciclo impar.

Volviendo a los matchings, en lugar de explorar todos los caminos M-alternantes preferiríamos contar con una estructura explícita en el grafo que impida que exista un matching con más aristas que M.

Cubrimiento por vértices

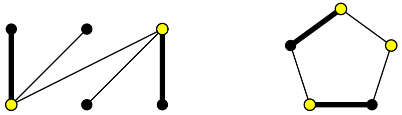
Un cubrimiento por vértices de un grafo G es un conjunto $Q \subseteq V(G)$ que cubre $E(G)$; es decir, que contiene al menos un extremo de cada arista de G.

Cubrimiento por vértices

En un grafo que representa una red de caminos, podemos pensar el problema de encontrar un cubrimiento con cardinalidad mínima como el problema de ubicar el mínimo número de guardias en los extremos de los caminos de manera que cada camino esté vigilado desde al menos uno de sus extremos por un guardia.

Como ningún vértice puede cubrir dos aristas distintas de un matching, el tamaño de todo cubrimiento por vértices es al menos el tamaño de cada matching. Por lo tanto, si contáramos con un matching y un cubrimiento por vértices del mismo tamaño esto demostraría que són optimos.

Ejemplo de matchings y cubrimientos por vértices



Teorema de Kőnig

Teorema de Kőnig (1931)

Si G es un grafo bipartito, entonces el tamaño de un matching máximo en G es igual al tamaño de un cubrimiento por vértices mínimo de G.

Demostración.

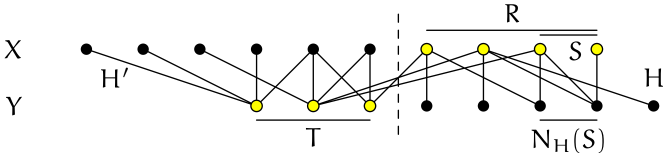
Sea G un X, Y-bigrafo. Si M es un matching de G y Q es un cubrimiento por vértices de G, como distintas aristas de M tienen que ser cubiertas por vértices distintos de Q, entonces $|M| \leqslant |Q|$.

Dado un cubrimiento por vértices de tamaño mínimo Q, vamos a construir un matching M tal que $|M| = |Q|$, lo que prueba que la igualdad siempre se puede alcanzar. $(\ldots)$

Teorema de Kőnig

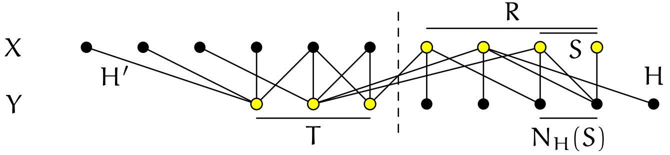
Demostración (cont.)

Particionamos Q en $R = Q \cap X$ e $T = Q \cap Y$ y sean H y H' los subgrafos de G inducidos por $R \cup (Y - T)$ y $T \cup (X - R)$, respectivamente.



Vamos a usar el Teorema de Hall para mostrar que H tiene un matching M_1 que satura todo R y H' tiene un matching M_2 que satura todo T. Esto probará que existe $M = M_1 \cup M_2$, un matching de G de tamaño $|R| + |T| = |Q|$, como queremos. $(\ldots)$

Teorema de Kőnig



Demostración (cont.)

Como $R \cup T$ es un cubrimiento por vértices, G no tiene aristas que unan $Y - T$ con $X - R$. Para cada $S \subseteq R$, consideramos $N_H(S)$, que está contenido en $Y - T$ (porque $V(H) \cap T = \emptyset$).
Afirmamos que $|N_H(S)| \geqslant |S|$. En efecto, si $|N_H(S)| < |S|$ entonces podríamos reemplazar $N_H(S)$ por S en Q y obtener un cubrimiento por vértices de menor tamaño que Q porque $N_H(S)$ cubre todos las aristas incidentes en S que no están cubiertas por T.
Luego, la minimalidad de Q implica que $|N_H(S)| \geqslant |S|$ para cada $S \subseteq R$ y el Teorema de Hall nos asegura que H tiene un matching M_1 que satura todo R. Aplicando el mismo razonamiento a H' se obtiene un matching M_2 en H' que satura T, como queríamos. $\square$

Relaciones mín-máx

Consideremos un problema de minimización **N** y un problema de maximización **M** sobre una clase $\mathcal{C}$ de instancias (por ejemplo, sobre todos los grafos). Diremos que **N** y **M** forman un par dual si, para cada instancia $\mathrm{I} \in \mathcal{C}$, el valor óptimo de **M** para I es menor o igual al valor óptimo de **N** para la misma instancia I. Por ejemplo, los problemas de cubrimiento por vértices mínimo y de matching máximo sobre grafos bipartitos forman un par dual.

Una relación mín-máx es un teorema que establece la igualdad entre los valores óptimos de un par dual de problemas de optimización sobre todas las instancias. El Teorema de Kőnig es una relación mín-máx entre los problemas de cubrimiento por vértices mínimo y de matching máximo en grafos bipartitos.

Si vale una relación mín-máx, la existencia de candidatos a soluciones del mismo tamaño para los problemas **N** y **M** en una instancia demuestra que ambos candidatos son soluciones óptimas para dicha instancia. Así, las relaciones mín-máx pueden proveernos certificados de optimalidad para los problemas involucrados.

Conjuntos independientes y cubrimientos

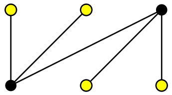
Para grafos bipartitos, no solo los matchings están relacionados con los cubrimientos por vértices, sino que también los conjuntos independientes están relacionados con los cubrimientos por aristas.

Número de independencia

El número de independencia de un grafo es el tamaño máximo de un conjunto independiente de vértices.

Observación

El número de independencia de un grafo bipartito no siempre es igual al tamaño del mayor de los conjuntos de una bipartición.



Conjuntos independientes y cubrimientos

Así como ningún vértice cubre más de una arista de un matching, ninguna arista contiene dos vértices de un mismo conjunto independiente. Esto nos lleva a definir el siguiente problema dual de cubrimiento.

Cubrimiento por aristas

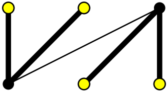
Un cubrimiento por aristas de G es un conjunto L de aristas de G que cubren a todos los vértices de G; es decir, tales que cada vértice de G es incidente a al menos una arista de L.

Observación

No todos los grafos tienen cubrimientos por aristas. Solo aquellos que no tienen vértices aislados tienen cubrimientos por aristas.

Conjuntos independientes y cubrimientos

Ejemplo



Las cuatros aristas incidentes en los vértices del conjunto independiente forman un cubrimiento por aristas. Los otros vértices quedan también cubiertos sin necesidad de aristas adicionales.

Conjuntos independientes y cubrimientos

Notación

Para los tamaños óptimos de los conjuntos independientes y de cubrimiento que fuimos definiendo introducimos la siguiente notación:

- $\alpha(G)$ es el tamaño de un conjunto independiente máximo.
- $\alpha'(G)$ es el tamaño de un matching máximo.
- $\beta(G)$ es el tamaño de un cubrimiento por vértices mínimo.
- $\beta'(G)$ es el tamaño de un cubrimiento por aristas mínimo.

Con esta notación, el Teorema de Kőnig establece que $\alpha'(G) = \beta(G)$ para todo grafo bipartito G.

Como ninguna arista puede cubrir dos vértices de un conjunto independiente, la desigualdad $\alpha(G) \leqslant \beta'(G)$ es inmediata. Vamos a probar que vale $\alpha(G) = \beta'(G)$ para todo grafo bipartito G sin vértices aislados.

Conjuntos independientes y cubrimientos

Notación

Si $S \subseteq V(G)$, vamos a denotar por $\bar{S}$ el conjunto $V(G) - S$.

Lema

En un grafo G, $S \subseteq V(G)$ es un conjunto independiente si y sólo si $\bar{S}$ es un cubrimiento por vértices. En particular,

$$\alpha(G) + \beta(G) = |V(G)|.$$

Demostración.

Si S es un conjunto independiente entonces cada arista es incidente en al menos un vértice de $\overline{S}$, es decir, $\overline{S}$ es un cubrimiento por vértices. Recíprocamente, si $\overline{S}$ cubre todas las aristas entonces no hay aristas que unan dos vértices de S, es decir, S es un conjunto independiente. Por lo tanto, un conjunto independiente máximo es el complemento de un cubrimiento por vértices mínimo y $\alpha(G) + \beta(G) = |V(G)|$. $\square$

Teorema de Gallai

Teorema (Gallai, 1959)

Si G es un grafo sin vértices aislados entonces

$$\alpha'(G) + \beta'(G) = |V(G)|.$$

Demostración.

A partir de un matching máximo M, vamos a construir un cubrimiento por aristas de tamaño $|V(G)| - |M| = |V(G)| - \alpha'(G)$.
Esto probará que el tamaño mínimo $\beta'(G)$ de un cubrimiento por aristas de G satisface $\beta'(G) \leqslant |V(G)| - \alpha'(G)$.
Luego, a partir de un cubrimiento por aristas mínimo L, vamos a construir un matching de tamaño $|V(G)| - |L| = |V(G)| - \beta'(G)$.
Esto probará que el tamaño máximo $\alpha'(G)$ de un matching de G satisface $\alpha'(G) \geqslant |V(G)| - \beta'(G)$.
La validez del teorema quedará probada entonces como consecuencia de ambas desigualdades. (..)

Teorema de Gallai

Demostración (cont.)

Sea M un matching máximo de G. Construimos un cubrimiento por aristas de G agregando a M una arista incidente a cada vértice no saturado (lo cual es posible porque G no tiene vértices aislados). Como usamos una arista por cada vértice, salvo en el caso de las aristas de M que sirven para dos vértices, entonces el tamaño total de este cubrimiento por aristas es $|V(G)| - |M|$.

