Algoritmos para problemas de PLEM

¿Cómo se resuelve un PLEM?

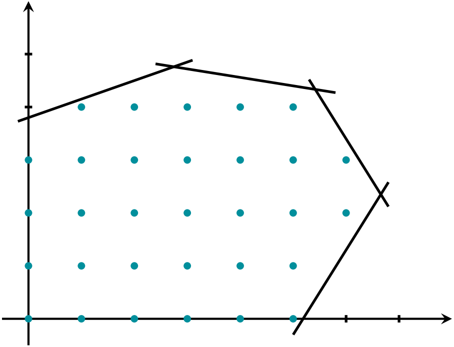
Técnicas:

- enumeración completa: considerar todas las posibles combinaciones de las variables enteras y para cada combinación resolver el PPL correspondiente.
- algoritmos de planos de corte
- enumeración inteligente: algoritmos *Branch-and-Bound*
- planos de corte + *Branch-and-Bound*: algoritmos *Branch-and-Cut*
- algoritmos de generación de columnas
- generación de columnas + *Branch-and-Bound*: algoritmos *Branch-and-Price*
- planos de corte + generación de columnas + *Branch-and-Bound*: algoritmos *Branch-and-Cut-and-Price*
- programación dinámica
- heurísticas y metaheurísticas

Algoritmos para PLEM

- En el peor caso los algoritmos exactos pueden requerir un tiempo de cómputo de crecimiento exponencial en el "tamaño" del problema que se quiere resolver.
- Problemas del mismo "tamaño" pueden requerir tiempo de cómputo muy diferentes.
- La dificultad depende del tipo de restricciones, y algunas veces de los valores de los coeficientes (c_j, a_{ij}, b_i).

Puntos factibles de un PLE

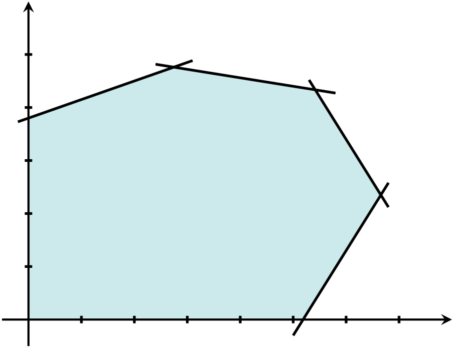


Relajación de un PLEM

Es un PL o PLEM "fácil de resolver" tal que el valor óptimo es una cota superior (prob. Max) del valor óptimo del PLEM.

- Modificando la región factible: La región factible del modelo original está contenida dentro de la del modelo relajado.
- Modificando la función objetivo

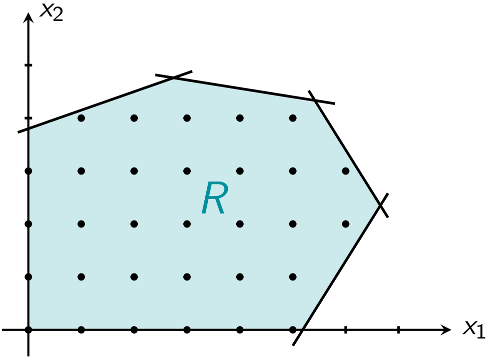
Relajación lineal de un PLEM



Región factible de la relajación lineal

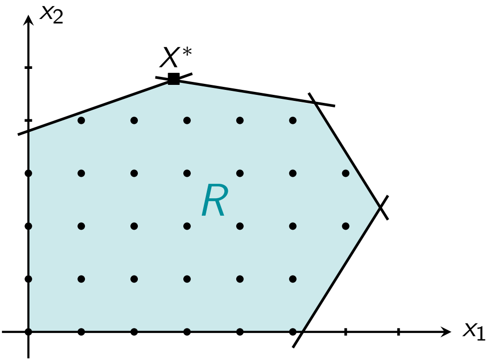
óptimo de la relajación lineal *mejor o igual* que óptimo del PLEM

Algoritmos Branch & Bound



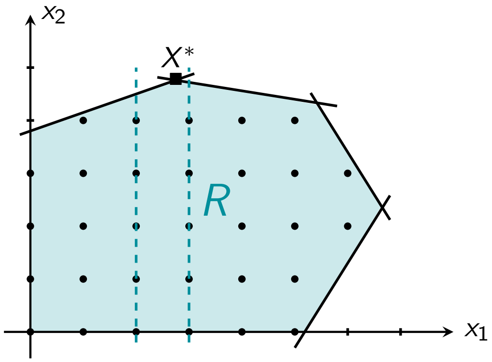
Región factible de la relajación lineal

Algoritmos Branch & Bound



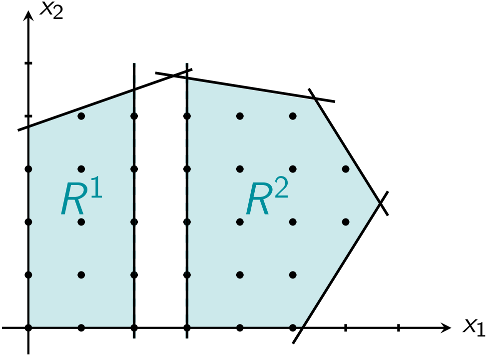
Región factible de la relajación lineal

Algoritmos Branch & Bound



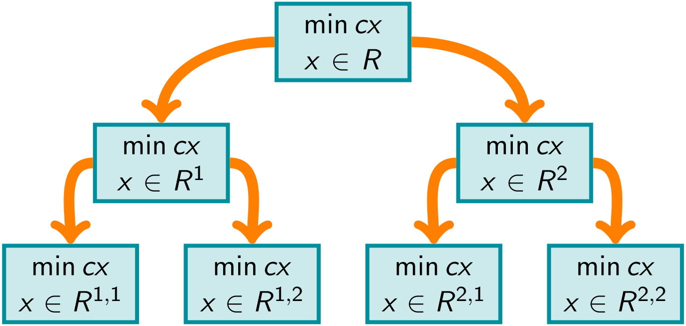
Región factible de la relajación lineal

Algoritmos Branch & Bound



Dos nuevos subproblemas

Algoritmos Branch & Bound



Algoritmos Branch & Bound

Esquema general:

- Árbol de enumeración con la raíz correspondiente al problema original.
- A cada nodo del árbol le corresponde un subproblema.
- Argumentos de dominancia y factibilidad permiten podar ramas del árbol.
- En cada nodo del árbol calculamos una cota del óptimo restringido a ese espacio de soluciones (el óptimo seguro NO es mejor que esa cota).
- Si la cota es peor que la mejor solución obtenida hasta el momento, no es necesario explorar esa parte del árbol.
- Para obtener esa cota en cada nodo, resolvemos la relajación lineal de la formulación asociada a ese nodo.

Algoritmos Branch & Bound

Un nodo puede llegar a no tener sucesores (poda) por:

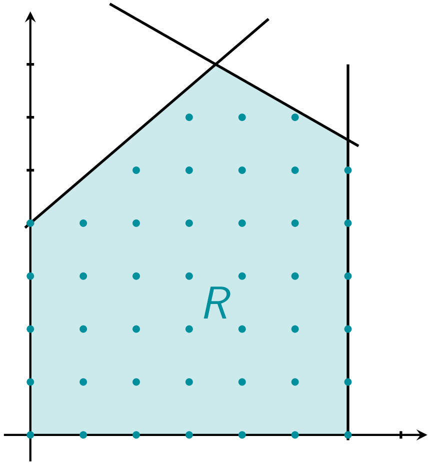
- El subproblema es no factible.
- El subproblema tiene una solución no entera pero con un valor de la función objetivo peor que el de la mejor solución: se descarta.
- El subproblema tiene una solución entera.

Algoritmos Branch & Bound

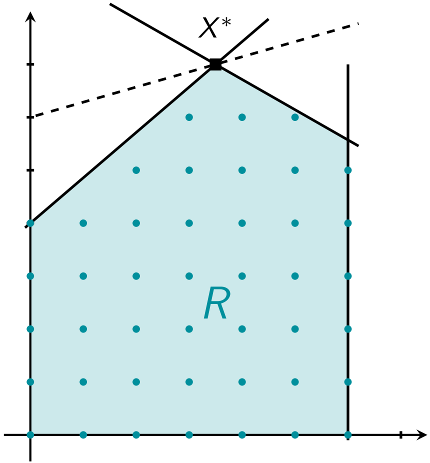
Cuando el suproblema tiene una solución entera:

- Si su valor de función objetivo es mejor que el de la mejor solución entera ya obtenida se convierte en mejor solución.
- Si su valor de función objetivo es peor que el de la mejor solución actual se descarta.

