El problema de pre-asignación

Cinvestav

Marcos César Vargas Magaña
Estudiante de doctorado
Departamento de Matemáticas
CINVESTAV

Index

Introducción

$O(n\,C \cdot VC(n,m))$	Egerváry (1931)
$O(2^n\, n^2)$	Easterfield (1946)
$O(n\,C \cdot DC(n,m,C))$	Robinson (1949)
$O(n^4)$	Kuhn (1955), Hungarian method
$O(n^2\, m)$	Iri (1960)
$O(n^3)$	Dinits and Kronrod (1969)
$O(n \cdot SP_+(n,m,C))$	Edmonds and Karp (1970), Tomizawa (1971)
$O(n^{3/4}\, m \log C)$	Gabow (1985)
$O(\sqrt{n}\, m \log(nC))$	Gabow and Tarjan (1989) (cf. Orlin and Ahuja (1992))
$O(\sqrt{n}\, m\, C)$	Kao, Lam, Sung, and Ting (1999)
$O(\sqrt{n}\, m\, C \log_n (n^2/m))$	Kao, Lam, Sung, and Ting (2001)

Figura : Análisis de complejidad de algunos algritmos que resuelven el problema de asignación. Donde n es el número de vértices y m es el número de aristas, C es el máximo de los valores absolutos de los costos, $VC(n,m)$ es la complejidad en tiempo de encontrar un cubierta por vértices mínima en una gráfica bipartita, $DC(n,m,C)$ es la complejidad en tiempo de encontrar un circuito dirigido en una gráfica dirigida y $SP_+(n,m,C)$ es la complejidad en tiempo encontrar un camino más corto en una gráfica dirigida con costos enteros no negativos.

Algoritmos que veremos

1. **El algoritmo de subastas**. Originado en 1979, es intuitivo y fácil de entender, ha sido **extendido a otros problemas** de flujos en redes tales como el problema de transporte, el problema del camino más corto, el problema del flujo máximo, etc. Una de las ventajas del algoritmo es que puede ser implementado en **entornos paralelos**. Su **complejidad** computacional en tiempo es $\mathbf{O(n\,m}\log(\mathbf{nC}))$, donde n es el número de vertices, m el número de aristas y C el costo absoluto máximo.
   1. *Algoritmo de subastas ingenuo*.
   2. *Algoritmo de subastas*.
   3. *Algoritmo de subastas con ϵ-escalamiento*.
2. El Algoritmo fué **mejorado** y alcanza una complejidad de $O(\sqrt{n}\,m\log(nC))$.

Algoritmos que veremos

1. **El algoritmo de subastas**. Originado en 1979, es intuitivo y fácil de entender, ha sido **extendido a otros problemas** de flujos en redes tales como el problema de transporte, el problema del camino más corto, el problema del flujo máximo, etc. Una de las ventajas del algoritmo es que puede ser implementado en **entornos paralelos**. Su **complejidad** computacional en tiempo es $\mathbf{O(n\,m\,\log(nC))}$, donde n es el número de vertices, m el número de aristas y C el costo absoluto máximo.

    1. *Algoritmo de subastas ingenuo*.
    2. *Algoritmo de subastas*.
    3. *Algoritmo de subastas con ϵ-escalamiento*.

2. El Algoritmo fué **mejorado** y alcanza una complejidad de $O(\sqrt{n}\,m\,\log(nC))$.

Estadísticas

Las estadísticas que presentaremos fueron obtenidas de instancias aleatorias con distribucion uniforme, en una máquina con las siguientes características:

1. Procesador: 2.5 GHz Intel Core i7
2. Memoria: 8 GB 1333 MHz DDR3
3. Sistema Operativo: Mac OS X Lion 10.7

Gráficas bipartitas

$G = (X \cup Y, E)$ Denotará una gráfica bipartita.

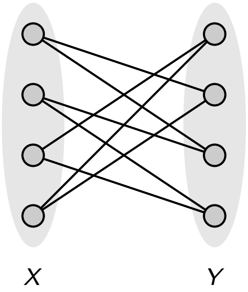


FIGURA : Una gráfica bipartita.

Emparejamientos

Sea $G = (X \cup Y, E)$ una gráfica bipartita.

1. Un conjunto de aristas $M \subseteq E$ que son disjuntas por pares es llamado **emparejamiento** o *asignación*.
2. Sea $c : E \to \mathbb{R}$ una *función de costos* sobre las aristas de G. El **costo** de un emparejamiento M está dado por
$$cost(M) = \sum_{e \in M} c(e). \tag{1}$$
3. Si $n = |X| = |Y|$, entonces un emparejamiento M de G se dice que es **perfecto** si cubre todos los vértices, es decir, $|M| = n$.

Emparejamientos

Sea $G = (X \cup Y, E)$ una gráfica bipartita.

1. Un conjunto de aristas $M \subseteq E$ que son disjuntas por pares es llamado **emparejamiento** o *asignación*.
2. Sea $c : E \to \mathbb{R}$ una *función de costos* sobre las aristas de G. El **costo** de un emparejamiento M está dado por

$$cost(M) = \sum_{e \in M} c(e). \tag{1}$$

3. Si $n = |X| = |Y|$, entonces un emparejamiento M de G se dice que es **perfecto** si cubre todos los vértices, es decir, $|M| = n$.

EMPAREJAMIENTOS

Sea $G = (X \cup Y, E)$ una gráfica bipartita.

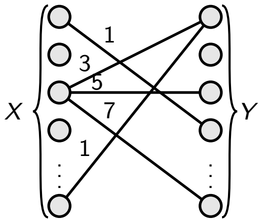
1. Un conjunto de aristas $M \subseteq E$ que son disjuntas por pares es llamado **emparejamiento** o *asignación*.
2. Sea $c : E \to \mathbb{R}$ una *función de costos* sobre las aristas de G. El **costo** de un emparejamiento M está dado por

$$cost(M) = \sum_{e \in M} c(e). \tag{1}$$

3. Si $n = |X| = |Y|$, entonces un emparejamiento M de G se dice que es **perfecto** si cubre todos los vértices, es decir, $|M| = n$.

El problema de asignación

1. Desde el punto de vista de teoría de gráficas, **una instancia del problema de asignación** consiste en una gráfica bipartita $G = (X \cup Y, E)$ y una *función de costos* $c : E \to \mathbb{R}$.
2. Si $|X| = |Y| = n$ entonces se dice que la instancia es **simétrica** o *balanceada*, de otra forma se dice que es **asimétrica** o *no balanceada*.
3. Si G tiene al menos un emparejamiento perfecto entonces la instancia es **factible**, de otra forma es **infactible**.
4. **El problema de asignación consiste** en encontrar un emparejamiento perfecto de costo **óptimo** (mínimo para nuestro caso) en una **instancia simétrica**.



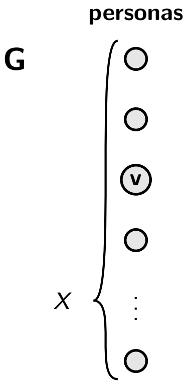


Figura : Una instancia del problema de asignación con **las personas**.

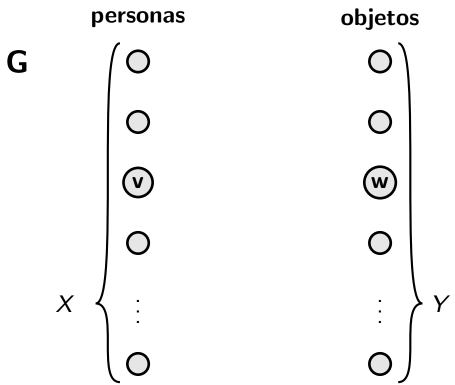


Figura : Una instancia del problema de asignación con **las personas, los objetos**.

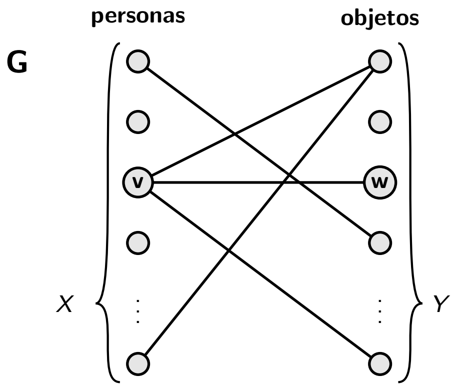


Figura : Una instancia del problema de asignación con **las personas, los objetos**.