Sea L un cubrimiento mínimo por aristas. Si ambas extremos de una arista e son extremos de aristas de L distintas de e entonces $e \notin L$, ya que $L - \{e\}$ seguiría siendo un cubrimiento por aristas. Por lo tanto, cada componente formada por aristas de L es una estrella (es decir, es un árbol con a lo sumo un vértice que no es una hoja). Sea k el número de esas componentes. Marcamos un vértice de grado máximo en cada una de esas componentes. Como L tiene una arista por cada vértice no marcado en cada estrella tenemos que $|L| = |V(G)| - k$. Formamos el matching de tamaño $k = |V(G)| - |L|$ eligiendo una arista de cada estrella en L. $\square$

Teorema de Kőnig para conjuntos independientes

Este teorema nos dice que en un grafo bipartito sin vértices aislados el tamaño de un conjunto independiente máximo es igual al tamaño mínimo de un cubrimiento por aristas.

Corolario (Kőnig, 1916)

Si G es un grafo bipartito sin vértices aislados entonces

$$\alpha(G) = \beta'(G).$$

Demostración.

Como G no tiene vértices aislados, por el lema y el Teorema de Gallai:

$$\alpha(G) + \beta(G) = |V(G)| = \alpha'(G) + \beta'(G).$$

Como G es bipartito, el Teorema de Kőnig nos asegura que $\alpha'(G) = \beta(G)$. Concluimos que $\alpha(G) = \beta'(G)$. $\square$

Dominación

En un cubrimiento por vértices, cada vértice cubre las aristas que son incidentes en él. Como las aristas incidentes en un vértice forman una estrella, el problema del cubrimiento por vértices mínimo puede describirse como el problema de cubrir todas las aristas con el menor número posible de estrellas. Muchas veces, en lugar de cubrir las aristas, se busca cubrir los vértices con la menor cantidad de estrellas.

Dominación

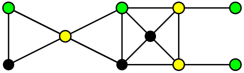
En un grafo G, un conjunto $S \subseteq V(G)$ es un conjunto dominante si cada vértice que no pertenece a S es vecino de un vértice en S. El número de dominación $\gamma(G)$ es el tamaño mínimo de un conjunto dominante de G.

La noción se debe a Berge (1962), el nombre a Ore (1962) y la notación a un survey de Cockayne y Hedetniemi (1977).

Número de dominación y cubrimiento por aristas

Ejemplo

En este grafo vemos dos conjuntos dominantes minimales: uno de tamaño 4 y otro de tamaño 3.



Resulta que $\gamma = 3$.

Observaciones

- Todo cubrimiento por vértices es un conjunto dominante y, por lo tanto, $\gamma(G) \leqslant \beta(G)$.
- La diferencia entre el número de dominación y el tamaño de un cubrimiento por vértices mínimo puede ser tan grande como se quiera. Por ejemplo, para K_n: $\gamma(K_n) = 1$ pero $\beta(K_n) = n - 1$.

Dominación

Cuando se estudia el problema de dominación como problema extremal, se trata de obtener cotas en términos de otros parámetros tales como la cantidad de vértices y el grado mínimo. Un vértice de grado k se domina a sí mismo y a otros k vértices. Por lo tanto en un grafo k-regular todo conjunto dominante tiene al menos $|V(G)|/(k+1)$ vértices.
Vamos a ver un resultado que prueba que para todo grafo con grado mínimo k, existe un algoritmo goloso que encuentra un conjunto dominante con una cantidad de vértices no muy superior a esa.

Vecindad cerrada

Si G es un grafo, denotamos por $N_G[v]$ al conjunto $N_G(v) \cup \{v\}$ que consiste en v y sus vecinos en G. Dicho conjunto se suele llamar la vecindad cerrada de v en G y consiste en aquellos vértices dominados por v (que son a su vez los vértices que dominan a v).

Cota superior para el número de dominación

Teorema (Arnautov, 1974; Payan, 1975)

Todo grafo con n vértices y de grado mínimo k tiene un conjunto dominante de tamaño a lo sumo

$$\frac{n}{k+1}(1 + \ln(k+1))$$

Demostración (Alon, 1990).

Sea G un grafo con grado mínimo k. Dado $S \subseteq V(G)$, tal que el conjunto U de los vértices no dominados por S satisface $U \neq \emptyset$. Afirmamos que algún vértice y fuera de S domina al menos a $(k+1)|U|/n$ vértices de U. En efecto, como cada vértice de G tiene al menos k vecinos, $\sum_{v \in U} |N_G[v]| \geqslant (k+1)|U|$. Por el principio del palomar, alguno de los n vértices de G está en al menos $(k+1)|U|/n$ de esas $|U|$ vecindades cerradas. Más aún, como y aparece en vecindades cerradas de elementos no dominados por S, entonces $y \notin S$, lo que prueba la afirmación. (...)

Cota superior para el número de dominación

Demostración (cont.)

Iterativamente seleccionamos un vértices que domina la mayor cantidad de los vértices no dominados. Recién probamos que si en algún paso hay r vértices no dominados entonces en el siguiente paso quedarán a lo sumo $r(1-(k+1)/n)$ vértices no dominados.

Por lo tanto, después de $(n/(k+1))\ln(k+1)$ pasos empezando con $S=\emptyset$, usando la desigualdad $1-x<e^{-x}$ para todo $x\neq 0$, vemos que el número de vértices no dominados es a lo sumo

$$n\left(1-\frac{k+1}{n}\right)^{\frac{n}{k+1}\ln(k+1)} < ne^{-\ln(k+1)} = \frac{n}{k+1}.$$

Los vértices ya elegidos junto con estos vértices no dominados forman un conjunto dominante de tamaño a lo sumo

$$\frac{n}{k+1}\ln(k+1)+\frac{n}{k+1}=\frac{n}{k+1}(1+\ln(k+1)). \qquad \square$$

Variantes de dominación

Muchas variantes del concepto de dominación son estudiadas.

Variantes de dominación

Un conjunto dominante S de G se dice:

- un conjunto dominante conexo si $G[S]$ es conexo,
- un conjunto independiente dominante si $G[S]$ no tiene aristas,
- un conjunto dominante total si $G[S]$ no tiene vértices aislados.

Cada cada variante agrega alguna restricción sobre la definición usual de dominación, los conjuntos dominantes de estos tipos son al menos tan grandes como $\gamma(G)$.

Conjuntos independientes dominantes

El caso de los conjuntos independientes dominantes es especial porque son equivalentes a los conjuntos independientes maximales.

Lema

Un conjunto de vértices en un grafo es un conjunto independiente dominante si y sólo si es un conjunto independiente maximal.

Demostración.

Entre todos los conjuntos independientes, S es maximal si y sólo si todo vértice fuera de S tiene un vecino en S, lo que equivale a que S es un conjunto dominante. $\square$

Conjuntos independientes dominantes en grafos sin claw

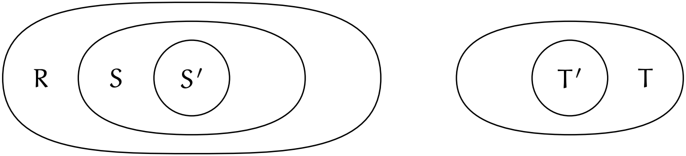
Recordemos que el claw es el grafo bipartito completo $K_{1,3}$.
Un grafo se dice libre de claw si no contiene al grafo claw como subgrafo inducido.

Teorema (Allan-Laskar, 1978)

Todo grafo libre de claw tiene un conjunto independiente dominante de tamaño $\gamma(G)$.

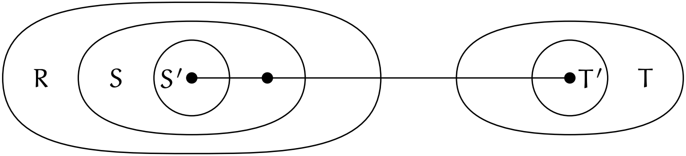
Demostración.

Sea S un conjunto dominante mínimo de un grafo G. Sea S' un subconjunto independiente maximal de S y sea R el conjunto $N_G[S'] = N_G(S') \cup S'$ de vértices dominados por S'. Por la maximalidad de S', $S \subseteq R$. Análogamente, si $T = V(G) - R$ y T' es un subconjunto independiente maximal de T entonces T' domina T.



(...)

Conjuntos independientes en grafos sin claw



Demostración (cont.)

Como T' no contiene vecinos en S', $S' \cup T'$ es un conjunto independiente. Esto prueba que $S' \cup T'$ es un conjunto independiente dominante.

Solo falta probar que $|S' \cup T'| \leqslant \gamma(G)$. Cada vértice de $S - S'$ tiene a lo sumo un vecino en T' porque tiene un vecino en S', $S' \cup T'$ es independiente y G es libre de claw. Como S es dominante, cada vértice de T' tiene al menos un vecino en $S - S'$. Luego, $|T'| \leqslant |S - S'|$. Concluimos que

$$|S' \cup T'| \leqslant |S'| + |S - S'| = |S| = \gamma(G). \qquad \square$$

Matching máximo en grafos bipartitos

Para encontrar un matching máximo, podemos buscar iterativamente caminos aumentantes que permitan incrementar el tamaño del matching que tenemos.

Dado un matching M en un X, Y-bigrafo G, vamos a buscar construir caminos M-aumentantes desde cada vértice no saturado por M en X. Solo hace falta buscar caminos partiendo desde X porque todo camino aumentante tiene longitud impar y por lo tanto tiene uno de sus extremos en X y el otro en Y. Vamos a buscar desde todos los vértices de X no saturados por M simultáneamente. Empezando con un matching de tamaño 0, $\alpha'(G)$ aplicaciones de la búsqueda de un camino aumentante producen un matching máximo.

