UNIVERSIDAD DE PANAMÁ

VICERRECTORÍA DE INVESTIGACIÓN Y POSTGRADO

PROGRAMA DE MAESTRÍA EN MATEMÁTICA

MÉTODOS DE PLANO DE CORTE PARA LA SOLUCIÓN DE PROBLEMAS DE PROGRAMACIÓN ENTERA

WESLEY E. JONES A.

Tesis presentada como uno de los requisitos para optar por el título de Maestría en Matemática, opción Investigación de Operaciones

Panamá, República de Panamá

2023

Agradecimiento

A Dios, primeramente, por abrir las puertas, una y otra vez, que parecían estar cerradas y por poner en mi camino a las personas indicadas para seguir adelante.

A todos mis profesores, por su dedicación para dar sentido a cada uno de los cursos que estudié durante el desarrollo del programa.

Al profesor Eloy Rico quien me ayudó en las primeras etapas de la tesis.

Al Dr. José Del Rosario Garrido por su apoyo y por facilitar algunos de los libros de la bibliografía y atender oportunamente a mis consultas.

A la Dra. Manuela Foster por su persistencia en enseñarme a fijarme en los detalles.

A la Dra. Magaly de Chial, quien me motivó a seguir adelante en el momento más difícil.

A mi familia por su apoyo incondicional en las etapas más difíciles durante el desarrollo del programa de maestría.

A mis compañeros, por apoyarme y alentarme en los momentos en los que necesité ayuda.

Dedicatoria

Al Dr. José Del Rosario Garrido, por motivarme, apoyarme y guiar mis pasos durante todo el proceso de formación en el posgrado desde el primer día.

Contenido

Introducción

En la Investigación de Operaciones se resuelven problemas relacionados con la toma de decisiones para mejorar el funcionamiento de un sistema. Uno de los tipos de problemas más estudiados en este campo, es el que requiere determinar un conjunto de valores que maximicen o minimicen una función lineal llamada función objetivo sujeta a un sistema de restricciones lineales.

A menudo, en estos problemas existen situaciones en las que las variables de decisión solo admiten valores enteros. Este tipo de problemas surge prácticamente en todas las áreas de aplicación de la programación matemática. Cuando en un problema lineal algunas o todas las variables de decisión tienen que ser enteras se le clasifica como **problema de programación entera**.

Existe un gran número de problemas prácticos que se pueden modelar como problemas enteros, por ejemplo: el problema de la mochila, el problema del viajero, la planificación de la producción, los problemas de secuenciación, programación y enrutamiento, el diseño de redes de telecomunicaciones, la planificación de horarios de clases, etc.

De hecho, según Kaufmann & Henry-Labordere (1977), "para efectos prácticos, los problemas de programación lineal con soluciones enteras tienen una importancia aún mayor que los problemas clásicos de programación lineal" (p.1).

Desde su introducción por George Danzig en 1947, el método simplex ha demostrado ser un algoritmo poderoso y eficiente para resolver problemas lineales. No obstante, no siempre se puede usar para resolver problemas enteros. Por eso han surgido diferentes técnicas para resolver este tipo de problemas.

Uno de los métodos utilizados para resolver problemas de programación entera se conoce como planos de corte, introducido por R. E. Gomory en 1958. Desde entonces, los planos de corte han sido estudiados en profundidad y aplicados en diversas situaciones.

Este trabajo explorará tres algoritmos de corte propuestos por Gomory:

- el algoritmo fraccionario dual para problemas enteros,
- el algoritmo fraccionario dual para problemas mixtos, y
- el algoritmo dual completamente entero.

En el primer capítulo se dará una introducción a la programación lineal entera, definiendo sus características básicas y algunos conceptos acerca de las principales técnicas de solución de problemas enteros.

En el Capítulo 2 se desarrollan los Métodos de Plano de Corte, incluyendo un breve antecedente histórico, una definición de plano de corte y una explicación del método simplex dual lexicográfico que es fundamental para la demostración de la convergencia de estos métodos. Para cada uno de los tres algoritmos, se detallará el procedimiento, un ejemplo y la demostración de convergencia.

En el Capítulo 3 se compararán los tres métodos destacando sus similitudes, diferencias y limitaciones.

En el Capítulo 4 se formulará y resolverá un problema práctico de programación entera utilizando uno de los métodos explicados.

Finalmente, se incluyen conclusiones y recomendaciones de temas para profundizar en estudios posteriores.

Capítulo 1

Generalidades sobre programación entera

1.1 Clasificación de los problemas enteros

Uno de los problemas más estudiados en la investigación de operaciones son los problemas de programación lineal. Estos son problemas que buscan la maximización o minimización de una función objetivo lineal sujeta a restricciones lineales. Si le agregamos la restricción que algunas o todas las variables tienen que ser enteras, entonces tenemos un problema de programación entera.

Muchos problemas prácticos tienen que ser modelados como enteros debido a que en el mundo real los elementos involucrados son indivisibles. Por ejemplo, no tiene sentido asignar 10.7 personas, o construir 1.3 aeronaves, ni adjudicar 3.4 proyectos.

Cuando todas las variables de decisión tienen que ser enteras, se trata de un problema de programación **entera**. Cuando solo algunas están restringidas a valores enteros, el problema es de programación **mixta**. Cuando las variables solo pueden adoptar los valores 0 o 1, se le llama de programación **booleana**.

1.2 Algunas técnicas de solución de problemas enteros

En 1947, el matemático norteamericano George B. Dantzig desarrolló el método simplex que se constituyó en un algoritmo muy eficaz para resolver problemas de programación lineal, manteniéndose en la actualidad como el método más extendido para resolver esta clase de problemas, aun los que tienen gran cantidad de variables de decisión y restricciones.

A primera vista, pareciera que un problema de programación entera pudiera ser resuelto con el método simplex y luego redondear el resultado al entero más cercano. Pero esto puede no ser una solución, porque el resultado redondeado puede no ser factible o no ser óptimo. Por lo que el método de redondeo solo puede ser considerado cuando se sabe que las variables de decisión tienen que tomar valores muy grandes, de forma que el redondeo producirá un error muy pequeño (Malón, 2020, pág. 2).

Ciertos problemas de programación entera se pueden resolver como lineales usando el **método simplex**. Un ejemplo de ello son los problemas cuya matriz de restricciones es unimodular. Se dice que una matriz es unimodular si la determinante de cada sub-matriz cuadrada es 0, 1 o -1. Las soluciones a tales problemas se pueden obtener omitiendo las restricciones de enteros y resolviendo el problema con el método simplex[1]. Los problemas de transporte, asignación y flujo de red mínimo pertenecen a esta clase de problemas enteros que se pueden resolver utilizando el método simplex.