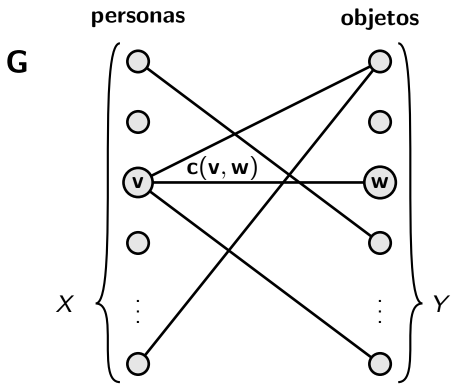


Figura : Una instancia del problema de asignación con **las personas, los objetos, los costos**.

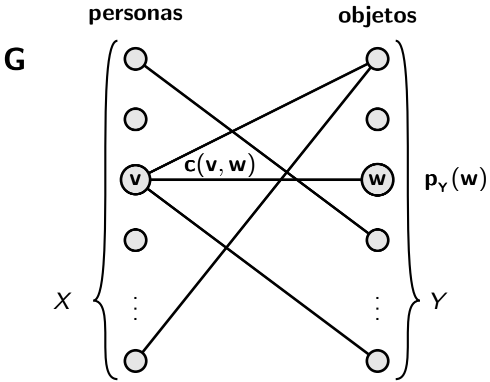


FIGURA : Una instancia del problema de asignación con **las personas, los objetos, los costos, los precios**.

Bases del algoritmo de subastas

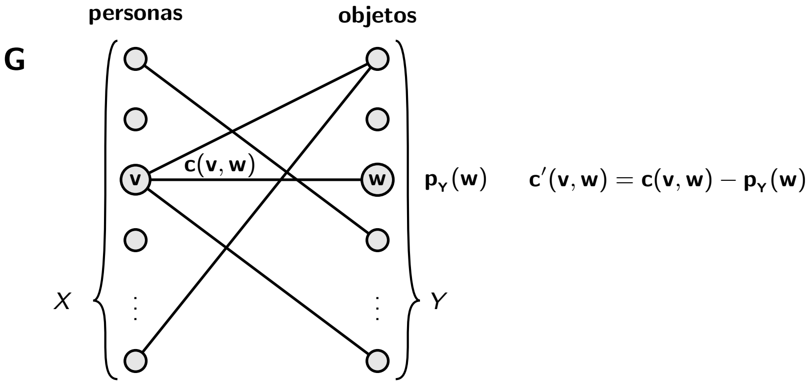
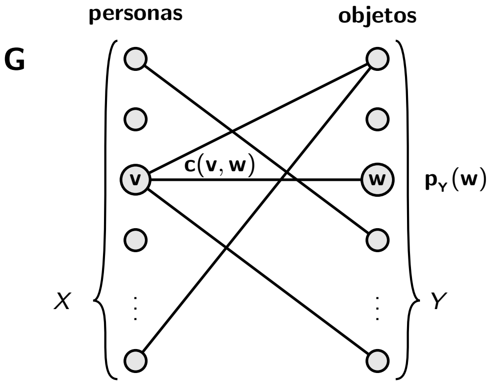


FIGURA : Una instancia del problema de asignación con **las personas, los objetos, los costos, los precios y las respectivas pérdidas**.

Bases del algoritmo de subastas



$$c'(v,w) = c(v,w) - p_Y(w)$$

$$c'(v,w) = \min_{vz \in E}\{c'(v,z)\}$$

Figura : Una instancia del problema de asignación con **las personas, los objetos, los costos, los precios y las respectivas pérdidas**.

Algoritmo de subastas ingenuo

Un emparejamiento $M \subseteq E$ y un vector de precios $p_Y : Y \to \mathbb{R}$ satisfacen la condición de **holgura complementaria** (*condición HC*), si

$$c'(v,w) = \min_{vz \in E} \{c'(v,z)\} \quad \forall vw \in M. \tag{2}$$

Teorema

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ una instancia simétrica del problema de asignación. **Si M es un emparejamiento perfecto que satisface la condición HC con algún vector de precios $p_Y : Y \to \mathbb{R}$, entonces M es de costo mínimo.**

Algoritmo de subastas ingenuo

Un emparejamiento $M \subseteq E$ y un vector de precios $p_Y : Y \to \mathbb{R}$ satisfacen la condición de **holgura complementaria** (*condición HC*), si

$$c'(v,w) = \min_{vz \in E} \{c'(v,z)\} \quad \forall vw \in M. \tag{2}$$

Teorema

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ una instancia simétrica del problema de asignación. **Si M es un emparejamiento perfecto que satisface la condición HC con algún vector de precios $p_Y : Y \to \mathbb{R}$, entonces M es de costo mínimo.**

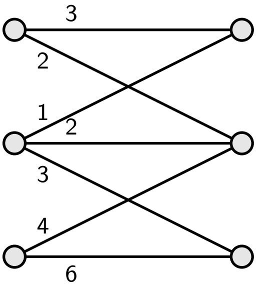
El pseudocódigo del algoritmo de subastas ingenuo

El algoritmo **inicia con un emparejamiento vacio** M **y un vector de precios arbitrario** p_Y. A continuación ejecutamos los siguientes pasos en ciclo mientras que el emparejamiento no cubra todos los vértices.

1. **Tomar una persona** v **no asignada** bajo el emparejamiento.
2. **Buscar los objetos** w y z que le den la menor y la segunda menor pérdida a v, respectivamente.
3. **Si existe** una persona v' asignada al objeto w entonces **desasignamos** a v', quitando $v'w$ del emparejamiento.
4. **Asignamos** v a w, poniendo vw en el emparejamiento.
5. **Calculamos** el valor de decremento $\gamma = c'(v,z) - c'(v,w) \geq 0$.
6. **Decrementamos el precio** de w: $p_Y(w) = p_Y(w) - \gamma$.

Ejemplo

p_Y $\qquad$ $c'(v, w)$

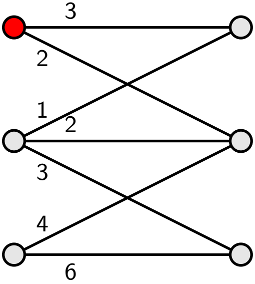


0

0

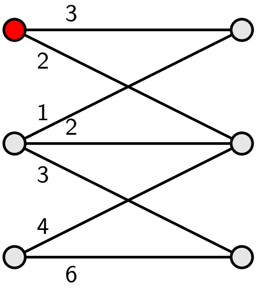
0

Ejemplo



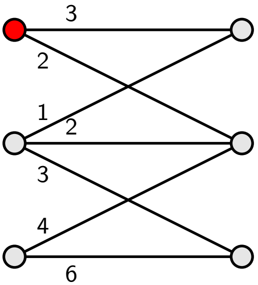
p_Y	$c'(v,w)$
0	**3**
0	**2**
0	

Ejemplo



p_Y	$c'(v,w)$
0	3
0	2
0	

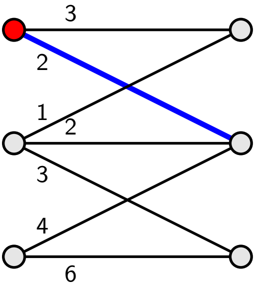
Ejemplo



p_Y	$c'(v, w)$
0	3
0	2
0	

$$\gamma = \mathbf{3} - \mathbf{2} = \mathbf{1}$$

Ejemplo

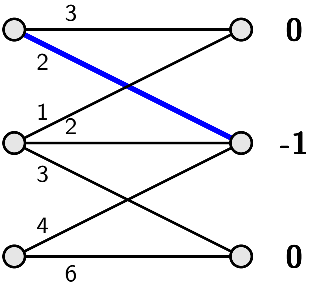


	p_Y	$c'(v,w)$
	$\mathbf{0}$	3
	$\mathbf{0}$	2
	$\mathbf{0}$	

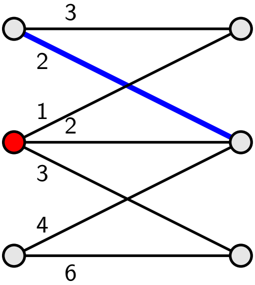
$$\gamma = \mathbf{3} - \mathbf{2} = \mathbf{1}$$