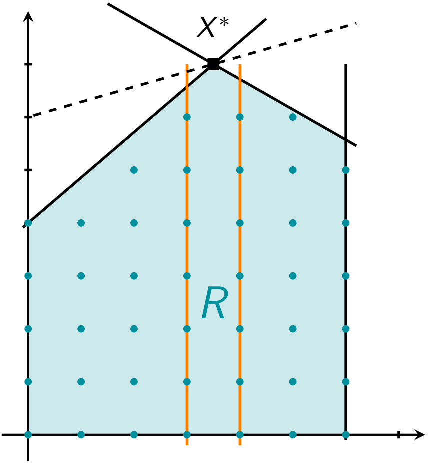
Algoritmos Branch & Bound



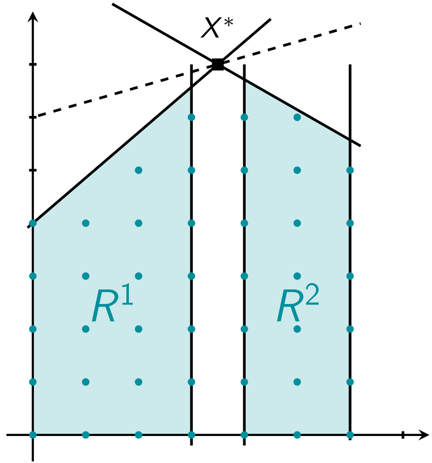
Algoritmos Branch & Bound



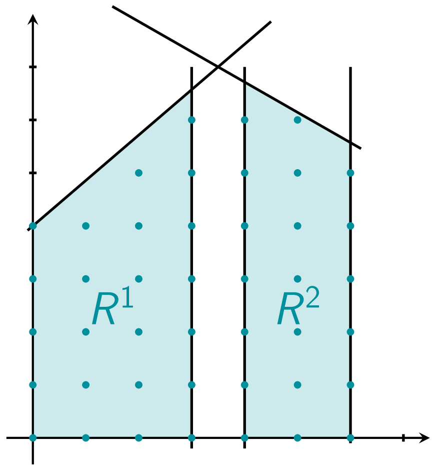
Algoritmos Branch & Bound



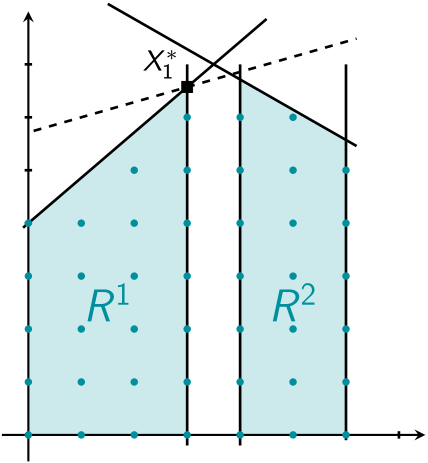
Algoritmos Branch & Bound



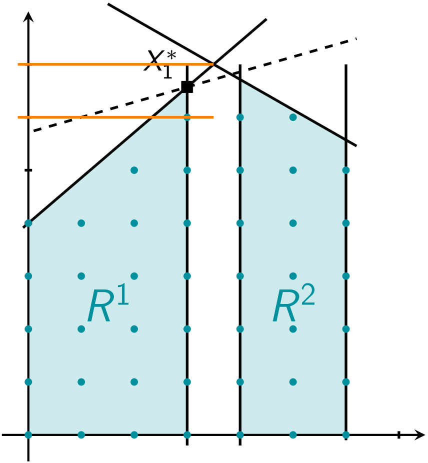
Algoritmos Branch & Bound



Algoritmos Branch & Bound

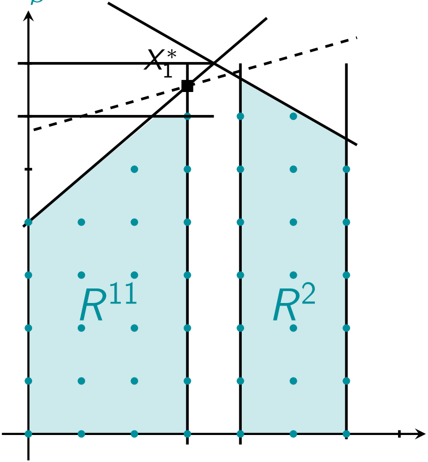


Algoritmos Branch & Bound



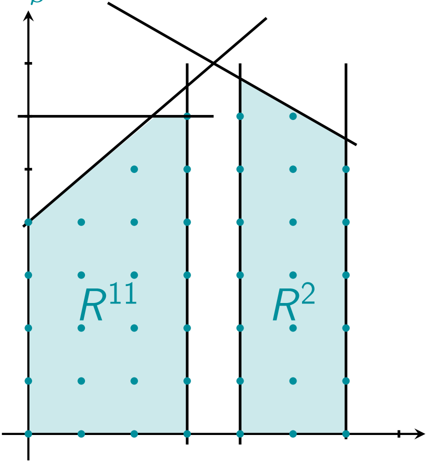
Algoritmos Branch & Bound

$R^{12} = \emptyset$



Algoritmos Branch & Bound

$R^{12} = \emptyset$



Algoritmos Branch & Bound

Esquema:

1. Elegir un nodo no explorado del árbol (nodos abiertos)
2. Resolver la relajación lineal asociada al nodo
3. Si el valor óptimo de la relajación es peor que la mejor solución hasta el momento, cerrar el nodo (poda)
4. Si la solución es entera, cerrar nodo y actualizar mejor solución si tiene mejor valor
5. Si existe x_i^* variable fraccionaria, abrir dos hijos: $x_i \leq \lfloor x_i^* \rfloor$ y $x_i^* \geq \lceil x_i^* \rceil$ (ramificación). Volver a 1.

Algoritmos Branch & Bound: Recorrido del árbol

Reglas de selección de próximo nodo a explorar:

- Profundidad: nodos que están en nivel más bajo en el árbol. Evita que la lista de nodos abiertos crezca y "permitiría" encontrar soluciones factibles rápidamente.
- Mejor cota: nunca dividiríamos un nodo con cota inferior mayor que el óptimo del problema.
- Combinación/Criterios heurísticos
- Criterios ad-hoc

Algoritmos Branch & Bound: Estrategia de ramificación

Reglas de ramificación:

- Variable más fraccionaria
- Variable menos fraccionaria
- Variable con mayor importancia "económica"
- Por restricciones
- Pseudo-costos
- Criterios ad-hoc

Algoritmos Branch & Bound

Problemas que se pueden presentar:

- Puede ser necesario explorar un gran número de subproblemas
- Si la búsqueda es a lo ancho, puede ser necesaria un gran cantidad de memoria.
- Si la búsqueda es en profundidad, se puede gastar mucho tiempo explorando un callejón sin "salida".

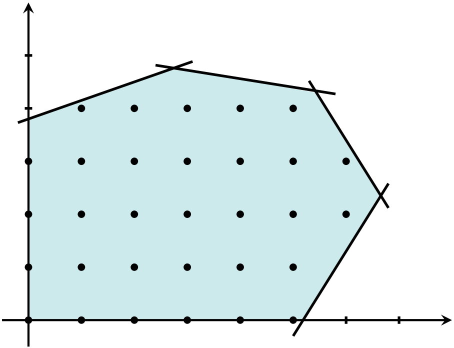
Algoritmos Branch & Bound

- Si todas las variables enteras tienen cota superior e inferior el árbol es finito: en cada subproblema alguna de las variables tiene rango estrictamente menor que su padre.

- Ejemplo de problema infactible pero que no termina:

$$
\begin{array}{rrcl}
\text{Min} & 0 & & \\
\text{sa} & 3x - 3y & \geq & 1 \\
 & 3x - 3y & \leq & 2 \\
 & x, y & \in & \mathbb{Z}
\end{array}
$$

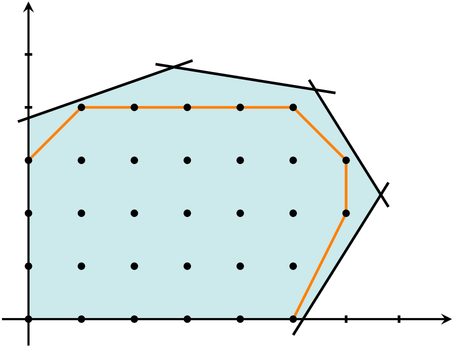
Algoritmos de Planos de Corte



Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Algoritmos de Planos de Corte



Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

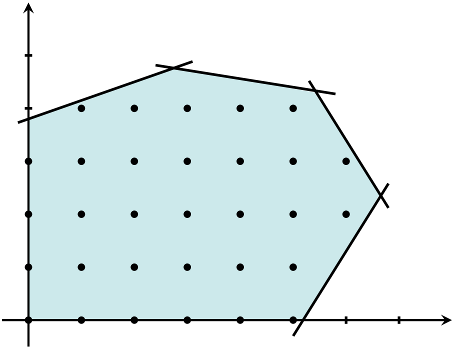
Algoritmos de Planos de Corte

- Cápsula convexa del conjunto de soluciones factibles, *conv*(S), es el conjunto de todos los puntos que son combinación convexa de los puntos de S (menor convexo que contiene a S):

$$conv(S) = \{x \in \mathbb{R}^n \text{ tal que } x = \sum_{x_i \in S} \alpha_i x_i, \sum \alpha_i = 1, \alpha_i \geq 0\}$$

- Si conociéramos *conv*(S) podemos resolver un PLEM mediante un algoritmo para PL, ya que los extremos de *conv*(S) son puntos enteros.
- Pero para los problemas NP-Difícil no es esperable encontrar una descripción completa de *conv*(S), ni que la cantidad de restricciones necesarias para describirla sea polinomial.
- Vamos a tratar de acercanos a *conv*(S).