A veces los problemas enteros se pueden resolver con **técnicas de enumeración**. La intención de estos métodos es enumerar exhaustivamente todos los posibles candidatos a la solución del problema de programación entera y escoger el candidato factible que optimiza la función objetivo. Su desventaja principal estriba en que su complejidad aumenta rápidamente con el número de variables y restricciones. Por ejemplo, en un problema sencillo de asignación de tres candidatos a tres posiciones, las posibles combinaciones son $3! = 6$. Examinar esta pequeña cantidad de posibilidades no es un reto. No obstante, resolver el mismo problema para 100 candidatos a sendas posiciones, requeriría examinar $100! = 9.3 \times 10^{157}$ opciones. Explorar

[1] Una demostración puede hallarse en Mathur & Salkin, (1989), p. 74.

todas y cada una de las opciones está completamente fuera del alcance aun de las supercomputadoras más potentes.

1.3 Las principales técnicas de solución de problemas enteros

La mayoría de los problemas de programación entera no se pueden resolver usando sólo el método simplex. En esos casos hay que recurrir a otras técnicas. Los enfoques principales son: las técnicas de ramificación y acotación y las técnicas de plano de corte.

El primer paso en ambas técnicas consiste en abordar el problema como si fuera un problema de programación lineal usando el método simplex sin la restricción de que las variables sean enteras. El resultado de este problema lineal asociado se le llama solución al **problema relajado.**

1.3.1 Los métodos de ramificación y acotación ("Branch and Bound")

Esta técnica se basa en subdividir sistemáticamente la región factible del problema relajado y hacer evaluaciones del problema entero con base en estas subdivisiones. La ventaja de estos métodos es que son fáciles de entender y programar en una computadora. Además, se pueden utilizar tanto para problemas mixtos como enteros. Su desventaja principal estriba en que su complejidad aumenta rápidamente según el número de variables y restricciones. Por eso, no son muy eficientes para problemas grandes y complejos (Kaufmann & Henry-Labordere, 1977, p. 75).

1.3.2 Los métodos de plano de corte

La idea básica de los métodos de plano de corte es deducir desigualdades suplementarias, llamados "cortes", que, al ser añadidos a las restricciones existentes, reducen la región factible del problema relajado, de modo que la solución entera óptima se convierta en un punto extremo y, por lo tanto, se pueda encontrar mediante el poderoso y eficiente método simplex. A diferencia de los métodos de ramificación y acotación, los métodos de plano de corte no subdividen la región factible, sino que trabajan con un solo problema lineal, que se va refinando con cada corte. Su desventaja principal radica en que el número de cortes, aunque finito, puede ser grande, lo que limita la rapidez de su convergencia.

1.3.3 Los métodos de ramificación y corte ("Branch and Cut")

A mediados de la década de 1990, Gérard Cornuéjols y sus colegas demostraron que los cortes de Gomory eran muy efectivos en combinación con el método de branch-and-bound (Cornuéjols, 2005). Esta combinación recibe el nombre de ramificación y corte. Hoy día, todos los solucionadores de problemas enteros comerciales utilizan cortes de Gomory de una forma u otra. (Wikipedia)

Capítulo 2

Algoritmos de Plano de Corte

2.1 Conceptos Generales

Los algoritmos de plano de corte son herramientas para resolver problemas enteros. El primero en proponer esta técnica fue el matemático norteamericano Ralph E. Gomory (Gomory, Outline of an Algorithm for Integer Solutions to Linear Programs. Bulletin of the American Mathematical Society, 64(5), 275–278., 1958).

Cuando él presentó el algoritmo dual fraccionario, el impacto fue enorme e inmediato, porque demostró que se podía reducir la programación entera a una secuencia de soluciones de problemas lineales.

También demostró que el algoritmo convergía. Es relativamente simple crear cortes que remueven una porción del área factible sin remover puntos enteros. Lo difícil es probar que un número finito de tales cortes producirán un problema lineal cuya solución óptima es un número entero. El corte de Gomory fue el primer algoritmo desarrollado para la programación entera que podía demostrarse que convergía en un número finito de pasos.

Poco después, Gomory extendió el procedimiento para resolver problemas mixtos (Gomory, An Algorithm for the Mixed Integer Problem. Rand Report, RM-2597, 1960). También creó otro algoritmo para resolver el problema del redondeo (Gomory, All-integer integer programming algorithm, 1960).

Desde entonces los cortes de Gomory han jugado un papel fundamental en la teoría de la programación entera.

2.1.1 El problema lineal asociado

El incremento en la complejidad de los métodos de solución a los problemas enteros comparados con el método simplex, se debe al hecho que la solución óptima ya no se encuentra en un punto extremo, lo que era la clave de la eficiencia del método simplex. Es por ello que los métodos de corte comienzan con la relajación de las condiciones de integridad y el uso del algoritmo simplex. Al resultado de este problema lineal asociado se le llama **solución óptima relajada**.

La solución óptima relajada nos da información importante para el problema entero:

- Si no es factible, entonces el problema entero tampoco lo es.
- Si es entera, entonces el problema entero está resuelto.
- Si no es entera, su valor de Z representa la cota al valor de Z del problema entero.

2.1.2 Definición de plano de corte

Un plano de corte es un hiperplano que separa la solución óptima del problema relajado del conjunto de soluciones factibles enteras. Es decir, una vez aplicado el corte, la solución óptima no entera obtenida del problema lineal asociado queda excluida de la región factible. Esto, en efecto, reduce la región factible, pero sin excluir ninguna solución entera factible.

En todos los métodos de corte se producen nuevas restricciones llamados "cortes" que se añaden a las originales para eventualmente producir un problema lineal cuya solución óptima entera corresponde a la del problema entero.

Considere, por ejemplo, el siguiente problema:

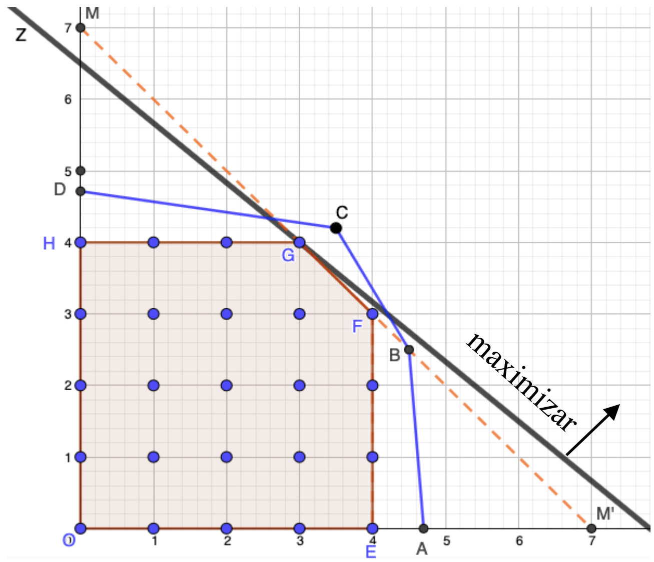
Maximizar $\boldsymbol{5x_1 + 6x_2}$

$$x_1 + 7x_2 \leq 33$$

$10x_1 + 6x_2 \leq 60$

$50x_1 + 4x_2 \leq 235$

$x_1, x_2 \geq 0$ y x_1 y x_2 entero



Las restricciones definen la región convexa OABCD. El punto C (3.5, 4.2) es la solución óptima del problema lineal relajado.

