Métodos iterativos para Optimización Combinatorial

J.A. Soto, Universidad de Chile.

Escuela de Primavera

21–23 Oct, 2013

Outline

Outline

Introducción

- Argumento iterativo, muy exitoso en la última década, para diseñar algoritmos exactos y de aproximación para diversos problemas de optimización combinatorial.
- *"Iterative Methods in Combinatorial Optimization"*
  por Lau, Ravi y Singh.
  `http://research.microsoft.com/en-us/um/people/mohits/book.html`.

Conceptos básicos

Optimización Combinatorial

Optimización sobre familias discretas con gran estructura.

Problemas "P-fáciles"

Algoritmos a tiempo polinomial en el tamaño de la entrada.

- Caminos más cortos.
- Emparejamientos máximos.
- Árboles de costo mínimo.

Problemas "NP-difíciles"

Bajo conjetura $P \neq NP$, no admiten algoritmos polinomiales.

- Caminos más largos.
- Asignación Generalizada.
- Árboles de Steiner.

Conceptos básicos

Optimización Combinatorial

Optimización sobre familias discretas con gran estructura.

Modelar con Programación Lineal / Entera.

Problemas "P-fáciles"

Algoritmos a tiempo polinomial en el tamaño de la entrada.

- Caminos más cortos.
- Emparejamientos máximos.
- Árboles de costo mínimo.

Algoritmos exactos

Problemas "NP-difíciles"

Bajo conjetura $P \neq NP$, no admiten algoritmos polinomiales.

- Caminos más largos.
- Asignación Generalizada.
- Árboles de Steiner.

Algoritmos aproximados

Programación lineal entera

Muchos de estos problemas se pueden modelar como:

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in \{0,1\}^n \end{aligned}$$

Programación lineal entera

Muchos de estos problemas se pueden modelar como:

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in \{0,1\}^n \end{aligned}$$

$\Rightarrow$Relajar$\Rightarrow$

Programa Lineal

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in [0,1]^n \end{aligned}$$

Programación lineal entera

Muchos de estos problemas se pueden modelar como:

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in \{0,1\}^n \end{aligned}$$

Difíciles de resolver

=Relajar⇒

Programa Lineal

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in [0,1]^n \end{aligned}$$

Alg. Polinomiales para resolver PL dados A, b y c.

Encontrar x^* fraccional (punto extremo del políedro).

Programación lineal entera

Muchos de estos problemas se pueden modelar como:

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in \{0,1\}^n \end{aligned}$$

=Relajar⇒

Programa Lineal

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \in [0,1]^n \end{aligned}$$

Difíciles de resolver

Alg. Polinomiales para resolver PL dados A, b y c.

Solución aproximada.

⇐Redondear=

Encontrar x^* fraccional (punto extremo del políedro).

¿Problemas difíciles?

Técnica útil para muchos problemas NP-difíciles.

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- Redondear x^*.

¿Problemas difíciles?

Técnica útil para muchos problemas NP-difíciles.

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- Redondear x^*.

¿Cómo redondear?

¿Problemas difíciles?

Técnica útil para muchos problemas NP-difíciles.

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- Redondear x^*.

¿Cómo redondear?

- Redondeo simple (ej. Fijar en 1 todos los $x_i \geq 1/2$)
- Redondeo aleatorizado (Fijar en 1 con probabilidad x_i).
- Método Primal-Dual (iterar entre soluciones primales y duales).
- Redondeo iterativo (Iniciado por Jain '98)
- Relajación iterativa (Iniciado por Singh, Lau y otros '07)

¿Problemas fáciles?

Para muchos problemas en P:

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- ¡x^* resulta ser integral!

¿Problemas fáciles?

Para muchos problemas en P:

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- ¡x^* resulta ser integral!

¿Cómo probamos que x^* es integral?

¿Problemas fáciles?

Para muchos problemas en P:

- Formular como Programa Entero.
- Encontrar punto extremo óptimo x^* de relajación.
- ¡x^* resulta ser integral!

¿Cómo probamos que x^* es integral?

- Total Unimodularidad.
- Total Dual Integralidad.
- Total Dual Laminaridad.
- Esquemas Primal - Dual.
- Relajación Iterativa (Iniciado por Singh, Lau y otros '07)

Ideas generales

Redondeo (NP-difícil) / *Relajación* (P)

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

Ideas generales

Redondeo (NP-difícil) / *Relajación* (P)

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

Relajación iterativa (NP-difícil)

- - Encontrar problema base en P con puntos extremos integrales.
  - Agregar restricciones al problema base e intentar esquema anterior.
  - cambio: Demostrar que existe una coordenada con alto valor x_i (redondear) o una restricción "poco violada" (relajar).

Outline

Álgebra Lineal: Recuerdo

Dado $A \in \mathbb{R}^{n \times d}$ una matriz a coeficientes reales.

- Vectores filas: $A_1, \ldots, A_n$.
- Vectores columnas: $A[\cdot, 1], \ldots, A[\cdot, d]$.
- Rango fila: $\dim(\text{span}(A_i)_{i=1}^n)$
  $=$ número máximo de vectores filas linealmente independientes.
- Rango columna: $\dim(\text{span}(A[\cdot, j])_{j=1}^d))$
  $=$ número máximo de vectores columnas l.i.

Álgebra Lineal: Recuerdo

Dado $A \in \mathbb{R}^{n \times d}$ una matriz a coeficientes reales.

- Vectores filas: $A_1, \ldots, A_n$.
- Vectores columnas: $A[\cdot, 1], \ldots, A[\cdot, d]$.
- Rango fila: $\dim(\mathrm{span}(A_i)_{i=1}^n)$
  $=$ número máximo de vectores filas linealmente independientes.
- Rango columna: $\dim(\mathrm{span}(A[\cdot, j])_{j=1}^d))$
  $=$ número máximo de vectores columnas l.i.

Lema elemental

$\mathrm{rk}(A) :=$ Rango fila de $A =$ Rango columna de A.

Programación lineal

$A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, $c, x \in \mathbb{R}^d$.

$$\begin{aligned} \text{mín } & c^\top x \\ \text{s.a. } & Ax \geq b \\ & x \geq 0 \end{aligned}$$

$$P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}.$$

Programación lineal

$A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, $c, x \in \mathbb{R}^d$.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \geq 0 \end{aligned}$$

$$P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}.$$

- P es un **poliedro** (no acotado en este caso).

Programación lineal

$A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, $c, x \in \mathbb{R}^d$.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \geq 0 \end{aligned}$$

$$P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}.$$

- P es un **poliedro** (no acotado en este caso).
- **Solución factible**: Puntos x de P. **Solución óptima**: Factible y de minimo costo (mínimo $c^\top x$).

Programación lineal

$A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, $c, x \in \mathbb{R}^d$.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \geq 0 \end{aligned}$$

$$P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}.$$

- P es un **poliedro** (no acotado en este caso).
- **Solución factible**: Puntos x de P. **Solución óptima**: Factible y de minimo costo (mínimo $c^\top x$).
- **Punto extremo**: Un punto x de P es un punto extremo si no existe $y \neq 0$ tal que $x + y \in P$ y $x - y \in P$. (equiv. si no puede escribirse como combinación convexa de otros puntos de P)

Programación lineal

$A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, $c, x \in \mathbb{R}^d$.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &Ax \geq b \\ &x \geq 0 \end{aligned}$$

$$P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}.$$

- P es un **poliedro** (no acotado en este caso).
- **Solución factible**: Puntos x de P. **Solución óptima**: Factible y de minimo costo (mínimo $c^\top x$).
- **Punto extremo**: Un punto x de P es un punto extremo si no existe $y \neq 0$ tal que $x + y \in P$ y $x - y \in P$. (equiv. si no puede escribirse como combinación convexa de otros puntos de P)
- Para $x \in P$, definimos $J(A, x) = \{A_i \colon A_i x = b_i\}$ la **colección de filas de A ajustadas en x^***.

PL: Lemas que necesitaremos (y probaremos)

Existe óptimo extremo.

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Lema de Rango

Sea $x^* \in P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}$ punto extremo positivo ($x_i^* > 0$, $\forall i$).
Entonces: la cantidad máxima de filas l.i. en $J(A, x^*)$ es igual al número de coordenadas de x^*

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.
- $A^= y = A^=(x^* + y) - A^= x^* \geq 0$ y por simetría $A^= y = 0$.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.
- $A^= y = A^=(x^* + y) - A^= x^* \geq 0$ y por simetría $A^= y = 0$.
- (opt.) $c^\top x^* \leq c^\top (x^* + y)$ y $c^\top x^* \leq c^\top (x^* - y)$. Luego $c^\top y = 0$

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.
- $A^= y = A^=(x^* + y) - A^= x^* \geq 0$ y por simetría $A^= y = 0$.
- (opt.) $c^\top x^* \leq c^\top (x^* + y)$ y $c^\top x^* \leq c^\top (x^* - y)$. Luego $c^\top y = 0$
- Como $x^* + y \geq 0$ y $x^* - y \geq 0$, tenemos que $x^*_j = 0$ implica $y_j = 0$

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.
- $A^= y = A^=(x^* + y) - A^= x^* \geq 0$ y por simetría $A^= y = 0$.
- (opt.) $c^\top x^* \leq c^\top (x^* + y)$ y $c^\top x^* \leq c^\top (x^* - y)$. Luego $c^\top y = 0$
- Como $x^* + y \geq 0$ y $x^* - y \geq 0$, tenemos que $x^*_j = 0$ implica $y_j = 0$
- S.p.g. y tiene una coordenada negativa. Sea $z = x^* + \lambda y \in P$ con λ lo más grande posible.

Programación lineal (2): Importante

Si $\min\{c^\top x \colon Ax \geq b, x \geq 0\}$ es finito, entonces se alcanza en x^* punto extremo de P.

Sketch de demostración.

- x^*: solución óptima, con mayor $\big[\#$ coordenadas $0 + \#J(A, x^*)\big]$.
- Si x^* no es punto extremo existe $y \neq 0$ con $x^* + y, x^* - y \in P$.
- Sea $A^=$ la submatriz de A con filas en $J(A, x^*)$.
- $A^= y = A^=(x^* + y) - A^= x^* \geq 0$ y por simetría $A^= y = 0$.
- (opt.) $c^\top x^* \leq c^\top (x^* + y)$ y $c^\top x^* \leq c^\top (x^* - y)$. Luego $c^\top y = 0$
- Como $x^* + y \geq 0$ y $x^* - y \geq 0$, tenemos que $x^*_j = 0$ implica $y_j = 0$
- S.p.g. y tiene una coordenada negativa. Sea $z = x^* + \lambda y \in P$ con λ lo más grande posible.
- z es óptimo y las restricciones ajustadas de x^* son ajustadas para z también. Pero, z tiene o bien un cero, o una restricción ajustada extra. (contradicción).

