Matchings Pesados

Daniel Penazzi

26 de mayo de 2021

Tabla de Contenidos

Minimizar suma

- Dado un grafo bipartito con partes iguales y pesos en los lados que representan costos queremos hallar un matching perfecto que minimize la suma de los costos.
- Ya vimos como hacer para minimizar el mayor costo, pero el algoritmo que resuelve este caso es mucho mas complicado.
- El algoritmo que daremos se llama el algoritmo Húngaro, y fue creado en 1955 por Kuhn.
- ¿Porqué “Húngaro” ?

Minimizar suma

- Khun lo nombró de esa forma en en honor a unos húngaros que sentaron las bases para el algoritmo, decadas antes que se creara el algoritmo.
- Uno de esos húngaros era Kőnig, del cual ya hablamos.
- El otro era Jenő Egerváry.
- Egerváry tambien se suicidó, en 1958, por problemas durante la ocupación soviética de Hungria en esa epoca

Minimizar suma

- Dado que un matching perfecto induce una biyección entre las filas y las columnas de la matriz que representa al grafo, esto es equivalente a lo siguiente:
  - Dada una matriz C $n \times n$, encontrar una permutación $\sigma : \{1, 2, ..., n\} \mapsto \{1, 2, ..., n\}$ tal que:

$$\sum_{i=1}^{n} C_{i,\sigma(i)} \leq \sum_{i=1}^{n} C_{i,\tau(i)}$$

    para toda permutación $\tau : \{1, 2, ..., n\} \mapsto \{1, 2, ..., n\}$
- Hay un caso superfácil.

Minimizar suma

Propiedad Obvia

Supongamos que C es una matriz de costos C tal que $C_{i,j} \geq 0 \forall i,j$. Sea σ una permutación de $\{1,2,...,n\}$ tal que $C_{i,\sigma(i)} = 0 \forall i$. Entonces el matching asociado a σ minimiza la suma de los costos.

- Prueba: obvio
- Pues si $C_{i,j} \geq 0 \forall i,j$ entonces todo matching tendrá suma al menos igual a 0 y como $C_{i,\sigma(i)} = 0 \forall i$ entonces $\sum_{i=1}^{n} C_{i,\sigma(i)} = 0$.
- fin prueba.

Minimizar suma

- Ok, superfantástico, si encontramos un “matching de ceros" asi, sabemos que minimiza la suma de costos.
- pero ¿cuáles son las posibilidades de que podamos encontrar una matriz con un “matching de ceros"?
- Pej, si C no tiene ceros, esto no nos sirve de nada.
- Pero tenemos otra propiedad casi obvia que hace que la anterior propiedad obvia nos sirva mucho.

Otra propiedad (casi) obvia

Propiedad Casi Obvia

Supongamos que C es una matriz de costos C $n \times n$ y sea $\tilde{C}$ una matriz que se obtiene de C restandole una constante a una fila o columna. Entonces todo matching que minimize la suma de costos respecto de C también minimiza la suma de costos respecto de $\tilde{C}$ (y viceversa).

- Prueba:
- Haremos la prueba para el caso de restarle una constante a una fila. (el caso de restarle a una columna es similar, aunque hay que aclarar que la simetria de los casos viene de que estamos suponiendo que la cantidad de filas es igual a la cantidad de columnas).

Prueba de la propiedad casi obvia

- Supongamos entonces para concretar que $\tilde{C}$ se obtiene de C restandole la constante r a la fila k-ésima.
- Es decir:

$$\tilde{C}_{i,j} = \begin{cases} C_{i,j} & \text{si } i \neq k \\ C_{k,j} - r & \text{si } i = k \end{cases}$$

- Sea σ una permutación asociada a un matching cualquiera.

$$\begin{aligned} \sum_{i=1}^{n} \tilde{C}_{i,\sigma(i)} &= \left(\sum_{i \neq k} C_{i,\sigma(i)} \right) + C_{k,\sigma(k)} - r \\ &= \left(\sum_{i=1}^{n} C_{i,\sigma(i)} \right) - r \end{aligned}$$

Prueba de la propiedad casi obvia

- Si σ minimiza la suma respecto de los costos C, tenemos que $\sum_{i=1}^{n} C_{i,\sigma(i)} \leq \sum_{i=1}^{n} C_{i,\tau(i)}$ para toda permutación τ.
- Entonces

$$\sum_{i=1}^{n} \tilde{C}_{i,\sigma(i)} = \left(\sum_{i=1}^{n} C_{i,\sigma(i)}\right) - r \leq \left(\sum_{i=1}^{n} C_{i,\tau(i)}\right) - r = \sum_{i=1}^{n} \tilde{C}_{i,\tau(i)}$$

- Y σ tambien mimimiza respecto de $\tilde{C}$.
- La vuelta es de la misma forma. Fin.