Ejemplo

p_Y $\qquad$ $c'(v, w)$

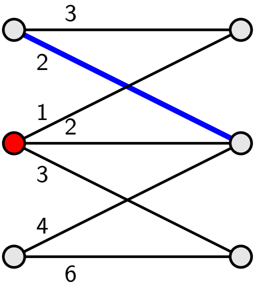


Ejemplo



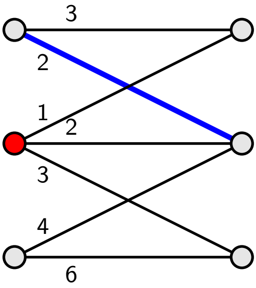
p_Y	$c'(v, w)$
0	**1**
-1	**3**
0	**3**

Ejemplo



p_Y	$c'(v,w)$
0	**1**
-1	**3**
0	**3**

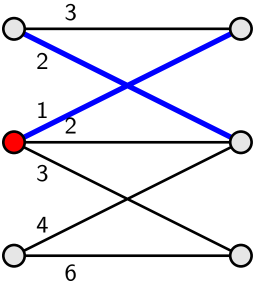
Ejemplo



p_Y	$c'(v,w)$
0	**1**
-1	**3**
0	**3**

$$\gamma = \mathbf{3} - \mathbf{1} = \mathbf{2}$$

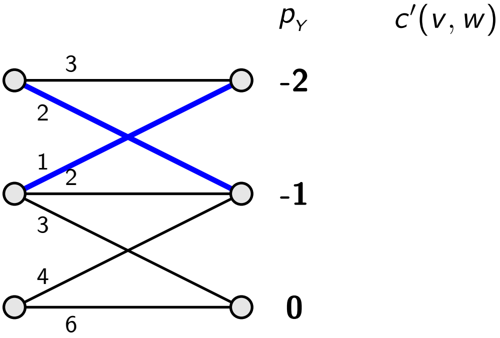
Ejemplo



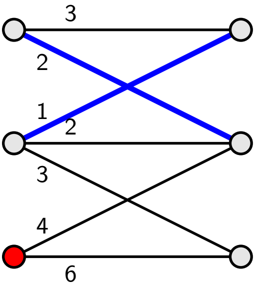
p_Y	$c'(v,w)$
0	**1**
-1	**3**
0	**3**

$$\gamma = \mathbf{3} - \mathbf{1} = \mathbf{2}$$

EJEMPLO

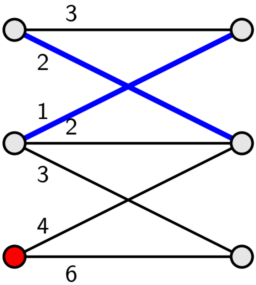


Ejemplo



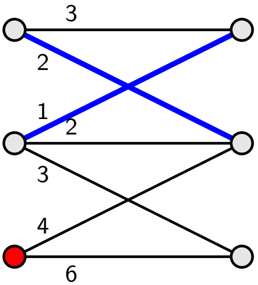
p_Y	$c'(v, w)$
-2	
-1	**5**
0	**6**

Ejemplo



	p_Y	$c'(v, w)$
	-2	
	-1	**5**
	0	**6**

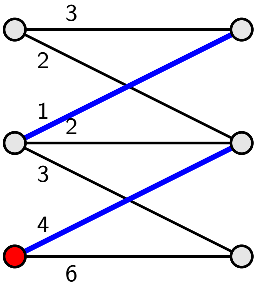
Ejemplo



p_Y	$c'(v,w)$
-2	
-1	**5**
0	**6**

$$\gamma = \mathbf{6} - \mathbf{5} = \mathbf{1}$$

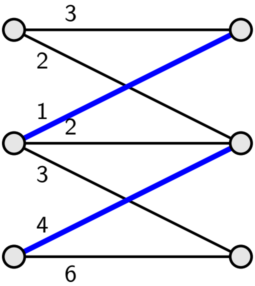
Ejemplo



p_Y	$c'(v,w)$
-2	
-1	**5**
0	**6**

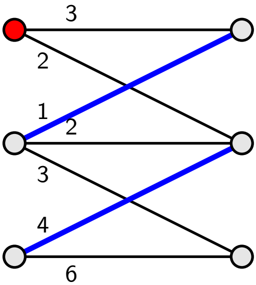
$$\gamma = \mathbf{6} - \mathbf{5} = \mathbf{1}$$

Ejemplo



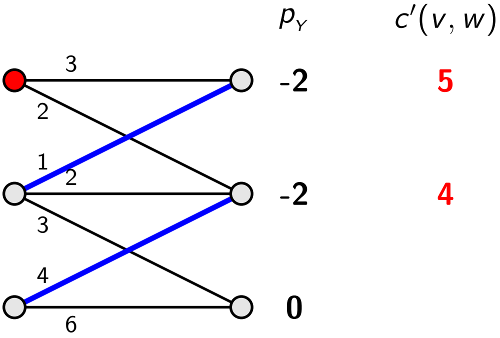
	p_Y	$c'(v, w)$
	-2	
	-2	
	0	

Ejemplo

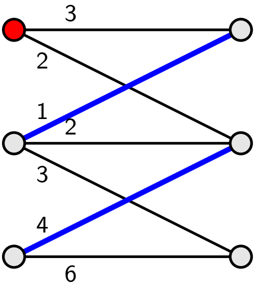


	p_Y	$c'(v, w)$
	-2	**5**
	-2	**4**
	0	

Ejemplo



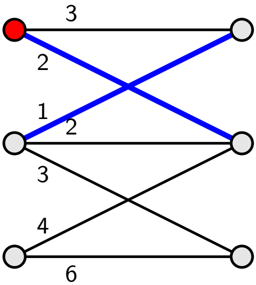
Ejemplo



p_Y	$c'(v,w)$
-2	5
-2	4
0	

$$\gamma = \mathbf{5} - \mathbf{4} = \mathbf{1}$$

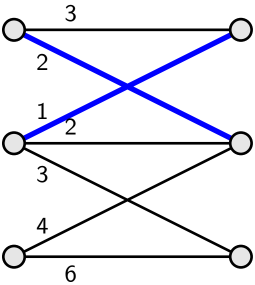
Ejemplo



	p_Y	$c'(v,w)$
	-2	**5**
	-2	**4**
	0	

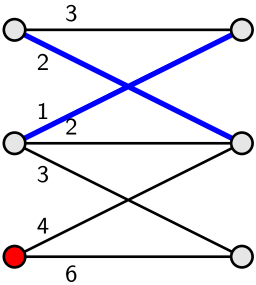
$$\gamma = \mathbf{5} - \mathbf{4} = \mathbf{1}$$

