Fundamentos de Programación Entera

3. Problemas resolubles eficientemente

Carlos Testuri – Fernando Islas

Departamento de Investigación Operativa – Instituto de Computación
Facultad de Ingeniería – Universidad de la República

2012-2023

Contenido

Problema con resolución eficiente

Se dice que un problema tiene resolución eficiente cuando existen algoritmos que permiten resolver sus instancias mediante un uso de los recursos tiempo y espacio (memoria) que no crece exponencialmente en función de las dimensiones de las instancias.

El uso del tiempo se mide a partir del conteo de instrucciones empleadas en resolver el problema. Durante el curso solo se medirá el uso el tiempo.

A modo de ejemplo introductorio se pude decir que un algoritmo es *eficiente* si en *el peor caso* (instancia) requiere una cantidad de orden polinomial de instrucciones en función de la dimensión de la instancia.

Propiedades de problemas con resolución eficiente

En la determinación de si un problema tiene resolución eficiente se cumplen en general las propiedades:

1. *de optimización eficiente*. Existe un algoritmo eficiente que lo resuelve.
2. *existencia de dual-fuerte*. Existe un problema dual-fuerte asociado que permite determinar condiciones de optimalidad.
3. *separación eficiente*. Existe un algoritmo eficiente que permite determinar un plano de separación entre la región factible y un punto exterior a su casco convexo.
4. *existencia de casco convexo explícito*. Es conocida una formulación del casco convexo de la región factible del problema.

Propiedades de problemas con resolución eficiente

En la determinación de si un problema tiene resolución eficiente se cumplen en general las propiedades:

1. *de optimización eficiente*. Existe un algoritmo eficiente que lo resuelve.
2. *existencia de dual-fuerte*. Existe un problema dual-fuerte asociado que permite determinar condiciones de optimalidad.
3. *separación eficiente*. Existe un algoritmo eficiente que permite determinar un plano de separación entre la región factible y un punto exterior a su casco convexo.
4. *existencia de casco convexo explícito*. Es conocida una formulación del casco convexo de la región factible del problema.

Propiedades de problemas con resolución eficiente

En la determinación de si un problema tiene resolución eficiente se cumplen en general las propiedades:

1. *de optimización eficiente*. Existe un algoritmo eficiente que lo resuelve.
2. *existencia de dual-fuerte*. Existe un problema dual-fuerte asociado que permite determinar condiciones de optimalidad.
3. *separación eficiente*. Existe un algoritmo eficiente que permite determinar un plano de separación entre la región factible y un punto exterior a su casco convexo.
4. *existencia de casco convexo explícito*. Es conocida una formulación del casco convexo de la región factible del problema.

Propiedades de problemas con resolución eficiente

En la determinación de si un problema tiene resolución eficiente se cumplen en general las propiedades:

1. *de optimización eficiente*. Existe un algoritmo eficiente que lo resuelve.
2. *existencia de dual-fuerte*. Existe un problema dual-fuerte asociado que permite determinar condiciones de optimalidad.
3. *separación eficiente*. Existe un algoritmo eficiente que permite determinar un plano de separación entre la región factible y un punto exterior a su casco convexo.
4. *existencia de casco convexo explícito*. Es conocida una formulación del casco convexo de la región factible del problema.

Problemas enteros con solución mediante relajación a programación lineal

Dado el problema (IP)

$$\max\{c^\tau x : Ax \leq b, x \in \mathbb{Z}^n_+\}$$

con A y b enteros.

¿Cuándo es que su relajación a programación lineal,

$$\max\{c^\tau x : Ax \leq b, x \in \mathbb{R}^n_+\},$$

tiene solución óptima entera?

Matrices totalmente unimodulares

Una matriz cuadrada es unimodular si su determinante es -1 o $+1$.

Se dice que una *matriz es totalmente unimodular* (TU) si el determinante de cada submatriz cuadrada de ella es -1, 0, o $+1$.

Proposición (1)

Una matriz A es TU si

1. *sus elementos son* -1, 0, *o* $+1$,
2. *cada columna contiene a lo sumo dos elementos no nulos,* ***y***
3. *existe una bipartición* (M_1, M_2) *del conjunto de filas tal que para cada columna j que contiene dos elementos no nulos cumple* $\sum_{i \in M_1} a_{ij} = \sum_{i \in M_2} a_{ij}$.