Algoritmos de Planos de Corte

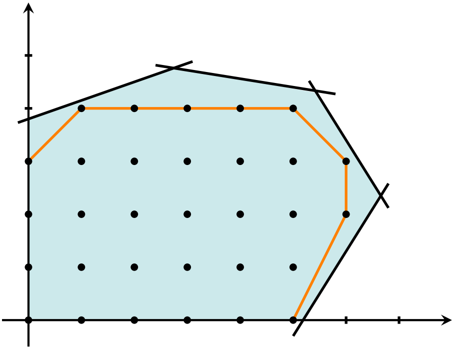


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

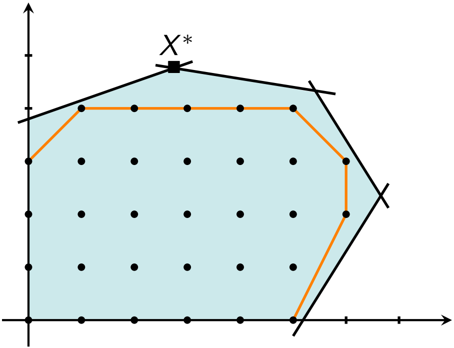


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

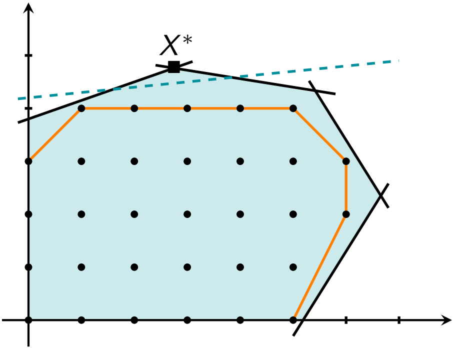


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

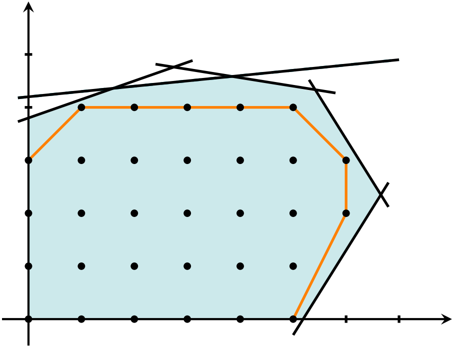


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

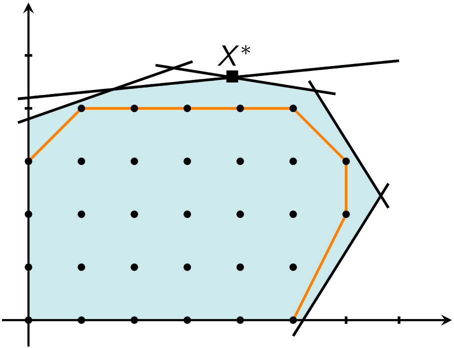


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

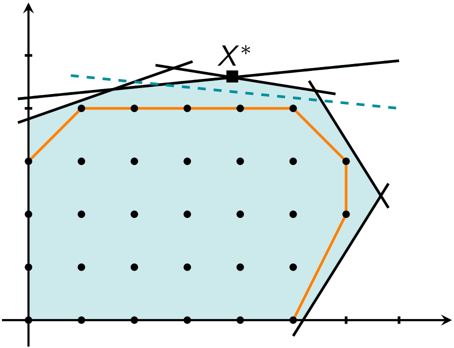


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte

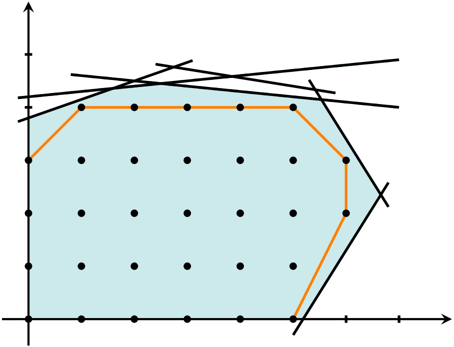


Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

Algoritmos de Planos de Corte



Conjunto de soluciones factibles: S

Región celeste: Región factible de la relajación lineal, R

Región naranja: Cápsula convexa del conjunto de soluciones factibles, $conv(S)$

¿Qué es un plano de corte?

- Una desigualdad satisfecha por toda solución factible, desigualdad válida.
- No es parte de la actual formulación.
- No es satisfecha por la solución óptima de la relajación lineal actual.
- Tiene un algoritmo para encontrarlo (separación).

Algoritmos de separación

- Una desigualdad, $\Pi x \leq \Pi_0$, está violada por un punto x^* si $\Pi x^* > \Pi_0$.
- Para implementar un algoritmo de planos de corte, necesitamos detectar cuando una desigualdad válida es violada por la solución óptima de una relajación dada.
- Esto es, dado una solución fraccionaria x^* de una relajación lineal $R \setminus conv(S)$ tenemos que encontrar una desigualdad que separe x^* de $conv(S)$.
- Los algoritmos que buscan estas desigualdades violadas se llaman algoritmos de separación.

Algoritmos de separación

Estos algoritmos pueden ser:

- Exactos: Dada una clase de desigualdades, el procedimiento toma $x^* \in R$ como entrada y retorna una o más desigualdades de esa clase violadas por x^* o prueba que no existe.

- Heurísticos: Similar pero retorna una o más desigualdades de esa clase violadas por x^* o un mensaje de falla. Es decir, puede existir una desigualdad de esa clase violada pero el procedimiento no es capaz de encontrarla.

Algoritmos de planos de corte

El algoritmo continúa hasta que:

- Una solución entera es encontrada.
- El programa lineal es infactible (significa que el PLEM es infactible).
- No se pudo identificar un corte violado (no se conoce la descripción completa de la cápsula convexa o los algoritmos de separación no son exactos).

Planos de corte

Clasificación:

- De propósito general.
- De relajaciones del problema.
- Específicos del problema.

Planos de corte generales

- Sólo se basan en la condición de integrabilidad de las variables.
- Pueden ser utilizados para cualquier PLE.
- Son muy débiles.
- Cortes de Gomory, disyuntivos 0-1.

Planos de corte de relajaciones del problema

- Una desigualdad válida para una relajación de un problema también es válida para el propio problema.
- Derivar desigualdades válidas para relajaciones comunes.

Planos de corte particulares

- Explotan las características propias del problema.
- Estudio poliedral.

Cortes disyuntivos

$S = \{x \in \mathbb{Z}^n, x \geq 0, Ax \leq b\}$, y $S = \bigcup_{i=1}^{r} S_i$.

Si

$$\Pi^i x \leq \Pi_0^i$$

es desigualdad válida $\forall x \in S_i$, $i = 1, \ldots, r$ y

$$\Pi \leq \Pi^i \text{ y } \Pi_0 \geq \Pi_0^i, i = 1, \ldots, r$$

entonces la desigualdad

$$\Pi x \leq \Pi_0$$

es válida para S.

Separación de cortes disyuntivos para $r = 2$ (cortes *split*)

Sea $S = \{x \in \mathbb{Z}^n, x \geq 0, Ax \leq b\}$, x^* la solución óptima de R, con $x_i^* \notin \mathbb{Z}$ y consideremos la disyunción *split* elemental:

$$x_i \leq \lfloor x_i^* \rfloor \text{ ó } x_i \geq \lfloor x_i^* \rfloor + 1$$

Si el valor óptimo de

$$\begin{array}{ll}
\text{Max} & \Pi x^* - \Pi_0 \\
\text{s.a.} & \Pi \leq uA + u_0 e_i \\
& \Pi \leq vA - v_0 e_i \\
& \Pi_0 \geq bu + \lfloor x_i^* \rfloor u_0 \\
& \Pi_0 \geq bv - (\lfloor x_i^* \rfloor + 1) v_0 \\
& u, u_0, v, v_0 \geq 0 \qquad u, v \in \mathbb{R}^m, \quad u_0, v_0 \in \mathbb{R} \\
& \Pi, \Pi_0 \text{ libres} \qquad\qquad \Pi \in \mathbb{R}^n, \quad \Pi_0 \in \mathbb{R}
\end{array}$$

es mayor que 0, entonces $\Pi x \leq \Pi_0$ es una desigualdad válida para S violada por x^*.

Planos de corte de Gomory

Sea x^* la solución óptima de R, x_k^* fraccionaria con $k \in I$ y su correspondiente fila en el diccionario óptimo:

$$x_k = x_k^* - \sum_{j \in N} \bar{a_{kj}} x_j.$$

Entonces

- Puro ($C = \emptyset$)

$$\bar{f_k} \leq \sum_{j \in N} \bar{f_{k_j}} x_j$$

- Mixto ($C \neq \emptyset$)

$$\bar{f_k} \leq \sum_{j \in N \cap I} \bar{f_{kj}} x_j + \sum_{j \in N \cap C, \bar{a_{kj}} > 0} \bar{a_{kj}} x_j + \sum_{j \in N \cap C, \bar{a_{kj}} < 0} \frac{f_k}{1 - f_k} \bar{a_{kj}} x_j$$

donde $\bar{f_k} = \bar{b_k} - \lfloor \bar{b_k} \rfloor$ y $\bar{f_{kj}} = \bar{a_{kj}} - \lfloor \bar{a_{kj}} \rfloor$, son desigualdades válidas para S.