Por el Teorema de Kőnig, sabemos que en un grafo bipartito es posible probar que un matching es máximo exhibiendo un cubrimiento por vértices del mismo tamaño.

Algoritmo de camino aumentante

Algoritmo de camino aumentante

Entrada: Un X, Y-bigrafo G, un matching M en G y el conjunto U de vértices en X no saturados por M.
Idea: Explorar los caminos M-alternantes desde U, donde $S \subseteq X$ y $T \subseteq Y$ son los vértices alcanzables. A medida que los vértices son alcanzados, almacenar los vértices desde los que fueron alcanzados.
Inicialización: $S = U$ y $T = \emptyset$. S no tiene ningún vértice marcado.
Iteración: Si S tiene vértices sin marcar, seleccionar un vértice sin marcar $x \in S$. Explorar x, considerando cada $y \in N(x)$ tal que $xy \notin M$. Si y no está saturado, terminar y **devolver** un camino aumentante desde U hasta y. Si no, y está unido a algún vértice $w \in X$ a través de una arista del matching. En este caso, incluir y en T (y almacenar que es alcanzable desde x) e incluir w en S (como alcanzable desde y). Una vez exploradas todas las aristas que inciden en x, marcar x e iterar.
Si S tiene todos sus vértices marcados, parar y **devolver** M como un matching máximo y $T \cup (X - S)$ como un cubrimiento mínimo.

Algoritmo de camino aumentante

Teorema

Aplicando repetidamente el algoritmo de camino aumentante a un grafo bipartito, se obtienen un matching y un cubrimiento por vértices del mismo tamaño.

Demostración.

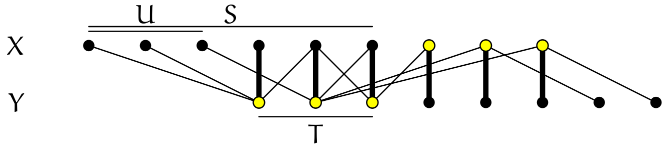
Solo debemos verificar que el algoritmo de camino aumentante devuelve un camino aumentante o un cubrimiento por vértices de tamaño M.

Si el algoritmo devuelve un camino aumentante, no hay nada que probar. Si no, termina con todos los vértices de S marcados y afirmando que $Q = T \cup (X - S)$ es un cubrimiento por vértices de tamaño $|M|$. Debemos probar que dicha afirmación es correcta. $(\ldots)$

Algoritmo de camino aumentante

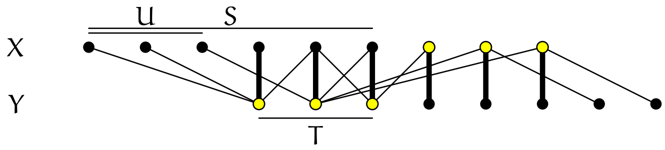
Demostración (cont.)

La situación es la siguiente (los vértices resaltados son los de Q):



Para mostrar que $Q = T \cup (X - S)$ es un cubrimiento por vértices, alcanza con mostrar que no hay aristas que unan S a $Y - T$. Un camino M-aumentante entra a X solo a través de aristas de M. Luego, cada vértice x de $S - U$ está en pareja vía M con un vértice de T. Por lo tanto, no hay ninguna arista de M que una un vértice de S con un vértice de $Y - T$. Pero tampoco hay una arista uniendo S con $Y - T$ que no pertenezca a M: cuando un camino alcanza $x \in S$, puede continuar a lo largo de cualquier arista que no esté en M, así que al explorar x todos sus vecinos pasan a T. Como el algoritmo marcó todos los vértices de S antes de terminar, todas las aristas que salen de S van hacia T. $(\ldots)$

Algoritmo de camino aumentante



Demostración.

Hemos probado que $Q = T \cup (X - S)$ es un cubrimiento por vértices. Ahora vamos a probar que Q tiene tamaño $|M|$. El algoritmo puso en T únicamente vértices saturados, porque si el algoritmo hubiese encontrado un vértice alcanzable en Y no saturado por M, habría reportado un camino aumentante. Como $U \subseteq S$, también cada vértice de $X - S$ está saturado y las aristas de M incidentes en $X - S$ no pueden tener su otro extremo en T. Por lo tanto M tiene al menos $|T| + |X - S| = |Q|$ aristas. Pero como un matching no puede ser más grande que un cubrimiento por vértices entonces $|M| = |Q|$. $\square$

Algoritmo de camino aumentante

- Sea G un X, Y-bigrafo con n vértices y m aristas.
- Vamos a suponer que G no tiene vértices aislados. (Si los tuviera, estos pueden removerse en tiempo $O(n)$.)
- Como $\alpha'(G) \leqslant n/2$ entonces obtenemos un matching máximo aplicando el algoritmo de camino aumentante a lo sumo $n/2$ veces.
- Cada aplicación explora cada vértice de X a lo sumo una vez (justo antes de marcarlo). Por lo tanto, cada arista se considera a lo sumo una vez.
- Por lo tanto, el tiempo de ejecución de una aplicación del algoritmo de camino aumentante es $O(m)$. (Que asciende a $O(n + m)$ si incluimos la remoción de los vértices aislados.)
- Concluimos que se puede encontrar un matching máximo de G en tiempo $O(nm)$.
- Para grafos densos, esta cota es $O(n^3)$.

Matchings estables

En lugar de optimizar el tamaño de un matching, podemos intentar optimizar las preferencias.

Por ejemplo, supongamos que, dados n hombres y n mujeres, quisiéramos establecer n "matrimonios estables", cada uno de ellos formado por un hombre y una mujer. Si un hombre x y una mujer a están formando parejas con otras personas, pero x prefiere a a frente a su actual pareja y a prefiere a x frente a su actual pareja, entonces tienen un incentivo para dejar sus actuales parejas para formar ellos una nueva pareja. En esta situación decimos que (x, a) es un par inestable.

Matching estable

Un matching perfecto se dice un matching estable si no tiene pares inestables.

Matrimonios estables

Ejemplo

Dados los hombres x, y, z, w y las mujeres a, b, c, d con las siguientes escalas de preferencias:

$$
\begin{array}{ccc}
x : a > b > c > d & \qquad & a : z > x > y > w \\
y : a > c > b > d & & b : y > w > x > z \\
z : c > d > a > b & & c : w > x > y > z \\
w : c > b > a > d & & d : x > y > z > w
\end{array}
$$

admite el siguiente matching estable $\{xa, yb, zd, wc\}$.

En el paper "College admission and the stability of marriage", Gale y Shapley (1962) probaron que un conjunto de matrimonios estables siempre existe y puede hallarse usando un algoritmo relativamente simple. En el algoritmo, hombres y mujeres no juegan roles simétricos, y vamos a discutir la importancia de esto más adelante.

Algoritmo de las propuestas de Gale y Shapley

Algoritmo de propuestas de matrimonio de Gale y Shapley

Entrada: Escala de preferencias de cada uno de los hombres y mujeres, respecto de las personas del sexo opuesto.
Idea: Construir un matching estable usando propuestas de matrimonio, recordando quién rechazó a quién una tal propuesta.
Iteración: Cada hombre le propone matrimonio a la mujer que se ubica más alto en su escala de preferencias y que no lo haya rechazado con anterioridad.
Si alguna mujer recibe más de una propuesta, cada mujer que recibe más de una propuesta le responde "quizás" a aquel entre aquellos que le realizaron una propuesta que se ubica más alto en su propia escala de preferencias, y rechaza a todos los demás que le hayan realizado una propuesta.
Repetir hasta que cada mujer recibe exactamente una propuesta.
Salida: Los matrimonios formados por cada mujer y el único hombre que le hizo una propuesta en la última iteración.

Algoritmo de las propuestas de Gale y Shapley

Volvamos a considerar el ejemplo anterior.

Ejemplo

Dados los hombres x, y, z, w y las mujeres a, b, c, d con las siguientes escalas de preferencias:

$$
\begin{array}{cc}
x : a > b > c > d & a : z > x > y > w \\
y : a > c > b > d & b : y > w > x > z \\
z : c > d > a > b & c : w > x > y > z \\
w : c > b > a > d & d : x > y > z > w
\end{array}
$$

Algoritmo de las propuestas de Gale y Shapley

Analicemos el algoritmo de Gale y Shapley en dicho ejemplo paso a paso:

- **Iteración 1:** Propuestas: xa, ya, zc y wc. En consecuencia: a le dice 'quizás' a x y rechaza a y, mientras que c le dice 'quizás' a w y rechaza a z.
- **Iteración 2:** Propuestas: xa, yc, zd y wc. En consecuencia: a responde 'quizás a x, c responde 'quizás' a w y rechaza a y, y d responde 'quizás' a z.
- **Iteración 3:** Propuestas: xa, yb, zd y wc. El algoritmo devuelve estas parejas como matrimonios estables.

Algoritmo de las propuestas de Gale y Shapley

Teorema (Gale y Shapley, 1962)

El algoritmo de Gale y Shapley devuelve un matching estable.

Demostración.

Observemos que toda mujer que haya rechazado una propuesta de matrimonio se ubica en lo más alto en la escala de preferencias del último hombre al que dijo "quizás" entre aquellas mujeres que no rechazaron a dicho hombre. Luego, como hay tantos hombres como mujeres, no puede darse que al finalizar una iteración del algoritmo un hombre x haya sido rechazado por todas las mujeres. Más aún, el algoritmo en algún punto termina porque la cantidad total de pares hombre-mujer donde la mujer no rechazó al hombre decrece estrictamente con cada iteración. (...)

Algoritmo de las propuestas de Gale y Shapley

Demostración (cont.)

