Investigación Operativa

Programación Lineal Entera - Branch & Bound

Nazareno Faillace Mullen

Departamento de Matemática - FCEN - UBA

Problema Lineal Entero

- Programación Lineal:
  - Simplex $\rightarrow$ complejidad exponencial (pero *usualmente es polinomial* [1])
  - Método de elipsoides $\rightarrow$ complejidad polinomial
  - Método de punto interior $\rightarrow$ complejidad polinomial

  $\Rightarrow$ resolver Problemas Lineales es "fácil".

- Programación Lineal Entera:
  - NP-Completo $\Rightarrow$ resolver problemas de Programación Lineal Entera es difícil.

Idea: resolver Programación Lineal Entera usando Programación Lineal.

[1] Spielman & Teng, *Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time*, 2001, https://arxiv.org/abs/cs/0111050

Problema Lineal Entero

El objetivo de la clase de hoy será resolver el problema:

$$\begin{aligned} \text{máx} \quad & c^t x \\ \text{s.a:} \quad & Ax \leq b \\ & x \in \mathbb{Z}^n_{\geq 0} \end{aligned}$$

$$\begin{aligned} \text{máx} \quad & c^t x \\ \text{s.a:} \quad & Ax \leq b \\ & x \in \mathbb{Z}^n_{\geq 0} \end{aligned}$$

Llamamos $\mathcal{S} = \{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\}$ la región de soluciones factibles.

Dos ideas para entender Branch & Bound:

1. Para todo $\mathcal{T}$ tal que $\mathcal{S} \subseteq \mathcal{T}$ vale: $z^*_{\mathcal{S}} \leq z^*_{\mathcal{T}}$
2. $z^*_{\mathcal{S}} = z^*_{conv(\mathcal{S})}$

Obs: hallar $z^*_{conv(\mathcal{S})}$ es un problema de Programación Lineal

Idea de la demo del punto 2:

Llamemos $\mathcal{S} = \left\{Ax \leq b, x \in \mathbb{Z}^n_{\geq 0}\right\}$. Supongamos por simplicidad que $\mathcal{S} \neq \varnothing$ y que además es acotado (y por ende tiene solo finitos puntos). Nosotros buscamos el valor de $z_{\mathsf{IP}} = \max\{c^t x : x \in \mathcal{S}\}$.

Como son solo finitos puntos, además de realizarse en algún punto que llamaremos x^*, tenemos que $c^t x^* = z_{\mathsf{IP}} < \infty$

Idea de la demo del punto 2:

Llamemos $\mathcal{S} = \left\{Ax \leq b, x \in \mathbb{Z}^n_{\geq 0}\right\}$. Supongamos por simplicidad que $\mathcal{S} \neq \varnothing$ y que además es acotado (y por ende tiene solo finitos puntos). Nosotros buscamos el valor de $z_{\text{IP}} = \text{máx}\left\{c^t x : x \in \mathcal{S}\right\}$.

Como son solo finitos puntos, además de realizarse en algún punto que llamaremos x^*, tenemos que $c^t x^* = z_{\text{IP}} < \infty$

Dada una combinación convexa de puntos de $\mathcal{S}$, el valor de la función objetivo está acotado por z_{IP}, pues si $\sum_{i=1}^{k} \lambda_i = 1$ (con $\lambda_i \geq 0 \quad \forall i$), entonces

$$c^t \left(\sum_{i=1}^{k} \lambda_i x_i\right) = \sum_{i=1}^{k} \lambda_i \left(c^t x_i\right) \leq \sum_{i=1}^{k} \lambda_i z_{\text{IP}} = z_{\text{IP}} = c^t x^*.$$

Idea de la demo del punto 2:

Llamemos $\mathcal{S} = \left\{Ax \leq b, x \in \mathbb{Z}^n_{\geq 0}\right\}$. Supongamos por simplicidad que $\mathcal{S} \neq \varnothing$ y que además es acotado (y por ende tiene solo finitos puntos). Nosotros buscamos el valor de $z_{\mathsf{IP}} = \text{máx}\{c^t x : x \in \mathcal{S}\}$.