Método de planos de corte de Gomory

1. Resolver la relajación lineal inicial, R, por el método SIMPLEX
2. Sea x^* la solución óptima de R. Si x_k^* es entera $\forall k \in I$, PARAR
3. Sea x_k^* fraccionaria con $k \in I$ y su correspondiente fila en el diccionario óptimo:

$$x_k = x_k^* + \sum_{j \in N} \bar{a_{kj}} x_j$$

4. Generar el plano de corte:

$$\bar{f_k} \leq \sum_{j \in N \cap I} \bar{f_{kj}} x_j + \sum_{j \in N \cap C, \bar{a_{kj}} > 0} \bar{a_{kj}} x_j + \sum_{j \in N \cap C, \bar{a_{kj}} < 0} \frac{f_k}{1 - f_k} \bar{a_{kj}} x_j$$

5. Agregar el plano de corte a R y reoptimizar (por SIMPLEX DUAL). Retornar a 2.

Método de planos de corte de Gomory

- Convergencia finita dada cierta regla para la elección de la variable fraccionaria en el Paso 3.
- Generalmente es necesario un gran número de planos de corte.
- Errores numéricos pueden generar soluciones incorrectas o que el programa falle.
- Recien se obtiene una solución factible al finalizar el algoritmo.

Método de planos de corte de Gomory

$$
\begin{array}{lrcl}
\text{Max} & -x_1 + 4x_2 & & \\
 & 2x_1 + x_2 & \leq & 10 \\
 & -2x_1 + 3x_2 & \leq & 8 \\
 & x_1, x_2 & \geq & 0 \\
 & x_1, x_2 & \in & \mathbb{Z}
\end{array}
$$

Método de planos de corte de Gomory

$$
\begin{array}{lrcl}
\text{Max} & -x_1 + 4x_2 & & \\
 & 2x_1 + x_2 & \leq & 10 \\
 & -2x_1 + 3x_2 & \leq & 8 \\
 & x_1, x_2 & \geq & 0 \\
 & x_1, x_2 & \in & \mathbb{Z}
\end{array}
$$

Diccionario óptimo de la relajación:

$$
\begin{array}{lcl}
x_1 & = & 11/4 + 3/8x_3 - 1/8x_4 \\
x_2 & = & 9/2 + 1/4x_3 + 1/4x_4 \\
z & = & 61/4 - 5/8x_3 - 9/8x_4
\end{array}
$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8 x_3 + 1/8 x_4$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$11/4 = x_1 - 3/8x_3 + 1/8x_4$ (por ser $x_j \geq 0$)

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8 x_3 + 1/8 x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad (\text{por ser } x_j \geq 0)$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$11/4 = x_1 - 3/8x_3 + 1/8x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4$ (por ser $x_j \geq 0$)

$$x_1 - x_3 + 0x_4$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8x_3 + 1/8x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad (\text{por ser } x_j \geq 0)$$

$$x_1 - x_3 + 0x_4 \qquad (\text{porque } x_j \in \mathbb{Z})$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8 x_3 + 1/8 x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad \text{(por ser } x_j \geq 0\text{)}$$

$$2 = \lfloor 11/4 \rfloor \geq x_1 - x_3 + 0 x_4 \qquad \text{(porque } x_j \in \mathbb{Z}\text{)}$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8x_3 + 1/8x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad \text{(por ser } x_j \geq 0\text{)}$$

$$2 = \lfloor 11/4 \rfloor \geq x_1 - x_3 + 0x_4 \qquad \text{(porque } x_j \in \mathbb{Z}\text{)}$$

Restándole la fila del diccionario:

$$\begin{array}{lrcl} +(& x_1 - x_3 + 0x_4 & \leq & 2) \\ -(& x_1 - 3/8x_3 + 1/8x_4 & = & 11/4) \\ \hline \end{array}$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8x_3 + 1/8x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad \text{(por ser } x_j \geq 0\text{)}$$

$$2 = \lfloor 11/4 \rfloor \geq x_1 - x_3 + 0x_4 \qquad \text{(porque } x_j \in \mathbb{Z}\text{)}$$

Restándole la fila del diccionario:

$$\begin{array}{rrcl}
+(& x_1 - x_3 + 0x_4 & \leq & 2) \\
-(& x_1 - 3/8x_3 + 1/8x_4 & = & 11/4) \\
\hline
 & -5/8x_3 - 1/8x_4 & \leq & \text{-}3/4
\end{array}$$

Método de planos de corte de Gomory

Derivemos un corte de Gomory sobre x_1:

$$11/4 = x_1 - 3/8x_3 + 1/8x_4 \geq x_1 + \lfloor -3/8 \rfloor x_3 + \lfloor 1/8 \rfloor x_4 \qquad \text{(por ser } x_j \geq 0\text{)}$$

$$2 = \lfloor 11/4 \rfloor \geq x_1 - x_3 + 0x_4 \qquad \text{(porque } x_j \in \mathbb{Z}\text{)}$$

Restándole la fila del diccionario:

$$\begin{array}{lrcl} +(& x_1 - x_3 + 0x_4 & \leq & 2) \\ -(& x_1 - 3/8x_3 + 1/8x_4 & = & 11/4) \\ \hline & -5/8x_3 - 1/8x_4 & \leq & \text{-}3/4 \end{array}$$

Entonces la desigualdad

$$3/4 \leq 5/8x_3 + 1/8x_4$$

es válida para S. Como en la solución óptima de la relajación lineal x_3 y x_4 valen 0 por ser variables no básicas, está violada por esa solución.

Método de planos de corte de Gomory

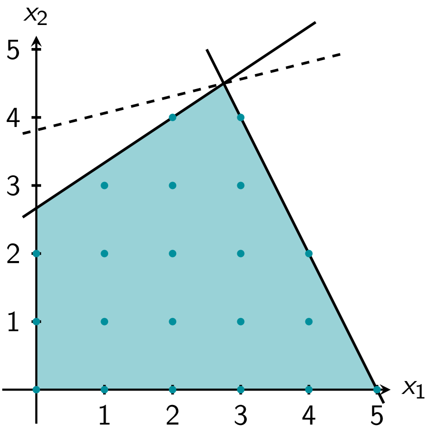
Lo agregamos:

$$
\begin{aligned}
x_1 &= 11/4 + 3/8x_3 - 1/8x_4 \\
x_2 &= 9/2 + 1/4x_3 + 1/4x_4 \\
x_5 &= -3/4 - 5/8x_3 - 1/8x_4 \\
z &= 61/4 - 5/8x_3 - 9/8x_4
\end{aligned}
$$

Método de planos de corte de Gomory

Gráficamente:

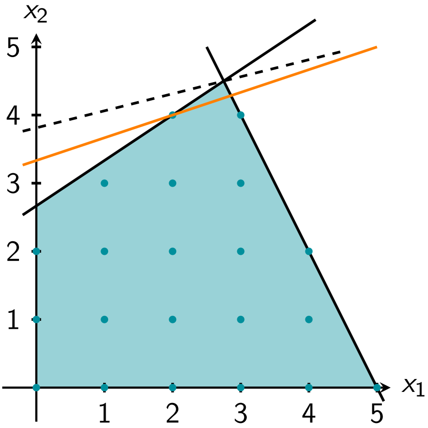
$$
\begin{array}{llrcl}
\text{Max} & z = -x_1 + 4x_2 & & & \\
\text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\
 & & -2x_1 + 3x_2 & \leq & 8 \\
 & & x_1, x_2 & \geq & 0
\end{array}
$$



Método de planos de corte de Gomory

Gráficamente:

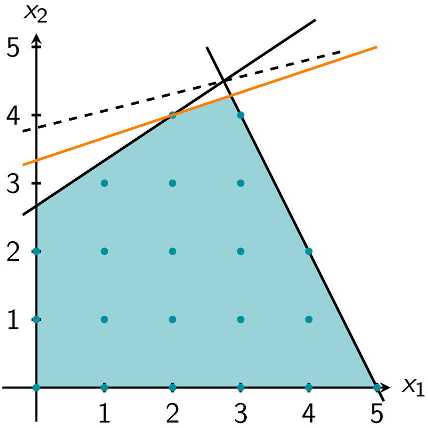
$$
\begin{array}{lrcl}
\text{Max} & z = -x_1 + 4x_2 & & \\
\text{s.a.} & 2x_1 + x_2 & \leq & 10 \\
 & -2x_1 + 3x_2 & \leq & 8 \\
 & -x_1 + 3x_2 & \leq & 10 \\
 & x_1, x_2 & \geq & 0
\end{array}
$$



Método de planos de corte de Gomory

Gráficamente:

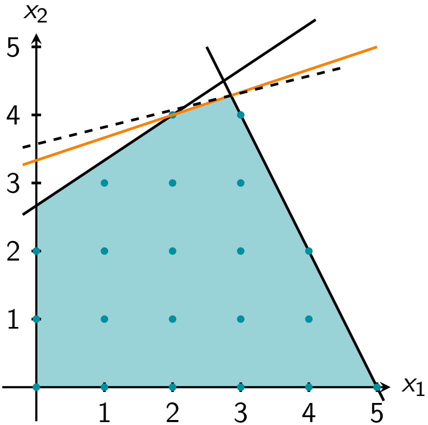
$$
\begin{array}{llrcl}
\text{Max} & z = -x_1 + 4x_2 & & & \\
\text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\
 & & -2x_1 + 3x_2 & \leq & 8 \\
 & & -x_1 + 3x_2 & \leq & 10 \\
 & & x_1, x_2 & \geq & 0
\end{array}
$$



Método de planos de corte de Gomory

Gráficamente:

$$\begin{array}{llrcl}
\text{Max} & z = -x_1 + 4x_2 & & & \\
\text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\
 & & -2x_1 + 3x_2 & \leq & 8 \\
 & & \color{orange}{-x_1 + 3x_2} & \color{orange}{\leq} & \color{orange}{10} \\
 & & x_1, x_2 & \geq & 0
\end{array}$$