Ejemplo



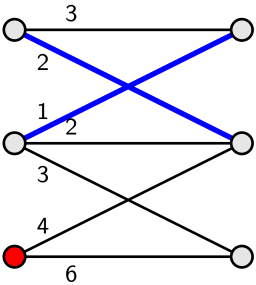
p_Y	$c'(v, w)$
-2
-3
0

Ejemplo



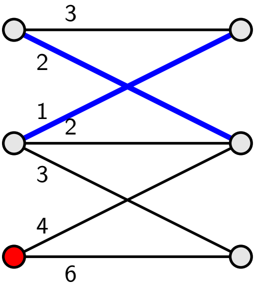
	p_Y	$c'(v, w)$
	-2	
	-3	**7**
	0	**6**

Ejemplo



	p_Y	$c'(v, w)$
	-2	
	-3	7
	0	6

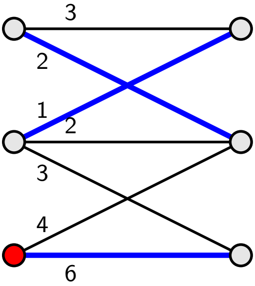
Ejemplo



p_Y	$c'(v, w)$
-2	
-3	**7**
0	**6**

$$\gamma = \mathbf{7} - \mathbf{6} = \mathbf{1}$$

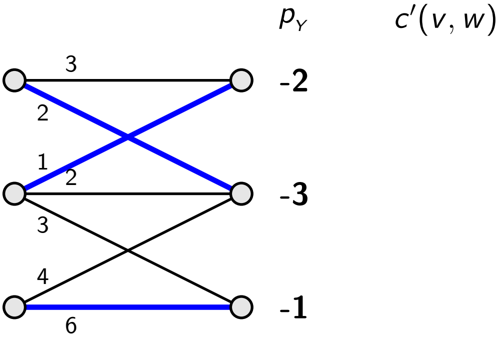
EJEMPLO



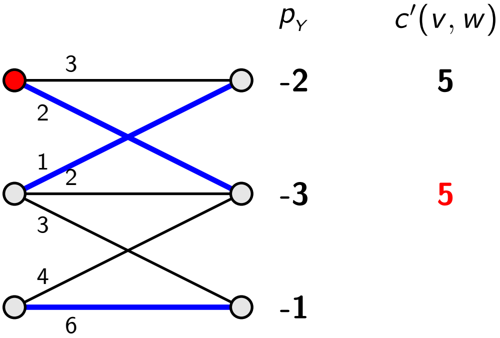
p_Y	$c'(v,w)$
-2	
-3	**7**
0	**6**

$$\gamma = \mathbf{7} - \mathbf{6} = \mathbf{1}$$

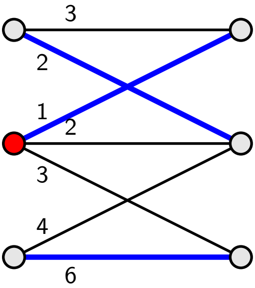
Ejemplo



Ejemplo

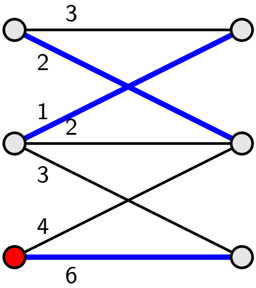


Ejemplo



	p_Y	$c'(v,w)$
	-2	**3**
	-3	**5**
	-1	**4**

Ejemplo



	p_Y	$c'(v,w)$
	-2	
	-3	**7**
	-1	**7**

¿Por qué funciona?

1. El siguiente **invarainte** se matiene en cada iteración del ciclo: "El emparejamiento M y el vector de precios p_Y satisfacen la condición HC".

1. Tomar una persona v no cubierta por el emparejamiento.
2. Sean w y z los objetos que le den la menor y la segunda menor pérdida a v, respectivamente.
3. if(existe una persona v' que esté asignada al objeto w)
4. desasignar a v'.
5. Asignamos v con w.
6. $\gamma = c'(v,z) - c'(v,w)$
7. $p_Y(w) = p_Y(w) - \gamma$

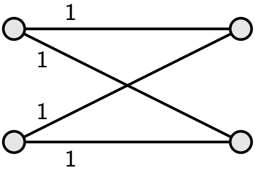
1. Explicar en el pizarrón.

Teorema

Si el algoritmo de subastas ingenuo termina *entonces regresará un emparejamiento perfecto de costo mínimo.*

¿Por qué es **INGENUO** este algoritmo?

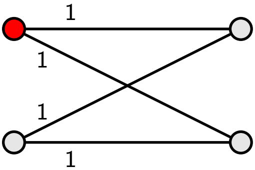
p_Y $\qquad$ $c'(v, w)$



0

0

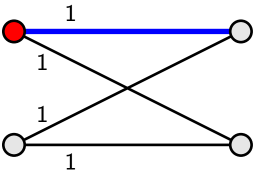
¿Por qué es **INGENUO** este algoritmo?



p_Y	$c'(v,w)$
$\mathbf{0}$	$\mathbf{1}$
$\mathbf{0}$	$\mathbf{1}$

$$\gamma = \mathbf{1} - \mathbf{1} = \mathbf{0}$$

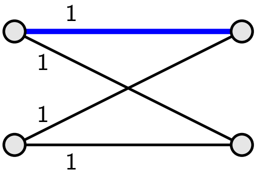
¿Por qué es **INGENUO** este algoritmo?



	p_Y	$c'(v,w)$
	0	**1**
	0	**1**

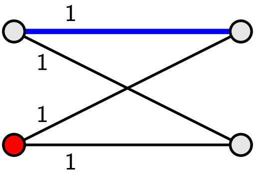
$$\gamma = \mathbf{1} - \mathbf{1} = \mathbf{0}$$

¿Por qué es **INGENUO** este algoritmo?



	p_Y	$c'(v, w)$
	0	
	0	

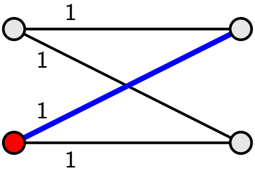
¿Por qué es **INGENUO** este algoritmo?



	p_Y	$c'(v,w)$
	$\mathbf{0}$	$\mathbf{1}$
	$\mathbf{0}$	$\mathbf{1}$

$$\gamma = \mathbf{1} - \mathbf{1} = \mathbf{0}$$

¿Por qué es **INGENUO** este algoritmo?

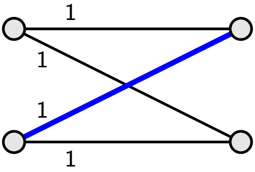


	p_Y	$c'(v, w)$
	$\mathbf{0}$	$\mathbf{1}$
	$\mathbf{0}$	$\mathbf{1}$

$$\gamma = \mathbf{1} - \mathbf{1} = \mathbf{0}$$

¿Por qué es **INGENUO** este algoritmo?

p_Y $\quad c'(v, w)$

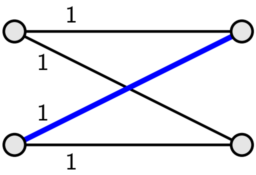


0

0

¿Por qué es **INGENUO** este algoritmo?

p_Y $\qquad$ $c'(v, w)$



0

0

Y así sucesivamente...

Algoritmo de subastas

Sea $\epsilon > 0$. Un emparejamiento $M \subseteq E$ y un vector de precios $p_Y : Y \to \mathbb{R}$ satisfacen la condición de ϵ-**holgura complementaria** (condición ϵ-HC), si

$$c'(v,w) \leq \min_{vz \in E} \{c'(v,z)\} + \epsilon \quad \forall vw \in M. \tag{3}$$

Un emparejamiento perfecto que satisface la condición ϵ-HC con algún vector de precios se dice que es un **emparejamiento ϵ-óptimo**.

Teorema

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ una instancia simétrica del problema de asignación con $n = |X| = |Y|$, $S \subseteq E$ un emparejamiento perfecto de costo mínimo y $\epsilon > 0$. **Si $M \subseteq E$ es un emparejamiento perfecto que satistace la condición ϵ-HC con algún vector de precios p_Y, entonces**

$$cost(M) \leq cost(S) + n\epsilon.$$

Algoritmo de subastas

Sea $\epsilon > 0$. Un emparejamiento $M \subseteq E$ y un vector de precios $p_Y : Y \to \mathbb{R}$ satisfacen la condición de ϵ-**holgura complementaria** (condición ϵ-HC), si

$$c'(v, w) \leq \min_{vz \in E} \{c'(v, z)\} + \epsilon \quad \forall vw \in M. \tag{3}$$

Un emparejamiento perfecto que satisface la condición ϵ-HC con algún vector de precios se dice que es un **emparejamiento ϵ-óptimo**.

Teorema

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ *una instancia simétrica del problema de asignación con* $n = |X| = |Y|$, $S \subseteq E$ *un emparejamiento perfecto de costo mínimo y* $\epsilon > 0$. **Si $M \subseteq E$ es un emparejamiento perfecto que satistace la condición ϵ-HC con algún vector de precios p_Y, entonces**

$$cost(M) \leq cost(S) + n\epsilon.$$

Corolario

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ una instancia simétrica del problema de asignación con $n = |X| = |Y|$ y sea $\epsilon < 1/n$. **Si M es un emparejamiento perfecto que satisface la condición ϵ-HC con algún vector de precios, entonces M es un emparejamiento perfecto de costo mínimo**.

Demostración.

De acuerdo al teorema, **si S es de costo mínimo**, entonces

$$cost(S) \leq cost(M) \leq cost(S) + n\epsilon < cost(S) + 1,$$

entonces $cost(S) = cost(M)$.

procedure min_subastas(ϵ, M, p_Y**)**

Primero el algoritmo **deshecha el emparejamiento** M. A continuación el algoritmo ejecuta los siguientes pasos en ciclo, mientras que el emparejamiento no cubra todos los vértices.