Los puntos mostrados dentro de OABCD representan puntos con componentes enteros y, por lo tanto, representan soluciones factibles al problema entero. Pero, puesto que las soluciones óptimas de un problema lineal siempre están en los bordes o extremos del espacio de solución, se observa claramente que ninguno de los vértices tiene componentes enteros y, por lo tanto, no son soluciones factibles del problema entero.

Supóngase ahora que la región factible del problema lineal se pudiese reducir a un casco convexo de sus puntos reticulares dentro de la región factible, como se muestra en el área sombreada OEFGH, con cortes como la recta MM'. Se puede considerar esta región sombreada como la región factible de otro problema lineal. De hecho, esta región corresponde al problema original con las siguientes restricciones adicionales:

$x_1 + x_2 \leq 7$ (recta MM')

$x_1 \leq 4$

$x_2 \leq 4$

Esta nueva región factible tiene tres características importantes: Primero, excluye la solución óptima relajada. Segundo, contiene todos los puntos factibles del problema entero original. Tercero: todos los puntos extremos de esta nueva región factible tienen componentes enteros.

Por lo tanto, la solución óptima del problema lineal modificado tiene componentes enteros y es también la solución óptima del problema entero original. En este caso, la solución, el punto G, con coordenadas (3,4), corresponde tanto al problema entero original como al problema lineal modificado.

Esta es la esencia de los métodos de plano de corte: sistemáticamente crear restricciones adicionales a la región factible y tratar de resolver el problema usando el método simplex hasta que la solución sea entera.

El algoritmo tiene las siguientes propiedades:

- Los cortes generados nunca excluirán de la nueva región factible un punto entero factible del problema entero original.
- Cada nuevo corte reducirá la región factible del problema lineal original.
- Se generarán las restricciones que lleven a la solución en un número finito de pasos.

2.1.3 Notación

Considere el problema entero con *n* variables y *m* restricciones:

$$\max\ Z = cx$$

$$s.a.\ Ax = b$$

$$x \in \mathbb{Z}_+^n$$

donde $A \in \mathrm{M}^{n\times m}$, con valores enteros, $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m$.

Dentro de la literatura de planos de corte, los elementos del conjunto de las restricciones y de la función objetivo suelen presentarse según la siguiente tabla, que es una transposición de la que se suele emplear en los algoritmos simplex primal y dual:

Variables	Constantes	Variables No-Básicas		
	1	$-x_1$	$\cdots$	$-x_n$
$x_0 =$	a_{00}	a_{01}	$\cdots$	a_{0n}
$x_1 =$	0	–1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_i =$	0	0	–1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_n =$	0	0	0	–1
$x_{n+1} =$	$a_{n+1,0}$	$a_{n+1,1}$	$\cdots$	$a_{n+1,n}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$x_{n+m} =$	$a_{n+m,0}$	$a_{n+m,1}$	$\cdots$	$a_{n+m,n}$

La fila 0 corresponde a la función objetivo con $x_0 = z$ y $c_j = a_{0j}$, $j = 1, \dots, n$

$$x_0 = a_{00} + \sum_{j=1}^{n} c_j(-x_j)$$

Las filas 1 hasta n corresponden a variables de decisión, con las ecuaciones triviales

$$x_j = -(-x_j)\ (j = 1, \dots, n).$$

Las filas n+1 hasta n+m representa las restricciones, donde x_{n+i} representa las variables de holgura y $a_{n+i,0} = b_i$.

$$x_{n+i} = a_{\mathrm{n+i,0}} + \sum_{J=1}^{n} a_{n+i,j}(-x_j)\ (\ i = 1\ \dots m)$$

Al introducir las ecuaciones triviales $x_j = -(-x_j)$ $(j = 1, \dots, n)$ como parte de las restricciones se puede usar notación matricial:

$$X = A\ (-x_n)$$

donde

$X = (x_j)$ es un vector columna de n+m+1 componentes (j = 0, 1, 2, …, n+m),

$A = (a_{ij})$ es una matriz $(n + m + 1)$ por $(n + 1)$, donde a_{ij} corresponde al coeficiente de la i-ésima fila y la j-ésima columna (i = 0, 1, 2, …, n+m), (j = 0, 1, 2, …, n+m).

Se usará $\propto_j$ $(j = 0,1, \ldots, n)$ para indicar el vector de la j-ésima columna.

$(-x_n)$ es un vector fila con sus componentes $(1, -x_1, -x_2, \ldots -x_n)$, siendo $x_1, x_2, \ldots x_n$ las variables no-básicas de la tabla inicial, y $x_{n+1}, \ldots, x_{n+m}$ las variables de holgura.

Según Hu (1970), "las razones para expresar cada variable en términos de $(-x_1), (-x_2), \ldots, (-x_n)$ son puramente históricas, pero ha llegado a ser la práctica estándar en programación entera" (p. 231).

Con cada iteración del método simplex, se escoge un nuevo conjunto de variables no básicas y la tabla cambia. Se usará el índice *t* para indicar que el valor corresponde a la t-ésima tabla simplex. Así, t = 0 indica que no se ha realizado ninguna iteración del algoritmo y, que la tabla corresponde a la tabla inicial del problema. La ecuación de la matriz entonces puede denotarse así:

$$X^t = A^t\,(-x_n^t) \qquad (2.1\text{-}1)$$

2.1.4 El método simplex dual lexicográfico

Los algoritmos de corte que se describirán utilizan el método simplex dual. Para garantizar que el algoritmo converja en un número finito de pasos, se requiere usar reglas para evitar que el método repita la selección de un conjunto de variables básicas ya utilizado en una iteración anterior. De allí el concepto de reglas lexicográficas.

Un vector es **lexicográficamente positivo** si su primer componente distinto de cero es positivo. Por ejemplo, el vector (0, 0, 3, –6, 5) es lexicográficamente positivo, mientras que el vector (0, –2, 8, 5, 9) es lexicográficamente negativo. Un vector **x** que es lexicográficamente positivo se denotará $x \succ^{L} 0$. De igual modo, un vector x_1 es **lexicográficamente menor** que x_2 (denotado $x_1 \prec^{L} x_2$) si el primer término distinto de cero de $x_1 - x_2$ es negativo. Por ejemplo, (0, –2, 8, 5, 9) es lexicográficamente menor que (0, 0, 3, –6, 5) porque su diferencia (0, –2, 5, 11, 4) es lexicográficamente negativo.