Observemos que la secuencia de hombres a los cuales una mujer dice “quizás” es creciente en su escala de preferencias. Esto sucede porque la mujer dice “quizás” al mismo hombre mientras que no aparezca una oferta mejor y porque una vez que una mujer le dice “quizás” a un hombre éste le sigue proponiendo matrimonio hasta que esta misma mujer lo rechace o el algoritmo termine y contraigan matrimonio.

Veamos entonces que el resultado es un matching estable. Si el resultado no fuera estable, entonces existiría un par inestable (x, a) donde x está en pareja con b e y está en pareja con a. Pero, como la mujer a dijo“quizás” en orden creciente según su propia escala de preferencias, x nunca le propuso matrimonio a a. Sin embargo, x le propuso matrimonio a b, para lo cual debió haber sido rechazado por a, ya que a se sitúa más alto en su escala de preferencias. Esta contradicción demuestra que la salida del algoritmo es un matching estable. $\square$

Algoritmo de las propuestas de Gale y Shapley

La asimetría entre hombres y mujeres en el algoritmo de las propuestas nos lleva a preguntarnos cuál de los dos grupos es más beneficiado por el algoritmo.

Cuando la primera elección de los hombres son todas distintas, todos obtienen la mujer que se ubica más alto en su propia escala de preferencias, mientras que las mujeres terminan casándose con el primero que les ofrece matrimonio independientemente de su propia escala de preferencias. Cuando el algoritmo se ejecuta con las mujeres ejerciendo el rol de formular las propuestas, se puede dar la situación opuesta en que cada mujer pudo escoger al marido que se ubica más alto en su escala de preferencias mientras que cada hombre contrajo matrimonio con la primer mujer que se lo propuso, independientemente de su propia escala de preferencias.

Si la mujer propone...

Ejemplo

Si volvemos al ejemplo anterior con los hombres x, y, z, w y las mujeres a, b, c, d con las escalas de preferencias:

$$
\begin{array}{ll}
x : a > b > c > d & \quad a : z > x > y > w \\
y : a > c > b > d & \quad b : y > w > x > z \\
z : c > d > a > b & \quad c : w > x > y > z \\
w : c > b > a > d & \quad d : x > y > z > w
\end{array}
$$

vemos que si el algoritmo de Gale y Shapley se ejecuta dejando que las mujeres sean las que formulen las propuestas, el resultado es el matching estable: $\{az, by, cw, dx\}$ donde cada mujer contrae matrimonio con el hombre que se ubica en lo más alto de su propia escala de preferencias.

La "conclusión" que sacamos es que las convenciones sociales tradicionales favorecerían a los hombres.

Otro tipo de matrimonios estables

Curiosamente, este algoritmo no se hizo célebre por su aplicabilidad para arreglar matrimonios.

Cada año, los graduados de facultades de medicina de Estados Unidos presentan sus escalas de preferencias respecto de los hospitales en los que querrían hacer sus residencias. A su vez, los hospitales tienen sus propias escalas de preferencias respecto de los graduados. Podemos modelar a un hospital con varias vacantes como varios hospitales con la misma escala de preferencias. El desorden en la asignación de residentes hizo que los hospitales tuvieran que diseñar e implementar un algoritmo diez años antes de la solución de Gale y Shapley. El resultado fue el National Resident Matching Program (NRMP), una corporación sin fines de lucro que proporciona un día uniforme para presentar las solicitudes y establece el procedimiento de asignación de residentes a hospitales.

¿Quién está más feliz con el resultado?

El que hace las propuestas, claro. Como los hospitales corrían el algoritmo, no es sorprendente que ellos jugaran el rol de hacer las propuestas y estaban más felices con los resultados. La disconformidad de los residentes con el NRMP llevó a que en 1998 se cambiara a un algoritmo donde los estudiantes hacen las propuestas. En 1998, el algoritmo procesó 35.823 candidatos para cubrir 22.451 vacantes.

Este tipo de asimetría está presente también en otros contextos: los empleadores son los que hacen las propuestas conocidas como "ofertas laborales".

Pero el más feliz de todos debería ser Lloyd S. Shapley, que obtuvo el Premio Nobel de Economía 2012 (compartido con Alvin E. Roth) y, en su caso, se debió principalmente al algoritmo que diseñaron con David Gale. (David Gale había fallecido en 2008.)

¿Quién está más feliz con el resultado?

Puede haber matrimonios estables distintos a los que se pueden encontrar con las dos variantes del algoritmo de Gale y Shapley.

Para buscar un matching estable “justo”, le podríamos dar a cada persona un número de puntos con los cuales puntuar sus preferencias. El peso del par (x, a) entonces podría ser la suma de los puntos que x le da a a más los puntos que a le da a x y el problema sería el de encontrar un matching de peso total máximo. A continuación veremos un algoritmo para resolver este problema. Sin embargo, un matching óptimo en este nuevo sentido puede no ser estable en el sentido de Gale y Shapley...

Matching bipartito pesado

Ejemplo

Una empresa tiene n campos y n plantas de procesamiento. Cada campo produce maíz cuyo procesamiento requiere de la capacidad de toda una planta. La ganancia que resulta de procesar la cosecha del campo i en la planta j es $w_{i,j}$. La empresa busca maximizar la ganancia total.

Podemos considerar el X, Y-bigrafo donde $X = \{x_1, \dots, x_n\}$ es el conjunto de los campos e $Y = \{y_1, \dots, y_n\}$ es el conjunto de las plantas de procesamiento. Asignamos a la arista $x_i y_j$ que une el campo x_i con la planta de procesamiento y_j el peso $w_{i,j}$. La empresa busca entonces un matching de peso total máximo.

Matching bipartito pesado

The Hungarian Method for the Assignment Problem (Kuhn, 2010)

“I spent the summer of 1953 at the Institute for Numerical Analysis which was housed on the U.C.L.A. campus. I was supported by the National Bureau of Standards and shared an office with Ted Motzkin, a pioneer in the theory of inequalities and one of the most scholarly mathematicians I have ever known. I had no fixed duties and spent the summer working on subjects that were of interest to me at the time, such as the traveling salesman problem and the assignment problem.

Matching bipartito pesado

The Hungarian Method for the Assignment Problem (Kuhn, 2010)

“The Institute for Numerical Analysis was the home of the SWAC (Standards Western Automatic Computer), which had been designed by Harry Huskey and had a memory of 256 words of 40 bits each on 40 Williamson tubes. The formulation of the assignment problem as a linear program was well known, but a 10 by 10 assignment problem has 100 variables in its primal statement and 100 constraints in the dual and so was too large for the SWAC to solve as a linear program. The SEAC (Standard Eastern Automatic Computer), housed in the National Bureau of Standards in Washington, could solve linear programs with about 25 variables and 25 constraints. The SEAC had a liquid mercury memory system which was extremely limiting.

Matching bipartito pesado

The Hungarian Method for the Assignment Problem (Kuhn, 2010)

"During that summer, I was reading König's book on graph theory. I recognized the following theorem of König to be a pre-linear programming example of duality: If the numbers of a matrix are 0's and 1's, then the minimum number of rows and columns that will contain all of the 1's is equal to the maximum number of 1's that can be chosen, with no two in the same row or column.
Indeed, the primal problem is the special case of an assignment problem in which the ratings of the individuals in the jobs are only 0's and 1's. In a footnote, König refers to a paper of E. Egerváry (in Hungarian), which seemed to contain the treatment of a more general case.

Matching bipartito pesado

The Hungarian Method for the Assignment Problem (Kuhn, 2010)

"When I returned to Bryn Mawr, where I was on the faculty in 1953, I took out a Hungarian grammar and a large Hungarian-English dictionary and taught myself enough Hungarian to translate Egerváry's paper. I then realized that Egerváry's paper gave a computationally trivial method for reducing the general assignment problem to a 0-1 problem. Thus, by putting the two ideas together, the Hungarian Method was born. I tested the algorithm by solving 12 by 12 problems with random 3-digit ratings by hand. I could do any such problem, with pencil and paper, in no more than 2 hours. This seemed to be much better than any other method known at the time."

On the origin of the Hungarian method (Kuhn, 1991)

"This must have been one of the last times when pencil and paper could beat the largest and fastest electronic computer in the world."

Matching bipartito pesado

Ejemplo (cont.)

El gobierno considera que se está produciendo demasiado maíz, así que busca detener completamente la producción de maíz por parte de la empresa. Para ello, ofrece pagar a la empresa la suma u_i si acepta no usar el campo i y la suma v_j si acepta no usar la planta de procesamiento j. Si $u_i + v_j < w_{i,j}$ entonces a la empresa no le conviene aceptar la oferta del gobierno ya que puede procesar la producción del campo i en la planta j y hacer más dinero.

Por lo tanto, para conseguir que la empresa detenga totalmente la producción debe asegurarse que $u_i + v_j \geqslant w_{i,j}$ para todos los pares i, j. Si el gobierno además quisiera minimizar el costo total para lograrlo, necesita encontrar dichos valores de modo que se minimice la suma $\sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$.

Matching bipartito pesado

Matching de peso máximo

Un transversal de una matriz $n \times n$ es un conjunto de n entradas de esa matriz consistente en una entrada de cada fila y una entrada de cada columna.
Al problema de encontrar un transversal con suma máxima se lo llama el problema de asignación.
Esta es la formulación matricial del problema del matching perfecto de peso total máximo, donde se asignan pesos no negativos $w_{i,j}$ a las aristas de $K_{n,n}$ y buscamos un matching perfecto M de peso total $w(M)$ máximo.