Minimizar suma

- Cómo usar las dos propiedades para encontrar un matching que minimize la suma:
  1. le restamos el minimo de cada fila a cada fila.
     - Esto nos asegura que hay al menos un 0 en cada fila.
  2. Luego de eso, le restamos el minimo de cada columna a cada columna.
     - Esto nos asegura que hay al menos un 0 en cada columna.
     - Y por lo anterior, entonces tenemos al menos un 0 en cada fila y cada columna.
  3. Buscamos un matching de 0s. (es decir, miramos el grafo en el cual dos vértices x, y son vécinos sólo si la entrada x, y de la matriz alterada es 0)
     - Si encontramos uno, minimiza la suma.
     - Si no....bueno, ahi se complica. Luego veremos como se hace en este caso.

Minimizar suma: ejemplo

- Ejemplo:

$$\begin{bmatrix} 1 & 7 & \infty & 4 \\ \infty & 5 & \infty & 7 \\ 4 & 7 & 4 & 8 \\ 9 & 4 & 7 & 5 \end{bmatrix}$$

- Restando el minimo de cada fila, tenemos:

$$\begin{bmatrix} 0 & 6 & \infty & 3 \\ \infty & 0 & \infty & 2 \\ 0 & 3 & 0 & 4 \\ 5 & 0 & 3 & 1 \end{bmatrix}$$

Ejemplo

- Ahora restamos el minimo de cada columna.
- Observemos que en las tres primeras columnas el mínimo es 0, asi que restar 0 hace que todo quede igual: el único cambio se ve en la última columna:

$$\begin{bmatrix} 0 & 6 & \infty & 2 \\ \infty & 0 & \infty & 1 \\ 0 & 3 & 0 & 3 \\ 5 & 0 & 3 & 0 \end{bmatrix}$$

- En este caso sencillo vemos que la diagonal forma un matching perfecto de ceros, asi que por las dos propiedades anteriores es un matching que minimiza la suma.
- El costo total es $1 + 5 + 4 + 5 = 15$.

Ejemplo 2

No pudimos extenderlo a un matching perfecto

$$\begin{array}{cccccc}
 & i & ii & iii & iv & \\
A & \infty & \color{blue}{[0]} & 4 & 2 & ii \\
B & 2 & 0 & \color{blue}{[0]} & 3 & iii \\
C & \color{blue}{[0]} & 0 & 4 & 0 & \\
D & \infty & 1 & 0 & 3 & s \\
 & & B & D & &
\end{array}$$

- $S = \{A, B, C\}$ tiene $\Gamma(S) = \{ii, iii\}$ asi que como $|S| > |\Gamma(S)|$, tenemos un certificado que no existe un matching de 0s

S

- Si no existe matching siempre tendremos un S con $|S| > |\Gamma(S)|$, y lo encontraremos algorítmicamente.
- Si miramos la prueba del teorema de Hall, vemos que el S es la intersección del corte con X.
- Si lo hacemos con el sistema de colas, entonces es fácil reconstruirlo.
- Si lo hacemos con este sistema de matrices, simplemente recordemos que como el corte final son todos los vértices que aparecen en la cola, tenemos que el corte final consiste de todas las filas y todas las columnas etiqueteadas. (mas el vértice “trucho”s, que no aparece si lo hacemos con la matriz):
- Asi que el S serán todas las filas etiqueteadas cuando paramos.

¿?