Cuando el algoritmo simplex dual incorpora reglas lexicográficas para evitar ciclos se le conoce como el Método Simplex Dual Lexicográfico (Mathur & Salkin, 1989, p. 60). Este difiere del algoritmo simplex dual habitual en que se establece una manera especial de determinar las variables salientes y entrantes. En el método simplex dual lexicográfico, se comienza y mantienen tablas simplex que son dualmente factibles y tienen todas las columnas no básicas lexicográficamente positivas. Esto se logra mediante el uso de una regla llamada de pivote lexicográfico, que explicaremos más adelante. Este enfoque asegura que el método simplex no cicle y, además, garantiza que la solución óptima del problema lineal, si existe, se encuentre en un número finito de pasos.

Selección del pivote en el algoritmo simplex dual lexicográfico:

Paso 1: Seleccione la fila pivote *k* de la variable que sale de la base usando el criterio

$$\min \ \{a_{i0}\} \text{ para toda } a_{i0} < 0. \tag{2.1-2}$$

Paso 2: Seleccione la columna pivote *p* de la variable que entra a la base usando el criterio

$$\frac{\propto_p}{|a_{kp}|} \prec^{L} \frac{\propto_j}{|a_{kj}|} \text{ para toda j tal que } a_{kp} < 0 \tag{2.1-3}$$

Paso 3: Actualice la tabla realizando las siguientes operaciones por columna:

$$\text{Para la columna 0:} \quad \propto_0^{t+1} = \propto_0^{t} - \frac{a_{k0}}{a_{kp}} \propto_p^{t}. \tag{2.1-4}$$

Para la columna pivote: $\propto_p^{t+1} = \frac{-1}{a_{kp}} \propto_p^t.$ (2.1-5)

Para las demás columnas: $\propto_j^{t+1} = \propto_j^t - \frac{a_{kj}}{a_{kp}} \propto_p^t.$ (2.1-6)

Estas operaciones transforman el elemento pivote en –1 y a todos los otros elementos de la fila pivote en 0.

Diferencias entre el algoritmo simplex primal y el simplex dual:

- En el algoritmo simplex primal, primero se selecciona la columna correspondiente a la variable que entra a la base y luego la fila correspondiente a la variable que sale para así determinar el pivote. En cambio, en el algoritmo simplex dual, primero se selecciona la fila de la variable que sale y luego la columna de la variable que entra. En el método simplex dual *lexicográfico* se agregan reglas lexicográficas en la selección de la columna pivote.
- En el algoritmo simplex primal, las operaciones se hacen usando la fila pivote, mientras que en el algoritmo simplex dual se usa la columna pivote.

2.2 El algoritmo dual fraccionario para problemas enteros

El algoritmo dual fraccionario, también llamado método de plano de corte Gomory es una técnica de corte desarrollada en 1958 por R. E. Gomory inicialmente para resolver problemas enteros. Luego, en 1960, él mismo lo extendió para resolver problemas mixtos.

2.2.1 Procedimiento

Paso 1: Resolver el problema lineal asociado ignorando la restricción que las variables tienen que ser enteras, usando el método simplex primal o simplex dual. Si la solución óptima es entera, el problema entero queda resuelto.

Paso 2: Añadir el "corte de Gomory" como una restricción adicional al final de la tabla. En la sección 2.2.2 se explica cómo generar el corte.

Paso 3: Re-optimizar la tabla aplicando el método simplex dual.

Paso 4: Repetir los pasos 2 y 3 hasta que todos los valores de la columna a_{i0} sean enteros.

2.2.2 Generación del corte

Selección de la fila generadora del corte

El método simplex produce un conjunto de ecuaciones de la forma (omitiendo el sub-índice de fila)

$$x = a_0 + \sum a_j \left(-x_j\right). \tag{2.2-1}$$

Donde x_j son las variables no-básicas actuales.

Aunque se puede usar cualquiera, existen dos criterios que, por lo general, se utilizan para escoger la fila generadora:

i) Escoger la fila que tenga el valor de a_0 con la parte fraccionaria más grande, o

ii) Escoger la primera fila que tenga un a_0 no entero.

Por lo general, la literatura recomienda usar el primer criterio porque tiende a producir cortes más efectivos, pero en la demostración de la convergencia del algoritmo (sección 2.2.4), se usará el segundo.

Generación del corte de Gomory

Sea $[a]$ el mayor entero $\leq a$ y sea f la parte fraccionaria:

$$f = a - [a]. \tag{2.2-2}$$

Al escribir la ecuación 2.2-1, usando $f_j = a_j - [a_j]$, $f_0 = a_0 - [a_0]$, se tiene:

$$x = [a_0] + f_0 + \sum ([a_j] + f_j)(-x_j) \tag{2.2-3}$$

Se reescribe la ecuación de modo que las partes enteras estén en el lado izquierdo y las fraccionarias en el lado derecho:

$$x - \sum [a_j](-x_j) - [a_0] = f_0 + \sum f_j(-x_j). \tag{2.2-4}$$

El lado derecho de esta ecuación es menor que 1, para cualquier punto entero en la región factible. Dado que el problema restringe las variables a números enteros, se cumple la desigualdad

$$f_0 + \sum f_j (-x_j) \leq 0, \tag{2.2-5}$$

la cual puede re-escribirse como

$$\sum f_i (-x_j) \leq -f_0 \tag{2.2-6}$$

Esta desigualdad excluye la solución factible básica y, por lo tanto, es un corte con las propiedades deseadas. Al introducir una nueva variable de holgura x_k, se agrega una nueva restricción al problema lineal, que constituye el **corte de Gomory**, a veces llamado corte fraccionario porque todos sus coeficientes son fracciones:

$$x_k = -f_0 - \sum f_i (-x_j) \tag{2.2-7}$$

Selección del pivote

El pivote se determina usando el método dual lexicográfico, según lo explicado en la sección 2.1.4. La fila pivote es precisamente el corte de Gomory (ecuación (2.2-7)), porque es la única fila con $a_0 < 0$. La columna pivote se determina aplicando la ecuación (2.1-3).

2.2.3 Ejemplo.

Se considera el siguiente problema de programación entera, que se resolverá atendiendo al procedimiento descrito:

Maximizar $\mathbf{5x_1 + 6x_2}$

sujeto a $\mathbf{10x_1 + 3x_2 \leq 52}$

$\mathbf{2x_1 + 3x_2 \leq 18}$

$\mathbf{x_1, x_2 \geq 0}$ y entero

Paso 1: Resolver el problema lineal asociado, usando el método simplex.

Las celdas resaltadas representan el pivote.