Como son solo finitos puntos, además de realizarse en algún punto que llamaremos x^*, tenemos que $c^t x^* = z_{\mathsf{IP}} < \infty$

Dada una combinación convexa de puntos de $\mathcal{S}$, el valor de la función objetivo está acotado por z_{IP}, pues si $\sum\limits_{i=1}^{k} \lambda_i = 1$ (con $\lambda_i \geq 0 \quad \forall i$), entonces

$$c^t \left(\sum_{i=1}^{k} \lambda_i x_i\right) = \sum_{i=1}^{k} \lambda_i \left(c^t x_i\right) \leq \sum_{i=1}^{k} \lambda_i z_{\mathsf{IP}} = z_{\mathsf{IP}} = c^t x^*.$$

Concluimos que $z_{\mathsf{IP}} = \text{máx}\{c^t x : x \in \mathcal{S}\} = \text{máx}\{c^t x : x \in \text{conv}(\mathcal{S})\}$, pues $x^* \in \mathcal{S} \subseteq \text{conv}(\mathcal{S})$. La diferencia está en que este último es un problema lineal (y la dificultad radica en describir conv($\mathcal{S}$), la cápsula convexa de $\mathcal{S}$)

Cápsula convexa - Ejemplo 2D

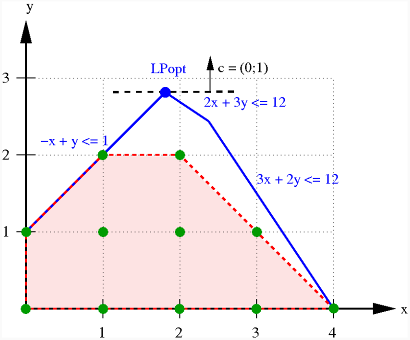
$$\begin{array}{lrcl} \text{máx} & y & & \\ \text{s.a:} & -x+y & \leq & 1 \\ & 3x+2y & \leq & 12 \\ & 2x+3y & \leq & 12 \\ & x,y \in \mathbb{Z}_{\geq 0} & & \end{array}$$

$$\begin{array}{lrcl} \text{máx} & y & & \\ \text{s.a:} & -x+y & \leq & 1 \\ & 3x+2y & \leq & 12 \\ & 2x+3y & \leq & 12 \\ & x,y \in \mathbb{R}_{\geq 0} & & \end{array}$$

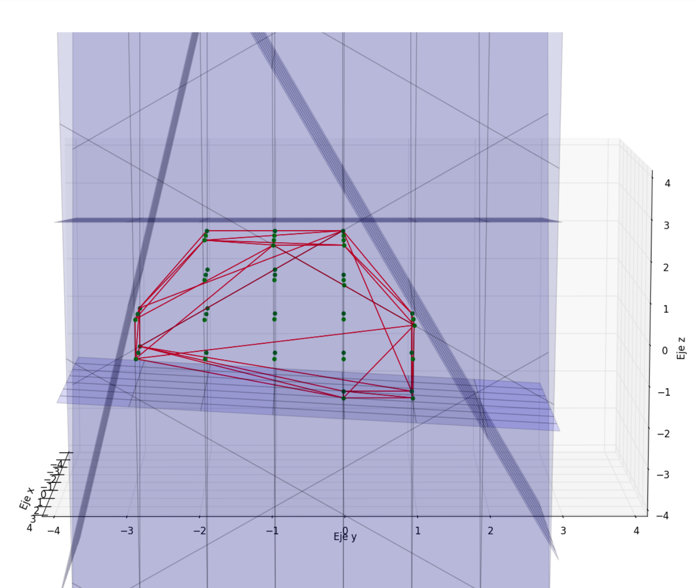
Cápsula convexa - Ejemplo 2D

$$\begin{array}{llcl} \text{máx} & y & & \\ \text{s.a:} & -x+y & \leq & 1 \\ & 3x+2y & \leq & 12 \\ & 2x+3y & \leq & 12 \\ & x,y \in \mathbb{Z}_{\geq 0} & & \end{array}$$

$$\begin{array}{llcl} \text{máx} & y & & \\ \text{s.a:} & -x+y & \leq & 1 \\ & 3x+2y & \leq & 12 \\ & 2x+3y & \leq & 12 \\ & x,y \in \mathbb{R}_{\geq 0} & & \end{array}$$



Cápsula convexa - Ejemplo 3D (gráfico)