- Ok, entonces no hay matching perfecto de ceros en la matriz.
- ¿y ahora que hacemos?
- Porque matching con suma mínima existe, ya que hay al menos un matching perfecto, hay una cantidad finita de ellos y de entre todos ellos alguno tiene suma mínima.
- Asi que hay que hacer algo.
- La estrategia del Húngaro es usar el S que es un certificado de que no existe matching como guia para seguir modificando la matriz, agregando mas 0s y buscamos matching de 0s.
- Si no se encuentra, se encontrará un nuevo S con $|S| > |\Gamma(S)|$, y seguimos.
- Probaremos que eventualmente se encuentra un matching de 0s.

Mas “cambio" de la matriz

- ¿Cómo es exactamente el cambio que debemos hacer en la matriz?
- Como dijimos, la idea es agregar mas ceros en forma “compatible”con la matriz original, usando la propiedad casi obvia.
- Es decir, restando constantes adecuadas en filas o columnas.
- El S con $|S| > |\Gamma(S)|$ nos esta diciendo que en las filas de S no hay suficientes ceros.
- Asi que nos esta indicando que ahi hay que agregar mas ceros.

Mas “cambio" de la matriz

- Pero no parece posible hacer eso, ni tampoco restar de columnas, pues ya tenemos al menos un cero en cada fila y cada columna, si restamos algo mas tendremos números negativos!
- Ahi entre el siguiente truco: en la propiedad casi obvia. nunca dijimos que la constante a restar tuviera que ser positiva.
- Asi que podriamos restar una constante negativa, que equivale a sumar una constante a una fila o columna.
- Pero esto parece una locura: quiero crear mas ceros, para que sumaria una constante?
- La idea es que como quiero crear mas ceros en las filas de S, debo restar una constante, pero eso haria que algunas entradas fuesen negativas, entonces sumamos constantes a algunas columnas para que no queden negativos.

Mas “cambio" de la matriz

- ¿En cuales columnas?
- En la columnas de $\Gamma(S)$ ya tenemos ceros, asi que queremos encontrar nuevos ceros en el complemento de $\Gamma(S)$.
- Entonces una idea seria tomar $m =$el mínimo de los elementos de S que estén en el complemento de $\Gamma(S)$ y restarselo a las filas de S.
- Como tomamos el mínimo de la intersección de las filas de S con el complemento de $\Gamma(S)$, en esa zona aparecerán “nuevos ceros” pero ningún número negativo.
- Pero en $\Gamma(S)$ pueden (y de hecho van a aparecer) negativos.

Mas “cambio" de la matriz

- En nuestro ejemplo, deberiamos ver los elementos de las filas A, B, D que estén en las columnas i, iv (el complemento de $\Gamma(S)$).
- Son ∞, 2 y 3, asi que el mínimo es 2.
- Restando 2 de las filas A,B y D nos queda:

$$\begin{array}{cccccc} & i & ii & iii & iv & \\ A & \infty & -2 & 2 & 0 & ii \\ B & 0 & -2 & -2 & 1 & iii \\ C & 0 & 0 & 4 & 0 & \\ D & \infty & -1 & -2 & 1 & s \\ & & B & D & & \end{array}$$

- Tenemos nuevos ceros en las filas A y B....pero tambien tenemos numeros negativos!

Mas “cambio" de la matriz

- Si tenemos números negativos ya no podemos garantizar que un matching de ceros tenga suma mínima, asi que no nos sirve.
- Debemos “eliminar" esos números negativos.
- Ahi es cuando sumaremos un número a ciertas columnas.
- Como dijimos y se ve en el ejemplo, las columnas con problemas son las de $\Gamma(S)$.
- Asi que sumamos el número que le restamos a las filas de S a las columnas de $\Gamma(S)$.

Mas “cambio" de la matriz

$$\begin{array}{cccccc} & i & ii & iii & iv & \\ A & \infty & 0 & 4 & 0 & ii \\ B & 0 & 0 & 0 & 1 & iii \\ C & 0 & 2 & 6 & 0 & \\ D & \infty & 1 & 0 & 1 & s \\ & & B & D & & \end{array}$$

- Y ahora tenemos una matriz con todos no negativos.
- Observar que “perdimos" un cero de la matriz anterior: el que estaba en la fila C, columna ii.
- Como estamos sumando m a las columnas de $\Gamma(S)$ y no le restamos nada a las filas del complemento de S, esto es algo que puede pasar.