□

Programación lineal (3): Lema técnico

Sea $x \in P$, $A^=$ la submatriz de A ajustadas para x, y $A^=_x$ la submatriz de $A^=$ con columnas asociadas a entradas > 0 de x.
x es punto extremo de P si y solo si todas las columnas de $A^=_x$ son l.i.

Programación lineal (3): Lema técnico

Sea $x \in P$, $A^=$ la submatriz de A ajustadas para x, y $A^=_x$ la submatriz de $A^=$ con columnas asociadas a entradas > 0 de x.
x es punto extremo de P si y solo si todas las columnas de $A^=_x$ son l.i.

Sketch de demostración.

- ($\Leftarrow$) Si x no es punto extremo, entonces existe $y \neq 0$ como en la demostración anterior con $A^=y = 0$ y tal que $x_j = 0 \Rightarrow y_j = 0$.
  Esto implica que la matriz $A^=_y$ tiene columnas l.d. y es submatriz de $A^=_x$.

□

Programación lineal (3): Lema técnico

Sea $x \in P$, $A^=$ la submatriz de A ajustadas para x, y $A^=_x$ la submatriz de $A^=$ con columnas asociadas a entradas > 0 de x.
x es punto extremo de P si y solo si todas las columnas de $A^=_x$ son l.i.

Sketch de demostración.

- ($\Leftarrow$) Si x no es punto extremo, entonces existe $y \neq 0$ como en la demostración anterior con $A^=y = 0$ y tal que $x_j = 0 \Rightarrow y_j = 0$.
  Esto implica que la matriz $A^=_y$ tiene columnas l.d. y es submatriz de $A^=_x$.
- ($\Rightarrow$) Si las columnas de $A^=_x$ son l.d. entonces existe $z \neq 0$ de la dimension adecuada tal que $A^=_x z = 0$. Tomemos $y \in \mathbb{R}^d$ igual a z completado con 0, de modo que $A^=y = 0$.
  Por construcción, $x_j = 0 \Rightarrow y_j = 0$. Es facil ver que para $\varepsilon > 0$ suficientemente pequeño, $x \pm \varepsilon y \geq 0$, $A(x \pm \varepsilon y) \geq b$, con lo que x no es punto extremo.

□

Programación lineal (4): Lema de Rango

Lema de Rango

Sea $x^* \in P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}$ punto extremo positivo ($x_i^* > 0$, $\forall i$).
Entonces: la cantidad máxima de filas l.i. en $J(A, x^*)$ es igual al número de coordenadas de x^*

Programación lineal (4): Lema de Rango

Lema de Rango

Sea $x^* \in P = \{x \in \mathbb{R}^d \colon Ax \geq b, x \geq 0\}$ punto extremo positivo ($x_i^* > 0$, $\forall i$). Entonces: la cantidad máxima de filas l.i. en $J(A, x^*)$ es igual al número de coordenadas de x^*

Demostración.

Las filas de $J(A, x^*)$ son las filas de $A_{x^*}^{=}$ del lema anterior. Por el lema, como x^* es punto extremo, las columnas de $A_{x^*}^{=}$ son l.i. Es decir, el rango columna de $A_{x^*}^{=}$ es d. Luego el número máximo de filas l.i. de J_x, que es el rango fila de A_x^*, es d. □

Observación

Estos resultados se pueden aplicar también a problemas de maximización, por ejemplo:

$$\begin{aligned} \text{máx } & c^\top x \\ \text{s.a. } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

ó

$$\begin{aligned} \text{máx } & c^\top x \\ \text{s.a. } & A^1 x \leq b^1 \\ & A^2 x = b^2 \\ & A^3 x \geq b^3 \\ & x \geq 0 \end{aligned}$$

Observación

Estos resultados se pueden aplicar también a problemas de maximización, por ejemplo:

$$\begin{aligned} &\text{máx } c^\top x \\ \text{s.a. } &Ax \leq b \\ &x \geq 0 \end{aligned}$$

ó

$$\begin{aligned} &\text{máx } c^\top x \\ \text{s.a. } &A^1 x \leq b^1 \\ &A^2 x = b^2 \\ &A^3 x \geq b^3 \\ &x \geq 0 \end{aligned}$$

Basta ver que el primer problema es equivalente a

$$\text{mín}\{(-c)^\top x \colon (-A)x \geq (-b), x \geq 0\},$$

y el segundo se puede transformar en el primero.

Resolución de Programas lineales

Importante: Hay algoritmos polinomiales para resolver programas lineales (i.e., para encontrar un punto extremo óptimo).
Esto se puede hacer en 2 casos.

- Cuando todos los datos (A, b y c) son dados de manera explícita. (polinomial en el número de bits para escribir las matrices).
- También se puede hacer cuando A, b y c no son los datos originales (veremos ejemplos después), sino que tenemos acceso a un oráculo de separación que, dado x^* es capaz de
  - Asegurar que x^* es solución factible ($x^* \in P$)
  - Si x^* no es factible, devuelve un hiperplano que separa x de P (i.e., una desigualdad tal que $a^\top x \leq b$ para todo $x \in P$, pero $a^\top x^* > b$).

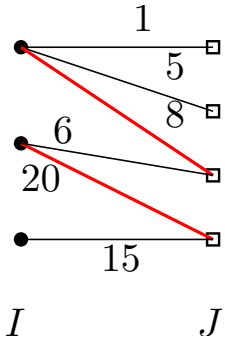
Outline

Problema de asignación

- Sea I un conjunto de máquinas y J un conjunto de trabajos.
- Cada máquina $i \in I$ es capaz de procesar algunos trabajos de J.
- $E = \{(i, j) \colon j \text{ puede procesar } i\}$.
- Procesar el trabajo j en la máquina i nos da una ganancia de $w_{i,j} \in \mathbb{R}$.
- Queremos encontrar una asignación (matching) M de trabajos a máquinas tal que cada máquina procese a lo más 1 trabajo y cada trabajo sea procesado en a lo más una máquina.
- Objetivo: Encontrar una asignación (un matching) M que maximize la ganancia total $\sum_{(i,j) \in M} w_{i,j}$

Problema de asignación: Ejemplo

I: máquinas, J: trabajos.



Problema de asignación: Modelo entero

I: máquinas, J: trabajos, $w\colon E \to \mathbb{R}$ pesos.

- Para $e \in E$, x_e variable que indica si $e \in M$.
- $x_e \in \{0, 1\}$.

Problema de asignación: Modelo entero

I: máquinas, J: trabajos, $w\colon E \to \mathbb{R}$ pesos.

- Para $e \in E$, x_e variable que indica si $e \in M$.
- $x_e \in \{0, 1\}$.
- $\forall i \in I\colon \displaystyle\sum_{j\colon (i,j)\in E} x_{ij} \leq 1.$

Problema de asignación: Modelo entero

I: máquinas, J: trabajos, $w\colon E \to \mathbb{R}$ pesos.

- Para $e \in E$, x_e variable que indica si $e \in M$.
- $x_e \in \{0, 1\}$.
- $\forall i \in I\colon \displaystyle\sum_{j\colon (i,j)\in E} x_{ij} \leq 1.$
- $\forall j \in J\colon \displaystyle\sum_{i\colon (i,j)\in E} x_{ij} \leq 1.$

Problema de asignación: Modelo entero

I: máquinas, J: trabajos, $w\colon E \to \mathbb{R}$ pesos.

- Para $e \in E$, x_e variable que indica si $e \in M$.
- $x_e \in \{0, 1\}$.
- $\forall i \in I\colon \displaystyle\sum_{j\colon (i,j)\in E} x_{ij} \leq 1.$
- $\forall j \in J\colon \displaystyle\sum_{i\colon (i,j)\in E} x_{ij} \leq 1.$
- Objetivo: $\max \sum_{(i,j)\in E} x_{i,j} w_{i,j} = w^\top x$

Problema de asignación: Modelo entero

$$\begin{aligned} & \max\ w^\top x \\ \text{s.a.} \quad & \sum_{j\colon (i,j)\in E} x_{i,j} \leq 1, \quad \forall i \in I. \\ & \sum_{i\colon (i,j)\in E} x_{i,j} \leq 1, \quad \forall j \in J. \\ & x_e \in \{0,1\} \end{aligned}$$

Problema de asignación: Modelo entero

$$\begin{aligned} & \text{máx } w^\top x \\ \text{s.a.} \quad & \sum_{e \in \delta_E(v)} x_e \leq 1, \quad \forall v \in I \cup J. \\ & x_e \in \{0,1\} \end{aligned}$$

Problema de asignación: Modelo entero

$$
\begin{aligned}
&\text{máx } w^\top x \\
\text{s.a. } \quad &x(\delta_E(v)) \leq 1, \quad \forall v \in I \cup J. \\
&x_e \in \{0,1\}
\end{aligned}
$$

Problema de asignación: Relajación

$$\begin{aligned} & \text{máx } w^\top x \\ \text{s.a. } & x(\delta(v)) \leq 1, \quad \forall v \in I \cup J. \\ & x_e \geq 0 \end{aligned}$$

Problema de asignación: Relajación

$$\begin{aligned} &\text{máx } w^\top x \\ \text{s.a. } &x(\delta(v)) \leq 1, \quad \forall v \in I \cup J. \\ &x_e \geq 0 \end{aligned}$$

Teorema:

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Primera aplicación del método iterativo:

Teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E : x(\delta_E(v)) \geq 1, \forall x \geq 0\}$, entonces existe $e \in E$ con $x_e^* \in \{0, 1\}$.

Primera aplicación del método iterativo:

Teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E : x(\delta_E(v)) \geq 1, \forall x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Algoritmo

Sea $M \leftarrow 0$. Repetir lo siguiente hasta terminar.

- x^* : punto extremo óptimo del problema actual.
  - Si $x^* = 0$ terminar.
- Agregar a M todas las aristas con $x^*_e = 1$.
- (reducción) Borrar todas las aristas e con $x^*_e = 0$ y para cada $e = (i, j)$ con $x^*_e = 1$, borrar i, j y todas sus aristas incidentes.