Relajación lineal de un problema entero

Dado un problema de PLE (Programación Lineal Entera) de la forma:

$$
\text{(IP)} \qquad \begin{array}{ll} \max & c^t x \\ \text{s.a:} & Ax \leq b \\ & x \in \mathbb{Z}^n_{\geq 0} \end{array}
$$

Dado un problema de PLE (Programación Lineal Entera) de la forma:

$$(\text{IP}) \qquad \begin{array}{ll} \text{máx} & c^t x \\ \text{s.a:} & Ax \leq b \\ & x \in \mathbb{Z}^n_{\geq 0} \end{array}$$

Definimos su *relajación lineal* al problema que se obtiene al relajar la condición de integralidad de las variables en cuestión:

$$(\text{LP}) \qquad \begin{array}{ll} \text{máx} & c^t x \\ \text{s.a:} & Ax \leq b \\ & x \in \mathbb{R}^n_{\geq 0} \end{array}$$

Relajación lineal de un problema entero

Dado un problema de PLE (Programación Lineal Entera) de la forma:

$$(\text{IP}) \qquad \begin{array}{ll} \max & c^t x \\ \text{s.a:} & Ax \leq b \\ & x \in \mathbb{Z}^n_{\geq 0} \end{array}$$

Definimos su *relajación lineal* al problema que se obtiene al relajar la condición de integralidad de las variables en cuestión:

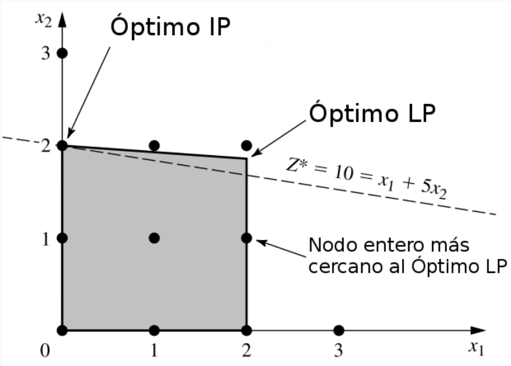
$$(\text{LP}) \qquad \begin{array}{ll} \max & c^t x \\ \text{s.a:} & Ax \leq b \\ & x \in \mathbb{R}^n_{\geq 0} \end{array}$$

Observación: Usando lo visto en las clases anteriores, tenemos las herramientas para encontrar el óptimo de la relajación lineal.

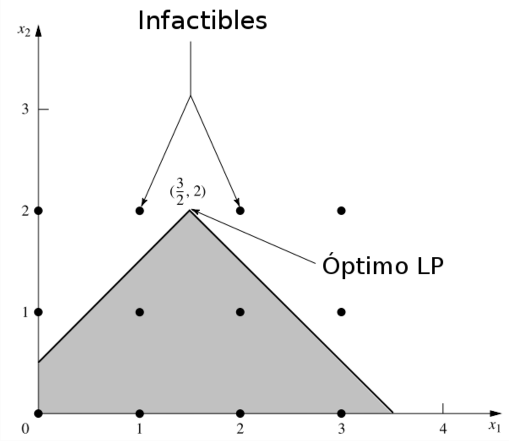
Redondear soluciones de la relajación lineal

No siempre redondear la solución de la relajación lineal es adecuado:

Podría no ser óptima



Podría no ser factible



Branch & Bound

La idea central del algoritmo, y que será uno de los criterios de corte a tener en cuenta, radica en la siguiente observación.

Branch & Bound

La idea central del algoritmo, y que será uno de los criterios de corte a tener en cuenta, radica en la siguiente observación.

Si la solución de la relajación lineal de un problema tiene todas sus coordenadas enteras, entonces también es solución al problema entero

Branch & Bound

La idea central del algoritmo, y que será uno de los criterios de corte a tener en cuenta, radica en la siguiente observación.

Si la solución de la relajación lineal de un problema tiene todas sus coordenadas enteras, entonces también es solución al problema entero

Recordemos que estamos maximizando. Si el óptimo de la relajación (que tiene coordenadas enteras) ocurre en x, tenemos que:

$$z_{\text{IP}} \underset{(1)}{\leq} z_{\text{LP}} = c^t x \underset{(2)}{\leq} z_{\text{IP}} \Rightarrow z_{\text{IP}} = z_{\text{LP}}$$

Branch & Bound

La idea central del algoritmo, y que será uno de los criterios de corte a tener en cuenta, radica en la siguiente observación.

