Métodos de Programación Lineal Entera

Alexander Olza Rodriguez

Diciembre de 2020

1. Introducción

Dada la dificultad de la programación lineal entera, a lo largo de los años se han propuesto diversos algoritmos para problemas enteros mixtos (MIP). Los nuevos métodos se apoyan en sus antecesores, introduciendo mejoras.

La bases de todos ellos pueden clasificarse en dos tipos: Estrategias de "Divide y Vencerás", como la ramificación, y estrategias de fortalecimiento de restricciones, como los cortes. A continuación se presenta un resumen de las técnicas básicas (B&B, Cutting planes) y de las combinaciones más utilizadas (B&C). Después se introducen los conceptos de la técnica Branch and Price, y su descendiente Branch and Cut and Price.

2. Técnicas Básicas

2.1. Cutting Planes

La dificultad de la PLE está relacionada con la diferencia entre región factible del MIP y la de su problema lineal relajado (LP). Una medida de dicha dificultad es la distancia entre la solución del problema relajado y la del MIP. Por eso, las técnicas de cortes añaden nuevas restricciones que generen una región factible relajada cada vez más parecida a la región factible del MIP. El procedimiento es el siguiente. Dado un MIP y su relajación lineal LP_0, empezamos con $i = 0$:

1. Se **resuelve** el LP_i. Si la solución es entera, FIN: Es la solución óptima del MIP. Si no, paso 2.
2. Se **añade** una nueva restricción, de forma que la solución del LP_i quede fuera de la región factible pero ninguna posible solución entera quede excluída. $i+ = 1$.Paso 1.

Esta estrategia se conoce como generación dinámica de filas. Esto se debe a que las restricciones lineales de los sucesivos LP_i se escriben de forma matricial como $A_i x \leq b_i$ con $A \in M_{m,n}, \quad x \in M_{n,1}, \quad b \in M_{m,1}$ donde n es el número de variables y m el número de restricciones, que aumenta cada vez que se introduce un corte.

Existen distintos modos de generar las nuevas restricciones. Uno de ellos es el llamado Cortes de Gomory.

2.2. Branch and Bound

Esta estrategia se basa en el principio de "Divide y Vencerás". En lugar de atacar el MIP original, se relaja la integralidad y se resuelve una secuencia ordenada, en forma de árbol, de LP con regiones factibles disjuntas. Las soluciones de estos LP establecen cotas superiores UB e inferiores LB que finalmente convergen a la solución del MIP. El algoritmo para minimizar $f(x)$ bajo restricciones e integralidad es:

1. **Resuelve** LP_0. Si la solución $x_0 \in \mathbb{N}$, es el óptimo: FIN. Si no, $LB = f(x_0)$, $UB = \infty$, $i = 0$. Paso 2.
2. **Ramifica**. $x_i = a.b \rightarrow$ Divide LP_i en dos subproblemas: $LP_{i+1} = LP_i \oplus \{x \leq a\}$ y $LP_{i+2} = LP_i \oplus \{x \geq a+1\}$[1]. Añádelos a la lista de nodos activos. Paso 3.

[1] Abuso de notación $\oplus$: Mismo problema con una restricción añadida.

3. **Elige un nodo** de la lista, LP_i. Resuelve. Si es factible, su solución es x_i. Paso 4. Si no, paso 5.

4. **Actualiza cotas.**

   a) Si $x_i \in \mathbb{N}$: Si $f(x_i) < UB$: $\quad UB = x_i$ sol. incumbente. Si no, Paso 5.

   b) Si no: Si $f(x_i) \in [LB, UB]$: $\quad LB = x_i$. Paso 2.

5. **Poda.** Si $x_i \in \mathbb{N}$, no caben restricciones adicionales. Se poda la rama por integralidad. Si no, si $f(x_i) \notin [LB, UB]$, se poda por cotas. Si LP_i es infactible, se poda por factibilidad. Paso 6.

6. **Optimalidad.** Si quedan nodos en la lista, Paso 3. Si no, FIN. Si hay solución incumbente, esta es óptima. Si no, el MIP es infactible.

Hay distintos criterios para elegir nodo en el Paso 3. Una búsqueda en anchura permite tratar problemas similares, ahorrando recursos. Una búsqueda en profundidad restringe rápidamente la región factible manteniendo el árbol más pequeño, ya que explora cada rama hasta poder podarla.