Primera aplicación del método iterativo:

Teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E : x(\delta_E(v)) \geq 1, \forall x \geq 0\}$,
entonces existe $e \in E$ con $x_e^* \in \{0, 1\}$.

Algoritmo

Sea $M \leftarrow 0$. Repetir lo siguiente hasta terminar.

- x^* : punto extremo óptimo del problema actual.
  - Si $x^* = 0$ terminar.
- Agregar a M todas las aristas con $x_e^* = 1$.
- (reducción) Borrar todas las aristas e con $x_e^* = 0$ y para cada $e = (i, j)$ con $x_e^* = 1$, borrar i, j y todas sus aristas incidentes.

(Simple): El teorema implica que el algoritmo devuelve una asignación M óptima.

Probemos el teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup Jx \geq 0\}$, y $x^* \neq 0$ entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

- Supongamos por contradicción que $0 < x^*_e < 1$ para todo $e \in E$.

Probemos el teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$,
y $x^* \neq 0$ entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

- Supongamos por contradicción que $0 < x^*_e < 1$ para todo $e \in E$.
- Lema de rango $\Rightarrow$ Existe un conjunto linealmente independiente de $|E|$ restricciones que son ajustadas en x.

  $W \subseteq I \cup J$, $|W| = |E|$ tal que
  - $\{x(\delta(v)) = 1 \colon v \in W\}$
  - $\{\chi(\delta(v)) \colon v \in W\}$ son l.i.

Probemos el teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$, y $x^* \neq 0$ entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

- Supongamos por contradicción que $0 < x^*_e < 1$ para todo $e \in E$.
- Lema de rango $\Rightarrow$ Existe un conjunto linealmente independiente de $|E|$ restricciones que son ajustadas en x.

  $W \subseteq I \cup J$, $|W| = |E|$ tal que
  - $\{x(\delta(v)) = 1 \colon v \in W\}$
  - $\{\chi(\delta(v)) \colon v \in W\}$ son l.i.
- **Afirmación**: $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$.

Afirmación: $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$

Primero: Para todo $v \in W$, $x(\delta_E(v)) = 1$.
Como $0 < x_e^* < 1$ tenemos que $d_E(v) \geq 2$.

Segundo:

$$2|W| = 2|E| = \sum_{v \in I \cup J} d_E(v) \geq \sum_{v \in W} d_E(v) \geq 2|W|.$$

Afirmación: $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$

Primero: Para todo $v \in W$, $x(\delta_E(v)) = 1$.
Como $0 < x_e^* < 1$ tenemos que $d_E(v) \geq 2$.

Segundo:

$$2|W| = 2|E| = \sum_{v \in I \cup J} d_E(v) \geq \sum_{v \in W} d_E(v) \geq 2|W|.$$

Última desigualdad: $d_E(v) = 2$ para $v \in W$.
Penúltima desigualdad: $d_E(v) = 0$, para $v \notin W$.

Demostración del teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Si no:

- $\{\chi(\delta(v)) \colon v \in W\}$ son l.i. (con $|W| = |E|$)
- $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$.

Demostración del teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Si no:

- $\{\chi(\delta(v)) \colon v \in W\}$ son l.i. (con $|W| = |E|$)
- $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$.

- (W, E) es unión de ciclos. Sea $(C, E(C))$ uno de ellos.

Demostración del teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Si no:

- $\{\chi(\delta(v)) \colon v \in W\}$ son l.i. (con $|W| = |E|$)
- $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$.

- (W, E) es unión de ciclos. Sea $(C, E(C))$ uno de ellos.
- Como el grafo es bipartito, C tiene un número par de aristas y

$$\chi(E(C)) = \sum_{v \in C \cap I} \chi(\delta(v)) = \sum_{v \in C \cap J} \chi(\delta(v)).$$

Demostración del teorema

Si x^* es un punto extremo de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$, entonces existe $e \in E$ con $x^*_e \in \{0, 1\}$.

Si no:

- $\{\chi(\delta(v)) \colon v \in W\}$ son l.i. (con $|W| = |E|$)
- $d_E(v) = 2$, para todo $v \in W$, $d_E(v) = 0$, para todo $v \notin W$.

- (W, E) es unión de ciclos. Sea $(C, E(C))$ uno de ellos.
- Como el grafo es bipartito, C tiene un número par de aristas y

$$\chi(E(C)) = \sum_{v \in C \cap I} \chi(\delta(v)) = \sum_{v \in C \cap J} \chi(\delta(v)).$$

- Contradice independencia lineal! $\square$

Resumen:

Gracias al teorema, existe algoritmo iterativo polinomial que encuentra un matching M con peso igual a $\max\{w^\top x \colon x \in P\}$.

Resumen:

Gracias al teorema, existe algoritmo iterativo polinomial que encuentra un matching M con peso igual a $\max\{w^\top x \colon x \in P\}$.

Corolario

Todos los puntos extremos de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$ son integrales!

Resumen:

Gracias al teorema, existe algoritmo iterativo polinomial que encuentra un matching M con peso igual a $\max\{w^\top x : x \in P\}$.

Corolario

Todos los puntos extremos de $P = \{x \in \mathbb{R}^E : x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$ son integrales!

(dem:) Si x^* es punto extremo, existe una dirección $w \in \mathbb{R}^E$ tal que x^* es el único máximo de $w^\top x$, $x \in P$.

Resumen:

Gracias al teorema, existe algoritmo iterativo polinomial que encuentra un matching M con peso igual a $\max\{w^\top x \colon x \in P\}$.

Corolario

Todos los puntos extremos de $P = \{x \in \mathbb{R}^E \colon x(\delta_E(v)) \geq 1, \forall v \in I \cup J x \geq 0\}$ son integrales!

(dem:) Si x^* es punto extremo, existe una dirección $w \in \mathbb{R}^E$ tal que x^* es el único máximo de $w^\top x$, $x \in P$.

Entonces, no es necesario iterar, basta resolver el programa lineal para encontrar M.

Outline

Recuerdo del esquema general

Redondeo (NP-difícil) / *Relajación* (P)

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

Recuerdo del esquema general

Redondeo (NP-difícil) / *Relajación* (P)

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

Relajación iterativa (NP-difícil)

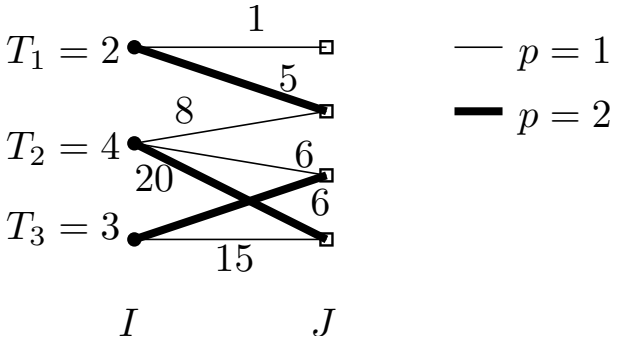
-
  - Encontrar problema base en P con puntos extremos integrales.
  - Agregar restricciones al problema base e intentar esquema anterior.
  - cambio: Demostrar que existe una coordenada con alto valor x_i (redondear) o una restricción "poco violada" (relajar).

Asignación Generalizada

- Sea I un conjunto de máquinas y J un conjunto de trabajos.
- $E = \{(i, j) : j \text{ puede procesar } i\}$,
- Cada máquina i puede ser usada por T_i unidades de tiempo.
- $p_{i,j}$: tiempo necesario para procesar j en i.
- $c_{i,j}$: costo de procesar j en i.
- Cada trabajo debe ser procesado en alguna máquina.
- Objetivo: Encontrar una asignación (un semi-matching) M que minimice el costo total $\sum_{(i,j)\in M} c_{i,j}$, respetando los tiempos de las máquinas.

Asignación General: Ejemplo

I: máquinas, J: trabajos.



Asignación General: Programa Entero

I: máquinas, J: trabajos, T_i tiempos, $p, c\colon E \to \mathbb{R}$ tiempos y costos.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &x(\delta_E(j)) = 1, \quad \forall j \in J. \\ &\sum_{e \in \delta_E(i)} p_e x_e \leq T_i, \quad \forall i \in I. \\ &x_e \in \{0,1\} \end{aligned}$$

Asignación General: Programa Entero

I: máquinas, J: trabajos, T_i tiempos, $p, c\colon E \to \mathbb{R}$ tiempos y costos.

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &x(\delta_E(j)) = 1, \quad \forall j \in J. \\ &\sum_{e \in \delta_E(i)} p_e x_e \leq T_i, \quad \forall i \in I. \\ &x_e \in \{0,1\} \end{aligned}$$

- El problema es NP-difícil por lo que suponemos que no hay algoritmo polinomial.
- La relajación natural NO es integral.

Aproximación

Supongamos que $C \in \mathbb{R}$ es el costo de una solución óptima (que no podemos encontrar eficientemente).

Shmoys y Tardos probaron que existe un algoritmo que devuelve una asignación M de trabajos a máquinas tal que:

- Todos los trabajos son asignados.
- Cada máquina i es usada a lo más $2T_i$ unidades de tiempo (es decir, el doble de lo que originalmente se pedía).
- El costo total de la asignación es $\leq C$.

Dicho algoritmo se puede ver como aplicación del método de relajación iterativa.

Asignación General: Programa Lineal modificado

$$
\begin{aligned}
PL(E, I'): \quad & \text{mín } c^\top x \\
\text{s.a.} \quad & x(\delta_E(j)) = 1, \quad \forall j \in J. \\
& \sum_{e \in \delta_E(i)} p_e x_e \leq T_i, \quad \forall i \in I'. \\
& x_e \geq 0 \\
& x_e = 0, \quad \text{si } p_{i,j} > T_i
\end{aligned}
$$

Notar que el costo del PL es menor o igual al costo del óptimo entero.