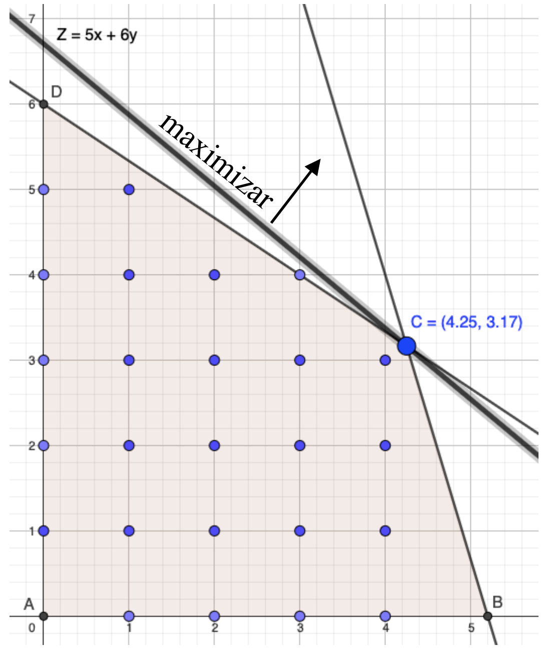
t_0	1	$-x_1$	$-x_2$
x_0	0	-5	-6
x_1	0	-1	0
x_2	0	0	-1
x_3	52	10	3
x_4	18	2	3

t_1	1	$-x_1$	$-x_4$
x_0	36	-1	2
x_1	0	-1	0
x_2	6	2/3	1/3
x_3	34	8	-1
x_4	0	0	-1

t_2	1	$-x_3$	$-x_4$
x_0	161/4	1/8	15/8
x_1	17/4	1/8	-1/8
x_2	19/6	-1/12	5/12

$\boldsymbol{x_3}$	0	-1	0
$\boldsymbol{x_4}$	0	0	-1

La solución óptima relajada es $\boldsymbol{Z = 40.25}$, con $\boldsymbol{x_1}$ = 4.25, $\boldsymbol{x_2}$ = 3.17, que corresponde al punto C en la figura.



Paso 2: Como la solución no es entera, se aplica un corte de Gomory. Arbitrariamente se escoge la ecuación correspondiente a x_1 como la fila generadora.

La ecuación correspondiente a x_1 es:

$$x_1 = \frac{17}{4} + \frac{1}{8}(-x_3) - \frac{1}{8}(-x_4)$$

Por la ecuación (2.2-2):

$$f_0 = \frac{17}{4} - \left[\frac{17}{4}\right] = 4\frac{1}{4} - 4 = \frac{1}{4}$$

$$f_3 = \frac{1}{8} - \left[\frac{1}{8}\right] = \frac{1}{8} - 0 = \frac{1}{8}$$

$$f_4 = -\frac{1}{8} - \left[-\frac{1}{8}\right] = -\frac{1}{8} - [-1] = \frac{7}{8}$$

Sustituyendo en la ecuación (2.2-7) se tiene el primer corte de Gomory:

$$\boldsymbol{x_5 = -\frac{1}{4} - \frac{1}{8}(-x_3) - \frac{7}{8}(-x_4)} \qquad (2.2\text{-}8)$$

Esta restricción se añade al final de la tabla $\boldsymbol{t_2}$, dando origen a la tabla 3:

$\boldsymbol{t_3}$	1	$\boldsymbol{-x_3}$	$\boldsymbol{-x_4}$	
$\boldsymbol{x_0}$	161/4	1/8	15/8	
$\boldsymbol{x_1}$	17/4	1/8	–1/8	←fila generadora

x_2	19/6	–1/12	5/12
x_3	0	–1	0
x_4	0	0	–1
x_5	$-1/4$	–1/8	–7/8

Paso 3: Luego de añadir esta restricción se continua con el método simplex dual lexicográfico explicado en la sección 2.1.4:

Según la ecuación (2.1-2) se escoge x_5 como la variable que sale de la base porque corresponde a la única fila con $a_{i0}^3 < 0$.

Según la ecuación (2.1-3) se escoge $x_3$como la variable que entra porque

$$\frac{\frac{1}{8}}{\left|-\frac{1}{8}\right|} \prec^L \frac{\frac{15}{8}}{\left|-\frac{7}{8}\right|}$$

La tabla se actualiza aplicando las ecuaciones (2.1-4), (2.1-5) y (2.1-6) y, para las columnas 0, 1 y 2 de la tabla 4, respectivamente:

$$\propto_0^4 = \propto_0^3 - 2 \propto_1^3$$

$$\propto_1^4 = 8 \propto_1^3$$

$$\propto_2^4 = \propto_2^3 - 7 \propto_1^3$$

Lo que da como resultado la siguiente tabla:

t_4	1	$-x_5$	$-x_4$
x_0	40	1	1
x_1	4	1	–1
x_2	10/3	–2/3	1
x_3	2	–8	7
x_4	0	0	–1
x_5	0	–1	0

Aquí se alcanza otra vez un valor óptimo con $Z = 40$ y $x_1 = 4$, $x_2 = 3.33$ que corresponde al punto E en la siguiente figura.

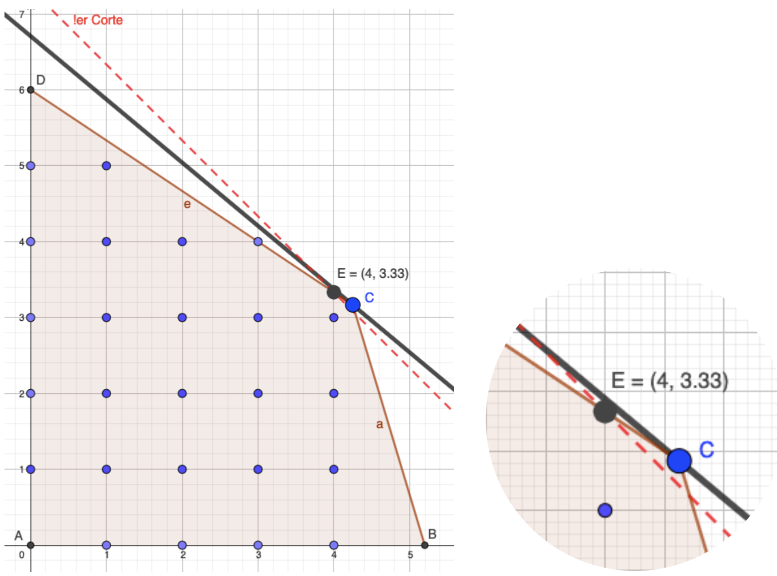
Para graficar el primer corte de Gomory, ecuación (2.2-8), lo expresamos en términos de las variables no básicas iniciales, $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$:

$$\boldsymbol{x_5 = -\frac{1}{4} + \frac{1}{8}x_3 + \frac{7}{8}\,x_4} \tag{2.2-8}$$

Reemplazando $\boldsymbol{x_3}$ y $\boldsymbol{x_4}$ de las ecuaciones originales y simplificando se tiene

$$\boldsymbol{3x_1 + 3x_2 + x_5 \ = 22} \tag{2.2-9}$$

Esta ecuación se muestra en la gráfica como una línea punteada. Aunque la línea excluye la solución óptima relajada (punto C) de la región factible, no se excluye ninguna solución entera dentro de la región de factibilidad ABCD del problema entero original.



