

Resumen - Programacion Lineal Entera

Profesores: Gonzalo Muñoz y Daniel Rossi

Auxiliares: Felipe Fierro, Felipe Hueitra, Anais Muñoz, Leonardo Navarro, Jimmy Pirul

Programación Lineal Entera (PLE)

En programación lineal entera (PLE), buscamos soluciones donde algunas o todas las variables deben tomar valores enteros. Esto difiere de la programación lineal (PL), donde se permite que las variables sean reales.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

1. Diferencias entre PL y PLE

Programación Lineal (PL):

- Se puede resolver con el método Simplex.
- El conjunto factible es convexo (un politopo).
- La solución se encuentra en un vértice del politopo.

Programación Lineal Entera (PLE):

- No se puede usar Simplex directamente.
- El conjunto factible es un subconjunto discreto (no convexo).
- La solución óptima no necesariamente está en un vértice del politopo relajado.

2. Relajación lineal

La **relajación lineal** de un problema entero se obtiene eliminando la restricción de enteridad:

$$x \in \mathbb{Z}^n \quad \longrightarrow \quad x \geq 0$$

Esto permite resolver el problema como un PL clásico.

Proposición: Si la solución óptima de la relajación lineal es entera, entonces también es solución óptima del problema entero. *Advertencia: esto no siempre ocurre.*

- **Nota:** Cuando se trabaja con variables binarias $x \in \{0, 1\}$, la relajación lineal correspondiente consiste en permitir que $x \in [0, 1]$.

3. Cotas y utilidad de la relajación lineal

Sea S_P el conjunto factible del problema entero y S_{PR} el de su relajación. Como:

$$S_P \subseteq S_{PR}$$

entonces la relajación lineal considera más soluciones posibles. En consecuencia, su valor óptimo es al menos tan bueno como el del problema entero. En problemas de **maximización** se cumple:

$$z_I \leq z^* \leq z_{RL}$$

donde:

- z_I : valor de una solución entera factible conocida.
- z^* : valor óptimo del problema entero.
- z_{RL} : valor óptimo de la relajación lineal.

Esta desigualdad define un **gap**, que indica cuán lejos está la mejor solución entera conocida del valor relajado.

Conclusión: La relajación lineal:

- Entrega una **cota superior** (en maximización) o inferior (en minimización) para el problema entero.
- Permite estimar, acotar y evaluar soluciones enteras.
- Puede coincidir con el óptimo entero si la solución resulta entera.

Branch and Bound

Branch and Bound (BnB) es un método exacto para resolver problemas de programación entera. Se basa en tres ideas principales: **relajación lineal**, **ramificación** y **acotación**.

1. Idea general

Dado un problema entero:

$$\begin{array}{ll} \min & c^T x \\ \text{s.a.} & Ax \leq b \\ & x \in \mathbb{Z}^n \end{array}$$

se comienza resolviendo su **relajación lineal** (sin la condición de enteridad). Si la solución óptima de la relajación es entera, hemos terminado. En caso contrario, se procede a dividir el problema en subproblemas, forzando a que al menos una variable fraccionaria tome un valor entero. A este proceso se le llama **ramificación**.

2. Ramificación

Si la solución relajada óptima es $\hat{x}$ y una de sus componentes es fraccionaria (por ejemplo, $\hat{x}_i = 1.3$), se crean dos subproblemas:

$$\text{(izquierda)} \quad x_i \leq \lfloor \hat{x}_i \rfloor \quad \text{(derecha)} \quad x_i \geq \lceil \hat{x}_i \rceil$$

Esto garantiza que se excluye la solución no-entera $\hat{x}$, pero se mantiene todo el espacio de soluciones enteras. Cada uno de estos subproblemas se resuelve nuevamente relajando la condición de enteridad.

3. Acotación y poda

Cada subproblema entrega una **cota inferior** (en minimización) o **cota superior** (en maximización) para su rama. Estas cotas se obtienen resolviendo la relajación lineal correspondiente.

Se pueden descartar ramas completas si ocurre alguna de estas condiciones:

- La relajación es **infactible**.
- La solución de la relajación es **entera** (posible candidata óptima).
- La cota de la rama es peor que la mejor solución entera encontrada hasta ahora (se realiza una **poda por acotación**).

4. Árbol de exploración

El proceso se visualiza como un árbol, donde:

- El nodo raíz corresponde al problema original relajado.
- Cada nodo hijo representa una versión restringida del problema (con una variable acotada).
- Se continúa expandiendo nodos hasta encontrar soluciones enteras viables, o hasta que todos los nodos hayan sido podados.

5. Gap y criterio de parada

En todo momento se mantiene:

$$\text{Mejor solución entera conocida} \leq \text{Valor óptimo del problema entero} \leq \text{Mejor cota relajada activa}$$

Esta brecha se denomina **gap**, y se puede usar como criterio de parada si es menor que una tolerancia dada.

Resumen de Branch and Bound:

1. Resolver la relajación lineal del problema.
2. Si la solución es entera, termina. Si no, seleccionar una variable fraccionaria y ramificar.
3. Resolver las relajaciones de los subproblemas.
4. Actualizar cotas y la mejor solución entera conocida.
5. Poda si corresponde. Repetir hasta cerrar todas las ramas o alcanzar el criterio de parada.