1. Tomar una persona v no cubierta por el emparejamiento.
2. Buscar los objetos w y z que le den la menor y la segunda menor pérdida a v, respectivamente.
3. Si existe una persona v' asignada al objeto w entonces desasignamos a v', quitando $v'w$ del emparejamiento.
4. Asignamos v a w, poniendo vw en el emparejamiento.
5. Calculamos el valor de decremento $\gamma = c'(v, z) - c'(v, w) \geq 0$.
6. Decrementamos el precio de w: $\boxed{p_Y(w) = p_Y(w) - \gamma - \epsilon}$.

El algoritmo termina regresando el emparejamiento M y el vector de precios p_Y.

TEOREMA

Si el procedimiento **min_subastas** *es aplicado a una instancia simétrica factible, entonces el procedimiento* **terminará** *y regresará un emparejamiento* ϵ*-optimo.*

1. Podemos correr el algoritmo directamente con $\epsilon = 1/(n+1)$ y así obtener un emparejamiento óptimo.

¿Por qué funciona?

Además de que el algoritmo matiene la siguiente invariante en el ciclo: "El emparejamiento M y el vector de precios p_Y satisfacen la condición ϵ-HC", tenemos

1. Cuando un objeto decrementa su precio, lo hace por al menos $\epsilon > 0$.
2. Sólo objetos que ya están asignados decrementan su precio.
3. Por lo tanto llegará el momento en el que todos los objetos que ya están asignados no serán atractivos para alguna persona de las que no están asignadas.
4. Entoces el emparejamiento irá incrementando su cardinalidad, uno a uno, hasta ser perfecto.

El problema

Guerras de precios

1. Mientras **más grande** sea ϵ, **menos tardará** la instancia en ser resuelta, pero el emparejamiento estará probablemente muy **lejos de ser óptimo**.
2. Mientras **más chico** sea ϵ, **más tardará** la instancia en ser resuelta, pero el emparejamiento estará muy **cerca de ser óptimo** (óptimo si $\epsilon < 1/n$).

La solución

El ϵ-escalamiento

1. Los **precios influyen** en el tiempo que tarda en resolverse una instancia.

El problema

Guerras de precios

1. Mientras **más grande** sea ϵ, **menos tardará** la instancia en ser resuelta, pero el emparejamiento estará probablemente muy **lejos de ser óptimo**.
2. Mientras **más chico** sea ϵ, **más tardará** la instancia en ser resuelta, pero el emparejamiento estará muy **cerca de ser óptimo** (óptimo si $\epsilon < 1/n$).

La solución

El ϵ-escalamiento

1. Los **precios influyen** en el tiempo que tarda en resolverse una instancia.

Sea $\{G = (X \cup Y, E),\ c : E \to \mathbb{Z}\}$ una instancia simétrica factible. Asumiendo que $C = \max_{vw \in E}\{|c(v, w)|\}$, $\alpha > 1$ y $\epsilon_f > 0$, el algoritmo está dado como sigue.

procedure min_epsilon_escalamiento(G, c, ϵ_f)

$\epsilon = C$

$\forall w \in Y,\ p_Y(w) = 0$

while$(\epsilon \geq \epsilon_f)$ do

    $\epsilon = \epsilon/\alpha$

    (M, p_Y)=min_subastas(ϵ, M, p_Y)

end

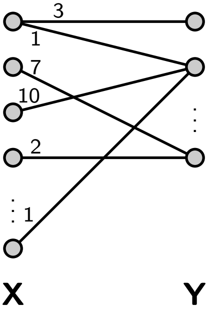
return M

Teorema

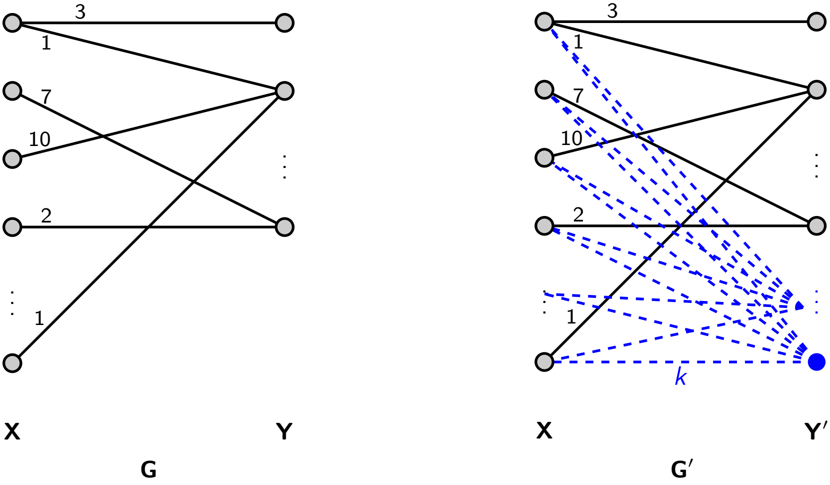
Si el procedimiento **min_epsilon_escalamiento** *es aplicado a una instancia simétrica factible entonces el procedimiento regresará un emparejamiento ϵ_f-óptimo.*

Como consecuencia, min_epsilon_escalamiento$(G, c, 1/n)$ regresará un emparejamiento óptimo.

1. El problema con los algoritmos que hemos presentado es que funcionan **solo con instancias simétricas**.
2. Si $n = |X| > |Y| = s$, ¿como encontramos un **emparejamiendo perfecto de costo óptimo**? asumiendo que existe.
3. La manera común de proceder es usando **técnicas de transformación**.
4. Aquí mostraremos dos transformaciones conocidas y **dos transformación nueva**.



Transformación de vértices artificiales

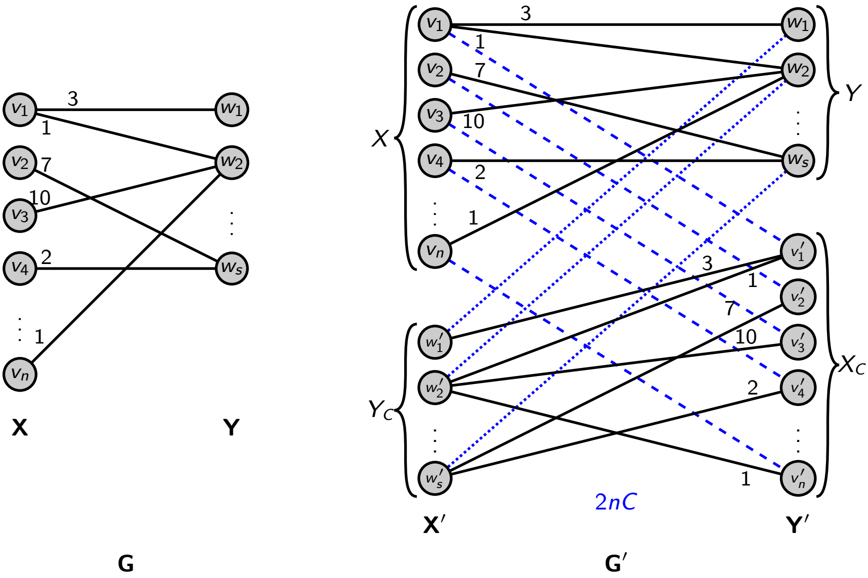


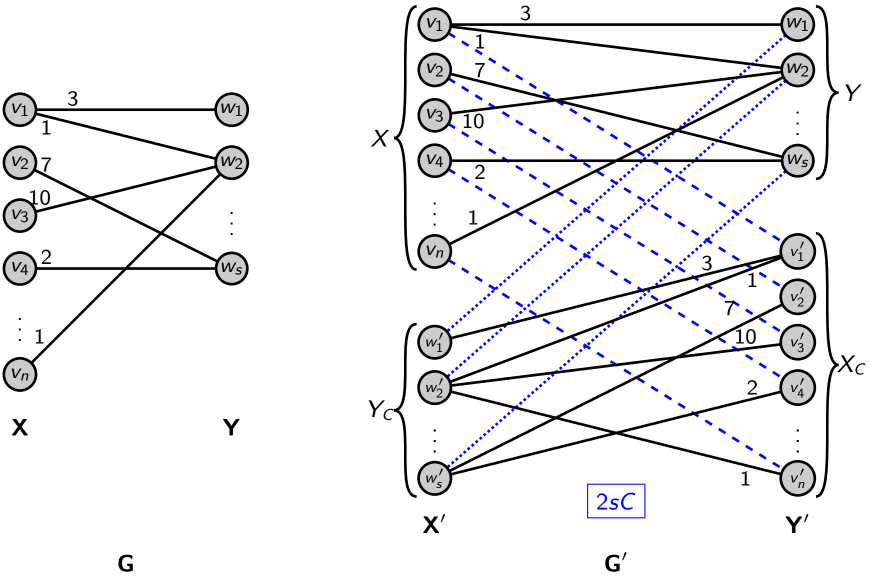
Teorema

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ *una instancia factible asimétrica y* $\{G', c'\}$ *su transformación de vértices artificiales.* **Si** M' **es un emparejamiento perfecto de costo óptimo** $\{G', c'\}$**, entonces** $M = M' \cap E$ **es un emparejamiento perfecto de un lado de costo óptimo en** $\{G, c\}$.