Aplicación de lema de rango

Si x^* es punto extremo de $P(E, I') = \{x \colon x(\delta_E(j)) = 1, \forall j \in J;$ $(px)(\delta_E(i)) \leq T_i, \forall i \in I', x \geq 0, x_e = 0,$ para e imposible$\}$, con todas sus coordenadas positivas, entonces

Aplicación de lema de rango

Si x^* es punto extremo de $P(E, I') = \{x \colon x(\delta_E(j)) = 1, \forall j \in J;$
$(px)(\delta_E(i)) \leq T_i, \forall i \in I', x \geq 0, x_e = 0,$ para e imposible$\}$, con todas sus coordenadas positivas, entonces
Existe un conjunto $J_o \subseteq J$ y un conjunto $I_o \subseteq I'$ tal que:

Aplicación de lema de rango

Si x^* es punto extremo de $P(E, I') = \{x \colon x(\delta_E(j)) = 1, \forall j \in J;$
$(px)(\delta_E(i)) \leq T_i, \forall i \in I', x \geq 0, x_e = 0,$ para e imposible$\}$, con todas sus coordenadas positivas, entonces
Existe un conjunto $J_o \subseteq J$ y un conjunto $I_o \subseteq I'$ tal que:

1. $(px)(\delta_E(i)) = T_i$ para todo $i \in I_o$. $\quad (\sum_{e \in \delta_E(i)} p_e x_e = T_i)$

Aplicación de lema de rango

Si x^* es punto extremo de $P(E, I') = \{x \colon x(\delta_E(j)) = 1, \forall j \in J;$ $(px)(\delta_E(i)) \leq T_i, \forall i \in I', x \geq 0, x_e = 0,$ para e imposible$\}$, con todas sus coordenadas positivas, entonces

Existe un conjunto $J_o \subseteq J$ y un conjunto $I_o \subseteq I'$ tal que:

1. $(px)(\delta_E(i)) = T_i$ para todo $i \in I_o$. $\quad (\sum_{e \in \delta_E(i)} p_e x_e = T_i)$
2. Las restricciones correspondientes a I_o y J_o son l.i.

Aplicación de lema de rango

Si x^* es punto extremo de $P(E, I') = \{x \colon x(\delta_E(j)) = 1, \forall j \in J;$
$(px)(\delta_E(i)) \leq T_i, \forall i \in I', x \geq 0, x_e = 0,$ para e imposible$\}$, con todas sus coordenadas positivas, entonces
Existe un conjunto $J_o \subseteq J$ y un conjunto $I_o \subseteq I'$ tal que:

1. $(px)(\delta_E(i)) = T_i$ para todo $i \in I_o$. $\quad (\sum_{e \in \delta_E(i)} p_e x_e = T_i)$
2. Las restricciones correspondientes a I_o y J_o son l.i.
3. $|I_o| + |J_o| = |E|$.

Consecuencia:

Teorema

Si x^* es punto extremo de $P(E, I')$, entonces al menos uno ocurre:

1. $\exists e \in E$ con $x^*_e \in \{0, 1\}$.
2. $\exists i \in I'$ con $d_E(i) = 1$.
3. $\exists i \in I'$ con $d_E(i) = 2$ y $x^*(\delta(i)) \geq 1$.

(dem:) Si ninguna condición ocurre, entonces $x^* > 0$; $d_E(j) \geq 2$, para todo $j \in J$, y todo $i \in I'$. Del lema anterior:

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$,

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$, $d_E(i) = 0$, para $i \in I \setminus I'$,

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$, $d_E(i) = 0$, para $i \in I \setminus I'$, $d_E(v) = 2$ para $v \in I' \cup J$.

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$, $d_E(i) = 0$, para $i \in I \setminus I'$, $d_E(v) = 2$ para $v \in I' \cup J$.

Sea C ciclo en $(I' \cup J, E)$,

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$, $d_E(i) = 0$, para $i \in I \setminus I'$, $d_E(v) = 2$ para $v \in I' \cup J$.

Sea C ciclo en $(I' \cup J, E)$,tenemos que:

$$\sum_{i \in I' \cap V(C)} x^*(\delta(i)) = \sum_{j \in J \cap V(C)} x^*(\delta(j)) = |J \cap V(C)| = |I' \cap V(C)|$$

Teorema (cont.)

$$|J_o| + |I_o| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in I'} d_E(i)}{2} = |J| + |I'| \geq |J_o| + |I_o|$$

Luego, $|J_o| = |J|$, $|I_o| = |I'|$, $d_E(i) = 0$, para $i \in I \setminus I'$, $d_E(v) = 2$ para $v \in I' \cup J$.

Sea C ciclo en $(I' \cup J, E)$,tenemos que:

$$\sum_{i \in I' \cap V(C)} x^*(\delta(i)) = \sum_{j \in J \cap V(C)} x^*(\delta(j)) = |J \cap V(C)| = |I' \cap V(C)|$$

Concluimos que existe $i \in I' \cap V(C)$ con $x^*(\delta(i)) \geq 1$.
(contradicción pues $d_E(i) = 2$).

Algoritmo

Teorema

Si $x^* > 0$ es punto extremo de $P(E, I')$, entonces o existe $e \in E$ con $x^*_e \in \{0, 1\}$, o existe $i \in I'$ con $d_E(i) = 1$, o con $d_E(i) = 2$ y $x^*(\delta(i)) \geq 1$.

(Shmoys y Tardos) Algoritmo

- Fijar $M \leftarrow \emptyset$, $I' \leftarrow I$.
- Repetir hasta que $J = \emptyset$.
  1. Sea x^* punto extremo óptimo de $LP(E, I')$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (reducción:) Si existe $e = (i, j) \in E$ con $x^*_{i,j} = 1$: $M \leftarrow M \cup \{(i, j)\}$, $J \leftarrow J \setminus \{j\}$, $T_i \leftarrow T_i - p_{i,j}$.
  4. (relajación:) Si existe máquina i con $d_E(i) = 1$, o una máquina con $d_E(i) = 2$, entonces $I' \leftarrow I \setminus \{i\}$.
- Devolver M.

Correctitud del algoritmo de Shmoys y Tardos:

El algoritmo iterativo anterior encuentra una asignación M con costo igual o menor al costo del problema original, pero donde cada máquina i se puede usar $2T_i$ unidades de tiempo.

Correctitud del algoritmo de Shmoys y Tardos:

El algoritmo iterativo anterior encuentra una asignación M con costo igual o menor al costo del problema original, pero donde cada máquina i se puede usar $2T_i$ unidades de tiempo.

$|I'| + |E|$ decrece por iteración, luego el algoritmo termina. Sea $y = x^* + \chi(M)$ en cada minuto.

Correctitud del algoritmo de Shmoys y Tardos:

El algoritmo iterativo anterior encuentra una asignación M con costo igual o menor al costo del problema original, pero donde cada máquina i se puede usar $2T_i$ unidades de tiempo.

$|I'| + |E|$ decrece por iteración, luego el algoritmo termina. Sea $y = x^* + \chi(M)$ en cada minuto.

- **Costo correcto**: Sea $G_o = (I \cup J, E_o)$ el grafo inicial y $C_o^* = c^\top x_o^*$ el costo óptimo inicial. En cada instante, $c^\top y = c^\top x^* + c(M) \leq C_o^*$. Luego, al final también $c(M) \leq C_o^*$

Correctitud del algoritmo de Shmoys y Tardos:

El algoritmo iterativo anterior encuentra una asignación M con costo igual o menor al costo del problema original, pero donde cada máquina i se puede usar $2T_i$ unidades de tiempo.

$|I'| + |E|$ decrece por iteración, luego el algoritmo termina. Sea $y = x^* + \chi(M)$ en cada minuto.

- **Costo correcto**: Sea $G_o = (I \cup J, E_o)$ el grafo inicial y $C_o^* = c^\top x_o^*$ el costo óptimo inicial. En cada instante, $c^\top y = c^\top x^* + c(M) \leq C_o^*$. Luego, al final también $c(M) \leq C_o^*$
- **Trabajos asignados**: Para todo $j \in J$, $y(\delta_{E_o}(j)) = x^*(\delta_E(j)) + \mathbb{1}(j \text{ es asignado en } M)$.

Correctitud del algoritmo de Shmoys y Tardos:

El algoritmo iterativo anterior encuentra una asignación M con costo igual o menor al costo del problema original, pero donde cada máquina i se puede usar $2T_i$ unidades de tiempo.

$|I'| + |E|$ decrece por iteración, luego el algoritmo termina. Sea $y = x^* + \chi(M)$ en cada minuto.

- **Costo correcto**: Sea $G_o = (I \cup J, E_o)$ el grafo inicial y $C_o^* = c^\top x_o^*$ el costo óptimo inicial. En cada instante, $c^\top y = c^\top x^* + c(M) \leq C_o^*$. Luego, al final también $c(M) \leq C_o^*$
- **Trabajos asignados**: Para todo $j \in J$, $y(\delta_{E_o}(j)) = x^*(\delta_E(j)) + \mathbb{1}(j \text{ es asignado en } M)$.
- **Tiempo de uso de máquina**: (?)

Correctitud (cont.)

Tomemos una iteración cualquiera. Sea T_i' el tiempo residual y $T_i(M)$ el tiempo ocupado en la máquina i por la asignación M actual.

Afirmación: En las iteraciones en que i no cambia de conjunto en $\{I', I \setminus I'\}$, $T_i' + T_i(M)$ no crece. (fácil)

Correctitud (cont.)

Tomemos una iteración cualquiera. Sea T'_i el tiempo residual y $T_i(M)$ el tiempo ocupado en la máquina i por la asignación M actual.

Afirmación: En las iteraciones en que i no cambia de conjunto en $\{I', I \setminus I'\}$, $T'_i + T_i(M)$ no crece. (fácil)

- Si al final $i \in I'$, $T_i(M) = T'_i + T_i(M) \leq$ cota inicial $= T_i$.

Correctitud (cont.)

Tomemos una iteración cualquiera. Sea T'_i el tiempo residual y $T_i(M)$ el tiempo ocupado en la máquina i por la asignación M actual.

Afirmación: En las iteraciones en que i no cambia de conjunto en $\{I', I \setminus I'\}$, $T'_i + T_i(M)$ no crece. (fácil)

- Si al final $i \in I'$, $T_i(M) = T'_i + T_i(M) \leq$ cota inicial $= T_i$.
- Tomemos el momento en que una máquina i sale de I':

Correctitud (cont.)

Tomemos una iteración cualquiera. Sea T_i' el tiempo residual y $T_i(M)$ el tiempo ocupado en la máquina i por la asignación M actual.

Afirmación: En las iteraciones en que i no cambia de conjunto en $\{I', I \setminus I'\}$, $T_i' + T_i(M)$ no crece. (fácil)

- Si al final $i \in I'$, $T_i(M) = T_i' + T_i(M) \leq$ cota inicial $= T_i$.
- Tomemos el momento en que una máquina i sale de I':
  - Si $d_E(i) = 1$. Hay un sólo j con $x_{ij}^* > 0$. Antes de borrar i, $\quad T_i' + T_i(M) \leq T_i$. Después de borrarlo, sólo j puede ser asignado a M. Luego al final, $T_i(M) \leq T_i + p_{ij} \leq 2T_i$.