3. Técnicas Compuestas

3.1. Branch and Cut

Es una combinación de las dos técnicas básicas. Sean MIP_0 y su relajación lineal LP_0:

1. **Inicializa.** Resuelve LP_0. Si es infactible, MIP_0 también, FIN. Si $x_0 \in \mathbb{N}$, es el óptimo, FIN. Si no, $LB = f(x_0)$ y $UB = \infty$, y se añade MIP_0 a la lista de nodos activos.

2. **Elige nodo.** Si hay, se elige un nodo activo i, Paso 3. Si no hay, el óptimo es $x^\star$, FIN.

3. Resuelve $LP_i(t)$. Si es infactible, podar, Paso 2. Si no, guardar sol. $x_i(t)$. Paso 4.

4. **Cortes**. Iteración $t+ = 1$. Si se puede, añadir cortes que excluyan $x_i(t)$ sin excluir ninguna sol. entera y añade el nuevo problema $LP_i(t)$ a la lista. Paso 3. Si no se puede, Paso 5.

5. **Poda.** Si $f(x_i(t)) \in [LB, UB]$: Si $x_i(t) \in \mathbb{N}$, Paso 6, y si no Paso 7. Si $f(x_i(t)) \notin [LB, UB]$, poda por cotas.

6. **Cota.** Tenemos $f(x_i(t)) \in [LB, UB]$, $x_i(t) \in \mathbb{N}$, sol. incumbente: $x^\star = x_i(t)$, $UB = f(x_i(t))$. Poda por integralidad. Paso 2.

7. **Ramifica.** Tenemos $f(x_i(t)) \in [LB, UB]$, pero $x_i(t) \notin \mathbb{N}$. Entonces, $LB = f(x_i(t))$. Ramifica, añade LP_{i+1}, LP_{i+2} a la lista. Paso 2.

3.2. Branch and Price

A continuación se describe la estrategia de generación de columnas. Partiendo del problema original P, se resuelve una secuencia de problemas restringidos, a los que se van añadiendo solamente las variables que mejoren el valor de la función objetivo. El proceso termina cuando no quedan variables ventajosas. Habitualmente, el último problema resuelto sigue teniendo menos variables que el original, y en ese caso la técnica es rentable.

El fundamento de esta técnica es el teorema de representación, que permite expresar cualquier punto de un poliedro como combinación de sus puntos extremos. Sea el MIP original, llamado P, de la forma $max \quad c(x)$ sujeto a $Ax \leq b$, $x \in S$, $x \in \mathbb{N}$, donde S es un conjunto acotado. Entonces, la región factible será el conjunto finito de puntos enteros en S, es decir, $S^\star = \{y_1, ..., y_p\}$. Por el citado teorema, $\forall y \in S^\star$, $y = \sum_{k=1}^{p} y_k \lambda_k$ con la restricción de convexidad $\sum_{k=1}^{p} \lambda_k = 1$ y $\lambda_k \in \{0, 1\}$. Aprovechando esto, se escribe la forma de Generación de Columnas de P, llamada Master Problem (MP):

$$
\begin{array}{rl}
max & \sum_k c_k \lambda_k \\
s.t. & \sum_k A y_k \leq b \\
\sum_{k=1}^{p} \lambda_k = 1, & \lambda_k \in \{0,1\}
\end{array}
$$

Y seleccionando un subconjunto de las variables λ_k en el MP se puede construir un Restricted Master Problem, RMP, donde las λ_k no seleccionadas están fijadas a cero.

Sea (π, α) una solución óptima al dual de un RMP, donde α es la variable dual asociada a la restricción $\sum_{k=1}^{p} \lambda_k = 1$ y π es el vector de variables duales asociadas al resto de restricciones. Entonces, se puede formular el Pricing Problem (PP):

$$
\begin{array}{rl}
max & [c(x) - \pi A x] - \alpha \\
 & s.t. x \in S^{\star}
\end{array}
$$

La solución del PP permite identificar la columna con mayor coste reducido. Es decir, en caso de que se añadiera una variable al RMP, identifica aquella con el mayor aumento asociado en su función objetivo (que estamos maximizando). Si el valor objetivo del PP (el coste reducido) es positivo, la columna seleccionada mejora la solución actual del RMP, y se añade. Si no, la solución actual del RMP es el óptimo del MP completo.