Con estos pesos, un cubrimiento (pesado) es una elección de pesos $u_1, \dots, u_n$ y $v_1, \dots, v_n$ de forma tal que $u_i + v_j \geqslant w_{i,j}$ para todos los i, j.
El costo $c(u, v)$ del cubrimiento (u, v) es $\sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$.
El problema del cubrimiento de peso minimo es aquel que consiste en encontrar un cubrimiento de costo $c(u, v)$ mínimo.

Matching máximo y cubrimiento mínimo: dualidad

Lema

Sea M un matching perfecto y (u, v) un cubrimiento pesado en un grafo bipartito pesado G. Entonces $c(u, v) \geqslant w(M)$.
Más aún, $c(u, v) = w(M)$ si y sólo si M consiste en aristas $x_i y_j$ tales que $u_i + v_j = w_{i,j}$. En tal caso, M y (u, v) son óptimos.

Demostración.

Como M satura todos los vértices entonces sumando las restricciones $u_i + v_j \geqslant w_{i,j}$ para todas las aristas $x_i y_j$ en el matching se obtiene que $c(u, v) \geqslant w(M)$, cualquiera sea el cubrimiento (u, v).
Más aún, $c(u, v) = w(M)$ si y sólo si cada una de las restricciones $u_i + v_j \geqslant w_{i,j}$ valen con igualdad para cada arista $x_i y_j$ de M.
Finalmente, como $c(u, v) \geqslant w(M)$ vale para todo cubrimiento (u, v) y para todo matching perfecto M, entonces $c(u, v) = w(M)$ implica que no hay ningún matching perfecto con peso mayor que M ni ningún cubrimiento con peso menor que (u, v); es decir, si $c(u, v) = w(M)$ entonces tanto M como (u, v) son óptimos. $\square$

Subgrafo de igualdad

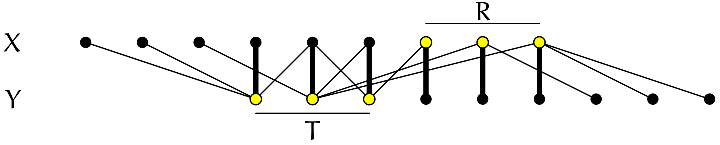
Como un matching y un cubrimiento son óptimos solo cuando las aristas del matching están cubiertas con igualdad, esto da lugar a un algoritmo. El subgrafo de igualdad juega un papel muy importante en ese algoritmo.

Definición

El subgrafo de igualdad para un cubrimiento pesado (u, v) es el subgrafo $G_{u,v}$ de $K_{n,n}$ formado por las aristas $x_i y_j$ tales que $u_i + v_j = w_{i,j}$.
El exceso de (u, v) para la arista $x_i y_j$ es $u_i + v_j - w_{i,j}$.

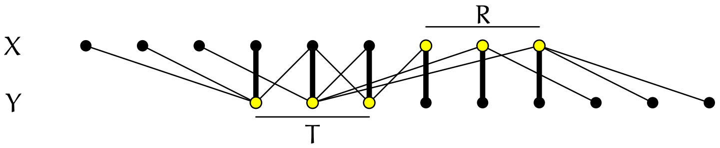
Matching bipartito de peso máximo

Si $G_{u,v}$ tiene un matching perfecto M entonces el peso de M es $\sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$ y, por el lema anterior, tenemos que M es óptimo y (u, v) también.
Si $G_{u,v}$ no tiene un matching perfecto, consideramos un matching máximo M y un cubrimiento por vértices Q del mismo tamaño de $G_{u,v}$ (por ejemplo, usando el algoritmo de camino aumentante).



Sean $R = Q \cap X$ y $T = Q \cap Y$. Entonces nuestro matching M de tamaño $|M| = |Q|$ consiste en $|R|$ aristas que unen R con $Y - T$ y $|T|$ aristas que unen T con $X - R$.

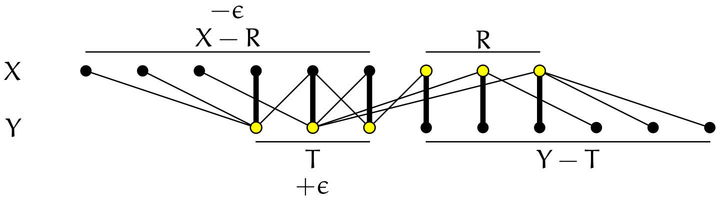
Matching bipartito de peso máximo



Para encontrar un matching más grande en el subgrafo de igualdad $G_{u,v}$, cambiamos (u, v) para incluir una arista entre $X - R$ e $Y - T$, manteniendo la igualdad sobre todas las aristas de M.

Las aristas que unen $X - R$ con $Y - T$ no están en $G_{u,v}$ porque tienen exceso $u_i + v_j - w_{i,j}$ positivo. Sea ϵ el exceso mínimo entre las aristas que unen $X - R$ con $Y - T$.

Matching bipartito de peso máximo



Reduciendo u_i en ϵ para todo $x_i \in X - R$, mantenemos la condición de cubrimiento para las aristas entre $X - R$ e $Y - T$ y a la vez incorporamos al menos una de ellas en el subgrafo de igualdad.

Para mantener la condición de cubrimiento en las aristas que unen $X - R$ con T, también incrementamos v_j en ϵ para cada $y_j \in T$.

Repetimos el procedimiento con el nuevo subgrafo de igualdad hasta que en algún momento obtenemos un cubrimiento cuyo grafo de igualdad tiene un matching perfecto.

Algoritmo húngaro

El algoritmo resultante fue llamado algoritmo húngaro por Harold Kuhn en honor a los trabajos de Kőnig y Egerváry en que se basó.

Algoritmo húngaro (Kuhn, 1955; Munkres, 1957)

Entrada: Una matriz pesos en las aristas de $K_{n,n}$ con bipartición $\{X, Y\}$ donde $X = \{x_1, \ldots, x_n\}$ e $Y = \{y_1, \ldots, y_n\}$.
Idea: Ir ajustando el cubrimiento (u, v) hasta que el subgrafo de igualdad $G_{u,v}$ tenga un matching perfecto.
Inicialización: Sea (u, v) el cubrimiento $u_i = \max_j w_{i,j}$ y $v_j = 0$.
Iteración: Encontrar un matching máximo M de $G_{u,v}$.
Si M no es perfecto, sea Q un cubrimiento por vértices de tamaño $|M|$ en $G_{u,v}$. Sean $R = X \cap Q$, $T = Y \cap Q$ y

$$\epsilon = \min\{u_i + v_j - w_{i,j} : x_i \in X - R, y_j \in Y - T\}.$$

Decrementar u_i en ϵ para cada $x_i \in X - R$ e incrementar v_j en ϵ para cada $y_j \in T$. Calcular el nuevo subgrafo de igualdad.
Repetir hasta que M sea un matching perfecto.
Salida: El matching perfecto M y el cubrimiento (u, v).

Algoritmo húngaro

Teorema (Kuhn, 1955; Munkres, 1957)

Si los pesos son racionales, el algoritmo húngaro encuentra un matching de peso máximo y un cubrimiento de peso mínimo.

Demostración.

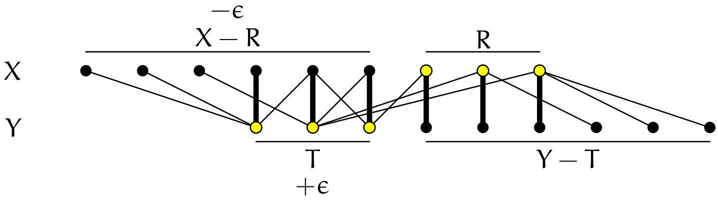
El algoritmo húngaro comienza con un cubrimiento. Puede terminar solo cuando el subgrafo de igualdad tiene un matching perfecto, lo que garantiza el mismo valor para el matching y el cubrimiento obtenidos.

Supongamos una iteración en la cual el subgrafo de igualdad no tiene un matching perfecto y sea (u, v) el cubrimiento al inicio de dicha iteración. Como ϵ es el mínimo de un conjunto finito de números positivos, $\epsilon > 0$. Denotamos (u', v') la nueva secuencia de costos asociadas a los vértices. $(\ldots)$

Algoritmo húngaro

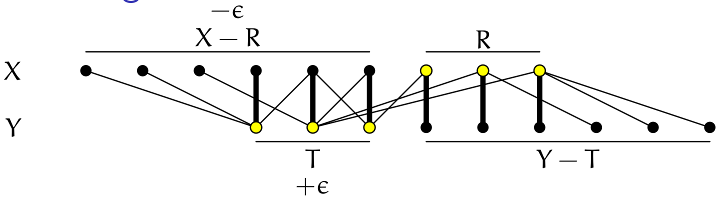
Demostración (cont.)

Veamos que (u', v') es un cubrimiento.



- El cambio en los pesos asociados a los vértices de $X - R$ y T hace que $u'_i + v'_j = u_i + v_j$ para las aristas entre $X - R$ y T y aquellas entre R e $Y - T$.
- Si $x_i \in R$ e $y_j \in T$ entonces $u'_i + v'_j = u_i + v_j + \epsilon$, y la arista $x_i y_j$ sigue igualmente cubierta.
- Si $x_i \in X - R$ e $y_j \in Y - T$ entonces $u'_i + v'_j$ es igual a $u_i + v_j - \epsilon$ y, por la elección de ϵ, $x_i y_j$ sigue cubierta. $(\ldots)$

Algoritmo húngaro



Demostración (cont.)