Correctitud (cont.)

Tomemos una iteración cualquiera. Sea T'_i el tiempo residual y $T_i(M)$ el tiempo ocupado en la máquina i por la asignación M actual.

Afirmación: En las iteraciones en que i no cambia de conjunto en $\{I', I \setminus I'\}$, $T'_i + T_i(M)$ no crece. (fácil)

- Si al final $i \in I'$, $T_i(M) = T'_i + T_i(M) \leq$ cota inicial $= T_i$.
- Tomemos el momento en que una máquina i sale de I':
  - Si $d_E(i) = 1$. Hay un sólo j con $x^*_{ij} > 0$. Antes de borrar i, $\quad T'_i + T_i(M) \leq T_i$. Después de borrarlo, sólo j puede ser asignado a M. Luego al final, $T_i(M) \leq T_i + p_{ij} \leq 2T_i$.
  - Si $d_E(i) = 2$. Hay j_1, j_2 con $x^*_{ij_1} > 0, x^*_{ij_1} > 0$ y $x^*_{ij_1} + x^*_{ij_2} \geq 1$. Luego al final,

$$\begin{aligned} T_i(M) &\leq T_i - T'_i + p_{ij_1} + p_{ij_2} \\ &\leq T_i + (1 - x^*_{ij_1})p_{ij_1} + (1 - x^*_{ij_2})p_{ij_2} \\ &\leq T_i + (2 - x^*_{ij_1} - x^*_{ij_2})T_i \leq 2T_i. \qquad \square \end{aligned}$$

Resumen

Redondeo (NP-difícil) / *Relajación* (P) – Problema de asignación

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

Resumen

Redondeo (NP-difícil) / *Relajación* (P) – Problema de asignación

- **Mientras problema actual no resuelto**
  - Encontrar punto extremo del PL actual.
  - Argumentar que existe una coordenada con valor x_i suficientemente alto (*o 1*).
  - Redondear dicho x_i a 1.
  - Eliminar variables 0-1 y repetir.

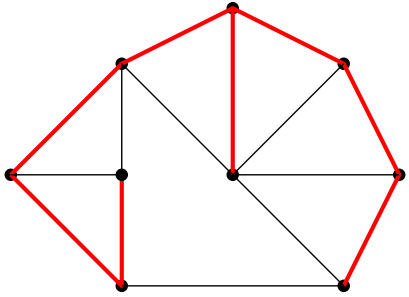
Relajación iterativa (NP-difícil) – Problema de asignación general

- - Encontrar problema base en P con puntos extremos integrales.
  - Agregar restricciones al problema base e intentar esquema anterior.
  - cambio: Demostrar que existe una coordenada con alto valor x_i (redondear) o una restricción "poco violada" (relajar).

Outline

Árboles cubridores mínimos

Dado un grafo $G = (V, E)$ conexo y $c\colon E \to \mathbb{R}$, costos en las aristas. Queremos encontrar un subgrafo $T = (V, F)$ **acíclico y conexo** (árbol) que toque todos los vértices y de costo total minimo.



Primer modelo: modelo de corte

- Supongamos que $c_e \geq 0$ por simpleza.

Primer modelo: modelo de corte

- Supongamos que $c_e \geq 0$ por simpleza.
- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.

Primer modelo: modelo de corte

- Supongamos que $c_e \geq 0$ por simpleza.
- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.
- Si $\emptyset \subsetneq S \subsetneq V$ es un conjunto de nodos, debe haber al menos una arista que cruce de S a $V \setminus S$:

$$x(\delta(S)) \geq 1.$$

Modelo de corte

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &x(\delta(S)) \geq 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\ &x \in \{0,1\}^E. \end{aligned}$$

Modelo de corte

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &x(\delta(S)) \geq 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\ &x \in \{0,1\}^E. \end{aligned}$$

Ventajas: Formulación simple (pero exponencial en el tamaño del grafo).
Se puede implementar oraculo de separación para su relajación lineal natural (corte mínimo se puede resolver en tiempo polinomial).

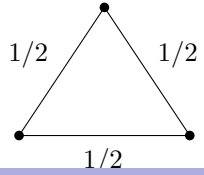
Modelo de corte

Programa Entero

$$\begin{aligned} &\text{mín } c^\top x \\ \text{s.a. } &x(\delta(S)) \geq 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\ &x \in \{0,1\}^E. \end{aligned}$$

Ventajas: Formulación simple (pero exponencial en el tamaño del grafo).
Se puede implementar oraculo de separación para su relajación lineal natural (corte mínimo se puede resolver en tiempo polinomial).

Desventajas: Relajación lineal NO integral.



Segundo modelo: modelo de eliminación de ciclos (o subtour)

- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.

Segundo modelo: modelo de eliminación de ciclos (o subtour)

- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.
- Si $\emptyset \subsetneq S \subsetneq V$ es un conjunto de nodos, llamemos $E(S)$ a las aristas con ambos extremos en S.

Segundo modelo: modelo de eliminación de ciclos (o subtour)

- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.
- Si $\emptyset \subsetneq S \subsetneq V$ es un conjunto de nodos, llamemos $E(S)$ a las aristas con ambos extremos en S.
- Para todo S no podemos tomar más que $|S| - 1$ aristas en $E(S)$ (si no, formamos algún ciclo).

Segundo modelo: modelo de eliminación de ciclos (o subtour)

- $x_e \in \{0, 1\}$ representa las aristas de T, por lo que queremos minimizar $c^\top x$.
- Si $\emptyset \subsetneq S \subsetneq V$ es un conjunto de nodos, llamemos $E(S)$ a las aristas con ambos extremos en S.
- Para todo S no podemos tomar más que $|S| - 1$ aristas en $E(S)$ (si no, formamos algún ciclo).
- En total debemos tomar $|V| - 1$ aristas.

Modelo de eliminación de ciclos

Programa Entero

$$
\begin{aligned}
& \text{mín } c^\top x \\
\text{s.a. } & x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\
& x(E(V)) = |V| - 1, \\
& x \in \{0,1\}^E.
\end{aligned}
$$

Modelo de eliminación de ciclos

Programa Entero

$$
\begin{aligned}
&\text{mín } c^\top x \\
\text{s.a. } &x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\
&x(E(V)) = |V| - 1, \\
&x \in \{0,1\}^E.
\end{aligned}
$$

Formulación simple y exponencial.
Se puede implementar oraculo de separación para su relajación (más complicado).

Modelo de eliminación de ciclos

Programa Entero

$$\begin{aligned} & \text{mín } c^\top x \\ \text{s.a. } & x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\ & x(E(V)) = |V| - 1, \\ & x \in \{0,1\}^E. \end{aligned}$$

Formulación simple y exponencial.
Se puede implementar oraculo de separación para su relajación (más complicado).

Teorema

La relajación lineal natural es integral.

Probaremos esto usando métodos iterativos.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

- Sea x un punto extremo de $P_{a.c.}$.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

- Sea x un punto extremo de $P_{a.c.}$.
- Sea $\mathcal{J}(x) = \{S \subseteq V \colon x(E(S)) = |S| - 1\}$ las desigualdades ajustadas para x.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

- Sea x un punto extremo de $P_{a.c.}$.
- Sea $\mathcal{J}(x) = \{S \subseteq V \colon x(E(S)) = |S| - 1\}$ las desigualdades ajustadas para x.
- Sea $\mathrm{span}(\mathcal{J}(x))$ al espacio vectorial generado por $\{\chi(E(S)) \colon S \in \mathcal{J}\}$.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

- Sea x un punto extremo de $P_{a.c.}$.
- Sea $\mathcal{J}(x) = \{S \subseteq V \colon x(E(S)) = |S| - 1\}$ las desigualdades ajustadas para x.
- Sea $\mathrm{span}(\mathcal{J}(x))$ al espacio vectorial generado por $\{\chi(E(S)) \colon S \in \mathcal{J}\}$.
- Cualquier base (es decir, conjunto l.i. maximal) de $\mathcal{J}(x)$ genera $\mathrm{span}(\mathcal{J}(x))$.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

- Sea x un punto extremo de $P_{a.c.}$.
- Sea $\mathcal{J}(x) = \{S \subseteq V \colon x(E(S)) = |S| - 1\}$ las desigualdades ajustadas para x.
- Sea $\mathrm{span}(\mathcal{J}(x))$ al espacio vectorial generado por $\{\chi(E(S)) \colon S \in \mathcal{J}\}$.
- Cualquier base (es decir, conjunto l.i. maximal) de $\mathcal{J}(x)$ genera $\mathrm{span}(\mathcal{J}(x))$.
- En general, $\mathrm{span}(\mathcal{J}(x))$ tiene muchísimas bases pero son difíciles de tratar. Si pudiéramos saber cual es el tamaño de una base podríamos usar el lema de rango para entender cuantas coordenadas no 0 tiene x.

Puntos extremos de $P_{a.c.}$

Sea $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$.

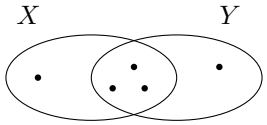
- Sea x un punto extremo de $P_{a.c.}$.
- Sea $\mathcal{J}(x) = \{S \subseteq V \colon x(E(S)) = |S| - 1\}$ las desigualdades ajustadas para x.
- Sea $\operatorname{span}(\mathcal{J}(x))$ al espacio vectorial generado por $\{\chi(E(S)) \colon S \in \mathcal{J}\}$.
- Cualquier base (es decir, conjunto l.i. maximal) de $\mathcal{J}(x)$ genera $\operatorname{span}(\mathcal{J}(x))$.
- En general, $\operatorname{span}(\mathcal{J}(x))$ tiene muchísimas bases pero son difíciles de tratar. Si pudiéramos saber cual es el tamaño de una base podríamos usar el lema de rango para entender cuantas coordenadas no 0 tiene x.
- Para estimar el tamaño de una base usaremos una técnica conocida como descruce.

Descruce (i): Supermodularidad

Lema

Para todo $X, Y \subseteq V$,

$$\chi(E(X)) + \chi(E(Y)) \leq \chi(E(X \cup Y)) + \chi(E(X \cap Y))$$



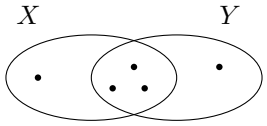
(dem): Pizarra.