Branch-and-Price combina la generación dinámica de columnas con la Ramificación. Existen muchos detalles en la implementación de este tipo de algoritmos. Por ejemplo, en lugar de optimizar directamente el PP pueden utilizarse métodos de aproximación. También pueden seleccionarse y añadirse varias columnas simultáneamente al RMP para reducir el número de iteraciones. Aún así, los pasos fundamentales son los siguientes:

1. **Inicializa**. Determina un RMP inicial con relajación lineal factible $LRMP$, Paso 2.
2. **RMP**. Resuelve el $LRMP$ actual. Paso 3.
3. **PP**. Dada la sol. del $LRMP$, resuelve el PP. Si el valor objetivo es positivo, Paso 4. Si no, el óptimo del $LRMP$ actual es también el óptimo del LMP, Paso 5.
4. **Añade** la columna identificada al $LRMP$. Paso 2.
5. **Actualiza cotas**. Tenemos una combinación de $\{\lambda_k\}$, que representan un x como posible solución incumbente de P.
   a) Si esta es factible (si $x \in \mathbb{N}$): Si es mejor que la actual, actualiza cota inferior y sol. incumbente. Paso 7 (poda por integralidad).
   b) Si no: Si procede actualiza cota superior, Paso 6. Si no, Paso 7 (poda por cotas)
6. **Ramifica** LP. Añade los subproblemas a la lista de nodos. Paso 8.
7. **Poda**. Paso 8.
8. **Elige un nodo**, Paso 1. Si no hay, FIN.

3.3. Branch and Cut and Price

Esta última variante incorpora planos de corte, ramificación y generación dinámica de columnas. La complejidad de implementación es grande, porque ha de hacerse de forma que estas técnicas no interfieran negativamente entre sí.

Primero se formula el MP y el LMP. Este se resuelve por generación de columnas, planteando sucesivos $LRMP$ y PP.

Cuando se alcanza el óptimo, la solución del LMP se retransforma al espacio de las variables originales, obteniendo así la solución del LP. Si esta no satisface la integralidad, se intenta añadir planos de corte.

Entonces se genera un nuevo LMP, que se optimiza otra vez mediante generación de columnas. Hay que asegurarse de que las nuevas variables duales, que emergen de las restricciones de corte, se consideren en los sucesivos PP. Si la solución del LMP sigue sin satisfacer la integralidad al trasladarse a la formulación original, se añaden más cortes o se ramifica.

En cada nodo del árbol se resolverá el correspondiente LMP, y si la solución no es entera se añadirán los cortes que sea posible, resolviendo los nuevos LMP antes de volver a ramificar. El esquema de los pasos es el siguiente:

1. **Inicializa**. Determina un RMP inicial con relajacion lineal factible $LRMP$, Paso 2.
2. **RMP**. Resuelve en $LRMP$ actual, Paso 3.
3. **PP**. Resuelve el PP. Si el valor objetivo es positivo, Paso 4. Si no, Paso 8.
4. **Añade** la columna identificada al $LRMP$. Paso 2.
5. **Cortes**. Si se puede, añade cortes que excluyan la solución fraccionaria actual del LMP. Esto formula un nuevo LMP. Paso 1. Si no, Paso 6.
6. **Ramifica** LP. Añade los dos nuevos MP resultantes a la lista de nodos. Paso 7.
7. **Elige un nodo**. Paso 1. Si no hay, FIN.
8. **Actualiza cotas**.
   a) Si la solución del $LRMP$ se traslada a una solución factible de P: Si es mejor que la actual, actualiza cota inferior y sol. incumbente. En cualquier caso, Paso 9 (poda por integralidad).
   b) Si no: Si procede, actualiza cota superior. Paso 5. Si no, Paso 9 (poda por cotas).
9. **Poda**. Paso 7.

Referencias

[1] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh y P. H. Vance, "Branch-and-Price: Column Generation for Solving Huge Integer Programs," *Operations Research*, vol. 46, n.º 3, págs. 316-329, 1998, ISSN: 0030364X, 15265463. dirección: http://www.jstor.org/stable/222825.

[2] J. Desrosiers y M. E. Lübbecke, "Branch-Price-and-Cut Algorithms," en *Wiley Encyclopedia of Operations Research and Management Science*. American Cancer Society, 2011, ISBN: 9780470400531. DOI: https://doi.org/10.1002/9780470400531.eorms0118. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470400531.eorms0118. dirección: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470400531.eorms0118.

[3] "Branch-and-Cut Approach," en *Applied Integer Programming*. John Wiley Sons, Ltd, 2009, cap. 12, págs. 305-333, ISBN: 9781118166000. DOI: https://doi.org/10.1002/9781118166000.ch12. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118166000.ch12. dirección: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118166000.ch12.

[4] G. Gamrath, "Generic Branch-Cut-and-Price," Tesis de mtría., 2010.