Matrices totalmente unimodulares

Una matriz cuadrada es unimodular si su determinante es -1 o $+1$.

Se dice que una *matriz es totalmente unimodular* (TU) si el determinante de cada submatriz cuadrada de ella es -1, 0, o $+1$.

Proposición (1)

Una matriz A es TU si

1. *sus elementos son -1, 0, o $+1$,*
2. *cada columna contiene a lo sumo dos elementos no nulos,* ***y***
3. *existe una bipartición (M_1, M_2) del conjunto de filas tal que para cada columna j que contiene dos elementos no nulos cumple $\sum_{i \in M_1} a_{ij} = \sum_{i \in M_2} a_{ij}$.*

Matrices totalmente unimodulares

Una matriz cuadrada es unimodular si su determinante es -1 o $+1$.

Se dice que una *matriz es totalmente unimodular* (TU) si el determinante de cada submatriz cuadrada de ella es -1, 0, o $+1$.

Proposición (1)

Una matriz A es TU si

1. *sus elementos son* -1, 0, *o* $+1$,
2. *cada columna contiene a lo sumo dos elementos no nulos,* ***y***
3. *existe una bipartición* (M_1, M_2) *del conjunto de filas tal que para cada columna j que contiene dos elementos no nulos cumple* $\sum_{i \in M_1} a_{ij} = \sum_{i \in M_2} a_{ij}$.

Matrices totalmente unimodulares: ejemplos

Matrices no TU

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

Matrices TU

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

Matrices totalmente unimodulares: ejemplos

Matrices no TU

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

Matrices TU

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

Problemas con matrices totalmente unimodulares

Las matrices TU permiten establecer condiciones en que la solución a un problema de programación lineal sea integral.

Proposición (2)

Dado el problema IP $\max\{c^\tau x : Ax \leq b, x \in \mathbb{Z}^n_+\}$.

Si A es totalmente unimodular y b es entero, entonces la relajación a programación lineal del problema IP, $\max\{c^\tau x : Ax \leq b, x \geq 0\}$*, tiene solución óptima entera que coincide con la solución de IP.*

Prueba.

Cada base B correspondiente a una solución factible del problema tiene determinante $\det(B) = -1$ o $+1$. Para resolver $x_B = B^{-1}b$ se tiene por regla de Cramer, que $B^{-1} = adjunto(B)/\det(B)$ o $x_i = \det(B_i)/\det(B)$, donde B_i es B con la columna i-ésima reemplazada por b. Dado que b es entero y $\det(B) = +1$ o -1, el resultado: x_B es entero. □

Propiedades de resolución eficiente en problemas con matrices TU

Para estos problemas se cumplen las propiedades de resolución eficiente:

1. Optimización eficiente: Sí (difícil de demostrar para caso general).
2. Dual-fuerte: el dual del problema lineal, $\min\{u^\tau b : u^\tau A \geq c^\tau, u \geq 0\}$, es un problema dual-fuerte.
3. Separación eficiente: se verifica al comprobar que se cumplen $Ax^* \leq b$ y $x^* \geq 0$.
4. Casco convexo explícito: el casco convexo del conjunto de soluciones factibles $\text{conv}(X) = \{Ax \leq b, x \geq 0\}$ es conocido.

Propiedades de resolución eficiente en problemas con matrices TU

Para estos problemas se cumplen las propiedades de resolución eficiente:

1. Optimización eficiente: Sí (difícil de demostrar para caso general).
2. Dual-fuerte: el dual del problema lineal, $\min\{u^\tau b : u^\tau A \geq c^\tau, u \geq 0\}$, es un problema dual-fuerte.
3. Separación eficiente: se verifica al comprobar que se cumplen $Ax^* \leq b$ y $x^* \geq 0$.
4. Casco convexo explícito: el casco convexo del conjunto de soluciones factibles $\text{conv}(X) = \{Ax \leq b, x \geq 0\}$ es conocido.