Descruce (i): Supermodularidad

Lema

Para todo $X, Y \subseteq V$,

$$\chi(E(X)) + \chi(E(Y)) \leq \chi(E(X \cup Y)) + \chi(E(X \cap Y))$$



(dem): Pizarra.

Corolario:

$$x(E(X)) + x(E(Y)) \leq x(E(X \cup Y)) + x(E(X \cap Y))$$

Descruce (ii)

Recordemos que $\mathcal{J}(x) = \{S \subseteq V : x(E(S)) = |S| - 1\}$ son las desigualdades ajustadas de un punto extremo x de $P_{a.c.}$.

Si $x > 0$, $X, Y \in \mathcal{J}(x)$ y $X \cap Y \neq \emptyset$ entonces

1. $X \cap Y$, $X \cup Y$ están en $\mathcal{J}(x)$.
2. $\chi(E(X)) + \chi(E(Y)) = \chi(E(X \cup Y)) + \chi(E(X \cap Y))$.

Descruce (ii)

Recordemos que $\mathcal{J}(x) = \{S \subseteq V : x(E(S)) = |S| - 1\}$ son las desigualdades ajustadas de un punto extremo x de $P_{a.c.}$.

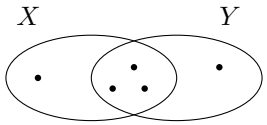
Si $x > 0$, $X, Y \in \mathcal{J}(x)$ y $X \cap Y \neq \emptyset$ entonces

1. $X \cap Y$, $X \cup Y$ están en $\mathcal{J}(x)$.
2. $\chi(E(X)) + \chi(E(Y)) = \chi(E(X \cup Y)) + \chi(E(X \cap Y))$.

(dem:) $$\begin{aligned} |X| - 1 + |Y| - 1 &= x(E(X)) + x(E(Y)) \\ &\leq x(E(X \cup Y)) + x(E(X \cap Y)) \\ &\leq |X \cup Y| - 1 + |X \cap Y| - 1 = |X| - 1 + |Y| - 1. \end{aligned}$$

Descruce (iii)

Dos conjuntos X e Y en V son intersectantes si $X \setminus Y$, $Y \setminus X$ y $X \cap Y \neq \emptyset$.

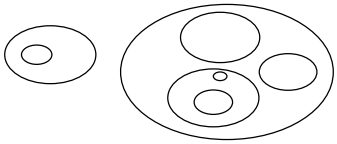


- Si X e Y son ajustados para x e intersectantes entonces $X \cup Y$ y $X \cap Y$ también son ajustados y las 4 restricciones asociadas son l.d.
- Idea: Eliminar intersecciones (descruzar).

Definición: Familia laminar

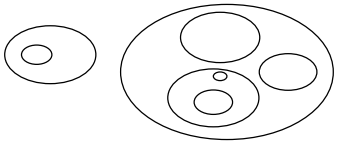
Una familia $\mathcal{L}$ de subconjuntos de V se llama laminar si no contiene dos conjuntos intersectantes.

Descruce (iv): Familias laminares



Sean $X, Y \in \mathcal{L}$ familia laminar. Si $X \cap Y \neq \emptyset$ entonces $X \subseteq Y$ o $Y \subseteq X$.

Descruce (iv): Familias laminares



Sean $X, Y \in \mathcal{L}$ familia laminar. Si $X \cap Y \neq \emptyset$ entonces $X \subseteq Y$ o $Y \subseteq X$.

Lemma: Laminar maximal es generador

Sea $\mathcal{L}$ una subfamilia laminar de $\mathcal{J}(x)$ maximal con $x > 0$ punto extremo. $\mathrm{span}(\mathcal{L}) = \mathrm{span}(\mathcal{J}(x))$.

- Supongamos que $\mathrm{span}(\mathcal{L}) \subsetneq \mathrm{span}(\mathcal{J}(x))$.
- Para $S \in \mathcal{J}(x) \setminus \mathcal{L}$, llamemos

$$\text{inter}(S, \mathcal{L}) = \#\{X \in \mathcal{L} \colon X \text{ y } S \text{ son intersectantes } \}$$

- Como $\mathrm{span}(\mathcal{L}) \subsetneq \mathrm{span}(\mathcal{J}(x))$, existe $S \in \mathcal{J}(x))$ con $\chi(E(S)) \notin \mathrm{span}(\mathcal{L})$.
- Elijamos S_o como aquel con mínimo inter$(S, \mathcal{L})$.
- Como $\mathcal{L} \cup \{S_o\}$ no es laminar, inter$(S_o, \mathcal{L}) \geq 1$.
- Sea T en $\mathcal{L}$ tal que S_o y T son intersectantes.
- Probaremos luego, que inter$(S_o \cap T, \mathcal{L})$ y inter$(S_o \cup T, \mathcal{L})$ son menores estrictos que inter$(S_o, \mathcal{L})$, con lo que $S_o \cap T$ y $S_o \cup T$ están en $\mathcal{L}$.

Laminar maximal es generador (2)

- Luego $S_o \cap T$, T y $S_o \cup T$ están en $\mathcal{L}$.
- Pero sabemos que $\chi(E(S_o)) = \chi(E(S_o \cup T)) + \chi(E(S_o \cap T)) - \chi(E(T))$.
- Es decir $S_o \in \mathrm{span}(\mathcal{L})$, lo que es una contradicción. $\square$

Laminar maximal es generador (2)

- Luego $S_o \cap T$, T y $S_o \cup T$ están en $\mathcal{L}$.
- Pero sabemos que $\chi(E(S_o)) = \chi(E(S_o \cup T)) + \chi(E(S_o \cap T)) - \chi(E(T))$.
- Es decir $S_o \in \mathrm{span}(\mathcal{L})$, lo que es una contradicción. $\square$

Falta aún probar que

inter$(S_o \cap T, \mathcal{L})$ y inter$(S_o \cup T, \mathcal{L})$ son menores estrictos que inter$(S_o, \mathcal{L})$

- Luego $S_o \cap T$, T y $S_o \cup T$ están en $\mathcal{L}$.
- Pero sabemos que $\chi(E(S_o)) = \chi(E(S_o \cup T)) + \chi(E(S_o \cap T)) - \chi(E(T))$.
- Es decir $S_o \in \mathrm{span}(\mathcal{L})$, lo que es una contradicción. $\square$

Falta aún probar que

inter$(S_o \cap T, \mathcal{L})$ y inter$(S_o \cup T, \mathcal{L})$ son menores estrictos que inter$(S_o, \mathcal{L})$

(dem:) Sea $R \in \mathcal{L}$ con $R \neq T$.

- $R \subsetneq T$ o $T \subsetneq R$ o $R \cap T \neq \emptyset$

- Luego $S_o \cap T$, T y $S_o \cup T$ están en $\mathcal{L}$.
- Pero sabemos que $\chi(E(S_o)) = \chi(E(S_o \cup T)) + \chi(E(S_o \cap T)) - \chi(E(T))$.
- Es decir $S_o \in \mathrm{span}(\mathcal{L})$, lo que es una contradicción. $\square$

Falta aún probar que

inter$(S_o \cap T, \mathcal{L})$ y inter$(S_o \cup T, \mathcal{L})$ son menores estrictos que inter$(S_o, \mathcal{L})$

(dem:) Sea $R \in \mathcal{L}$ con $R \neq T$.

- $R \subsetneq T$ o $T \subsetneq R$ o $R \cap T \neq \emptyset$
- ($\leq$): Si R es intersectante con $S_o \cap T$ (o $S_o \cup T$) entonces también lo es con S_o.

- Luego $S_o \cap T$, T y $S_o \cup T$ están en $\mathcal{L}$.
- Pero sabemos que $\chi(E(S_o)) = \chi(E(S_o \cup T)) + \chi(E(S_o \cap T)) - \chi(E(T))$.
- Es decir $S_o \in \mathrm{span}(\mathcal{L})$, lo que es una contradicción. $\square$

Falta aún probar que

inter$(S_o \cap T, \mathcal{L})$ y inter$(S_o \cup T, \mathcal{L})$ son menores estrictos que inter$(S_o, \mathcal{L})$

(dem:) Sea $R \in \mathcal{L}$ con $R \neq T$.

- $R \subsetneq T$ o $T \subsetneq R$ o $R \cap T \neq \emptyset$
- ($\leq$): Si R es intersectante con $S_o \cap T$ (o $S_o \cup T$) entonces también lo es con S_o.
- ($<$): T no es intersectante con $S_o \cup T$ o $S_o \cap T$. $\square$.

Resumen (descruce)

$$P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}.$$

$\mathcal{J}(x)$: restricciones ajustadas para x.

- Existe $\mathcal{L}$ laminar que genera $\mathcal{J}(x)$.

Resumen (descruce)

$$P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}.$$

$\mathcal{J}(x)$: restricciones ajustadas para x.

- Existe $\mathcal{L}$ laminar que genera $\mathcal{J}(x)$.
- Botando restricciones l.d. podemos asumir que $\mathcal{L}$ es base de $\mathrm{span}(\mathcal{J}(x))$.

Resumen (descruce)

$$P_{a.c.} = \{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}.$$

$\mathcal{J}(x)$: restricciones ajustadas para x.

- Existe $\mathcal{L}$ laminar que genera $\mathcal{J}(x)$.
- Botando restricciones l.d. podemos asumir que $\mathcal{L}$ es base de $\mathrm{span}(\mathcal{J}(x))$.
- **Lema de rango:**
  Si $x > 0$, el número de coordenadas de x es igual a $|\mathcal{L}|$.

Resumen (descruce)

$$P_{a.c.} = \{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}.$$

$\mathcal{J}(x)$: restricciones ajustadas para x.

- Existe $\mathcal{L}$ laminar que genera $\mathcal{J}(x)$.
- Botando restricciones l.d. podemos asumir que $\mathcal{L}$ es base de $\mathrm{span}(\mathcal{J}(x))$.
- **Lema de rango:**
  Si $x > 0$, el número de coordenadas de x es igual a $|\mathcal{L}|$.
- ¿Cuán grande puede ser $|\mathcal{L}|$?