1. **Ventajas y desventajas**.

Primera transformación de duplicación




v_1
v_2
v_3
v_4
v_n
w_1
w_2
w_s
3
1
7
10
2
1
X
Y
G
X
Y
Y_C
X_C
w'_1
w'_2
w'_s
v'_1
v'_2
v'_3
v'_4
v'_n
$2sC$
X′
Y′
G′

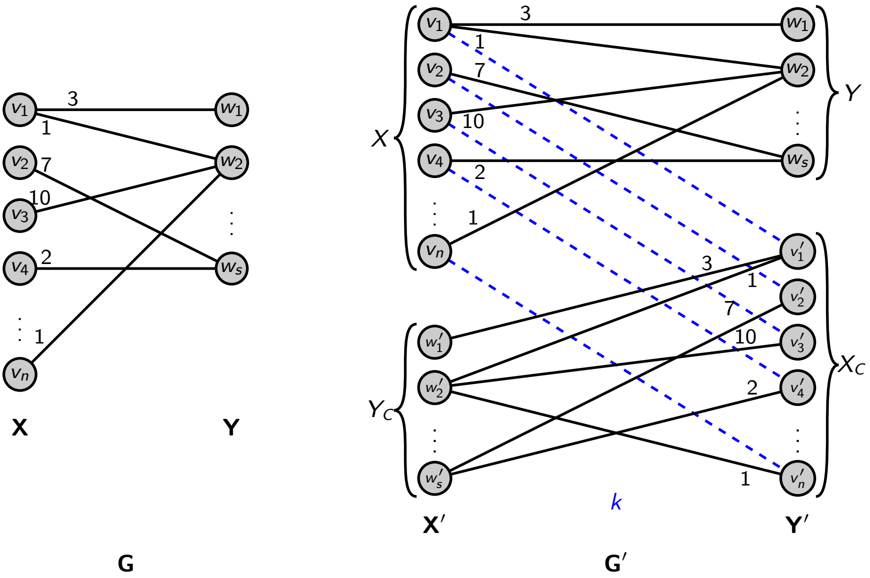
Teorema

Sea $\{G, c\}$ una instancia factible asimétrica y $\{G', c'_s\}$ su transformación de duplicación. **Si M' es un emparejamiento perfecto de costo mínimo en $\{G', c'_s\}$, entonces $M = M' \cap E$ es un emparejamiento perfecto de un lado de costo mínimo en $\{G, c\}$.**

Observaciones.

1. Ventajas y desventajas (si $C = 1250000$ y $s = 1000$, entonces $2sC = 2,500,000,000$).
2. $-2sC$ para el caso de **maximizar**.
3. $sC + 1$ para el caso de **minimizar** si los costos son **cero o positivos**.
4. $-sC$ para el caso de **maximizar** si los costos son **cero o positivos**.

Segunda transformación de duplicación



Teorema

Sea $\{G, c\}$ una instancia factible asimétrica y $\{G', c'\}$ su segunda transformación de duplicación. **Si M' es un emparejamiento perfecto de costo óptimo en $\{G', c'\}$, entonces $M = M' \cap E$ es un emparejamiento perfecto de un lado de costo mínimo en $\{G, c\}$.**

1. **ventajas y desventajas.**

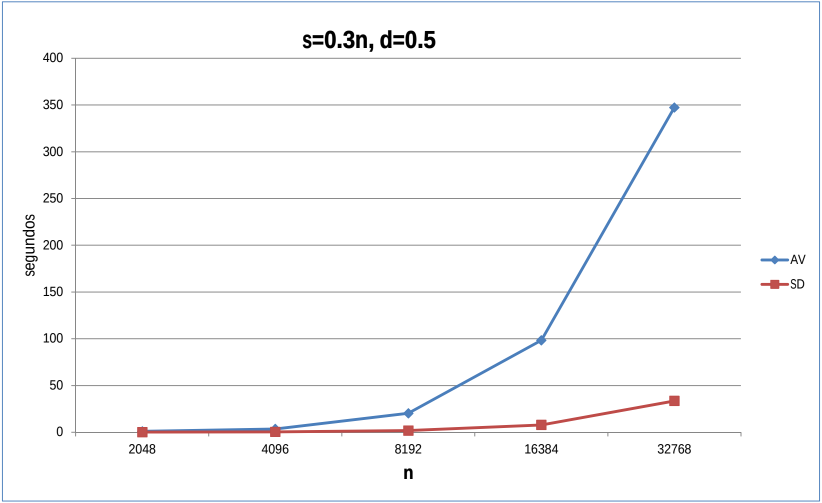


Figura : AV="Vértices Artificiales", SD="Segunda Transformación de Duplicación."

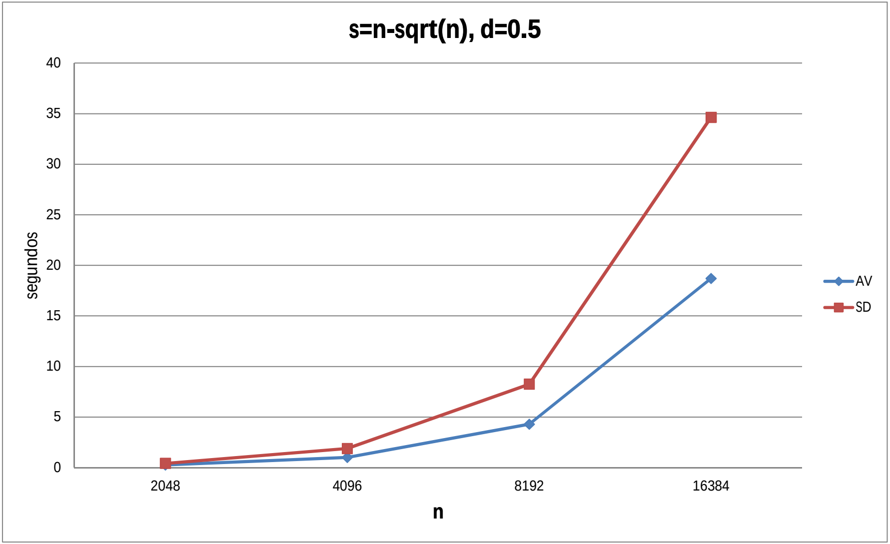


FIGURA : AV="Vértices Artificiales", SD="Segunda Transformación de Duplicación."