Método de planos de corte de Gomory

Diccionario óptimo de la relajación:

$$
\begin{aligned}
x_1 &= 20/7 + 3/7x_3 - 1/7x_5 \\
x_2 &= 30/7 + 1/7x_3 + 2/7x_5 \\
x_4 &= 6/7 + 3/7x_3 - 8/7x_4 \\
z &= 100/7 - 1/7x_3 - 9/7x_4
\end{aligned}
$$

Método de planos de corte de Gomory

Diccionario óptimo de la relajación:

$$\begin{aligned} x_1 &= 20/7 + 3/7x_3 - 1/7x_5 \\ x_2 &= 30/7 + 1/7x_3 + 2/7x_5 \\ x_4 &= 6/7 + 3/7x_3 - 8/7x_4 \\ z &= 100/7 - 1/7x_3 - 9/7x_4 \end{aligned}$$

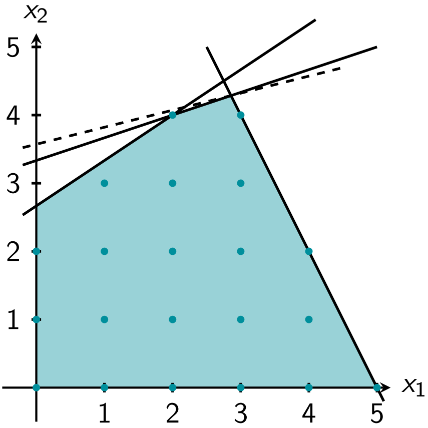
Derivemos un corte de Gómory sobre x_1:

$$3/7x_3 + 6/7x_5 \leq 6/7$$

Método de planos de corte de Gomory

Gráficamente:

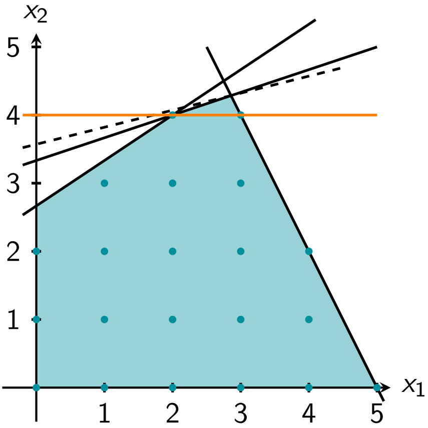
$$\begin{array}{llrcl} \text{Max} & z = -x_1 + 4x_2 & & & \\ \text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\ & & -2x_1 + 3x_2 & \leq & 8 \\ & & -x_1 + 3x_2 & \leq & 10 \\ & & x_1, x_2 & \geq & 0 \end{array}$$



Método de planos de corte de Gomory

Gráficamente:

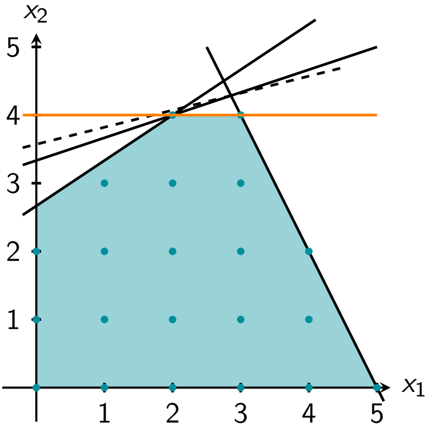
$$
\begin{array}{llrcl}
\text{Max} & z = -x_1 + 4x_2 & & & \\
\text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\
& & -2x_1 + 3x_2 & \leq & 8 \\
& & -x_1 + 3x_2 & \leq & 10 \\
& & +x_2 & \leq & 4 \\
& & x_1, x_2 & \geq & 0
\end{array}
$$



Método de planos de corte de Gomory

Gráficamente:

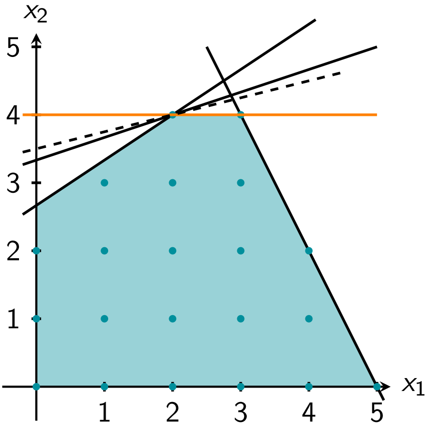
$$
\begin{array}{lrcl}
\text{Max} & z = -x_1 + 4x_2 & & \\
\text{s.a.} & 2x_1 + x_2 & \leq & 10 \\
 & -2x_1 + 3x_2 & \leq & 8 \\
 & -x_1 + 3x_2 & \leq & 10 \\
 & +x_2 & \leq & 4 \\
 & x_1, x_2 & \geq & 0
\end{array}
$$



Método de planos de corte de Gomory

Gráficamente:

$$\begin{array}{llrcl}
\text{Max} & z = -x_1 + 4x_2 & & & \\
\text{s.a.} & & 2x_1 + x_2 & \leq & 10 \\
& & -2x_1 + 3x_2 & \leq & 8 \\
& & -x_1 + 3x_2 & \leq & 10 \\
& & +x_2 & \leq & 4 \\
& & x_1, x_2 & \geq & 0
\end{array}$$



Método de planos de corte de Gomory

Lo agregamos:

$$
\begin{aligned}
x_1 &= 20/7 + 3/7x_3 - 1/7x_5 \\
x_2 &= 30/7 + 1/7x_3 + 2/7x_5 \\
x_4 &= 6/7 + 3/7x_3 - 8/7x_5 \\
x_5 &= -6/7 - 3/7x_3 - 6/7x_5 \\
z &= 100/7 - 1/7x_3 - 9/7x_5
\end{aligned}
$$

Método de planos de corte de Gomory

Lo agregamos:

$$
\begin{array}{lcl}
x_1 & = & 20/7 + 3/7x_3 - 1/7x_5 \\
x_2 & = & 30/7 + 1/7x_3 + 2/7x_5 \\
x_4 & = & 6/7 + 3/7x_3 - 8/7x_5 \\
x_5 & = & -6/7 - 3/7x_3 - 6/7x_5 \\
z & = & 100/7 - 1/7x_3 - 9/7x_5
\end{array}
$$

Diccionario óptimo de la relajación:

$$
\begin{array}{lcccccc}
x_1 & = & 2 & - & x_5 & + & x_6 \\
x_2 & = & 4 & + & & & 1/3x_6 \\
x_3 & = & 0 & - & 2x_5 & + & x_6 \\
x_4 & = & 2 & + & 2x_5 & - & 7/3x_6 \\
z & = & 14 & - & x_5 & - & 1/3x_6
\end{array}
$$

Método de planos de corte de Gomory

Lo agregamos:

$$
\begin{array}{rcl}
x_1 & = & 20/7 + 3/7x_3 - 1/7x_5 \\
x_2 & = & 30/7 + 1/7x_3 + 2/7x_5 \\
x_4 & = & 6/7 + 3/7x_3 - 8/7x_5 \\
x_5 & = & -6/7 - 3/7x_3 - 6/7x_5 \\
z & = & 100/7 - 1/7x_3 - 9/7x_5
\end{array}
$$

Diccionario óptimo de la relajación:

$$
\begin{array}{rcccccc}
x_1 & = & 2 & - & x_5 & + & x_6 \\
x_2 & = & 4 & + & & & 1/3x_6 \\
x_3 & = & 0 & - & 2x_5 & + & x_6 \\
x_4 & = & 2 & + & 2x_5 & - & 7/3x_6 \\
z & = & 14 & - & x_5 & - & 1/3x_6
\end{array}
$$

Solución entera!

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

S conjunto de puntos que cumplen:

$$\begin{aligned} 2x_1 - 3x_2 &\leq 13 \\ x_1 &\leq 6 \\ 3x_1 - 2x_2 &\leq 10 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in Z \end{aligned}$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

S conjunto de puntos que cumplen:

$$
\begin{array}{rcl}
2x_1 - 3x_2 & \leq & 13 \\
x_1 & \leq & 6 \\
3x_1 - 2x_2 & \leq & 10 \\
x_1, x_2 & \geq & 0 \\
x_1, x_2 & \in & \mathbb{Z}
\end{array}
$$

Si multiplicamos las restricciones por constantes positivas y las sumamos, obtenemos una desigualdad válida para S:

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

S conjunto de puntos que cumplen:

$$\begin{array}{rcl} 2x_1 - 3x_2 & \leq & 13 \\ x_1 & \leq & 6 \\ 3x_1 - 2x_2 & \leq & 10 \\ x_1, x_2 & \geq & 0 \\ x_1, x_2 & \in & \mathbb{Z} \end{array}$$

Si multiplicamos las restricciones por constantes positivas y las sumamos, obtenemos una desigualdad válida para S:

$$\begin{array}{rcl} 1/2(2x_1 - 3x_2 & \leq & 13) \\ + & & \\ 1(x_1 & \leq & 6) \\ + & & \\ 0(3x_1 - 2x_2 & \leq & 10) \\ \hline 2x_1 - 3/2x_2 & \leq & 25/2 \end{array}$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 3/2x_2 \leq 25/2.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$\lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2x_2 \leq 25/2.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 2x_2 \leq \lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2 x_2 \leq 25/2.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 2x_2 \leq \lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2x_2 \leq 25/2.$$

Como $x_1, x_2 \in \mathbb{Z}$, entonces

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 2x_2 \leq \lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2 x_2 \leq 25/2.$$

Como $x_1, x_2 \in \mathbb{Z}$, entonces

$$2x_1 - 2x_2 \in \mathbb{Z},$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 2x_2 \leq \lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2 x_2 \leq 25/2.$$

Como $x_1, x_2 \in \mathbb{Z}$, entonces

$$2x_1 - 2x_2 \in \mathbb{Z},$$

lo que implica que

$$2x_1 - 2x_2 \leq \lfloor 25/2 \rfloor = 12$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Como $x_1, x_2 \geq 0$,

$$2x_1 - 2x_2 \leq \lfloor 2 \rfloor x_1 + \lfloor -3/2 \rfloor x_2 \leq 2x_1 - 3/2 x_2 \leq 25/2.$$

Como $x_1, x_2 \in \mathbb{Z}$, entonces

$$2x_1 - 2x_2 \in \mathbb{Z},$$

lo que implica que

$$2x_1 - 2x_2 \leq \lfloor 25/2 \rfloor = 12$$

Por lo tanto

$$2x_1 - 2x_2 \leq 12$$

es una desigualdad válida para S.