Mas “cambio" de la matriz

- Pero no es un problema.
- ¿Por qué no es un problema?
- Porque el problema que tenemos es que las filas de S estan “compitiendo" por los pocos ceros de $\Gamma(S)$, mientras que los elementos del complemento de S tienen “suficientes" ceros en el complemento de $\Gamma(S)$ como para que perder algunos ceros en $\Gamma(S)$ los afecte.

Ceros del matching

- Volviendo a la matriz:
- Recordemos donde estaban nuestro 0s del matching.

$$
\begin{array}{cccccc}
 & i & ii & iii & iv & \\
A & \infty & \color{blue}{[0]} & 4 & 0 & ii \\
B & 0 & 0 & \color{blue}{[0]} & 1 & iii \\
C & \color{blue}{[0]} & 2 & 6 & 0 & \\
D & \infty & 1 & 0 & 1 & s \\
 & & B & D & &
\end{array}
$$

- Observemos que si bien perdimos un 0, ese cero que perdimos no formaba parte del matching.
- Veremos que esto siempre pasa y es clave para reducir la complejidad.

Ceros del matching

- Si no fuera asi, luego de cada cambio de matriz deberiamos empezar desde cero a buscar matchings de ceros, y todo el esfuerzo que se hizo hasta ese momento se habria perdido.
- Pero podemos simplemente continuar con el matching que traiamos.
- Simplemente hay que re-revisar las filas de S para detectar donde estan los nuevos ceros y etiquetear las columnas correspondientes para poder proseguir con el algoritmo.

Ceros del matching

extendemos el matching

$$
\begin{array}{cccccc}
 & i & ii & iii & iv & \\
A & \infty & 0 & 4 & [0] & ii \\
B & 0 & [0] & 0 & 1 & iii \\
C & [0] & 2 & 6 & 0 & \\
D & \infty & 1 & [0] & 1 & s \\
 & B & B & D & A &
\end{array}
$$

- Y tenemos un matching perfecto Aiv, Bii, Ci, $Diii$ de costo total 7+5+7+4=23 que por todo lo que hemos visto sabemos que debe ser de suma mínima.

- Enseguida daremos otro ejemplo para fijar las ideas.
- Todavia debemos probar que funciona, es decir, que siempre se detiene.
- Eso lo haremos en la siguiente clase, cuando calculemos su complejidad.
- Por ahora, repasemos como es el algoritmo completo.

Algoritmo Húngaro

1. Restar el mínimo de cada fila.
2. Restar el mínimo de cada columna.
3. Buscar matching perfecto de ceros.
4. Si se encuentra, listo. Si no, habrá un S con $|S| > |\Gamma(S)|$.
    1. Calcular $m =$mínimo de las entradas de la matriz que esten en las filas de S y las columnas del complemento de $\Gamma(S)$.
    2. Restar m de las filas de S.
    3. Sumar m a las columnas de $\Gamma(S)$.
    4. Continuar buscando matching perfecto de ceros en la nueva matriz.

Ejemplo final

- Grafo bipartito con partes $X = \{A, B, ...G\}$, $Y = \{a, b, ..., g\}$.

	a	b	c	d	e	f	g
A	1	2	3	4	1	2	3
B	4	7	9	9	4	9	7
C	7	9	9	9	2	2	9
D	3	3	4	5	3	2	4
E	1	5	7	8	4	1	8
F	2	7	7	7	2	4	7
G	2	7	8	9	5	2	9

- Restamos el mínimo de cada fila:

Ejemplo final

	a	b	c	d	e	f	g
A	0	1	2	3	0	1	2
B	0	3	5	5	0	5	3
C	5	7	7	7	0	0	7
D	1	1	2	3	1	0	2
E	0	4	6	7	3	0	7
F	0	5	5	5	0	2	5
G	0	5	6	7	3	0	7

- Ahora restamos el mínimo de cada columna:

Ejemplo final

	a	b	c	d	e	f	g
A	0	0	0	0	0	1	0
B	0	2	3	2	0	5	1
C	5	6	5	4	0	0	5
D	1	0	0	0	1	0	0
E	0	3	4	4	3	0	5
F	0	4	3	2	0	2	3
G	0	4	4	4	3	0	5