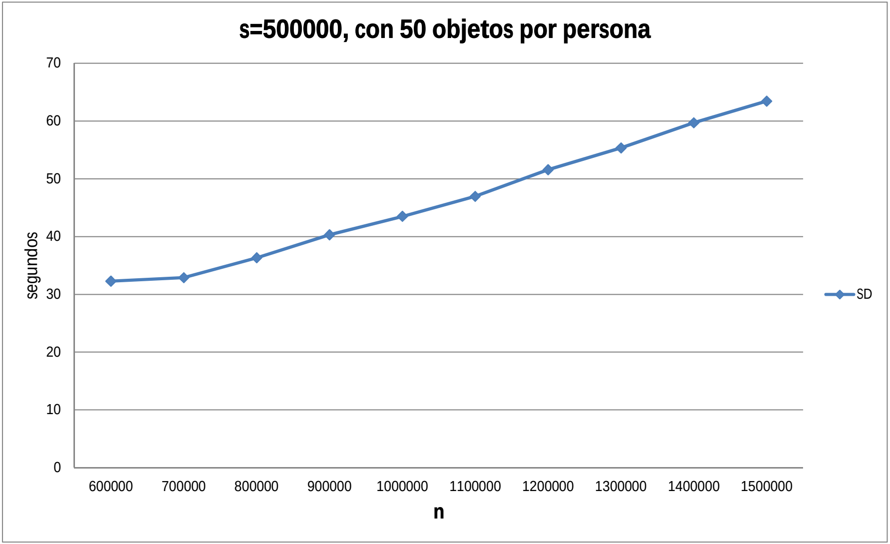
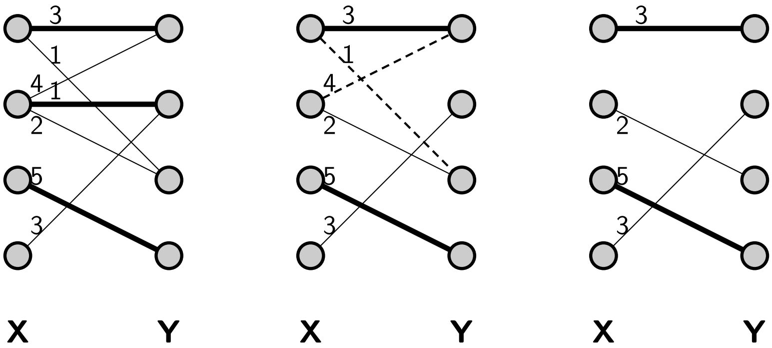


FIGURA : SD="Segunda Transformación de Duplicación."

El problema de preasignación

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica del problema de asignación y $M_{pre} \subseteq E$ un matching en G. El **problema de preasignación** consiste en encontrar un emparejamiento perfecto M de G **de costo óptimo** tal que $|M \cap M_{pre}|$ sea máximo.



(a) (b) (c)

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica y sea M un emparejamiento perfecto en G que satisface la **condición HC** con algún vector de precios p_Y. Es decir,

$$c'(v, w) = \min_{vz \in E} c'(v, z) \qquad \forall vw \in M. \tag{4}$$

DEFINICIÓN

Sea $G_{min}(M, p_Y) = (X \cup Y, E_{min}(M, p_Y))$ *una subgráfica de* G*, con*

$$E_{min}(M, p_Y) = \{vz \in E \mid c'(v, z) = c'(v, w), \text{ donde } vw \in M\}. \tag{5}$$

Algoritmo de preasignación ingenuo

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica y sea M un emparejamiento perfecto en G que satisface la **condición HC** con algún vector de precios p_Y. Es decir,

$$c'(v, w) = \min_{vz \in E} c'(v, z) \qquad \forall vw \in M. \tag{4}$$

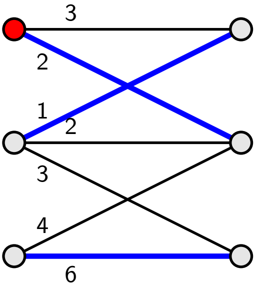
Definición

Sea $G_{min}(M, p_Y) = (X \cup Y, E_{min}(M, p_Y))$ una subgráfica de G, con

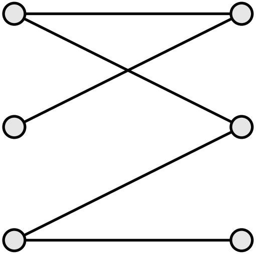
$$E_{min}(M, p_Y) = \{vz \in E \mid c'(v, z) = c'(v, w),\ \text{donde } vw \in M\}. \tag{5}$$

Ejemplo

p_Y	$c'(v, w)$
-2	5
-3	5
-1	

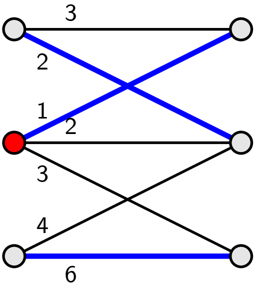


$G = (X \cup Y, E)$

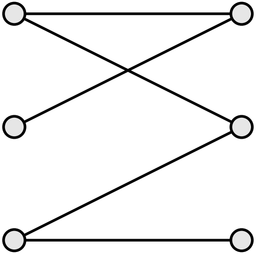


$G_{min}(M, p_Y) = (X \cup Y, E_{min})$

Ejemplo



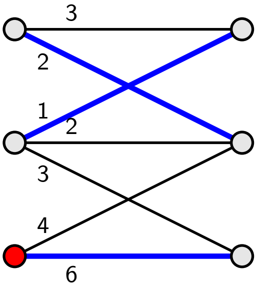
p_Y	$c'(v,w)$
-2	**3**
-3	**5**
-1	**4**



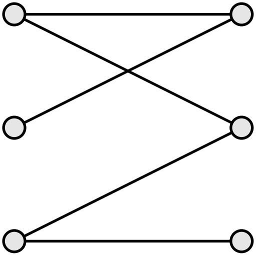
$G = (X \cup Y, E)$

$G_{min}(M, p_Y) = (X \cup Y, E_{min})$

Ejemplo



p_Y	$c'(v, w)$
-2	
-3	**7**
-1	**7**



$G = (X \cup Y, E)$ $\qquad$ $G_{min}(M, p_Y) = (X \cup Y, E_{min})$

Teorema

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica factible y M un emparejamiento perfecto de G que satisface la condición HC con algún vector de precios $p_Y : Y \to \mathbb{R}$. **Entonces S es un emparejamiento perfecto de costo mínimo en $\{G, c\}$ si y sólo si S es un emparejamiento perfecto en $G_{min}(M, p_Y)$.**

Para resolver el problema de preasignación solo asociamos a G_{min} la función de costos dada por

$$c_{min}(v, w) = \begin{cases} -1 & \text{si } vw \in M_{pre} \\ 0 & \text{de otra forma,} \end{cases} \tag{6}$$

y después **encontramos un emparejamiento perfecto de costo mínimo en la instancia** $\{G_{min}, c_{min}\}$.

Teorema

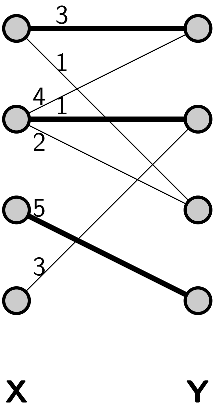
Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica factible y M un emparejamiento perfecto de G que satisface la condición HC con algún vector de precios $p_Y : Y \to \mathbb{R}$. **Entonces S es un emparejamiento perfecto de costo mínimo en $\{G, c\}$ si y sólo si S es un emparejamiento perfecto en $G_{min}(M, p_Y)$.**

Para resolver el problema de preasignación solo asociamos a G_{min} la función de costos dada por

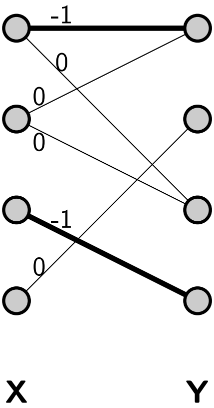
$$c_{min}(v, w) = \begin{cases} -1 & \text{si } vw \in M_{pre} \\ 0 & \text{de otra forma,} \end{cases} \tag{6}$$

y después **encontramos un emparejamiento perfecto de costo mínimo en la instancia $\{G_{min}, c_{min}\}$.**

Ejemplo



(a) La instancia (G, c, M_{pre})



(b) $G_{min}(M, p_Y)$ y c_{min}

El pseudocódigo de preasignación ingenuo

Procedure `min_ingenuo_preasignacion`(G, c, M_{pre})

    (M, p_Y)=`min_ingenuo_subastas`(G, c)

    $G_{min} = (X \cup Y, E_{min})$=`obtener_subgrafica`$(G, c, M, p_Y)$

    c_{min}=`crear_costos`(G_{min}, M_{pre})

    M_{sol}=`min_epsilon_escalamiento`$(G_{min}, c_{min}, 1/n)$

    `return` M_{sol}

`end.`

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica, con $n = |X|$, $\epsilon > 0$ y M un emparejamiento perfecto que satisface la **condición ϵ-HC** con algún vector de precios p_Y. Es decir,

$$c'(v, w) \leq \min_{vz \in E}\{c'(v, z)\} + \epsilon \qquad \forall vw \in M. \tag{7}$$

Definición

Sea $G^{\epsilon}_{min}(M, p_Y) = (X \cup Y, E^{\epsilon}_{min}(M, p_Y))$ una subgráfica de G, con

$$E^{\epsilon}_{min}(M, p_Y) = \{vz \in E \mid c'(v, z) < c'(v, w) + n\epsilon, \text{ donde } vw \in M\}. \tag{8}$$

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica, con $n = |X|$, $\epsilon > 0$ y M un emparejamiento perfecto que satisface la **condición ϵ-HC** con algún vector de precios p_Y. Es decir,

$$c'(v, w) \leq \min_{vz \in E}\{c'(v, z)\} + \epsilon \qquad \forall vw \in M. \tag{7}$$

Definición

Sea $G^{\epsilon}_{min}(M, p_Y) = (X \cup Y, E^{\epsilon}_{min}(M, p_Y))$ una subgráfica de G, con

$$E^{\epsilon}_{min}(M, p_Y) = \{vz \in E \mid c'(v, z) < c'(v, w) + n\epsilon, \ \textit{donde } vw \in M\}. \tag{8}$$

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia del problema de asignación, $\epsilon > 0$, M un emparejamiento perfecto que satisface la condición de ϵ-HC con un vector de precios p_Y y sea $G^{\epsilon}_{min}(M, p_Y)$ como la definimos antes.