Propiedades de resolución eficiente en problemas con matrices TU

Para estos problemas se cumplen las propiedades de resolución eficiente:

1. Optimización eficiente: Sí (difícil de demostrar para caso general).
2. Dual-fuerte: el dual del problema lineal, $\min\{u^\tau b : u^\tau A \geq c^\tau, u \geq 0\}$, es un problema dual-fuerte.
3. Separación eficiente: se verifica al comprobar que se cumplen $Ax^* \leq b$ y $x^* \geq 0$.
4. Casco convexo explícito: el casco convexo del conjunto de soluciones factibles $\mathrm{conv}(X) = \{Ax \leq b, x \geq 0\}$ es conocido.

Propiedades de resolución eficiente en problemas con matrices TU

Para estos problemas se cumplen las propiedades de resolución eficiente:

1. Optimización eficiente: Sí (difícil de demostrar para caso general).
2. Dual-fuerte: el dual del problema lineal, $\min\{u^\tau b : u^\tau A \geq c^\tau, u \geq 0\}$, es un problema dual-fuerte.
3. Separación eficiente: se verifica al comprobar que se cumplen $Ax^* \leq b$ y $x^* \geq 0$.
4. Casco convexo explícito: el casco convexo del conjunto de soluciones factibles $\mathrm{conv}(X) = \{Ax \leq b, x \geq 0\}$ es conocido.

Problema de flujo en red

Sea una *red de flujo* mediante un grafo dirigido $G = (V, A)$ con costo unitario de flujo c_{ij} y capacidad u_{ij} para todo arco $(i, j) \in A$, y demanda b_i para todo vértice $i \in V$. Donde i se denomina *fuente*, si $b_i \geq 0$, o *sumidero*, si $b_i \leq 0$. El *problema de flujo en red* consiste en determinar el flujo positivo que, cumpliendo las capacidades, satisface las demandas a mínimo costo.

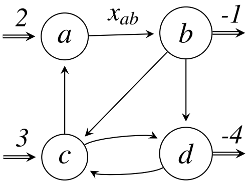
Dada la variable real de flujo x_{ij}, su formulación es

$$\begin{array}{cl} \min & \sum_{(i,j)\in A} c_{ij} x_{ij} \\ \text{s.a} & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = b_i, \quad \forall i \in V, \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A. \end{array}$$

Para que sea factible se debe cumplir que $\sum_{i\in V} b_i = 0$.
Es un problema con matriz TU.

Instancia de problema de flujo en red

Dado el grafo dirigido con demandas



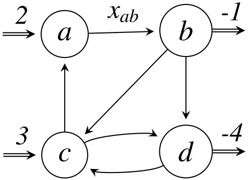
Se tienen las restricciones de *conservación de flujo* en vértices

Vértice	x_{ab}	x_{bc}	x_{bd}	x_{ca}	x_{cd}	x_{dc}	=	b
a	1	0	0	-1	0	0	=	2
b	-1	1	1	0	0	0	=	-1
c	0	-1	0	1	1	-1	=	3
d	0	0	-1	0	-1	1	=	-4

Los coeficientes de las variables en las restricciones de balance y en las de capacidad forman una matriz TU.

Instancia de problema de flujo en red

Dado el grafo dirigido con demandas



Se tienen las restricciones de *conservación de flujo* en vértices

Vértice	x_{ab}	x_{bc}	x_{bd}	x_{ca}	x_{cd}	x_{dc}	$=$	b
a	1	0	0	-1	0	0	=	2
b	-1	1	1	0	0	0	=	-1
c	0	-1	0	1	1	-1	=	3
d	0	0	-1	0	-1	1	=	-4

Los coeficientes de las variables en las restricciones de balance y en las de capacidad forman una matriz TU.

Flujo entero en problema de flujo en red

Proposición (3)

Dado el problema de flujo en red

$$\begin{array}{ll} \min & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ s.a & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = b_i, \quad \forall i \in V, \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A. \end{array}$$

Si las demandas (b_i) y las capacidades (u_{ij}) son enteras se cumplen:

1. *cada punto extremo (flujo) de la región factible es entero,*
2. *las restricciones (formulación) del problema establecen el casco convexo de los flujos enteros factibles.*

Casos particulares del problema de flujo en red

El problema de flujo en red abarca los problemas

1. *Problema de camino más corto*
   Dados los vértices de origen y destino del camino, consiste en encontrar el camino dirigido de costo mínimo desde el origen al destino (los costos no deben ser negativos).
2. *Problema de flujo máximo*
   Dados los vértices de origen y destino del flujo, consiste en determinar el máximo flujo desde el origen al destino.
3. *Problema de transporte*
   Clasificando los vértices entre proveedores y clientes, con ofertas y demandas de un único producto, respectivamente. Se tiene un costo unitario de transportar el producto entre cada proveedor y cliente. El objetivo es transportar el producto a costo total mínimo mientras se atienden las demandas mediante las ofertas.