Si la solución de la relajación lineal de un problema tiene todas sus coordenadas enteras, entonces también es solución al problema entero

Recordemos que estamos maximizando. Si el óptimo de la relajación (que tiene coordenadas enteras) ocurre en x, tenemos que:

$$z_{\text{IP}} \underset{(1)}{\leq} z_{\text{LP}} = c^t x \underset{(2)}{\leq} z_{\text{IP}} \Rightarrow z_{\text{IP}} = z_{\text{LP}}$$

(1) pues $\{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n_{\geq 0} : Ax \leq b\}$

(2) como x tiene todas sus coordenadas enteras: $x \in \{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\}$

La idea central del algoritmo, y que será uno de los criterios de corte a tener en cuenta, radica en la siguiente observación.

Si la solución de la relajación lineal de un problema tiene todas sus coordenadas enteras, entonces también es solución al problema entero

Recordemos que estamos maximizando. Si el óptimo de la relajación (que tiene coordenadas enteras) ocurre en x, tenemos que:

$$z_{\mathrm{IP}} \underset{(1)}{\leq} z_{\mathrm{LP}} = c^t x \underset{(2)}{\leq} z_{\mathrm{IP}} \Rightarrow z_{\mathrm{IP}} = z_{\mathrm{LP}}$$

(1) pues $\{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n_{\geq 0} : Ax \leq b\}$
(2) como x tiene todas sus coordenadas enteras: $x \in \{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\}$

Normalmente la solución de la relajación lineal de nuestro problema no tiene por qué dar un punto con todas sus coordenadas enteras. Si esto ocurre, vamos a particionar el espacio de búsqueda en otros más pequeños donde buscamos soluciones de la misma forma (¡sin dejar de lado vértices enteros al particionar!).

Ejemplo

$$
\begin{array}{lccc}
\text{máx} & 5x_1 + 7x_2 & & \\
\text{s.a:} & x_1 + x_2 & \leq & 6 \\
 & 5x_1 + 9x_2 & \leq & 45 \\
 & x_1, x_2 \in \mathbb{Z}_{\geq 0} & &
\end{array}
$$

Vamos a resolverlo eligiendo a x_2 como la primera variable a ramificar.

Ejercicio: resolverlo eligiendo a x_1 como la primera variable a ramificar.

Observemos qué información podemos deducir de un problema entero en función del resultado obtenido al resolver su relajación lineal.

Branch & Bound - Criterios de Poda

Observemos qué información podemos deducir de un problema entero en función del resultado obtenido al resolver su relajación lineal.

1. Si la *relajación lineal* es **infactible**, entonces nuestro problema en cuestión también es infactible.

Branch & Bound - Criterios de Poda

Observemos qué información podemos deducir de un problema entero en función del resultado obtenido al resolver su relajación lineal.

1. Si la *relajación lineal* es **infactible**, entonces nuestro problema en cuestión también es infactible.
2. Si la *relajación lineal* tiene un **óptimo con todas sus coordenadas enteras**, entonces el problema entero en cuestión está resuelto y el óptimo coincide.

Branch & Bound - Criterios de Poda

Observemos qué información podemos deducir de un problema entero en función del resultado obtenido al resolver su relajación lineal.

1. Si la *relajación lineal* es **infactible**, entonces nuestro problema en cuestión también es infactible.
2. Si la *relajación lineal* tiene un **óptimo con todas sus coordenadas enteras**, entonces el problema entero en cuestión está resuelto y el óptimo coincide.
3. **Toda solución entera** tendrá un **valor menor** al de la *relajación lineal*.

Branch & Bound - Criterios de Poda

A lo largo del algoritmo vamos a mantener una colección de problemas enteros a resolver, y el valor de la mejor solución entera obtenida hasta el momento (***valor incumbente***).

Branch & Bound - Criterios de Poda

A lo largo del algoritmo vamos a mantener una colección de problemas enteros a resolver, y el valor de la mejor solución entera obtenida hasta el momento (***valor incumbente***).

Esto nos da otra observación a tener en cuenta durante el algoritmo.

4. Si el valor **óptimo** de la *relajación lineal* es **menor al *valor incumbente***, entonces no hace falta terminar de resolver el problema entero en cuestión.