La solución óptima con este primer corte corresponde al punto E (4, 3.33). Como la solución aún no es completamente entera, se repiten los pasos 2 y 3, y se procede a aplicar un segundo corte, usando la ecuación correspondiente a x_2 como generadora, porque corresponde a la única fila con a_{i0}^{4} no entero.

$\boldsymbol{t_4}$	1	$\boldsymbol{-x_5}$	$\boldsymbol{-x_4}$
$\boldsymbol{x_0}$	40	1	1
$\boldsymbol{x_1}$	4	1	–1

x_2	10/3	–2/3	1	← fila generadora
x_3	2	–8	7	
x_4	0	0	–1	
x_5	0	–1	0	

$$x_2 = \frac{10}{3} + (-x_4) - \frac{2}{3}(-x_5)$$

Por la ecuación (2.2-2):

$$f_0 = \frac{10}{3} - \left\lfloor \frac{10}{3} \right\rfloor = \frac{10}{3} - 3 = \frac{1}{3}$$

$$f_4 = 1 - [1] = 1 - 1 = 0$$

$$f_5 = -\frac{2}{3} - \left\lfloor -\frac{2}{3} \right\rfloor = -\frac{2}{3} - [-\mathbf{1}] = -\frac{2}{3} + 1 = \frac{1}{3}$$

Reemplazando en la ecuación (2.2-7), se obtiene el segundo corte de Gomory:

$$\boldsymbol{x_6} = -\frac{\mathbf{1}}{\mathbf{3}} - \frac{\mathbf{1}}{\mathbf{3}}(-\boldsymbol{x_5}) \tag{2.2-10}$$

Esta restricción se añade al final de la tabla $\boldsymbol{t_4}$, dando origen a la tabla 5.

t_5	1	$-x_5$	$-x_4$
x_0	40	1	1
x_1	4	1	–1
x_2	10/3	–2/3	1
x_3	2	–8	7
x_4	0	0	–1
x_5	0	–1	0
x_6	$-1/3$	$-1/3$	0

Luego de añadir la restricción, se continua con el método simplex dual lexicográfico. Se aplica las ecuaciones (2.1-2) y (2.1-3) para escoger $\boldsymbol{x_6}$ como la variable que sale de la base y $\boldsymbol{x_5}$ como la que entra. La tabla se actualiza aplicando las ecuaciones (2.1-4), (2.1-5) y (2.1-6) para las columnas 0, 1 y 2 de la tabla 5, respectivamente:

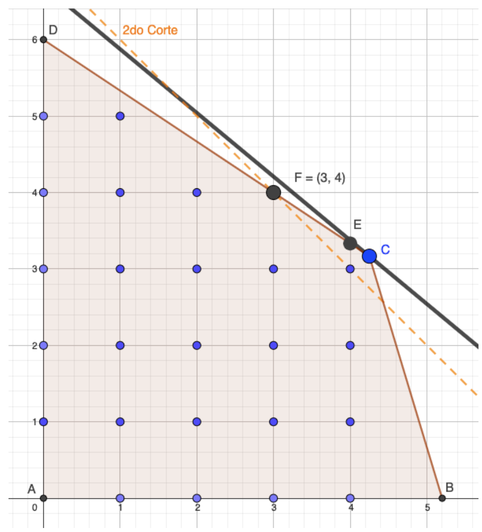
$$\propto_0^6 = \propto_0^5 - \propto_1^5$$

$$\propto_1^6 = 3 \propto_1^5$$

$$\propto_2^6 = \propto_2^5$$

Esto da como resultado la última tabla:

t_6	1	$-x_6$	$-x_4$
x_0	39	3	1
x_1	3	3	-1
x_2	4	-2	1
x_3	10	-24	7
x_4	0	0	-1
x_5	1	-3	0
x_6	0	-1	0



La solución final, entonces es: $\boldsymbol{Z = 39}$ y $\boldsymbol{x_1} =$ 3, $\boldsymbol{x_2} = 4$, que corresponde al punto F señalado en la figura.

Al expresar el segundo Corte de Gomory, ecuación (2.2-10), en términos de las variables $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$ se tiene:

$$x_1 + x_2 + x_6 = 7 \tag{2.2-11}$$

Esta ecuación se muestra como una línea de puntos en la figura. Se aprecia como la ecuación de corte excluye de la solución al punto óptimo anterior hallado (punto E) sin eliminar ninguno de los puntos factibles del problema original.

2.2.4 Convergencia del algoritmo

La demostración de la convergencia del algoritmo comienza asumiendo que se conoce algún límite inferior M para el valor de Z.

Supóngase también que, en congruencia con el método simplex dual lexicográfico, se usará la primera fila con una constante no entera como la fila generadora para el corte de Gomory. Eso incluye a la fila cero.

Ahora supongamos que el algoritmo NO es finito. Entonces, debido a que se está usando el método simplex dual lexicográfico, existe una secuencia infinita de tablas, cuya columna 0 decrece lexicográficamente. Es decir, $\boldsymbol{\propto_0^t \succ^L \propto_0^{t+1} \succ^L \propto_0^{t+2} \ldots}$

Dado que $\propto_0$ decrece lexicográficamente, su primer componente a_{00} debe ser no creciente y eventualmente permanecer fijo en un valor entero. De otro modo, la fila 0 genera el corte Gomory, que se utiliza como fila pivote.

$$x^t = -f_{00} + \sum_{j=1}^{n} \left(-f_{0j}\right)(-x_j) \geq 0 \qquad (2.2\text{-}12)$$

Si p es el índice de la columna de la fila pivote, entonces, según el método simplex dual, ecuación (2.1-4), tenemos:

$$a_{00}^{t+1} = a_{00}^{t} - \left(\frac{a_{0p}}{f_{0p}}\right) f_{00} \qquad (2.2\text{-}13)$$

El método dual simplex garantiza que $a_{0p} \geq 0$. Además, como f_{0p}, la parte fraccionaria de a_{0p}, es diferente de 0, se puede concluir que $a_{0p} > 0$.