Resumen (descruce)

$$P_{a.c.} = \{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}.$$
$$\mathcal{J}(x): \text{ restricciones ajustadas para } x.$$

- Existe $\mathcal{L}$ laminar que genera $\mathcal{J}(x)$.
- Botando restricciones l.d. podemos asumir que $\mathcal{L}$ es base de $\mathrm{span}(\mathcal{J}(x))$.
- **Lema de rango:**
  Si $x > 0$, el número de coordenadas de x es igual a $|\mathcal{L}|$.
- ¿Cuán grande puede ser $|\mathcal{L}|$?
- Todos los conjuntos en $\mathcal{L}$ tienen al menos 2 elementos de V. Por inducción se puede probar (ejercicio) que
$$|\mathcal{L}| \leq |V| - 1.$$

Puntos extremos positivos tienen hojas

Sea $x \in \mathbb{R}^E$ con $x_e > 0$ para todo e, punto extremo de $P_{a.c.} = \{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$, donde $G = (V, E)$ es conexo.

Teorema

Existe $v \in V$ con $d(v) = 1$.

Puntos extremos positivos tienen hojas

Sea $x \in \mathbb{R}^E$ con $x_e > 0$ para todo e, punto extremo de $P_{a.c.} = \{x \in \mathbb{R}^E \colon x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$, donde $G = (V, E)$ es conexo.

Teorema

Existe $v \in V$ con $d(v) = 1$.

(dem:) Si todo v tuviera grado ≥ 2, entonces

$$|E| = \frac{1}{2} \sum_{v \in V} d(v) \geq |V|.$$

Pero por lema de rango, $|E| = |\mathcal{L}| \leq |V| - 1$. $\square$

Algoritmo para árbol cubridor de peso mínimo

Algoritmo busca hojas:

- $F \leftarrow \emptyset$.
- Repetir hasta que $V = \emptyset$.
  - Encontrar punto extremo óptimo x del problema actual.
  - Borrar aristas e con $x_e = 0$.
  - Encontrar $v \in V$ con una sola arista vw incidente. $F \leftarrow F \cup \{vw\}$, $V \leftarrow V \setminus \{v\}$.
- Entregar (V, F).

Teorema

El algoritmo busca hojas encuentra un árbol de costo mínimo en tiempo polinomial.

1.- Factibilidad (F es árbol): inducción reversa.
2.- Costo. Se puede probar que si e se agrega a F entonces $x_e = 1$, luego lo que decrece la solución fraccional es igual a lo que crece la solución entera.

Consecuencias:

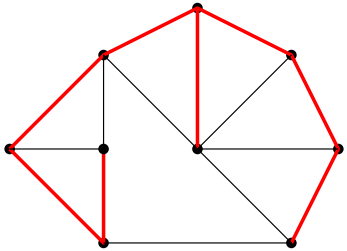
- Todos los puntos extremos de $P_{a.c.} = \{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x \geq 0\}$. son integrales.
- No necesitamos iterar!

Al igual que para el problema de asignación, una de las ventajas de usar una demostración de integralidad iterativa es que problemas NP-difíciles similares se pueden aproximar usando la misma idea de relajación iterativa.

Outline

Árboles cubridores mínimos con grado acotado (acmga)

Dado un grafo $G = (V, E)$ conexo, costos $c\colon E \to \mathbb{R}$ en las aristas y valores $B_v \geq 0$ en los vértices. Queremos encontrar un árbol cubridor $T = (V, F)$ con $d_F(v) \leq B_v$, de mínimo costo posible



Modelo Entero

Programa Entero para (acmga)

$$
\begin{aligned}
& \text{mín } c^\top x \\
\text{s.a. } & x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\
& x(E(V)) = |V| - 1, \\
& x(\delta(v)) \leq B_v, \quad \forall v \in V, \\
& x \in \{0,1\}^E.
\end{aligned}
$$

Modelo Entero

Programa Entero para (acmga)

$$
\begin{aligned}
&\text{mín } c^\top x \\
\text{s.a. } \quad & x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\
& x(E(V)) = |V| - 1, \\
& x(\delta(v)) \leq B_v, \quad \forall v \in V, \\
& x \in \{0,1\}^E.
\end{aligned}
$$

- El problema es NP-difícil por lo que suponemos que no hay algoritmo polinomial.
- La relajación natural NO es integral.

Aproximación

Supongamos que $C \in \mathbb{R}$ es el costo de una solución óptima.

- Goemans (2006) dio un algoritmo para encontrar un árbol de costo menor o igual que C pero donde las restricciones de grado se cambian por $B_v + 2$.

Aproximación

Supongamos que $C \in \mathbb{R}$ es el costo de una solución óptima.

- Goemans (2006) dio un algoritmo para encontrar un árbol de costo menor o igual que C pero donde las restricciones de grado se cambian por $B_v + 2$.
- Singh y Lau (2007) dieron uno donde las restricciones se cambian por $B_v + 1$.

Relajación

Para B vector de cotas de grado, $W \subseteq V$, llamemos $PL(G, B, W)$ al programa lineal:

$$\begin{aligned} & \text{mín } c^\top x \\ \text{s.a. } & x(E(S)) \leq |S| - 1, \quad \forall S \in 2^V \setminus \{\emptyset, V\} \\ & x(E(V)) = |V| - 1, \\ & x(\delta(v)) \leq B_v, \quad \forall v \in W, \\ & x \geq 0. \end{aligned}$$

y llamemos $P(G, B, W)$ al poliedro
$\{x \in \mathbb{R}^E : x(E(S)) \leq |S| - 1, \forall S; x(E) = |V| - 1; x(\delta(v)) \leq B_v, \forall v \in W; x \geq 0\}$.

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

1. $x(\delta(S)) = |S| - 1$, para todo $S \in \mathcal{L}$.
   $x(\delta(v)) = B_v$, para todo $v \in T$.

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

1. $x(\delta(S)) = |S| - 1$, para todo $S \in \mathcal{L}$.
   $x(\delta(v)) = B_v$, para todo $v \in T$.
2. $\{\chi(E(S)) \colon S \in \mathcal{L}\} \cup \{\chi(\delta(v)) \colon v \in T\}$ son l.i.

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

1. $x(\delta(S)) = |S| - 1$, para todo $S \in \mathcal{L}$.
   $x(\delta(v)) = B_v$, para todo $v \in T$.
2. $\{\chi(E(S)) \colon S \in \mathcal{L}\} \cup \{\chi(\delta(v)) \colon v \in T\}$ son l.i.
3. $|T| + |\mathcal{L}| = |E|$.

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

1. $x(\delta(S)) = |S| - 1$, para todo $S \in \mathcal{L}$.
   $x(\delta(v)) = B_v$, para todo $v \in T$.
2. $\{\chi(E(S)) \colon S \in \mathcal{L}\} \cup \{\chi(\delta(v)) \colon v \in T\}$ son l.i.
3. $|T| + |\mathcal{L}| = |E|$.

Aplicación de lema de rango

Si x^* es punto extremo de $P(G, B, W)$ con todas sus coordenadas positivas, entonces Existe un conjunto $T \subseteq W$ y una familia $\mathcal{L} \subseteq 2^V$ laminar tal que:

1. $x(\delta(S)) = |S| - 1$, para todo $S \in \mathcal{L}$.
   $x(\delta(v)) = B_v$, para todo $v \in T$.
2. $\{\chi(E(S)) \colon S \in \mathcal{L}\} \cup \{\chi(\delta(v)) \colon v \in T\}$ son l.i.
3. $|T| + |\mathcal{L}| = |E|$.

Gracias a esto probaremos el siguiente lema:

Si x^* es como antes, entonces una de los siguientes ocurre:

- Existe un vértice $v \in V$ con $d(v) = 1$.
- Existe un vértice $v \in W$ con $d(v) \leq 3$.

Demostración del lema

Si ninguna condición se satisface: $d(v) \geq 2$ para $v \in V$ y $d(v) \geq 4$ para $v \in W$.
Luego,

$$|E| = \frac{1}{2} \sum_{v \in V} d(v) \geq |V \setminus W| + 2|W| = |V| + |W|.$$

Demostración del lema

Si ninguna condición se satisface: $d(v) \geq 2$ para $v \in V$ y $d(v) \geq 4$ para $v \in W$.
Luego,

$$|E| = \frac{1}{2} \sum_{v \in V} d(v) \geq |V \setminus W| + 2|W| = |V| + |W|.$$

Pero del lema de rango y la cota para familia laminares sin singletons, sabemos que

$$|E| = |T| + |\mathcal{L}| \leq |W| + |V| - 1, \quad \square$$

Algoritmo iterativo con relajación aditiva de 2 unidades.

(Goemans) Algoritmo

- Fijar $W \leftarrow V$.
- Mientras $|V| \geq 2$.
  1. Sea x^* punto extremo óptimo de $LP(G, B, W)$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (reducción:) Si existe $v \in V$ con una sola arista $e = vw$ incidente: $F \leftarrow F \cup \{vw\}$, $V \leftarrow V \setminus \{v\}$, $W \leftarrow W \setminus \{v\}$, $B_w \leftarrow B_w - 1$.
  4. (relajación:) Si existe $v \in W$ con $d(v) \leq 3$, fijar $W \leftarrow W \setminus \{v\}$.
- Devolver (V, F).

Correctitud del algoritmo de Goemans:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 2$.

Correctitud del algoritmo de Goemans:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 2$.

$|E| + |V| + |W|$ decrece por iteración, luego el algoritmo termina cuando $|V| < 2$ (sin aristas). Sea $y = x^* + \chi(F)$ en cada minuto.

Correctitud del algoritmo de Goemans:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 2$.

$|E| + |V| + |W|$ decrece por iteración, luego el algoritmo termina cuando $|V| < 2$ (sin aristas). Sea $y = x^* + \chi(F)$ en cada minuto.

- **Costo correcto**: El costo $c^\top y$ solo puede decrecer.

Correctitud del algoritmo de Goemans:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 2$.

$|E| + |V| + |W|$ decrece por iteración, luego el algoritmo termina cuando $|V| < 2$ (sin aristas). Sea $y = x^* + \chi(F)$ en cada minuto.

- **Costo correcto**: El costo $c^\top y$ solo puede decrecer.
- **Factible (árbol)**: Por inducción (en reverso: siempre agregamos hojas)

Correctitud del algoritmo de Goemans:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 2$.

$|E| + |V| + |W|$ decrece por iteración, luego el algoritmo termina cuando $|V| < 2$ (sin aristas). Sea $y = x^* + \chi(F)$ en cada minuto.

- **Costo correcto**: El costo $c^\top y$ solo puede decrecer.
- **Factible (árbol)**: Por inducción (en reverso: siempre agregamos hojas)
- **Cota de grados**: (?)