Branch & Bound - Algoritmo

Juntando todas estas ideas, podemos formular un algoritmo para resolver un problema lineal entero, para el cual introducimos la siguiente notación.

Branch & Bound - Algoritmo

Juntando todas estas ideas, podemos formular un algoritmo para resolver un problema lineal entero, para el cual introducimos la siguiente notación.

- z^i_{IP} al *valor incumbente* de la mejor solución encontrada
- $\mathcal{L}$ a la lista de problemas que quedan por resolver
- $\mathcal{S}^i$ será la i-ésima región a considerar y $\bar{z}^i_{\mathrm{R}}$ su *cota incumbente*

Branch & Bound - Algoritmo

Juntando todas estas ideas, podemos formular un algoritmo para resolver un problema lineal entero, para el cual introducimos la siguiente notación.

- z^i_{IP} al *valor incumbente* de la mejor solución encontrada
- $\mathcal{L}$ a la lista de problemas que quedan por resolver
- $\mathcal{S}^i$ será la i-ésima región a considerar y $\bar{z}^i_{\text{R}}$ su *cota incumbente*

1. $\mathcal{L} = \{\mathcal{S}\}, z^0_{\text{IP}} = -\infty$
2. Mientras $\mathcal{L}$ no esté vacío: (Si $\mathcal{L} = \varnothing$ terminamos)
   2.1 Sacar algún $\mathcal{S}^i \in \mathcal{L}$. Resolver su relajación con óptimo z^i_R en x^i_R (si la relajación es infactible no hace falta hacer nada).
   2.2 Si $z^i_R > z^i_{\text{IP}}$ y x^i_R entera: Actualizar la solución y el *valor incumbente*. Sacar de $\mathcal{L}$ todos los problemas $\mathcal{S}^j$ con $\bar{z}^j_{\text{R}} \leq z^i_{\text{R}} = z^{i+1}_{\text{IP}}$
   2.3 Si $z^i_R > z^i_{\text{IP}}$ y x^i_R no es entera : Particionar $\mathcal{S}^i$ en $\{\mathcal{S}^i_j\}^k_{j=1}$, y agregar a $\mathcal{S}^i_j$ a $\mathcal{L}$ con $\bar{z_R}^i_j = z^i_R$

Branch & Bound - Algoritmo

Juntando todas estas ideas, podemos formular un algoritmo para resolver un problema lineal entero, para el cual introducimos la siguiente notación.

- z^i_{IP} al *valor incumbente* de la mejor solución encontrada
- $\mathcal{L}$ a la lista de problemas que quedan por resolver
- $\mathcal{S}^i$ será la i-ésima región a considerar y $\bar{z}^i_{\mathsf{R}}$ su *cota incumbente*

1. $\mathcal{L} = \{\mathcal{S}\}$, $z^0_{\mathsf{IP}} = -\infty$
2. Mientras $\mathcal{L}$ no esté vacío: (Si $\mathcal{L} = \varnothing$ terminamos)
   2.1 Sacar algún $\mathcal{S}^i \in \mathcal{L}$. Resolver su relajación con óptimo z^i_R en x^i_R (si la relajación es infactible no hace falta hacer nada).
   2.2 Si $z^i_R > z^i_{\mathsf{IP}}$ y x^i_R entera: Actualizar la solución y el *valor incumbente*. Sacar de $\mathcal{L}$ todos los problemas $\mathcal{S}^j$ con $\bar{z}^j_{\mathsf{R}} \leq z^i_{\mathsf{R}} = z^{i+1}_{\mathsf{IP}}$
   2.3 Si $z^i_R > z^i_{\mathsf{IP}}$ y x^i_R no es entera : Particionar $\mathcal{S}^i$ en $\{\mathcal{S}^i_j\}^k_{j=1}$, y agregar a $\mathcal{S}^i_j$ a $\mathcal{L}$ con $\bar{z_R}^i_j = z^i_R$

Branch & Bound - Particionar Problemas

Dado un problema, la forma que utilizaremos para obtener un subproblema a partir del mismo será agregando una restricción.

En el $i-$ésimo paso, si tenemos $x_R^i \notin \mathcal{S}^i$, entonces tiene alguna coordenada que no es entera. Sea j tal que $x_{R_j^i} \notin \mathbb{Z}$, particionamos $\mathcal{S}^i$ en dos subproblemas $\mathcal{S}_1^i$ y $\mathcal{S}_2^i$ agregando las restricciones $x_j \leq \lfloor x_{R_j^i} \rfloor$ en $\mathcal{S}_1^i$ y $x_j \geq \lceil x_{R_j^i} \rceil$ en $\mathcal{S}_2^i$.