Ejemplo final

$$\begin{array}{c|ccccccc|c}
 & a & b & c & d & e & f & g & \\
A & 0 & 0 & [0] & 0 & 0 & 1 & 0 & a \\
B & 0 & 2 & 3 & 2 & [0] & 5 & 1 & e \\
C & 5 & 6 & 5 & 4 & 0 & [0] & 5 & f \\
D & 1 & [0] & 0 & 0 & 1 & 0 & 0 & \\
E & [0] & 3 & 4 & 4 & 3 & 0 & 5 & s \\
F & 0 & 4 & 3 & 2 & 0 & 2 & 3 & s \\
G & 0 & 4 & 4 & 4 & 3 & 0 & 5 & s \\
 & E & A & A & A & F & E & A &
\end{array}$$

La columna c etiqueteada esta libre, extendemos el matching.

Ejemplo final 2da extension del matching

Extendimos el match en un lado, quedan dos mas, asi que reseteamos las etiquetas y continuamos

.

Ejemplo final 2da extension del matching

	a	b	c	d	e	f	g	
A	0	0	[0]	0	0	1	0	
B	0	2	3	2	[0]	5	1	
C	5	6	5	4	0	[0]	5	
D	1	[0]	0	0	1	0	0	
E	[0]	3	4	4	3	0	5	
F	0	4	3	2	0	2	3	s
G	0	4	4	4	3	0	5	s

.

Ejemplo final 2da extension del matching

$$
\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & 0 & 0 & [0] & 0 & 0 & 1 & 0 & \\
B & 0 & 2 & 3 & 2 & [0] & 5 & 1 & e \\
C & 5 & 6 & 5 & 4 & 0 & [0] & 5 & f \\
D & 1 & [0] & 0 & 0 & 1 & 0 & 0 & \\
E & [0] & 3 & 4 & 4 & 3 & 0 & 5 & a \\
F & 0 & 4 & 3 & 2 & 0 & 2 & 3 & s \\
G & 0 & 4 & 4 & 4 & 3 & 0 & 5 & s \\
 & F & & & & F & G & &
\end{array}
$$

No pudimos etiquetear mas columnas, el algoritmo se detiene (momentaneamente) Debemos “cambiar la matriz”.

Ejemplo final Cambio de matriz

Tenemos que encontrar S con $|S| > |\Gamma(S)|$

Ejemplo final Cambio de matriz

Tenemos que encontrar S con $|S| > |\Gamma(S)|$
Por lo que vimos antes, el S que queremos son las filas etiqueteadas

Ejemplo final Cambio de matriz

Tenemos que encontrar S con $|S| > |\Gamma(S)|$
Por lo que vimos antes, el S que queremos son las filas etiqueteadas

$$\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & 0 & 0 & [0] & 0 & 0 & 1 & 0 & \\
B & 0 & 2 & 3 & 2 & [0] & 5 & 1 & e \\
C & 5 & 6 & 5 & 4 & 0 & [0] & 5 & f \\
D & 1 & [0] & 0 & 0 & 1 & 0 & 0 & \\
E & [0] & 3 & 4 & 4 & 3 & 0 & 5 & a \\
F & 0 & 4 & 3 & 2 & 0 & 2 & 3 & s \\
G & 0 & 4 & 4 & 4 & 3 & 0 & 5 & s \\
 & F & & & & F & G & &
\end{array}$$

$S = \{B, C, E, F, G\}$ Mirando sus vecinos vemos que $\Gamma(S) = \{a, e, f\}$
y vemos que efectivamente $|S| > |\Gamma(S)|$

Ejemplo final Cambio de matriz

- Tenemos que encontrar el mínimo de $S \times \overline{\Gamma(S)}$.
- En nuestro caso seria de $\{B, C, E, F, G\} \times \{b, c, d, g\}$.
- Para ver bien lo que hacemos a mano, deberiamos "tachar" todos los elementos que no estan ahi.
- Haciendolo a mano esto es relativamente fácil: tachamos las filas que no estan en S y las columnas que si estan en $\Gamma(S)$.
- Es decir, tachamos las filas no etiqueteadas y las columnas que si estan etiqueteadas.
- Quedaria lo siguiente:

Ejemplo final Cambio de matriz: encontrando m

	a	b	c	d	e	f	g	
A	0	0	[0]	0	0	1	0	
B	0	2	3	2	[0]	5	1	e
C	5	6	5	4	0	[0]	5	f
D	1	[0]	0	0	1	0	0	
E	[0]	3	4	4	3	0	5	a
F	0	4	3	2	0	2	3	s
G	0	4	4	4	3	0	5	s
	F				F	G		

Ejemplo final Cambio de matriz: encontrando m

$$
\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & * & * & * & * & * & * & * & \\
B & * & 2 & 3 & 2 & * & * & 1 & e \\
C & * & 6 & 5 & 4 & * & * & 5 & f \\
D & * & * & * & * & * & * & * & \\
E & * & 3 & 4 & 4 & * & * & 5 & a \\
F & * & 4 & 3 & 2 & * & * & 3 & s \\
G & * & 4 & 4 & 4 & * & * & 5 & s \\
 & F & & & & F & G & &
\end{array}
$$

Asi que el mínimo que buscamos es $m = 1$.

Ejemplo final Cambio de matriz, restando y sumando m

- Tenemos que restar $m = 1$ a S y sumarle lo mismo a $\Gamma(S)$.
- Observemos que a los elementos de:
  1. $S \times \overline{\Gamma(S)}$: se les resta m, no se les suma nada.
  2. $S \times \Gamma(S)$: se les resta y suma m, asi que quedan como están.
  3. $\overline{S} \times \overline{\Gamma(S)}$ no se les resta ni suma nada, quedan iguales.
  4. $\overline{S} \times \Gamma(S)$: no se les resta nada, se les suma m
- Asi que es relativamente fácil cambiar la matriz.
  1. Restar m de $S \times \overline{\Gamma(S)}$
  2. Sumar m a $\overline{S} \times \Gamma(S)$.

Ejemplo final Cambio de matriz, restando y sumando m

Sumar $m = 1$ a $\overline{S} \times \Gamma(S)$

$$
\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & 1 & 0 & [0] & 0 & 1 & 2 & 0 & \\
B & 0 & 1 & 2 & 1 & [0] & 5 & 0 & e \\
C & 5 & 5 & 4 & 3 & 0 & [0] & 4 & f \\
D & 2 & [0] & 0 & 0 & 2 & 1 & 0 & \\
E & [0] & 2 & 3 & 3 & 3 & 0 & 4 & a \\
F & 0 & 3 & 2 & 1 & 0 & 2 & 2 & s \\
G & 0 & 3 & 3 & 3 & 3 & 0 & 4 & s \\
 & F & & & & F & G & &
\end{array}
$$

Ejemplo final Cambio de matriz, restando y sumando m

La columna esta libre, asi que extendemos el matching mirando las etiquetas

	a	b	c	d	e	f	g	
A	1	0	[0]	0	1	2	0	
B	0	1	2	1	0	5	[0]	e
C	5	5	4	3	0	[0]	4	f
D	2	[0]	0	0	2	1	0	
E	[0]	2	3	3	3	0	4	a
F	0	3	2	1	[0]	2	2	s
G	0	3	3	3	3	0	4	s
	F				F	G	B	

Ejemplo final Cambio de matriz, restando y sumando m

La columna esta libre, asi que extendemos el matching mirando las etiquetas

$$
\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & 1 & 0 & [0] & 0 & 1 & 2 & 0 & \\
B & 0 & 1 & 2 & 1 & 0 & 5 & [0] & e \\
C & 5 & 5 & 4 & 3 & 0 & [0] & 4 & f \\
D & 2 & [0] & 0 & 0 & 2 & 1 & 0 & \\
E & [0] & 2 & 3 & 3 & 3 & 0 & 4 & a \\
F & 0 & 3 & 2 & 1 & [0] & 2 & 2 & s \\
G & 0 & 3 & 3 & 3 & 3 & 0 & 4 & s \\
 & F & & & & F & G & B &
\end{array}
$$

Queda una sola fila sin matchear, pero antes de seguir, una discusión teórica.

No se pierden ceros del matching

- ¿Cómo estamos seguros de que nunca perderemos ningún cero del matching con el cual veniamos?
- Pues los ceros que se pueden perder al cambiar la matriz estan todos en las filas de $\overline{S}$ con columnas $\Gamma(S)$, y ninguno de esos lugares está en el matching.
- ¿Por qué? Pues porque si hubiera algún vértice $y \in \Gamma(S)$ que esté en el matching, digamos matcheado con $x \in X$, entonces y habría puesto a x en la cola, y x estaría en S, no en $\overline{S}$.
- Asi que lo que pasa con nuestro ejemplo pasará siempre.