Cota de grados

En cada iteración donde $v \in W$, llamemos B'_v a la cota de grado residual actual. Es fácil ver que siempre $B'_v + d_F(v) = B_v$. Cuando v sale de W, el grafo residual tiene a lo más 3 aristas incidentes. Luego lo peor que puede pasar es que las tres sean elegidas y en ese caso $d_F(v) \leq B_v - B'_v + 3 \leq B_v + 2$.

Algoritmo iterativo con relajación aditiva de 1 unidades.

(Singh y Lau) Algoritmo

- Fijar $W \leftarrow V$.
- Mientras $W \neq \emptyset$.
  1. Sea x^* punto extremo óptimo de $LP(G, B, W)$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (relajación:) Si existe $v \in W$ con $d(v) \leq B_v + 1$, fijar $W \leftarrow W \setminus \{v\}$.
- Devolver (V, F).

Algoritmo iterativo con relajación aditiva de 1 unidades.

(Singh y Lau) Algoritmo

- Fijar $W \leftarrow V$.
- Mientras $W \neq \emptyset$.
  1. Sea x^* punto extremo óptimo de $LP(G, B, W)$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (relajación:) Si existe $v \in W$ con $d(v) \leq B_v + 1$, fijar $W \leftarrow W \setminus \{v\}$.
- Devolver (V, F).

Teorema (Singh y Lau)

Si x^* es punto extremo óptimo de $LP(G, B, W)$ positivo y $W \neq \emptyset$, entonces existe $v \in W$ con $d(v) \leq B_v + 1$.

Demostración del teorema de Singh y Lau

Método de la moneda fraccional

- Supongamos por contradicción que para todo $v \in W$, $d(v) \geq B_v + 2$. Sea T y $\mathcal{L}$ como en el lema de rango.
- Dar una moneda a cada $e \in E$.
- Cada $e \in E$ le da
  1. x_e monedas al menor conjunto $S \in \mathcal{L}$ que contiene ambos extremos.
  2. $\frac{1-x_e}{2}$ monedas a sus dos extremos u y v.
- Probemos que: Cada $S \in \mathcal{L}$ y cada $v \in T$ recibe al menos una moneda, y que "sobran monedas".
- Con esto $|E| > |\mathcal{L}| + |T|$, lo que contradice el lema de rango.

Cada $S \in \mathcal{L}$ recibe al menos una moneda.

Sea $S \in \mathcal{L}$, $R_1, \ldots, R_k$ sus "hijos" en $\mathcal{L}$. Sea $A = E(S) \setminus \bigcup_{i=1}^{k} E(R_i)$. S recibe exactamente $x(A) = \sum_{e \in A} x(e)$ monedas.

Cada $S \in \mathcal{L}$ recibe al menos una moneda.

Sea $S \in \mathcal{L}$, $R_1, \ldots, R_k$ sus "hijos" en $\mathcal{L}$. Sea $A = E(S) \setminus \bigcup_{i=1}^{k} E(R_i)$. S recibe exactamente $x(A) = \sum_{e \in A} x(e)$ monedas.Como todos los conjuntos de $\mathcal{L}$ son ajustados, tenemos que:

$$\begin{aligned} x(A) &= x(E(S)) - \sum_{i=1}^{k} x(E(R_i)) \\ &= |S| - 1 - \sum_{i=1}^{k} (|R_i| - 1) = (|S| - \sum_{i=1}^{k} |R_i|) + (k-1) \in \mathbb{Z} \end{aligned}$$

Cada $S \in \mathcal{L}$ recibe al menos una moneda.

Sea $S \in \mathcal{L}$, $R_1, \ldots, R_k$ sus "hijos" en $\mathcal{L}$. Sea $A = E(S) \setminus \bigcup_{i=1}^{k} E(R_i)$. S recibe exactamente $x(A) = \sum_{e \in A} x(e)$ monedas.Como todos los conjuntos de $\mathcal{L}$ son ajustados, tenemos que:

$$\begin{aligned} x(A) &= x(E(S)) - \sum_{i=1}^{k} x(E(R_i)) \\ &= |S| - 1 - \sum_{i=1}^{k} (|R_i| - 1) = (|S| - \sum_{i=1}^{k} |R_i|) + (k-1) \in \mathbb{Z} \end{aligned}$$

Luego $x(A)$ es un entero no negativo. Pero si $x(A) = 0$ entonces $\chi(E(S)) = \sum_{i=1}^{k} \chi(E(R_i))$ (contradice l.i.) Por lo tanto $x(A) \geq 1$.

Cada $v \in T$ recibe al menos una moneda.

Cada $e = uw$ le da $(1 - x_e)/2$ monedas a sus extremos. Todo $v \in T$ satisface $x(\delta(v)) = B_v$. Además, supusimos que $d(v) \geq B_v + 2$ para todo $v \in W \supseteq T$.

Cada $v \in T$ recibe al menos una moneda.

Cada $e = uw$ le da $(1 - x_e)/2$ monedas a sus extremos. Todo $v \in T$ satisface $x(\delta(v)) = B_v$. Además, supusimos que $d(v) \geq B_v + 2$ para todo $v \in W \supseteq T$. Luego cada $v \in T$ recibe

$$\sum_{e \in \delta(v)} \frac{1 - x_e}{2} = \frac{d(v) - B_v}{2} \geq 1.$$

monedas.

Sobran monedas

- Si V no es parte de $\mathcal{L}$ entonces como E es conexo, alguna arista $e \in E$ no pertenece a ningún conjunto de $\mathcal{L}$ luego su parte x_e "se pierde". Concluimos que $V \in \mathcal{L}$.

Sobran monedas

- Si V no es parte de $\mathcal{L}$ entonces como E es conexo, alguna arista $e \in E$ no pertenece a ningún conjunto de $\mathcal{L}$ luego su parte x_e "se pierde". Concluimos que $V \in \mathcal{L}$.
- Todas las aristas e incidentes en $v \in V \setminus T$ deben tener $x_e = 1$ (de otro modo, v recibiría $(1 - x_e)/2$ de alguna de ellas, lo que se "pierde".)

Sobran monedas

- Si V no es parte de $\mathcal{L}$ entonces como E es conexo, alguna arista $e \in E$ no pertenece a ningún conjunto de $\mathcal{L}$ luego su parte x_e "se pierde". Concluimos que $V \in \mathcal{L}$.
- Todas las aristas e incidentes en $v \in V \setminus T$ deben tener $x_e = 1$ (de otro modo, v recibiría $(1 - x_e)/2$ de alguna de ellas, lo que se "pierde".)
- Notar que si $x_e = 1$ y $e = uv$, entonces $\{u, v\}$ es ajustado y luego $E(\{u, v\}) \in \operatorname{span}(\mathcal{L})$.

Sobran monedas

- Si V no es parte de $\mathcal{L}$ entonces como E es conexo, alguna arista $e \in E$ no pertenece a ningún conjunto de $\mathcal{L}$ luego su parte x_e "se pierde". Concluimos que $V \in \mathcal{L}$.
- Todas las aristas e incidentes en $v \in V \setminus T$ deben tener $x_e = 1$ (de otro modo, v recibiría $(1 - x_e)/2$ de alguna de ellas, lo que se "pierde".)
- Notar que si $x_e = 1$ y $e = uv$, entonces $\{u, v\}$ es ajustado y luego $E(\{u, v\}) \in \operatorname{span}(\mathcal{L})$.
- Concluimos que:

$$\begin{aligned} 2\chi(E(V)) &= \sum_{v \in V} \chi(\delta(v)) = \sum_{v \in T} \chi(\delta(v)) + \sum_{v \in V \setminus T} \chi(\delta(v)) \\ &= \sum_{v \in T} \chi(\delta(v)) + \sum_{v \in V \setminus T} \sum_{e = uv \in \delta(v)} \chi(E(\{u, v\}). \end{aligned}$$

Esto contradice la independencia lineal de $\{\chi(E(S)) \colon S \in \mathcal{L}\} \cup \{\chi(\delta(v) \colon v \in T)\}$.

De vuelta al algoritmo de Singh y Lau

(Singh y Lau) Algoritmo

- Fijar $W \leftarrow V$.
- Mientras $W \neq \emptyset$.
  1. Sea x^* punto extremo óptimo de $LP(G, B, W)$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (relajación:) Si existe $v \in W$ con $d(v) \leq B_v + 1$, fijar $W \leftarrow W \setminus \{v\}$.
- Devolver (V, E).

De vuelta al algoritmo de Singh y Lau

(Singh y Lau) Algoritmo

- Fijar $W \leftarrow V$.
- Mientras $W \neq \emptyset$.
  1. Sea x^* punto extremo óptimo de $LP(G, B, W)$.
  2. Eliminar de E todos e con $x^*_e = 0$.
  3. (relajación:) Si existe $v \in W$ con $d(v) \leq B_v + 1$, fijar $W \leftarrow W \setminus \{v\}$.
- Devolver (V, E).

Teorema (Singh y Lau)

Si x^* es punto extremo óptimo de $LP(G, B, W)$ positivo y $W \neq \emptyset$, entonces existe $v \in W$ con $d(v) \leq B_v + 1$.

Correctitud del algoritmo de Singh y Lau:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 1$.

Correctitud del algoritmo de Singh y Lau:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 1$.

$|E| + |V| + |W|$ decrece por iteración, hasta que $|W| = 0$. En ese momento, $LP(G, B, W)$ es igual a la relajación de árbol generador mínimo sin restricciones extras. Como eso es integral, (V, E) al final es un árbol.

Correctitud del algoritmo de Singh y Lau:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 1$.

$|E| + |V| + |W|$ decrece por iteración, hasta que $|W| = 0$. En ese momento, $LP(G, B, W)$ es igual a la relajación de árbol generador mínimo sin restricciones extras. Como eso es integral, (V, E) al final es un árbol.

- **Costo correcto**: El costo $c^\top x^*$ solo puede decrecer.

Correctitud del algoritmo de Singh y Lau:

El algoritmo iterativo anterior encuentra un árbol $T = (V, F)$ con costo igual o menor al costo del problema original, pero donde cada vértice i tiene grado a lo más $B_i + 1$.

$|E| + |V| + |W|$ decrece por iteración, hasta que $|W| = 0$. En ese momento, $LP(G, B, W)$ es igual a la relajación de árbol generador mínimo sin restricciones extras. Como eso es integral, (V, E) al final es un árbol.

- **Costo correcto**: El costo $c^\top x^*$ solo puede decrecer.
- **Cota de grados**: Solo eliminamos v de W si $d(v) \leq B_v + 1$. Al final todos los vértices se eliminan de W. $\square$