Branch & Bound - Particionar Problemas

Dado un problema, la forma que utilizaremos para obtener un subproblema a partir del mismo será agregando una restricción.

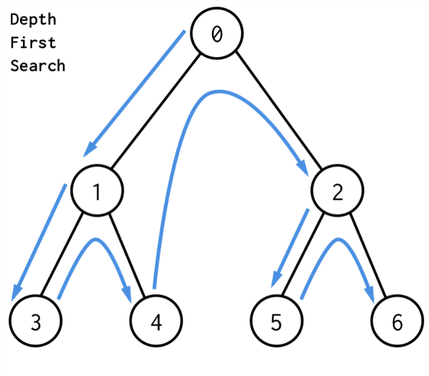
En el $i-$ésimo paso, si tenemos $x_R^i \notin \mathcal{S}^i$, entonces tiene alguna coordenada que no es entera. Sea j tal que $x_{R_j^i} \notin \mathbb{Z}$, particionamos $\mathcal{S}^i$ en dos subproblemas $\mathcal{S}_1^i$ y $\mathcal{S}_2^i$ agregando las restricciones $x_j \leq \lfloor x_{R_j^i} \rfloor$ en $\mathcal{S}_1^i$ y $x_j \geq \lceil x_{R_j^i} \rceil$ en $\mathcal{S}_2^i$.

Branch & Bound - Elección del próximo problema

La más utilizada es la *búsqueda en profundidad*, donde el próximo problema a considerar es un subproblema del último problema visto. En caso de terminar de considerar un problema, volvemos al último que vimos que todavía tenga subproblemas por considerar.

Branch & Bound - Elección del próximo problema

La más utilizada es la *búsqueda en profundidad*, donde el próximo problema a considerar es un subproblema del último problema visto. En caso de terminar de considerar un problema, volvemos al último que vimos que todavía tenga subproblemas por considerar.



Otro ejemplo

$$
\begin{array}{rrl}
\text{máx} & 2x_1 + 3x_2 & \\
\text{s.a:} & -\frac{2}{3}x_1 + x_2 & \leq \frac{5}{2} \\
 & \frac{1}{3}x_1 + x_2 & \leq \frac{9}{2} \\
 & 2x_1 + x_2 & \leq 14 \\
 & x_1, x_2 \in \mathbb{Z}_{\geq 0} &
\end{array}
$$

Branch & Bound - Particionar Problemas

Imaginemos ahora que en el problema en cuestión nuestras variables son binarias y tenemos una restricción de la siguiente forma (se las conoce como GUB, de *Generalized Upper Bound constraint* y surgen naturalmente al modelar problemas):

$$\sum_{j \in \mathcal{Q}} x_j = 1$$

Al resolver la relajación lineal, obtenemos que $0 < x_k < 1$ con $k \in \mathcal{Q}$, ¿Parece sensato separar en los problemas con $x_k = 1$ y $x_k = 0$?

Branch & Bound - Particionar Problemas

Imaginemos ahora que en el problema en cuestión nuestras variables son binarias y tenemos una restricción de la siguiente forma (se las conoce como GUB, de *Generalized Upper Bound constraint* y surgen naturalmente al modelar problemas):

$$\sum_{j \in \mathcal{Q}} x_j = 1$$

Al resolver la relajación lineal, obtenemos que $0 < x_k < 1$ con $k \in \mathcal{Q}$, ¿Parece sensato separar en los problemas con $x_k = 1$ y $x_k = 0$?

Sin ninguna información extra, al considerar el subproblema con $x_k = 0$ tenemos un problema muy similar al que teníamos originalmente. En general conviene dividir a la región factible en subproblemas de tamaño similar, por ejemplo, podemos tomar $\mathcal{Q} = \mathcal{Q}_1 \overset{d}{\cup} \mathcal{Q}_2$ y $\#\mathcal{Q}_1 \sim \#\mathcal{Q}_2$ y separar en subproblemas que agreguen la restricción

$$\sum_{j \in \mathcal{Q}_i} x_j = 0$$