Por lo tanto,

$$a_{0p} = [a_{0p}] + f_{0p} > f_{0p} \qquad (2.2\text{-}14)$$

$$a_{0p} > f_{0p}$$

y

$$\frac{a_{0p}}{f_{0p}} \geq 1$$

Por lo tanto,

$$a_{00}^{t+1} \leq \ a_{00}^{t} \ - \ f_{00}$$

Por la ecuación (2.2-2) se sabe que $a_{00}^{t} \ - \ f_{00} = [{a_{00}}^{t}]$. Y, por lo tanto,

$$a_{00}^{t+1} \leq [{a_{00}}^{t}] \qquad (2.2\text{-}15)$$

Esto significa que cada vez que la fila 0 se usa como la fila fuente para el corte de Gomory el valor de a_{00} decrecerá por lo menos hasta el siguiente entero. Si a_{00} decrece por una cantidad entera, entonces, después de un número finito de iteraciones quedará por debajo de M (la cota inferior asumida). Por lo tanto, si el algoritmo no es finito, entonces a_{00} deberá permanecer fijo en algún valor entero para todo $t \ > t_0$. Asumamos que este sea el caso.

Se fija la atención ahora en a_{10}. Usando el mismo argumento, a_{10} no puede permanecer fijo en un valor no entero pues esta fila ahora se convierte en la fila fuente para el corte Gomory, dado que a_{00} es ahora un entero. Usando el mismo argumento anterior, a_{10}^{t+1} decrecerá por lo menos al siguiente entero. Por lo tanto, a_{10} permanecerá fijo en un número o se volverá negativo después de un número finito de iteraciones.

Pero si a_{1p} es negativo eso implica que a_{0p} es positivo; lo cual hará que a_{00} tenga que decrecer, lo cual contradice el supuesto de que se mantiene fijo. Y si $a_{1j} \geq 0$ para todo $j = 1, \dots n,$ esto implica que el programa no es factible.

Por lo tanto, la única posibilidad es que a_{10} permanezca fijo después de un número finito de iteraciones.

Se puede repetir el mismo argumento para el segundo, tercer, hasta el componente n+m. Después de un número finito de iteraciones, sus valores permanecerán fijos en un número entero.

2.3 El algoritmo dual fraccionario para problemas mixtos

El algoritmo para problemas mixtos es esencialmente igual que para problemas enteros. La única diferencia radica en la manera de generar los cortes de Gomory.

2.3.1 Procedimiento

Paso 1: Al igual que con los problemas enteros, el método se inicia resolviendo el problema lineal asociado. Si la solución óptima es entera para las variables restringidas a ser enteras, el problema mixto queda resuelto.

Paso 2: Añadir el "corte de Gomory" como una restricción adicional al final de la tabla. En la sección 2.3.2 se explica cómo generar el corte para problemas mixtos.

Paso 3: Aplicar el método simplex dual para re-optimizar la tabla.

Paso 4: Repita los pasos 2 y 3 hasta que todos los valores de la columna a_{i0} para las x_i que tienen que ser enteras, efectivamente lo sean.

2.3.2 Generación del corte de Gomory para problemas mixtos

El uso del método simplex en el Paso 1, produce un conjunto de ecuaciones de la forma (omitiendo los subíndices de fila):

$$x = a_0 + \sum a_j \, (-x_j), \tag{2.3-1}$$

donde x_j son las variables no-básicas actuales. Seleccione una ecuación con un a_0 no entero, para un x_i restringido a entero. Esta es la fila generadora.

Dado que x tiene que ser entero, $x \equiv 0$ (mod 1). Según la ecuación (2.2-2), sea $[a_0]$ el mayor entero $\leq a_0$ y sea la parte fraccionaria $f_0 = a_0 - [a_0]$. Dado que $a_0 > 0$ y no entero, $f_0 > 0$. Por lo tanto,

$$x \equiv a_0 + \sum a_j \left(-x_j\right) \text{ (mod 1).} \tag{2.3-2}$$

Al sumar $0 \equiv -[a_0]$ nos queda:

$$0 \equiv f_0 + \sum a_j \left(-x_j\right) \text{ (mod 1)} \tag{2.3-3}$$

Esta relación puede reescribirse como

$$\sum a_j \left(x_j\right) \equiv f_0 \text{ (mod 1)} \tag{2.3-4}$$

Toda solución entera tiene que satisfacer esta relación. Se dividen los índices de las variables no-básicas en N^+ para $a_{ij} \geq 0$; y N^- de otro modo.

$$\sum_{j \in N^+} a_j x_j + \sum_{j \in N^-} a_j x_j \equiv f_0 \text{ (mod 1)} \tag{2.3-5}$$

El lado izquierdo de esta relación es positivo o es negativo. Se exploran cada una de estas opciones:

- Si es positivo: para poder diferir de f_0 por una cantidad entera tiene que ser igual a $1 + f_0$, o $2 + f_0$, etc. Por lo tanto, se puede escribir la ecuación:

$$\sum_{j \in N^+} a_j x_j \geq \sum_{j \in N^+} a_j x_j + \sum_{j \in N^-} a_j x_j \geq f_0 \tag{2.3-6}$$

- Si es negativo: para poder diferir de f_0 por una cantidad entera tiene que ser igual a $-1 + f_0$, o $-2 + f_0$, etc. Por lo tanto, se puede escribir la ecuación:

$$\sum_{j \in N^-} a_j x_j \leq \sum_{j \in N^+} a_j x_j + \sum_{j \in N^-} a_j x_j \leq -1 + f_0 \tag{2.3-7}$$

Al multiplicar ambos lados por el número negativo $\frac{f_0}{f_0 - 1}$ nos queda

$$\frac{f_0}{f_0 - 1} \sum_{j \in N^-} a_j x_j \geq f_0 \tag{2.3-8}$$

En ambos casos, las expresiones son $\geq f_0$, por lo que es cierto que

$$\sum_{j \in N^+} a_j\, x_j \; + \frac{f_0}{f_0 - 1} \sum_{j \in N^-} a_j\, x_j \geq \; f_0 \tag{2.3-9}$$

Toda solución entera debe satisfacer esta desigualdad, pero la solución actual no la cumple, ya que al sustituir $x_j = 0$ para toda j, el lado izquierdo es cero. Nótese que al derivar esta desigualdad sólo se usó el hecho de que x tenía que ser entero y que x_j tenía que ser no negativo. Por lo tanto, si algunos x_j están restrigidos a ser enteros, esta desigualdad sigue siendo válida.

Al introducir una nueva variable de holgura x_k, se agrega una nueva restricción al problema lineal, a saber:

$$x_k = \; -f_0 + \sum_{j \in N^+} a_j\, x_j \; + \frac{f_0}{f_0 - 1} \sum_{j \in N^-} a_j\, x_j \tag{2.3-10}$$

Esta restricción se añade al final de la tabla y se continúa con el Paso 3.