Casos particulares del problema de flujo en red

El problema de flujo en red abarca los problemas

1. *Problema de camino más corto*
   Dados los vértices de origen y destino del camino, consiste en encontrar el camino dirigido de costo mínimo desde el origen al destino (los costos no deben ser negativos).
2. *Problema de flujo máximo*
   Dados los vértices de origen y destino del flujo, consiste en determinar el máximo flujo desde el origen al destino.
3. *Problema de transporte*
   Clasificando los vértices entre proveedores y clientes, con ofertas y demandas de un único producto, respectivamente. Se tiene un costo unitario de transportar el producto entre cada proveedor y cliente. El objetivo es transportar el producto a costo total mínimo mientras se atienden las demandas mediante las ofertas.

Casos particulares del problema de flujo en red

El problema de flujo en red abarca los problemas

1. *Problema de camino más corto*
   Dados los vértices de origen y destino del camino, consiste en encontrar el camino dirigido de costo mínimo desde el origen al destino (los costos no deben ser negativos).
2. *Problema de flujo máximo*
   Dados los vértices de origen y destino del flujo, consiste en determinar el máximo flujo desde el origen al destino.
3. *Problema de transporte*
   Clasificando los vértices entre proveedores y clientes, con ofertas y demandas de un único producto, respectivamente. Se tiene un costo unitario de transportar el producto entre cada proveedor y cliente. El objetivo es transportar el producto a costo total mínimo mientras se atienden las demandas mediante las ofertas.

Problema de camino más corto

Dados los vértices de origen (s) y destino (t), se establece demanda unitaria entre ellos, y se interpreta que $x_{ij} = 1$ si el arco (i,j) pertenece al camino.

Se tiene la formulación

$$
\begin{array}{lll}
z = & \min & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
 & \text{s.a} & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = 1, \quad i = s, \\
 & & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = 0, \quad \forall i \in V\setminus\{s,t\}, \\
 & & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = -1, \quad i = t, \\
 & & x_{ij} \geq 0 \quad \text{entera}, \quad \forall (i,j) \in A.
\end{array}
$$

¿Qué establece el valor óptimo del problema, z?

Dual del problema de camino más corto

Dadas variables duales π, el problema dual de su relajación a programación lineal (LP) es

$$\begin{array}{rll} w^{LP} = & \max & \pi_s - \pi_t \\ & \text{s.a} & \pi_i - \pi_j \leq c_{ij} \quad \forall (i,j) \in A. \end{array}$$

Dado que la matriz del problema primal es TU se cumple la dualidad fuerte.

¿Tiene el problema múltiples soluciones?
Si π_i, para todo $i \in V$, es una solución dual factible, entonces dado el escalar δ, $\pi_i + \delta$, para todo $i \in V$, también es solución.

Si se establece que $\pi_t = 0$. ¿Qué relación hay entre π_s y z.

Dual del problema de camino más corto

Dadas variables duales π, el problema dual de su relajación a programación lineal (LP) es

$$
\begin{array}{rll}
w^{LP} = & \max & \pi_s - \pi_t \\
& \text{s.a} & \pi_i - \pi_j \leq c_{ij} \quad \forall (i,j) \in A.
\end{array}
$$

Dado que la matriz del problema primal es TU se cumple la dualidad fuerte.

¿Tiene el problema múltiples soluciones?
Si π_i, para todo $i \in V$, es una solución dual factible, entonces dado el escalar δ, $\pi_i + \delta$, para todo $i \in V$, también es solución.

Si se establece que $\pi_t = 0$. ¿Qué relación hay entre π_s y z.

Dual del problema de camino más corto

Dadas variables duales π, el problema dual de su relajación a programación lineal (LP) es

$$\begin{array}{rll} w^{LP} = & \max & \pi_s - \pi_t \\ & \text{s.a} & \pi_i - \pi_j \leq c_{ij} \quad \forall (i,j) \in A. \end{array}$$

Dado que la matriz del problema primal es TU se cumple la dualidad fuerte.