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}_+^m \Longrightarrow$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}^m_+ \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij} x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}^m_+ \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij}x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \geq 0$, $\Longrightarrow$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}_+^m \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij} x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \geq 0$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}^m_+ \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij}x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \geq 0$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \in \mathbb{Z}^n$, $\Longrightarrow$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}^m_+ \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij} x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \geq 0$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \in \mathbb{Z}^n$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \in \mathbb{Z},$$

Desigualdades Chvátal-Gomory (para $C = \emptyset$)

Dado

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m, x \geq 0\}.$$

Sea $\mu \in \mathbb{R}^m_+ \Longrightarrow$

$$\sum_{i=1}^{m} \mu_i \sum_{j=1}^{n} a_{ij}x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \geq 0$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \leq \sum_{i=1}^{m} \mu_i b_i \text{ es válida para } S.$$

Como $x \in \mathbb{Z}^n$, $\Longrightarrow$

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \in \mathbb{Z},$$

y por lo tanto

$$\sum_{j=1}^{n} \lfloor \sum_{i=1}^{m} \mu_i a_{ij} \rfloor x_j \leq \lfloor \sum_{i=1}^{m} \mu_i b_i \rfloor \text{ es válida para } S.$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Dada un conjunto de puntos S que cumplen una restricción tipo mochila

$$S = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}$$

con $a_i \geq 0$ y $C \subset \{1, \ldots, n\}$ tal que

$$\sum_{i \in C} a_i > b.$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Dada un conjunto de puntos S que cumplen una restricción tipo mochila

$$S = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}$$

con $a_i \geq 0$ y $C \subset \{1, \ldots, n\}$ tal que

$$\sum_{i \in C} a_i > b.$$

La desigualdad de cubrimiento

$$\sum_{i \in C} x_i \leq |C| - 1$$

es válida para S.

Desigualdades de cubrimiento (para $C = \emptyset$)

Ejemplo:

$$S = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

Como $11 + 6 + 6 > 19$,

Desigualdades de cubrimiento (para $C = \emptyset$)

Ejemplo:

$$S = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

Como $11 + 6 + 6 > 19$,

$$x_1 + x_2 + x_3 \leq 2$$

es desigualdad válida para S.

Desigualdades de cubrimiento (para $C = \emptyset$)

Ejemplo:

$$S = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

Como $11 + 6 + 6 > 19$,

$$x_1 + x_2 + x_3 \leq 2$$

es desigualdad válida para S.

También

$$x_1 + x_2 + x_6 \leq 2$$

$$x_1 + x_5 + x_6 \leq 2$$

$$x_3 + x_4 + x_5 + x_6 \leq 3$$

son desigualdades válidas para S.

Desigualdades de cubrimiento (para $C = \emptyset$)

La desigualdad $x_3 + x_4 + x_5 + x_6 \leq 3$ se puede extender a

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ que domina a la anterior.

Desigualdades de cubrimiento (para $C = \emptyset$)

La desigualdad $x_3 + x_4 + x_5 + x_6 \leq 3$ se puede extender a

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ que domina a la anterior.

Proposición

Sea $C \subset \{1, \ldots, n\}$ *tal que* $\sum_{i \in C} a_i > b$, *entonces*

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

es válida para S, *con* $E(C) = C \cup \{j : a_j \geq a_i \ \forall i \in C\}$.

Desigualdades de cubrimiento (para $C = \emptyset$)

La desigualdad $x_3 + x_4 + x_5 + x_6 \leq 3$ se puede extender a

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ que domina a la anterior.

Proposición

Sea $C \subset \{1, \ldots, n\}$ *tal que* $\sum_{i \in C} a_i > b$*, entonces*

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

es válida para S*, con* $E(C) = C \cup \{j : a_j \geq a_i \ \forall i \in C\}$*.*

En realidad, esta desigualdad también está dominada por

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3.$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.
  - Si $x_1 = 0$, es válida para todo α_1.

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.
  - Si $x_1 = 0$, es válida para todo α_1.
  - Si $x_1 = 1$, queremos el mayor α_1 tal que se cumpla

    $$\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3 \quad \forall x \in \{6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 - 11$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.
  - Si $x_1 = 0$, es válida para todo α_1.
  - Si $x_1 = 1$, queremos el mayor α_1 tal que se cumpla

    $$\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3 \quad \forall x \in \{6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 - 11$$

    Para eso planteamos

    $$\max\ z_1 = x_3 + x_4 + x_5 + x_6$$

    $$\begin{aligned} \text{s.a } 6x_3 + 5x_4 + 5x_5 + 4x_6 &\leq 8 \\ x_3, x_4, x_5, x_6 &\in \{0, 1\} \end{aligned}$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.
  - Si $x_1 = 0$, es válida para todo α_1.
  - Si $x_1 = 1$, queremos el mayor α_1 tal que se cumpla

    $$\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3 \quad \forall x \in \{6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 - 11$$

    Para eso planteamos

    $$\max z_1 = x_3 + x_4 + x_5 + x_6$$

    $$\begin{array}{rcl} \text{s.a } 6x_3 + 5x_4 + 5x_5 + 4x_6 & \leq & 8 \\ x_3, x_4, x_5, x_6 & \in & \{0,1\} \end{array}$$

    Como $z_1 = 1$, entonces $\alpha_1 =$

Desigualdades de cubrimiento (para $C = \emptyset$)

Veamos como la podemos derivar:

- Sabemos que $x_3 + x_4 + x_5 + x_6 \leq 3$ es válida y queremos incorporar a x_1 en la desigualdad.

  Para eso necesitamos saber el mayor valor de α_1 tal que
  $$\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$$
  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.
  - Si $x_1 = 0$, es válida para todo α_1.
  - Si $x_1 = 1$, queremos el mayor α_1 tal que se cumpla

    $$\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3 \quad \forall x \in \{6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 - 11$$

    Para eso planteamos

    $$\max z_1 = x_3 + x_4 + x_5 + x_6$$

    $$\begin{array}{rcl} \text{s.a } 6x_3 + 5x_4 + 5x_5 + 4x_6 & \leq & 8 \\ x_3, x_4, x_5, x_6 & \in & \{0, 1\} \end{array}$$

    Como $z_1 = 1$, entonces $\alpha_1 = 3 - 1 = 2$.

Desigualdades de cubrimiento (para $C = \emptyset$)

- Ahora sabemos que $2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S y queremos incorporar a x_2 a la desigualdad. Buscamos el mayor α_2 tal que

$$2x_1 + \alpha_2 x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$. Razonando de igual manera, plantemos

Desigualdades de cubrimiento (para $C = \emptyset$)

- Ahora sabemos que $2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S y queremos incorporar a x_2 a la desigualdad. Buscamos el mayor α_2 tal que

$$2x_1 + \alpha_2 x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$. Razonando de igual manera, plantemos

$$\begin{aligned} \max\ z_2 = 2x_1 + x_3 + x_4 + x_5 + x_6 \\ \text{s.a } 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 &\leq 13 \\ x_1, x_3, x_4, x_5, x_6 &\in \{0, 1\} \end{aligned}$$

Desigualdades de cubrimiento (para $C = \emptyset$)

- Ahora sabemos que $2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S y queremos incorporar a x_2 a la desigualdad. Buscamos el mayor α_2 tal que

$$2x_1 + \alpha_2 x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$. Razonando de igual manera, plantemos

$$\begin{aligned} \max\ z_2 = 2x_1 + x_3 + x_4 + x_5 + x_6 & & \\ \text{s.a}\ 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 & \leq & 13 \\ x_1, x_3, x_4, x_5, x_6 & \in & \{0,1\} \end{aligned}$$

Como $z_2 = 2$, entonces $\alpha_2 =$

Desigualdades de cubrimiento (para $C = \emptyset$)

- Ahora sabemos que $2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S y queremos incorporar a x_2 a la desigualdad. Buscamos el mayor α_2 tal que

$$2x_1 + \alpha_2 x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$. Razonando de igual manera, plantemos

$$\begin{aligned} \max\ z_2 = 2x_1 + x_3 + x_4 + x_5 + x_6 & & \\ \text{s.a } 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 & \leq & 13 \\ x_1, x_3, x_4, x_5, x_6 & \in & \{0,1\} \end{aligned}$$

Como $z_2 = 2$, entonces $\alpha_2 = 3 - 2 = 1$.