Lema

Si S es un emparejamiento perfecto en $G^{\epsilon}_{min}(M, p_Y)$, entonces S es tal que $cost(S) < cost(M) + n^2\epsilon$.

Lema

Si S es un emparejamiento perfecto en G que no está totalmente contenido en $G^{\epsilon}_{min}(M, p_Y)$, entonces $cost(S) > cost(M)$.

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia del problema de asignación, $\epsilon > 0$, M un emparejamiento perfecto que satisface la condición de ϵ-HC con un vector de precios p_Y y sea $G^{\epsilon}_{min}(M, p_Y)$ como la definimos antes.

Lema

Si S es un emparejamiento perfecto en $G^{\epsilon}_{min}(M, p_Y)$, entonces S es tal que $cost(S) < cost(M) + n^2\epsilon$.

Lema

Si S es un emparejamiento perfecto en G que no está totalmente contenido en $G^{\epsilon}_{min}(M, p_Y)$, entonces $cost(S) > cost(M)$.

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia del problema de asignación, $\epsilon > 0$, M un emparejamiento perfecto que satisface la condición de ϵ-HC con un vector de precios p_Y y sea $G^{\epsilon}_{min}(M, p_Y)$ como la definimos antes.

Lema

Si S es un emparejamiento perfecto en $G^{\epsilon}_{min}(M, p_Y)$, entonces S es tal que $cost(S) < cost(M) + n^2\epsilon$.

Lema

Si S es un emparejamiento perfecto en G que no está totalmente contenido en $G^{\epsilon}_{min}(M, p_Y)$, entonces $cost(S) > cost(M)$.

Teorema

Sea $\{G = (X \cup Y, E), c : E \to \mathbb{Z}\}$ una instancia simétrica, $\epsilon \leq 1/n^2$, M un emparejamiento perfecto que satisface la condición ϵ-HC con algún vector de precios p_Y y sea $G^{\epsilon}_{min}(M, p_Y)$ como definimos antes. Entonces S **es un emparejamiento perfecto de costo mínimo en $\{G, c\}$ si y sólo si S es un emparejamiento perfecto en $G^{\epsilon}_{min}(M, p_Y)$.**

Demostración

1. M es de costo mínimo, porque $\epsilon < 1/n^2 < 1/n$.
2. ($\Rightarrow$) Del lema anterior, si S no está comletamente contenido en G^{ϵ}_{min}, entonces $cost(S) > cost(M)$.
3. ($\Leftarrow$) Del lema anterior y dado que $\epsilon \leq 1/n^2$,

$$cost(M) \leq cost(S) < cost(M) + n^2\epsilon \leq cost(M) + 1.$$

El pseudocódigo de ϵ-preasignación

Procedure min_epsilon_preasignacion(G, c, M_{pre})

    (M, p_Y)=min_epsilon_escalamiento$(G, c, 1/n^2)$

    G^{ϵ}_{min}=obtener_subgrafica(G, c, M, p_Y, ϵ)

    c^{ϵ}_{min}=crear_costos$(G^{\epsilon}_{min}, M_{pre})$

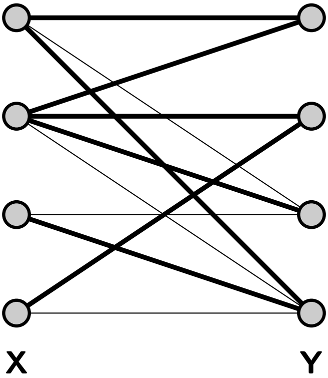
    M_{sol}=min_epsilon_escalamiento$(G^{\epsilon}_{min}, c^{\epsilon}_{min}, 1/n)$

    return M_{sol}

end.

1. $\log(n^2 C) = 2\log(nC) - \log(C) < 2\log(nC)$.
2. $O(n\, m \log(nC)) + O(m) + O(m) + O(n\, |E^{\epsilon}_{min}| \log(n)) = O(n\, m \log(nC))$.
3. Con una constante muy baja.
4. En general, si un algoritmo resuelve el problema de asignación en tiempo $O(f(n, m, C, \ldots))$, entonces **la complejidad en tiempo de resolver el problema de pre-asignación, usando este algoritmo, también será** $O(f(n, m, C, \ldots))$.

Como corolario



Sin ninguna modificación, los algoritmos de preasignación resuelven esta generalización del problema de preasignación, donde cada persona puede tener varios objetos preasignados y el objetivo es maximizar el numero de personas asignadas a uno de sus objetos preasignados, también manteniendo la optimalidad del costo total.

Estadísticas del algoritmo de ϵ-preasignación

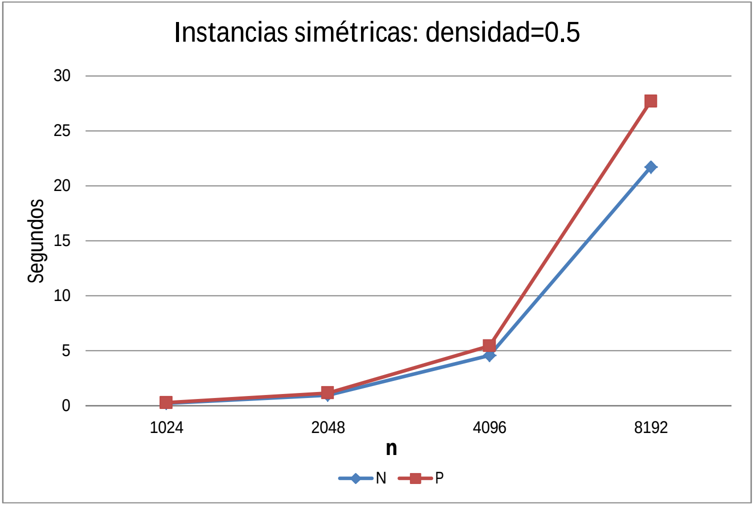


Figura : N='Asignación normal', P='Preasignación'. Ambos en las mismas instancias de cada caso.

Comentarios

Logramos conseguir una forma de combinar la condición de ϵ-Holgura Complementaria con el algoritmo de pre-asignación ingenuo. De esta forma eliminamos las desventajas individuales que tienen el algoritmo de pre-asignación ingenuo y el de ϵ-preasignación. Los pasos de este nuevo algoritmo combinado son:

1. Usar el algoritmo de ϵ-escalamiento para obtener un emparejamiento perfecto M y un vector de precios P_Y que satisfagan la condición ϵ-HC con algún $\epsilon < \frac{1}{n+1}$.
2. Usando M, modificamos el vector p_Y de forma que ahora se satisfaga la condición HC. Esto se hace en tiempo $O(n)$.
3. Usamos el algoritmo de pre-asignación ingenuo con M y p_Y.

Conclusiones

1. El algoritmo de Bertsekas es muy **bueno en la práctica** y **versatil** porque permite resolver el problema de preasignación.
2. **El problema de preasignación se puede resolver de manera muy eficiente**.
3. **El algoritmo de subastas no es el único que se ajusta para devolver un emparejamiento y un vector de precios que satisfaga las condiciones de holgura complementaria**. Como ejemplo tenemos al algoritmo de Goldberg & Kennedy.
4. **La solución a los problemas de pre-asignación no depende de un algoritmo en específico**, solo depende de las condiciones de holgura complementaria.

Fin