El algoritmo termina solo cuando el subgrafo de igualdad tiene un matching perfecto, así que alcanza con probar que termina. Como estamos suponiendo que los pesos $w_{i,j}$ son racionales, multiplicando los pesos por su mínimo común denominador da lugar a un problema equivalente en donde los pesos de las aristas son enteros. Luego el cubrimiento inicial del algoritmo es entero y los excesos también, por lo que los sucesivos cubrimientos obtenidos por el algoritmo húngaro son también enteros. Como en cada paso se reduce el costo del cubrimiento en una cantidad entera positiva (porque $|X - R| > |T|$), después de un número finito de pasos se da la igualdad. $\square$

Algoritmo húngaro

- Cuando los pesos son reales, el algoritmo aún funciona si obtenemos los cubrimientos por vértices en el grafo de igualdad más cuidadosamente.
- Vamos a mostrar que el algoritmo termina en a lo sumo n^2 iteraciones. Como cada iteración toma tiempo $O(n^2)$, esto probará además que el algoritmo se puede ejecutar en tiempo $O(n^4)$ (Munkres, 1957).
- Como las aristas de M se mantienen en el nuevo grafo de igualdad, el tamaño del matching en el grafo de igualdad nunca decrece.
- Como el tamaño del matching puede incrementar a lo sumo n veces, alcanza con mostrar que se incrementa en a lo sumo n iteraciones.
- Determinaremos el máximo matching M de $G_{u,v}$ aplicando el algoritmo de camino aumentante, lo que nos provee en la última iteración con un cubrimiento por vértices Q del mismo tamaño que M.

Algoritmo húngaro

- Más precisamente, determinaremos los conjuntos $S \subseteq X$ y $T \subseteq Y$ alcanzables mediante caminos M-alternantes desde el conjunto U formado por los vértices M-insaturados de X. Y definiremos $Q = R \cup T$ donde $R = X - S$.
- Aplicando un paso del algoritmo húngaro usando el cubrimiento por vértices Q se mantiene la igualdad en M y todas las aristas en caminos M-alternantes desde U. (Las aristas entre R y T desaparecen del grafo de igualdad, pero no tiene importancia.)
- En cada iteración, el grafo de igualdad gana al menos una arista entre S e $Y - T$, lo que produce un camino M-aumentante o agrega al menos un vértice nuevo a T dejando a U invariante.
- Como T puede contener a lo sumo n vértices, luego de n iteraciones obtenemos un matching más grande en el subgrafo de igualdad, como queríamos probar.

Matching bipartito de peso máximo

- La hipótesis de que los pesos de las aristas son no negativos no es restrictiva porque podemos sumarle a todas las aristas una misma constante para conseguirlo.
- Las hipótesis de que el grafo bipartito es balanceado (es decir $|X| = |Y|$) y completo no es restrictiva porque (una vez que todas las aristas tienen pesos no negativos) podemos agregar vértices para balancearlo y aristas con peso cero hasta hacerlo completo.
- A continuación, mostraremos que la igualdad $w(M) = c(u, v)$ se alcanza incluso si se exige que u y v sean no negativos.

Matching bipartito de peso máximo

Para ello vamos a utilizar el siguiente resultado.

Teorema (Dulmage y Mendelsohn, 1958)

Sea G un X, Y-bigrafo y sean $X' \subseteq X$ e $Y' \subseteq Y$. Entonces, G tiene un matching que satura $X' \cup Y'$ si y sólo si G tiene un matching M que satura X' y un matching N que satura Y'.

Bosquejo de la demostración.

El 'sólo si' vale trivialmente. Probaremos la recíproca.
Podemos asumir, sin pérdida de generalidad, que $E(G) = M \cup N$.
Entonces es posible probar que alguna componente de G es un camino que une un vértice N-insaturado $x' \in X'$ con un vértice M-insaturado $y' \in Y'$. Resulta que dicho camino tiene longitud par y une vértices en distintos lados de la bipartición, una contradicción. $\square$

Matching bipartito de peso máximo

Teorema (Egerváry, 1931)

Sea G un X, Y-bigrafo con pesos no negativos en las aristas. Entonces el máximo peso de un matching de G es igual al costo de un cubrimiento mínimo (u, v) de G tal que u y v son no negativos.

Demostración.

Sea M un matching de peso máximo y sea (u, v) un cubrimiento de costo mínimo entre aquellos en los que u y v son no negativos.
Ya sabemos, $w(M) \leqslant c(u, v)$ y vamos a probar la igualdad.
Sea $G_{u,v}$ el subgrafo de igualdad, sea X' el conjunto de vértices x_i para los cuales $u_i > 0$ y sea Y' el conjunto de los vértices y_j para los cuales $v_j > 0$.
Si $G_{u,v}$ tiene un matching M que satura X' e Y' simultáneamente entonces se da la igualdad $w(M) = (u, v)$. Por lo tanto, supongamos, por el absurdo, que $G_{u,v}$ no tiene un matching M que satura X' e Y' simultáneamente. $(\ldots)$

Matching bipartito de peso máximo

Demostración (cont.)

Por el teorema anterior y simetría, podemos suponer que no existe un matching de $G_{u,v}$ que cubra todo X'. Por el teorema de Hall, existe un subconjunto $S \subseteq X'$ tal que $|N_G(S)| < |S|$. Luego, existe un $\epsilon > 0$ tal que si se reduce u_i en ϵ para todos los vértices $x_i \in S$ y se incrementa v_j en ϵ para todos los vértices $y_j \in N_G(S)$ entonces los nuevos valores u y v siguen siendo no negativos. Pero entonces (u, v) sigue siendo un cubrimiento pero con costo menor al inicial. Esta contradicción prueba el teorema. $\square$

Matching máximo en grafos bipartitos más rápido

- El costo temporal $O(mn)$ para encontrar un matching máximo en grafos bipartitos que probamos para el algoritmo de camino aumentante puede mejorarse.
- Hopcroft y Karp (1973) probaron que si los sucesivos caminos aumentantes se toman de longitud mínima cada vez entonces sus longitudes son no decrecientes y los caminos de igual longitud no comparten vértices entre sí.
- Luego, es posible diseñar un algoritmo por etapas donde en cada etapa, usando breadth-first search (BFS) simultáneamente desde todos los vértices no saturados de X, encontraremos potencialmente varios caminos del mismo tamaño examinando una sola vez el conjunto de aristas.
- Más aún, $O(\sqrt{n})$ tales etapas son suficientes.
- Esto permite encontrar un matching máximo en cualquier grafo bipartito con n vértices y m aristas en tiempo $O(\sqrt{n}m)$.
- Para grafos densos, esta cota es $O(n^{2.5})$.

Algoritmo de Hopcroft y Karp

Observación

Si M es un matching de tamaño r y existe otro matching M^* de tamaño $s > r$ entonces hay al menos $s - r$ caminos M-aumentantes que no comparten vértices entre sí. Estos caminos pueden hallarse en $M \triangle M^*$.

Algoritmo de Hopcroft y Karp

El siguiente lema implica que la secuencia de longitudes de caminos en sucesivos aumentos de longitud mínima es no decreciente.

Lema

Si P es un camino M-aumentante de longitud mínima y P' es un camino $(M \triangle E(P))$-aumentante, entonces

$$|E(P')| \geqslant |E(P)| + 2|E(P) \cap E(P')|.$$

Demostración.

Por definición, $M \triangle E(P)$ es el matching obtenido al aumentar M a través del camino M-aumentante P. Sea N el matching $(M \triangle E(P)) \triangle E(P')$ obtenido al aumentar $M \triangle E(P)$ a través del camino $(M \triangle E(P))$-aumentante P'.

Como $|N| = |M| + 2$ entonces la observación anterior garantiza que $M \triangle N$ contiene dos caminos M-aumentantes P_1 y P_2 que no comparten vértices. Por hipótesis, cada uno de estos caminos es al menos tan largo como P. (...)

Algoritmo de Hopcroft y Karp

Demostración (cont.)

Como N se obtiene a partir de M intercambiando las aristas de P y luego las de P', una arista pertenece exactamente a uno de los matchings M y N si y sólo si pertenece a exactamente a uno de los caminos P y P'. Por lo tanto, $M \triangle N = E(P) \triangle E(P')$. Esto implica que

$$|E(P) \triangle E(P')| = |M \triangle N| \geqslant |E(P_1)| + |E(P_2)| \geqslant 2|E(P)|.$$

Luego

$$2|E(P)| \leqslant |E(P) \triangle E(P')| = |E(P)| + |E(P')| - 2|E(P) \cap E(P')|.$$

Concluimos que

$$|E(P')| \geqslant |E(P)| + 2|E(P) \cap E(P')|. \qquad \square$$

Algoritmo de Hopcroft y Karp

Lema

Si comenzamos con un matching vacío y $P_1, P_2, \ldots$ es una lista de caminos aumentantes de longitud mínima sucesivos, entonces los caminos de igual longitud no comparten vértices entre sí.

Demostración.

La demostración es por el absurdo. Consideremos el par P_k y P_ℓ con $\ell > k$ más cercanos entre sí en la lista tales que tienen la misma longitud pero comparten vértices. Por el lema anterior, las longitudes de los caminos aumentantes más cortos sucesivos son no decrecientes. Luego, las longitudes de los caminos $P_k, \ldots, P_\ell$ son todas iguales. Por construcción, los caminos $P_{k+1}, \ldots, P_\ell$ no comparten vértices entre sí. $(\ldots)$

Algoritmo de Hopcroft y Karp

Demostración (cont.)

Sea M' el matching dado por los caminos aumentantes $P_1, \ldots, P_k$. Como los caminos $P_{k+1}, \ldots, P_\ell$ no comparten vértices entre sí entonces P_ℓ es un camino M'-aumentante. Por el lema anterior,

$$|E(P_\ell)| \geqslant |E(P_k)| + 2|E(P_\ell) \cap E(P_k)|.$$

Pero como $|E(P_\ell)| = |E(P_k)|$ entonces $E(P_\ell) \cap E(P_k) = \emptyset$.