Desigualdades de cubrimiento (para $C = \emptyset$)

- Por último, sabiendo que $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S, queremos encontrar el mayor α_7 tal que

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + \alpha x_7 \leq 3$$

  es válida en $\{11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$.

  Para eso plantemos

$$\begin{array}{rcl} \max\ z_7 = 2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & & \\ \text{s.a } 11x_1 + x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 & \leq & 18 \\ x_1, x_2, x_3, x_4, x_5, x_6 & \in & \{0,1\} \end{array}$$

  Como $z_7 = 3$, entonces $\alpha_7 = 3 - 3 = 0$.

Finalmente $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ es válida para S.

Desigualdades de cubrimiento (para $C = \emptyset$)

Procedimiento general:

Sea $j_1 \dots j_r$ un orden en $N - C$. Para $t = 1, \dots, r$ supongamos que ya *lifteamos* para $i = 1, \dots, t-1$, entonces

$$\sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$$

es válida en S, y ahora queremos ajustar sobre x_{j_t}, es decir encontrar el mayor α_{j_t} tal que

$$\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$$

sea válida en S.

Desigualdades de cubrimiento (para $C = \emptyset$)

Planteamos

$$\max\ z_t = \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j$$

$$\text{s.a.} \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \quad \leq \quad b - a_{j_t}$$

$$x \in \{0,1\}^{|C|+t-1}$$

Desigualdades de cubrimiento (para $C = \emptyset$)

Planteamos

$$\max z_t = \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j$$

$$\text{s.a.} \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \quad \leq \quad b - a_{j_t}$$

$$x \in \{0,1\}^{|C|+t-1}$$

y tomamos

$$\alpha_{j_t} = |C| - 1 - z_t.$$

Algoritmos de separación para desigualdades de cubrimiento

¿Cómo las separamos?

Sea x^* tq $0 \leq x_j^* \leq 1 \; \forall j \in \{1, \ldots, n\}$ con alguna coordenada fraccionaria. Buscamos C *cubrimiento* tal que

$$\sum_{j \in C} x_j^* > |C| - 1$$

Algoritmos de separación para desigualdades de cubrimiento

¿Cómo las separamos?

Sea x^* tq $0 \leq x_j^* \leq 1 \ \forall j \in \{1, \ldots, n\}$ con alguna coordenada fraccionaria. Buscamos C *cubrimiento* tal que

$$\sum_{j \in C} x_j^* > |C| - 1 \Longleftrightarrow \sum_{j \in C} (1 - x_j^*) < 1.$$

Algoritmos de separación para desigualdades de cubrimiento

¿Cómo las separamos?

Sea x^* tq $0 \leq x_j^* \leq 1 \ \forall j \in \{1, \ldots, n\}$ con alguna coordenada fraccionaria. Buscamos C *cubrimiento* tal que

$$\sum_{j \in C} x_j^* > |C| - 1 \Longleftrightarrow \sum_{j \in C} (1 - x_j^*) < 1.$$

Es decir, buscamos $C \subset N$ tq

$$\sum_{j \in C} a_j > b$$

$$\sum_{j \in C} (1 - x_j^*) < 1$$

Algoritmos de separación para desigualdades de cubrimiento

Esto lo podemos formular de la siguiente manera:

Sea $z_j \in \{0,1\}$ tq $z_j = \begin{cases} 1 & \text{si } j \in C \\ 0 & \text{si } j \notin C \end{cases}$

$$\min \sum_{j=1}^{n} (1 - x_j^*) z_j = \phi$$

$$s.a. \sum_{j=1}^{n} a_j z_j \geq b + 1 \quad (> b \in \mathbb{Z})$$

Si $\phi < 1$: tenemos un *cubrimiento* violado.

Si $\phi \geq 1$: no hay *cubrimiento* violado.

Algoritmos de separación para desigualdades de cubrimiento

Ejemplo:
$X = \{x \in \{0,1\}^6 : 45x_1 + 46x_2 + 79x_3 + 54x_4 + 53x_5 + 125x_6 \leq 178\}$

Solución óptima de la relajación: $x^* = (0, 0, 3/4, 1/2, 1, 0)$

Problema de separación:

$$\min z_1 + z_2 + 1/4 z_3 + 1/2 z_4 + 0 z_5 + z_6 = \phi$$

$$\text{s.a. } 45z_1 + 46z_2 + 79z_3 + 54z_4 + 53z_5 + 125z_6 \geq 179 > 178$$

$$z_1, z_2, z_3, z_4, z_5, z_6 \in \{0, 1\}$$

Como $\phi = 3/4 < 1$, $z^* = (0, 0, 1, 1, 1, 0) \Longrightarrow$ des. cover violada:

$$x_3 + x_4 + x_5 \leq 2.$$

Desigualdades por redondeo

- Puro: Si $S = \{y \in \mathbb{Z} : y \leq b\}$ entonces

$$y \leq \lfloor b \rfloor$$

  es desigualdad válida para S.

Desigualdades por redondeo

- Puro: Si $S = \{y \in \mathbb{Z} : y \leq b\}$ entonces

$$y \leq \lfloor b \rfloor$$

  es desigualdad válida para S.

- Entero-mixto (*mixed-integer rounding*)
  Dado un conjunto de puntos

$$S = \{(y, x) \in \mathbb{Z} x \mathbb{R} : y \leq b + x,\ x, y \geq 0, b \notin \mathbb{Z}\}$$

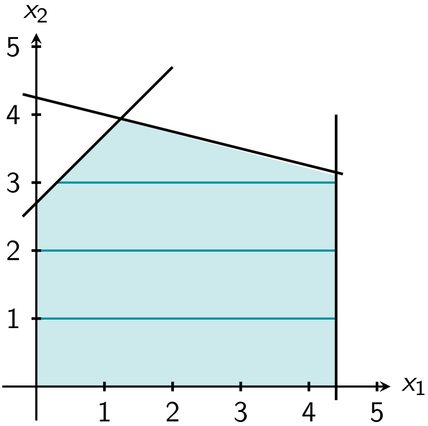
  la desigualdad

$$y \leq \lfloor b \rfloor + \frac{x}{1 - f}$$

  donde $f = b - \lfloor b \rfloor$ es válida para S.

Desigualdades por redondeo entero-mixto

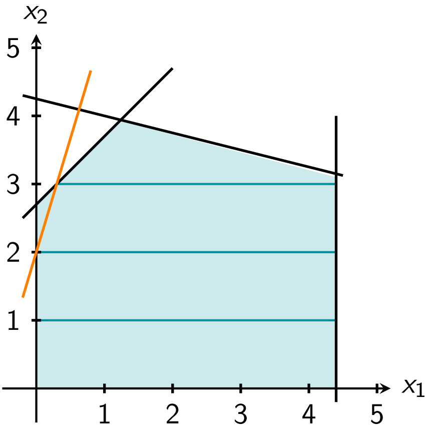
$$
\begin{aligned}
-x_1 + x_2 &\leq 2.7 \\
x_1 + 4x_2 &\leq 17 \\
x_1, x_2 &\geq 0 \\
x_2 &\in \mathbb{Z}
\end{aligned}
$$



Desigualdades por redondeo entero-mixto

$$
\begin{array}{rcl}
-x_1 + x_2 & \leq & 2.7 \\
x_1 + 4x_2 & \leq & 17 \\
x_1, x_2 & \geq & 0 \\
x_2 & \in & \mathbb{Z}
\end{array}
$$

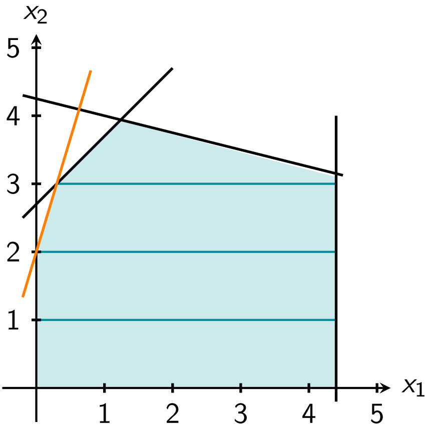
$x_2 \leq 2 + 1/(1 - 0.7)x_1$



Desigualdades por redondeo entero-mixto

$$
\begin{aligned}
-x_1 + x_2 &\leq 2.7 \\
x_1 + 4x_2 &\leq 17 \\
x_1, x_2 &\geq 0 \\
x_2 &\in \mathbb{Z}
\end{aligned}
$$

$x_2 \leq 2 + 1/(1 - 0.7)x_1$



Desigualdades particulares - Conjunto independiente

Dado $G = (V, X)$ buscamos el conjunto de independiente de mayor cardinalidad.