Ejemplo final, ultima extensión del matching

Tachamos filas no etiqueteadas y columnas etiqueteadas

	a	b	c	d	e	f	g	
A	1	0	[0]	0	1	2	0	
B	0	1	2	1	0	5	[0]	
C	5	5	4	3	0	[0]	4	f
D	2	[0]	0	0	2	1	0	
E	[0]	2	3	3	3	0	4	a
F	0	3	2	1	[0]	2	2	e
G	0	3	3	3	3	0	4	s
	G				C	G		

Ejemplo final, ultima extensión del matching

Tachamos filas no etiqueteadas y columnas etiqueteadas

$$\begin{array}{cccccccccc}
 & a & b & c & d & e & f & g & \\
A & * & * & * & * & * & * & * & \\
B & * & * & * & * & * & * & * & \\
C & * & 5 & 4 & 3 & * & * & 4 & f \\
D & * & * & * & * & * & * & * & \\
E & * & 2 & 3 & 3 & * & * & 4 & a \\
F & * & 3 & 2 & 1 & * & * & 2 & e \\
G & * & 3 & 3 & 3 & * & * & 4 & s \\
 & G & & & & C & G & &
\end{array}$$

$m = 1$ otra vez

Ejemplo final, ultima extensión del matching

La columna d recientemente etiqueteada está libre, podemos extender el matching

	a	b	c	d	e	f	g	
A	2	0	[0]	0	2	3	0	
B	1	1	2	1	1	6	[0]	
C	5	4	3	2	[0]	0	3	f
D	3	[0]	0	0	3	2	0	
E	[0]	1	2	2	3	0	3	a
F	0	2	1	[0]	0	2	1	e
G	0	2	2	2	3	[0]	3	s
	G			F	C	G		

Ejemplo final, ultima extensión del matching

Matching queda:

	a	b	c	d	e	f	g	
A	2	0	[0]	0	2	3	0	Ac
B	1	1	2	1	1	6	[0]	Bg
C	5	4	3	2	[0]	0	3	Ce
D	3	[0]	0	0	3	2	0	Db
E	[0]	1	2	2	3	0	3	Ea
F	0	2	1	[0]	0	2	1	Fd
G	0	2	2	2	3	[0]	3	Gf

Ejemplo final, ultima extensión del matching

Mirando la matriz original:

	a	b	c	d	e	f	g	
A	1	2	[3]	4	1	2	3	Ac : 3
B	4	7	9	9	4	9	[7]	Bg : 7
C	7	9	9	9	[2]	2	9	Ce : 2
D	3	[3]	4	5	3	2	4	Db : 3
E	[1]	5	7	8	4	1	8	Ea : 1
F	2	7	7	[7]	2	4	7	Fd : 7
G	2	7	8	9	5	[2]	9	Gf : 2

suma mínima:25

Comentarios

- En este ejemplo, teniamos 3 lados sin matchear luego del matching inicial
  1. El primero lo logramos matchear con el algoritmo simple de extender el matching.
  2. El segundo tuvimos que cambiar la matriz.
  3. El tercero no alcanzó con el primer cambio y tuvimos que volver a cambiar la matriz
- Pero tanto [2] como [3] al cambiar la matriz pudimos extender el matching en al menos un lado.
- Esto NO SIEMPRE va a pasar.

Comentarios

- Es decir, podemos cambiar la matriz y aun asi no poder extender el matching ni en un lado.
- Esto podria haber pasado en el ejemplo con algún cambio pequeño que hiciera que la fila F estuviera matcheada, de forma tal que la única que quedara sin matchear luego de [1] fuera la G, y al hacer primer cambio de matriz no hubieramos podido extender el matching ni siquiera en un lado.
- Asi que puede ser que luego de un cambio de matriz debamos tener que volver a cambiar la matriz, una o mas veces extras hasta poder extender el matching en un lado.
- La clase que viene veremos que eso puedo pasar a lo sumo n veces, y por eso el algoritmo termina.