Sin embargo, veamos que P_k y P_ℓ deben compartir aristas. Por construcción, cada vértice de P_k es saturado por M' usando una arista de P_k. Luego, para cada vértice v común de P_k y P_ℓ, la arista de M' que satura a v es compartida por P_k y P_ℓ.

Esta contradicción muestra que no existe un par de caminos P_k y P_ℓ de igual longitud y que compartan vértices, lo que prueba la validez del lema. $\square$

Algoritmo de Hopcroft y Karp

Entrada: Un X, Y-bigrafo G.
Inicialización: $M = \emptyset$.
Iteración: Hallar un conjunto maximal de caminos M-aumentantes de longitud mínima $P_1, \ldots, P_q$ que son disjuntos por vértices y aumentar M a través de los caminos $P_1, \ldots, P_q$, es decir, cambiar M por $M \triangle (E(P_1) \cup \cdots \cup E(P_q))$.
Repetir hasta que no haya más caminos aumentantes.
Salida: El matching máximo M y un cubrimiento por vértices del mismo tamaño.

Algoritmo de Hopcroft y Karp

Lema

Si el grafo bipartito de entrada tiene n vértices, el algoritmo de Hopcroft y Karp termina en a lo sumo $O(\sqrt{n})$ iteraciones.

Demostración.

Alcanza con probar que el número total de iteraciones es a lo sumo $2\lfloor\sqrt{n/2}\rfloor + 2$.

Sean $P_1, \ldots, P_s$ todos los sucesivos caminos aumentantes de longitud mínima encontrados por el algoritmo, donde $s = \alpha'(G) \leqslant n/2$. Como los caminos de igual longitud no comparten vértices, cada P_{i+1} es un camino aumentante para el matching M_i obtenido usando los caminos aumentantes $P_1, \ldots, P_i$. $(\ldots)$

Algoritmo de Hopcroft y Karp

Demostración (cont.)

Alcanza con probar la siguiente condición más fuerte: *si* $P_1, \ldots, P_s$ *son caminos* M*-aumentantes sucesivos partiendo desde un matching vacío hasta llegar a un matching máximo, el número de longitudes distintas entre* $P_1, \ldots, P_s$ *es a lo sumo* $2\lfloor\sqrt{s}\,\rfloor + 2$.

Sea $r = s - \lceil\sqrt{s}\,\rceil$. Como $|M_r| = r$ y s es el tamaño de un matching máximo M, por la observación, existen al menos $s - r$ caminos M_r-aumentantes que no comparten vértices entre sí. Luego, el más corto de estos caminos usa a lo sumo $\lfloor r/(s-r)\rfloor$ aristas de M_r. Entonces $|E(P_{r+1})| \leqslant 2\lfloor r/(s-r)\rfloor + 1$. Como $r < s$ y $s - r \geqslant \sqrt{s}$ entonces $r/(s-r) < \sqrt{s}$. Por lo tanto, los caminos hasta P_r tienen todos longitud a lo sumo $2\lfloor\sqrt{s}\,\rfloor + 1$. Además, por definición de r, el número de caminos $P_{r+1}, \ldots, P_s$ es $\lceil\sqrt{s}\,\rceil$.

Como hay $\lfloor\sqrt{s}\,\rfloor + 1$ naturales impares hasta $2\lfloor\sqrt{s}\,\rfloor + 1$, los caminos $P_1, \ldots, P_r$ tienen a lo sumo $\lfloor\sqrt{s}\,\rfloor + 1$ longitudes distintas. Junto con los últimos $\lceil\sqrt{s}\,\rceil$ caminos, se llega a lo sumo a $\lfloor\sqrt{s}\,\rfloor + 1 + \lceil\sqrt{s}\,\rceil \leqslant 2\lfloor\sqrt{s}\,\rfloor + 2$ longitudes distintas en total. $\square$

Algoritmo de Hopcroft y Karp

- Afirmamos que cada iteración del algoritmo de Hopcroft y Karp puede completarse en tiempo $O(m)$.
- Sea M un matching no máximo de G y sea U el conjunto de vértices de X no saturados por M.
- Sea $S \cup T$ el conjunto de vértices de G alcanzables desde U mediante caminos M-alternantes, donde $S \subseteq X$ y $T \subseteq Y$.
- Procediendo como en el algoritmo de camino aumentante, es posible hallar (realizando BFS simultáneamente desde todo U simultáneamente) en tiempo $O(m)$, para cada $v \in S \cup T$, la distancia mínima de U a v mediante caminos M-alternantes, que denotamos $d(v)$.
- Más aún, en el mismo tiempo es posible también determinar para cada $v \in (S - U) \cup T$, los vertices inmediatamente anteriores a v en caminos M-alternantes de U a v de longitud mínima y almacenar su número que denotaremos por $\text{pred}(v)$.
- La longitud mínima ℓ de un camino M-aumentante es el mínimo de $d(v)$ sobre los vértices M-insaturados $v \in T$.

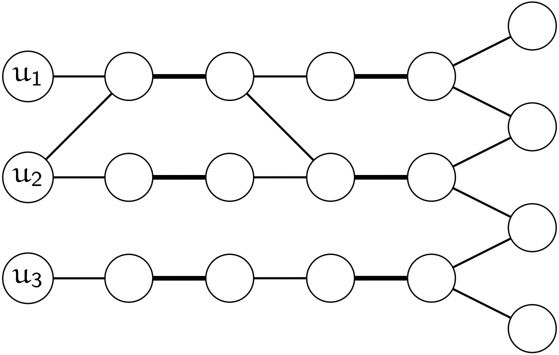
Algoritmo de Hopcroft y Karp

Algoritmo de caminos aumentantes mínimos disjuntos

Entrada: Un X, Y-bigrafo G con un matching M.
Inicialización: Sean $\mathcal{U}$, $\mathcal{S}$, T, $d(v)$, ℓ y $\mathrm{pred}(v)$ como en la transparencia anterior. Fijamos $\mathcal{P} = \emptyset$ y todo vértice como inexplorado.
Iteración: Considerar un vértice M-insaturado $v \in \mathsf{T}$ inexplorado tal que $d(v) = \ell$ y reconstruir un camino M-alternante P hasta v desde algún vértice $u \in \mathcal{U}$. Agregamos P a $\mathcal{P}$ y agregamos todos los vértices de P a una cola Q de vértices a ser borrados de G. Mientras haya un elemento z en la cola Q, lo marcamos como explorado y borramos todas las aristas del grafo incidentes en él. Cada vez que borramos una arista zz' tal que $d(z') = d(z) + 1$, decrementamos $\mathrm{pred}(z')$ en 1 y si $\mathrm{pred}(z')$ se vuelve 0 agregamos z' a la cola Q.
Repetir hasta que no haya más vértices M-insaturados en T que no hayan sido inexplorados.
Salida: Devolver $\mathcal{P}$.

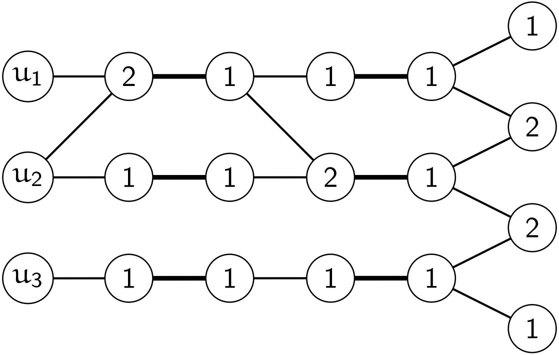
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



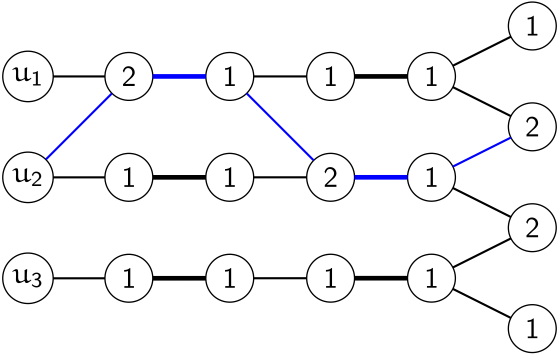
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



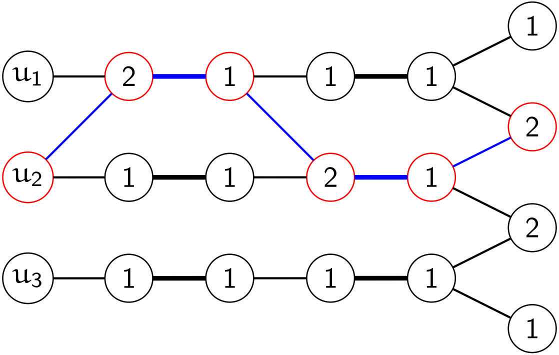
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



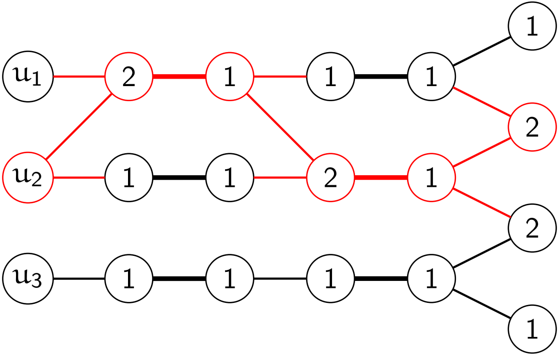
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



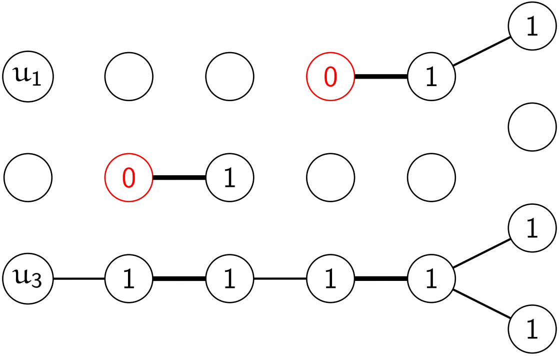
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