Variable:

$$x_i = \begin{cases} 1 & \text{si } v_i \text{ pertenece al conjunto independiente} \\ 0 & \text{si no} \end{cases}$$

$$\begin{array}{lll} \text{Max} & \displaystyle\sum_{v_i \in V} x_i & \\ \text{sa} & x_i + x_j \leq 1 & \forall (v_i, v_j) \in X \\ & x_i \in \{0, 1\} & \forall v_i \in V \end{array}$$

Desigualdades particulares - Conjunto independiente

- Si $C \subseteq V$ es una clique

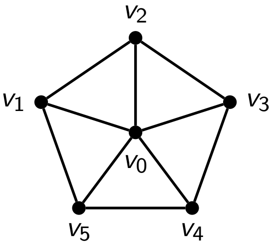
Desigualdades particulares - Conjunto independiente

- Si $C \subseteq V$ es una clique $\Longrightarrow \sum_{j \in C} x_j \leq 1$ es una desigualdad válida.
- Si H es un agujero impar

Desigualdades particulares - Conjunto independiente

- Si $C \subseteq V$ es una clique $\Longrightarrow \sum_{j \in C} x_j \leq 1$ es una desigualdad válida.
- Si H es un agujero impar $\Longrightarrow \sum_{j \in H} x_j \leq \frac{|H|-1}{2}$ es una desigualdad válida.

Ejemplo:



$$x_1 + x_2 + x_0 \leq 1$$

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$$

Desigualdades particulares - Conjunto independiente

- Si H es un rueda impar con centro v_0

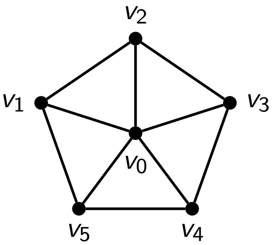
Desigualdades particulares - Conjunto independiente

- Si H es un rueda impar con centro $v_0 \Longrightarrow$

$$\frac{|H|-1}{2} x_0 + \sum_{j \in H} x_j \leq \frac{|H|-1}{2}$$

  es una desigualdad válida.

Ejemplo:



$$2x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$$

Desigualdades particulares - Viajante de comercio

- *2-matching*: Dados $H \subseteq V$, $3 \leq |H| \leq |V| - 1$, $\hat{E} \subseteq \delta(H)$ disjunto de cardinal k impar, $\hat{E} = \{W_i = (u_i, v_i), u_i \neq u_j, v_i \neq v_j, i, j = 1, \ldots, k\}$.

  Llamamos mango a H y dientes a $\hat{E}$. La desigualdad

  $$\sum_{e \in E(H)} x_e + \sum_{e \in \hat{E}} x_e \leq |H| + \lfloor \frac{|\hat{E}|}{2} \rfloor$$

  es válida para S, donde $E(H) = \{(u, v) \in X : u, v \in H\}$ y $\delta(H) = \{(u, v) \in X : u \in H, v \notin H\}$.

Desigualdades particulares - Viajante de comercio

- peine (comb): Dados $H, W_i, \ldots, W_k \subseteq V$ tales que:
  - $|H \cap W_i| \geq 1$, $i = 1, \ldots, k$
  - $|W_i \setminus H| \geq 1$, $i = 1, \ldots, k$
  - $2 \leq |W_i| \leq n - 2$, $i = 1, \ldots, k$
  - $W_i \cap W_j = \emptyset$, $i, j = 1, \ldots, k$, $i \neq j$
  - $k \geq 3$ impar

La desigualdad

$$\sum_{e \in E(H)} x_e + \sum_{i=1}^{k} \sum_{e \in E(W_i)} x_e \leq |H| + \sum_{i=1}^{k} (|W_i| - 1) - \frac{k+1}{2}$$

es válida para S.

Algoritmos Branch & Cut

Algoritmos Branch & Bound

$+$

Algoritmos de Planos de Corte

Previo al proceso de ramificación se considera la posibilidad de aplicar un algoritmo de planos de corte a la relajación lineal asociada al nodo (separación).

Algoritmos Branch & Cut

- Buena formulación
- Cotas Iniciales
- Preprocesamiento
- Separación
- Heurística Primal
- Estrategia de ramificación
- Estrategia de recorrido del árbol
- Fijado de variables
- IPC / *skip factor*

Algoritmo BC: Buena formulación

¿Que es una buena formulación?

- ¿La que tiene menos variables?
- ¿La que tiene menos restricciones?
- ¿La que tiene relajacion lineal más "ajustada"?

¿Cómo resolver más rápido las relajaciones lineales?

- Restricciones originales tratadas como planos de corte.
- Generación de columnas (variables) a demanda.

Algoritmo BC: Cotas iniciales

- Cualquier solución factible brinda una cota del óptimo del problema (el óptimo seguro es mejor o igual que esa cota).
- Buenas cotas permiten podar en mayor medida el árbol de enumeración y fijar variables.
- Obtenidas por heurísticas rápidas y eficientes.

Algoritmo BC: Preprocesamiento

- Fijado de variables:

$$
\begin{array}{lrcl}
\text{Min} & 5x_1 + 7x_2 + 10x_3 + 3x_4 + x_5 & & \\
\text{sa} & x_1 - x_2 + 5x_3 + x_4 - x_5 & \geq & 2 \\
 & -2x_1 + 6x_2 - 3x_3 - 2x_4 + 2x_5 & \geq & 0 \\
 & -x_2 + 2x_3 - 2x_4 - x_5 & \geq & 1 \\
 & x_1, x_2, x_3, x_4, x_5 & \in & \{0, 1\}
\end{array}
$$

  - Por la última restricción debe pasar:

$$2x_3 \geq 1 + x_2 + 2x_4 + x_5 \geq 1,$$

    y entonces:

$$x_3 \geq 1/2 \Longrightarrow x_3 = 1,$$

    y por lo tanto x_3 puede ser removida del PLEM.

Algoritmo BC: Preprocesamiento

- Fijado de variables:
  - Por la segunda restricción:

    $$6x_2 \geq 3 + 2x_1 + 2x_4 - 2x_5$$

    y entonces:

    $$x_2 \geq 1/6 \Longrightarrow x_2 = 1.$$

  - Por la última restricción, $-1 + 2 - 2x_4 - x_5 \geq 1$

    $$2x_4 \leq -x_5 \leq 0,$$

    y entonces:

    $$x_4 \leq 0 \Longrightarrow x_4 = 0.$$

  - Por la última restricción,

    $$x_5 \leq 0 \Longrightarrow x_5 = 0.$$

Algoritmo BC: Separación

Algoritmos de separación eficientes

- Tiempo polinomial
- Encuentre cortes si existen

Algoritmo BC: Heurística primal

- Observando la estructura de las soluciones fraccionarias, pueden deducirse buenas soluciones factibles que permitan podar en mayor medida el árbol de enumeración y fijar variables.
- Obtenidas por heurísticas rápidas y eficientes.

Algoritmo BC: Fijado de variables

A medida que el conjunto de soluciones es dividido por las estrategias de ramificación, algunas variables asumen un valor fijo.

Ejemplo: Restriccion original

$$x_1 + x_2 + x_3 \leq 1$$

Ramificación por

$$x_3 = 1$$

implica

$$x_1 = x_2 = 0.$$

Algoritmo BC: IPC / Skip Factor

¿Cuándo y cuántos cortes violados agregar a la formulación actual?

¿Cuándo y cuáles cortes borrar de la formulación actual?

- La generación de cortes toma tiempo (aun si no se ha tenido éxito).
- Los cortes incrementan el tamaño de la formulación (y entonces se incrementa el tiempo consumido en resolver las relajaciones).
- No generar cortes en todos los nodos del árbol de enumeración.
- Limitar el número de cortes generados por round de generación de cortes.
- Borrar cortes inactivos.

Algoritmo BC: IPC / Skip Factor

No generar cortes en todos los nodos:

- Sólo en el nodo raíz (cut-and-branch)
- Sólo en los primeros k niveles del árbol
- Sólo en los primeros k nodos evaluados (estrategia mejor cota)
- Cada k nodos evaluados (skip factor)

Borrar cortes inactivos:
Si el corte ha resultado holgado por k iteraciones consecutivas, entonces se borra de la formulación actual y se almacena en un pool de cortes.

Objetivo

Desarrollo de algoritmos tipo *Branch-and-Cut* para la resolución de los problemas abordados.

- modelar el problema mediante formulaciones de programación lineal entera
- evaluar la calidad de la cota inferior de la relajación lineal de estos modelos
- realizar un estudio poliedral de la formulación, derivando facetas y desigualdades válidas para el poliedro asociado
- utilizar estas desigualdades para el desarrollo de un algoritmo de planos de corte
- incorporar el procedimiento de planos de corte a un algoritmo *Branch-and-Cut*

Desarrollo de algoritmo *Branch & Cut*

- Etapa inicial de preprocesamiento que reduce el número de variables del modelo, permitiendo resolver instancias de mayor tamaño.
- Heurísticas iniciales y primales para el cálculo de cotas superiores que reducen el espacio de búsqueda.
- Procedimientos de separación rápidos y eficientes para varias de las familias de desigualdades válidas obtenidas de estudios poliedrales.
- Algoritmos de planos de corte que brinden buenas cotas inferiores.
- Estrategias de selección de variable de *branching* y recorrido del árbol que guien la búsqueda.

Evaluación de un algoritmo *Branch & Cut*

- ¿Vale la pena una implementación ad-hoc?
- ¿Por qué no usar un paquete de optimización?
- ¿Es superior la eficiencia de los cortes específicos?

¿Por qué “funciona” un *Branch & Cut*?

- Buena formulación.
- Excelentes cotas inferiores dadas por la fase de planos de corte.
- Excelentes cotas superiores obtenidas mediante la heurística primal.
- Buenas estrategias de recorrido del árbol y de ramificación.
- Preprocesamiento - Fijado de variables.