¿Tiene el problema múltiples soluciones?
Si π_i, para todo $i \in V$, es una solución dual factible, entonces dado el escalar δ, $\pi_i + \delta$, para todo $i \in V$, también es solución.

Si se establece que $\pi_t = 0$. ¿Qué relación hay entre π_s y z.

Dual del problema de camino más corto

Dadas variables duales π, el problema dual de su relajación a programación lineal (LP) es

$$
\begin{aligned}
w^{LP} = \quad & \max \quad \pi_s - \pi_t \\
& \text{s.a} \quad \pi_i - \pi_j \leq c_{ij} \quad \forall (i,j) \in A.
\end{aligned}
$$

Dado que la matriz del problema primal es TU se cumple la dualidad fuerte.

¿Tiene el problema múltiples soluciones?
Si π_i, para todo $i \in V$, es una solución dual factible, entonces dado el escalar δ, $\pi_i + \delta$, para todo $i \in V$, también es solución.

Si se establece que $\pi_t = 0$. ¿Qué relación hay entre π_s y z.

Problema de máximo flujo

Para modelar el problema se establece una circulación externa de todo el flujo sobre la red, adicionando un arco no-capacitado del vértice destino (t) al origen (s),

$$
\begin{array}{ll}
\max & x_{ts} \quad (\equiv -\min -x_{ts}) \\
\text{s.a} & \sum_{k:(i,k)\in A} x_{ik} - \sum_{k:(k,i)\in A} x_{ki} = 0, \quad \forall i \in V \\
& 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A.
\end{array}
$$

Su problema dual fuerte con variables π y λ correspondientes a las restricciones de balance y cota, respectivamente, es

$$
\begin{array}{ll}
\min & \sum_{(i,j)\in A} u_{ij}\lambda_{ij} \\
\text{s.a} & \pi_i - \pi_j + \lambda_{ij} \geq 0 \quad \forall (i,j) \in A, \\
& \pi_t - \pi_s \geq 1,
\end{array}
$$

denominado problema de *corte* $s - t$ *de capacidad mínima*.

Problema de transporte

Sean m proveedores y n clientes de un producto. El proveedor i puede suministrar s_i unidades y el cliente j demanda d_j unidades. Se asume que los totales suministrado y demandado coinciden. Al transportar del proveedor i al cliente j se incurre en un costo unitario c_{ij}.
El problema consiste en transportar a costo mínimo el producto desde los proveedores a los clientes.

$$
\begin{array}{ll}
\min & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \\
\text{s.a} & \sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1, ..., n, \\
& \sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, ..., m, \\
& x_{ij} \geq 0,\ i = 1, ..., m, j = 1, ..., n.
\end{array}
$$

El *problema de asignación* es un caso especial de este. En el cual la cantidad de proveedores y clientes coinciden, y todos ellos tienen una única unidad de suministro y demanda, respectivamente.

Arboles de expansión

Dado un grafo no dirigido $G = (V, E)$, un *árbol de expansión(cubrimiento)* de sus vértices es un subgrafo de él que contiene todos sus vértices y un subconjunto de sus aristas, $T = (V, E')$ con $E' \subseteq E$, tal que es conexo y no contiene ciclos.

Proposición (4)

Un grafo conexo $G = (V, E)$ con n vértices es un árbol, es equivalente a que

1. *tenga $n - 1$ aristas,*
2. *tenga un único camino entre todo par de vértices,*
3. *el agregado de una arista crea un ciclo,* ***o***
4. *la remoción de una arista hace que el grafo sea disconexo.*

Arbol de expansión minimal (MST): algoritmo de Kruskal

Dado un grafo no dirigido conexo $G = (V, E)$, donde $n = |V|$ y $m = |E|$, y un valor asociado a cada arista c_e, $e \in E$, se busca determinar un subgrafo árbol $T = (V, E')$, con $E' \subseteq E$, cuyo valor total de aristas sea mínimo.