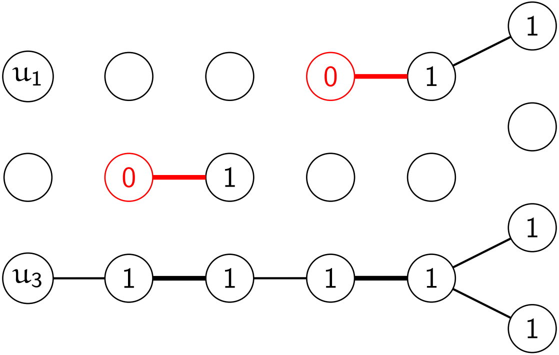
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



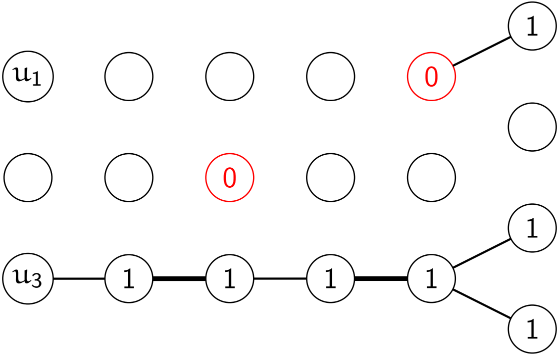
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



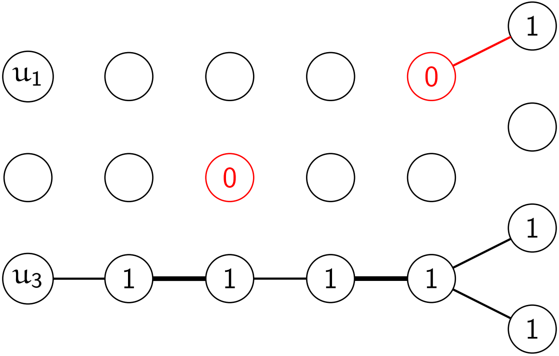
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



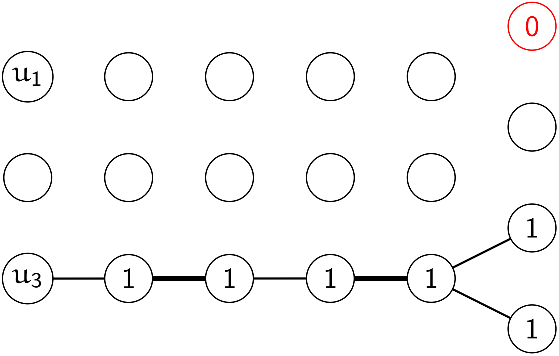
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



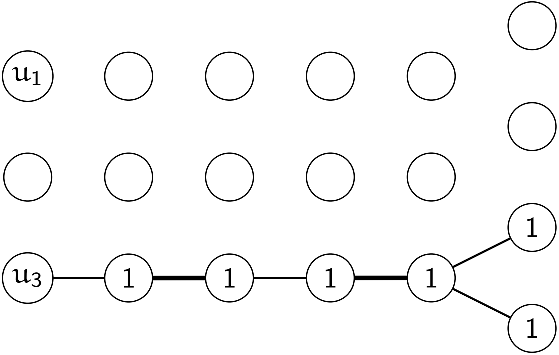
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



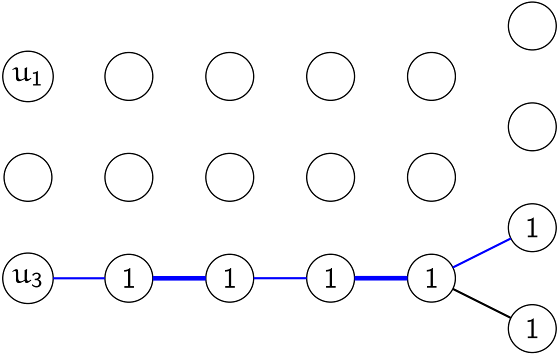
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



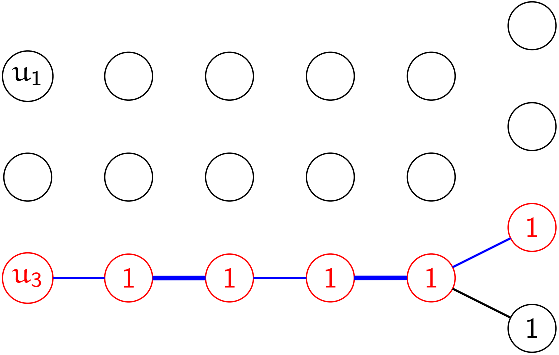
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



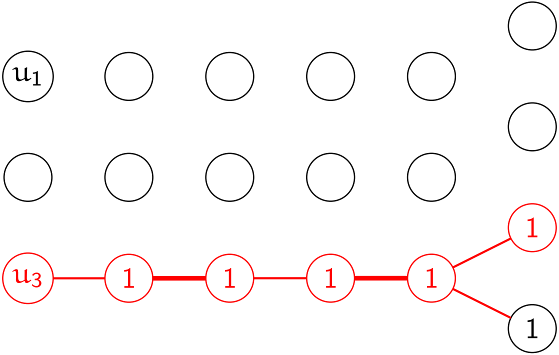
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



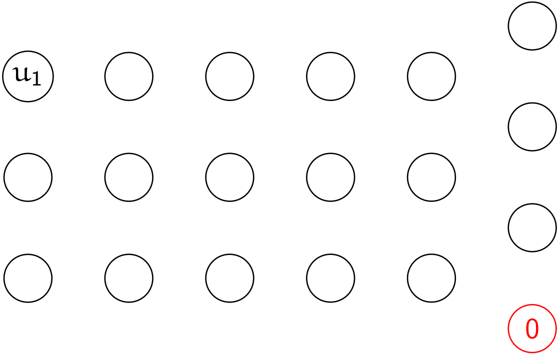
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



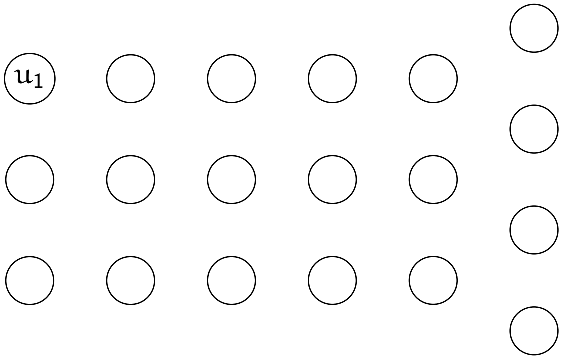
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



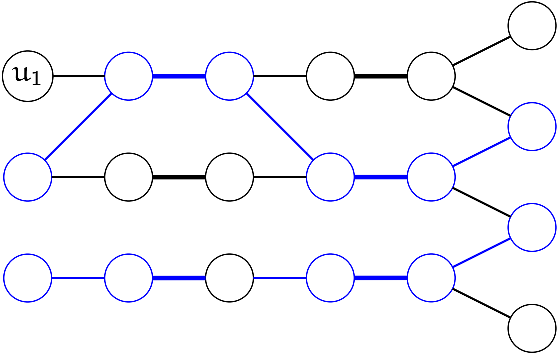
Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



Algoritmo de Hopcroft y Karp

Para analizar el comportamiento del algoritmo, alcanza con observar el subgrafo formado por todos los vértices que forman parte de caminos M-aumentantes de longitud mínima. Etiquetamos con números que indican el valor de pred. Quitamos dichas etiquetas cuando son explorados. Los vértices rojos son los que están en la cola Q.



Algoritmo de Hopcroft y Karp

Teorema (Hopcroft y Karp, 1973)

Si la entrada es un grafo bipartito con n vértices y m aristas entonces el algoritmo de Hopcroft y Karp puede ejecutarse en tiempo $O(\sqrt{n}m)$.

Demostración.

Por el lema anterior, el algoritmo de Hopcroft y Karp realiza $O(\sqrt{n})$ iteraciones. Y, a su vez, cada una de esas iteraciones se puede llevar adelante utilizando el algoritmo de caminos aumentantes mínimos disjuntos en tiempo $O(m)$, ya que el mismo realiza una cantidad $O(1)$ de operaciones por cada vértice y cada arista. $\square$

Matching de peso máximo en grafos arbitrarios

- El primero en dar un algoritmo polinomial (de tiempo $O(n^4)$) para hallar un matching de peso máximo en grafos arbitrarios fue Edmonds (1965).
- Gabow (1990) propuso para el mismo problema un algoritmo de tiempo $O(n(m + n \log n))$.
- El grafo de línea $L(G)$ de un grafo G tiene las aristas G como vértices y con dos aristas adyacentes en $L(G)$ si y sólo si son incidentes en G. El problema de hallar un matching de peso máximo es equivalente a hallar un conjunto independiente de peso máximo en el grafo de línea.
- Es fácil ver que los grafos de línea son grafos libres de claw.
- Recientemente se ha hallado un algoritmo de tiempo $O(n^2 \log n)$ que resuelve el problema de hallar un conjunto independiente de peso máximo en grafos libres de claw (Nobili y Sassano, 2015). Este resultado iguala la mejor complejidad temporal conocida para resolver el mismo problema en grafos de línea (Gabow, 1990). (Notar que si $G = L(H)$ entonces H tiene $|V(G)|$ aristas y $O(|V(G)|)$ vértices.)