Algoritmo de Kruskal

Inicialización. Sean $E^0 := \emptyset$, $T := (V, E^0)$, $k := 1$. Se ordenan las aristas según valor $c_1 \leq ... \leq c_m$.
Iteración. Si $T := (V, E^{k-1} \cup \{e_k\})$ es acíclico,
entonces $E^k := E^{k-1} \cup \{e_k\}$, si no $E^k := E^{k-1}$.
Si $|E^k| = n - 1$ *Parar* : $T := (V, E^k)$ es un árbol de expansión minimal.
En otro caso $k := k + 1$ y se retorna a *Iteración*.

Es un algoritmo ávido (opción óptima en cada paso) puede implementarse para ejecutar en tiempo $O(m \log n)$.
Se cumple la propiedad de optimización eficiente.

MST como problema de optimización: formulaciones alternativas

La variable x_e es uno si la arista $e \in E$ pertenece al árbol, y cero en caso contrario.

Para formular que un árbol de expansión sobre n vértices contiene $n-1$ aristas se tiene

$$\sum_{e \in E} x_e = n - 1.$$

Hay dos formulaciones alternativas que requieren además que

1. las aristas del subgrafo no formen un ciclo, denominada formulación de eliminación de ciclos (subtour),
2. el subgrafo sea conexo, denominada formulación de (imposición de conectividad mediante) conjunto de corte (cutset).

MST: formulación con eliminación de ciclos (subtour)

Se requiere que las aristas no formen ciclos.
Para todo subconjunto de vértices S de V, se debe cumplir que la cantidad de aristas incidentes a S es menor o igual que $|S|-1$.
Dado $S \subseteq V$, sean sus aristas incidentes $E(S) := \{e = \{i,j\} \in E | i,j \in S\}$.
Se requiere que

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subseteq V, S \neq \emptyset, V.$$

La formulación completa es

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.a} & \sum_{e \in E} x_e = n-1, \\ & \sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subseteq V, S \neq \emptyset, V, \\ & x_e \in \{0,1\}, \quad \forall e \in E. \end{array}$$

MST: formulación con conjunto de corte (cutset)

Se requiere que el subgrafo sea conexo
Para todo subconjunto de vértices S de V, se debe cumplir que haya al menos una arista que conecte S y $V \setminus S$.
Dado $S \subseteq V$, sean sus aristas de corte $\delta(S) := \{e = \{i,j\} \in E | i \in S, j \notin S\}$.
Se requiere que

$$\sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subseteq V, S \neq \varnothing, V.$$

La formulación completa es

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.a} & \sum_{e \in E} x_e = n - 1, \\ & \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subseteq V, S \neq \varnothing, V, \\ & x_e \in \{0, 1\}, \quad \forall e \in E. \end{array}$$

MST: bondad de las formulaciones alternativas

¿Es una de las formulaciones mejor que la otra?

La formulación subtour es mejor que la formulación cutset.

Proposición (5)

Sean P_{sub} y P_{cut} los poliedros correspondientes a las relajaciones lineales de las formulaciones subtour y cutset, respectivamente,

1. *Se cumple que $P_{sub} \subseteq P_{cut}$.*
2. *El poliedro P_{cut} puede tener puntos extremos fraccionarios.*

Proposición (6)

El poliedro P_{sub} es una formulación del casco convexo de las soluciones factibles enteras del problema MST.

MST: bondad de las formulaciones alternativas

¿Es una de las formulaciones mejor que la otra?

La formulación subtour es mejor que la formulación cutset.

Proposición (5)

Sean P_{sub} y P_{cut} los poliedros correspondientes a las relajaciones lineales de las formulaciones subtour y cutset, respectivamente,

1. *Se cumple que $P_{sub} \subseteq P_{cut}$.*
2. *El poliedro P_{cut} puede tener puntos extremos fraccionarios.*

Proposición (6)

El poliedro P_{sub} es una formulación del casco convexo de las soluciones factibles enteras del problema MST.

MST: bondad de las formulaciones alternativas

¿Es una de las formulaciones mejor que la otra?

La formulación subtour es mejor que la formulación cutset.

Proposición (5)

Sean P_{sub} y P_{cut} los poliedros correspondientes a las relajaciones lineales de las formulaciones subtour y cutset, respectivamente,

1. *Se cumple que $P_{sub} \subseteq P_{cut}$.*
2. *El poliedro P_{cut} puede tener puntos extremos fraccionarios.*

Proposición (6)

El poliedro P_{sub} es una formulación del casco convexo de las soluciones factibles enteras del problema MST.

MST: bondad de las formulaciones alternativas

Prueba 5.

1) **Si** $x \in P_{sub}$ **entonces** $x \in P_{cut}$**: demostración**

$$\sum_{e \in E(S)} x_e + \sum_{e \in \delta(S)} x_e + \sum_{e \in E(V \setminus S)} x_e = \sum_{e \in E} x_e$$

Para todo $x \in P_{sub}$ se tiene $\sum_{e \in E(S)} x_e \leq |S| - 1$ y $\sum_{e \in E(V \setminus S)} x_e \leq |V \setminus S| - 1$.
Dado que $\sum_{e \in E} x_e = n - 1$ se tiene que $\sum_{e \in \delta(S)} x_e \geq 1$. Por lo que $x \in P_{cut}$.

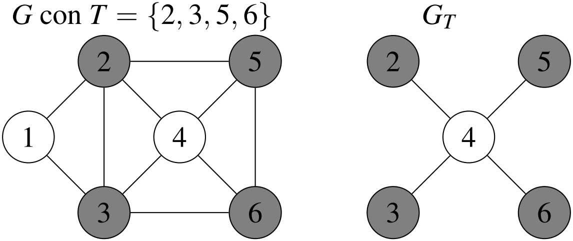
2) **Si** $x \in P_{cut}$ **entonces no tiene porque** $x \in P_{sub}$**: contraejemplo**
$V = \{A, B, C, D, E\}$ y
$E(v_1, v_2, c_{v_1,v_2}) = \{(A, B, 1), (A, C, 1), (B, C, 1), (B, D, 2), (C, E, 2), (D, E, 2)\}$
Tiene solución óptima: $x_E^* = \{1, 1, 0, 1, 1, 0\}$, con valor óptimo 6.
Una solución de P_{cut} es $x_E^{cut} = \{1, 1, 0.5, 0.5, 0.5, 0.5)\}$, pero esta solución no es factible para P_{sub} dado que para $S = \{A, B, C\}$ no se cumple $\sum_{e \in E(S)} x_e \leq |S| - 1$, $2.5 \not\leq 2$.

□

Arbol de Steiner

Dado un grafo no dirigido conexo $G = (V, E)$ y un subconjunto de vértices $T \subseteq V$ (terminales), se denomina *árbol de Steiner* al subgrafo acíclico de G, G_T, que conecta los vertices de T, posiblemente conteniendo vértices de $V \backslash T$.

Ejemplo



Notar que hay muchos árboles posibles.

Problema del árbol de Steiner

Dados valores en las aristas, $c_e \in E$, se denomina *problema del árbol de Steiner* al de determinar el árbol con menor valor total en sus aristas.

Sea la variable x_e que vale uno si la arista $e \in E$ pertenece al árbol, y cero en caso contrario, se tiene la formulación

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.a} & \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subseteq V, S \cap T \neq \varnothing, T, \\ & x_e \in \{0,1\}, \quad \forall e \in E. \end{array}$$

Es una generalización del problema del árbol óptimo: incluye el problema MST cuando $T = V$ y el problema de camino más corto cuando $|T| = 2$.

Aunque se conocen algoritmos de resolución eficiente para estos casos especiales (Kruskal y Dijkstra, respectivamente), **no se conoce algoritmo eficiente** para el problema del árbol de Steiner.

Problema del árbol de Steiner

Dados valores en las aristas, $c_e \in E$, se denomina *problema del árbol de Steiner* al de determinar el árbol con menor valor total en sus aristas.

Sea la variable x_e que vale uno si la arista $e \in E$ pertenece al árbol, y cero en caso contrario, se tiene la formulación

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.a} & \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subseteq V, S \cap T \neq \varnothing, T, \\ & x_e \in \{0, 1\}, \quad \forall e \in E. \end{array}$$

Es una generalización del problema del árbol óptimo: incluye el problema MST cuando $T = V$ y el problema de camino más corto cuando $|T| = 2$.

Aunque se conocen algoritmos de resolución eficiente para estos casos especiales (Kruskal y Dijkstra, respectivamente), **no se conoce algoritmo eficiente** para el problema del árbol de Steiner.