Sin embargo, se puede hacer más restrictivo este corte haciendo que los coeficientes a_j sean lo más pequeños posible, aprovechando el hecho que algunos de los x_j están restringidos a ser enteros.

Debido a que la ecuación de corte se derivó de la relación $\sum a_j \left(x_j\right) \equiv f_0$ (mod 1), ecuación (2.3-4), aumentar o disminuir a_j por una cantidad entera no destruirá esta relación de congruencia. Para los valores $a_j \geq 0$ el coeficiente más pequeño que se puede obtener es f_j. Para los valores $a_j < 0$ el coeficiente más pequeño que se puede obtener es $f_j - 1$. Por lo tanto, se quiere aumentar o disminuir el valor de cada a_j con

$$min(f_j, \frac{f_0}{1 - f_0}\left(1 - f_j\right)\,) \tag{2.3-11}$$

Dado que

$$f_j \leq \frac{f_0}{1-f_0}\left(1-f_j\right) \text{ si } f_j \leq f_0 \tag{2.3-12}$$

y

$$f_j > \frac{f_0}{1-f_0}\left(1-f_j\right) \text{ si } f_j > f_0 \tag{2.3-13}$$

Entonces, la ecuación de corte queda así:

$$x_k = -f_0 - \sum_j f_j^* \,(-x_j) \tag{2.3-14}$$

donde

$$f_j^* = a_j \qquad \text{si } a_j \geq 0 \text{ y } x_j \text{ no entero,}$$

$$f_j^* = \frac{f_0}{f_0-1} a_j \qquad \text{si } a_j < 0 \text{ y } x_j \text{ no entero,}$$

$$f_j^* = f_j \qquad \text{si } f_j \leq f_0 \text{ y } x_j \text{ entero,}$$

$$f_j^* = \frac{f_0}{1-f_0}(1-f_j) \qquad \text{si } f_j > f_0 \text{ y } x_j \text{ entero.}$$

2.3.3 Ejemplo

Se usará el mismo ejemplo resuelto en la sección 2.2.3 para un problema entero, pero esta vez con la exigencia de una solución mixta:

Maximizar $\mathbf{5x_1 + 6x_2}$

sujeto a

$\mathbf{10x_1 + 3x_2 \leq 52}$

$\mathbf{2x_1 + 3x_2 \leq 18}$

$\mathbf{x_1, x_2 \geq 0}$ y $\mathbf{x_1}$ entero

Paso 1: Resolver el problema lineal asociado.

Se vio en el ejemplo para el problema entero, que la solución óptima relajada corresponde a $\boldsymbol{Z = 40.25}$, con $\boldsymbol{x_1}$ = **4.25**, $\boldsymbol{x_2}$ = **3.17**, y que la tabla simplex con la solución relajada es:

t_2	1	$-x_3$	$-x_4$	
x_0	161/4	1/8	15/8	
x_1	17/4	1/8	–1/8	←Fila generadora
x_2	19/6	–1/12	5/12	
x_3	0	–1	0	
x_4	0	0	–1	

Paso 2: Como se requiere que la variable de decisión x_1 sea entera, se agrega un corte de Gomory mixto. La ecuación generadora para el corte se origina de la ecuación para x_1:

$$x_1 = \frac{17}{4} + \frac{1}{8}(-x_3) - \frac{1}{8}(-x_4)$$

Aplicando la ecuación (2.3-14) se tiene:

$$f_0 = \frac{17}{4} - \left[\frac{17}{4}\right] = \frac{1}{4}$$

$f_3 = a_3 = \frac{1}{8}$, dado que $a_3 \geq 0$ y x_3 no tiene que ser entero

$f_4 = \frac{f_0}{f_0 - 1} a_4 = \frac{\frac{1}{4}}{\frac{1}{4} - 1}\left(-\frac{1}{8}\right) = \frac{1}{24}$, dado que $a_4 < 0$ y x_4 no tiene que ser entero

Con lo que el corte queda así:

$$x_5 = -\frac{1}{4} + \frac{1}{8}(-x_3) + \frac{1}{24}(-x_4) \qquad (2.3\text{-}15)$$

Esta restricción se añade al final de la tabla 2 lo que da lugar a la tabla 3:

t_3	1	$-x_3$	$-x_4$
x_0	161/4	1/8	15/8
x_1	17/4	1/8	–1/8
x_2	19/6	–1/12	5/12
x_3	0	–1	0
x_4	0	0	–1
x_5	–1/4	1/8	1/24

Luego de añadir esta restricción, se continua con el método simplex dual lexicográfico. Se aplica las ecuaciones (2.1-2) y (2.1-3) para escoger $\boldsymbol{x}_5$ como la variable que sale de la base y $\boldsymbol{x}_3$ como la que entra. La tabla se actualiza aplicando las ecuaciones (2.1-4), (2.1-5) y (2.1-6) para las columnas 0, 1 y 2 de la tabla 3, respectivamente:

$$\propto_0^4 = \propto_0^3 - 2 \propto_1^3$$

$$\propto_1^4 = -8 \propto_1^3$$

$$\propto_2^4 = \propto_2^3 - \frac{1}{3} \propto_1^3$$

La última tabla revisada queda así:

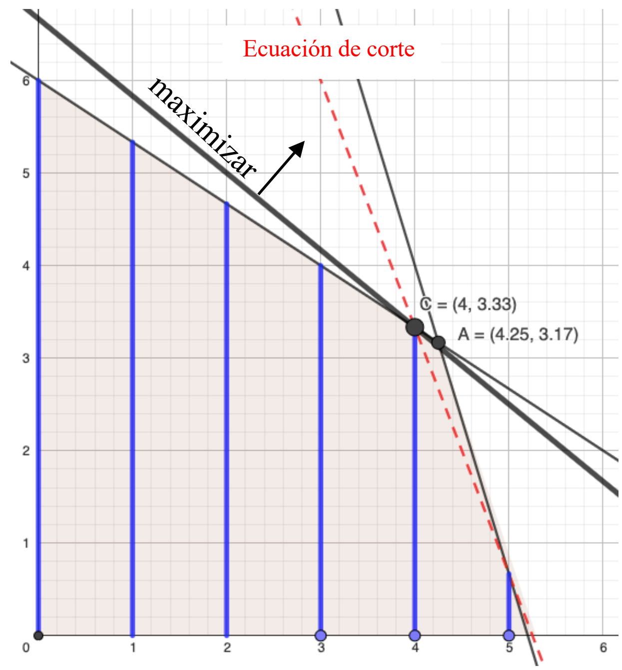
$\boldsymbol{t}_4$	1	$\boldsymbol{x}_5$	$\boldsymbol{x}_4$
$\boldsymbol{x}_0$	40	1	11/6
$\boldsymbol{x}_1$	4	1	–1/6
$\boldsymbol{x}_2$	10/3	–2/3	4/9
$\boldsymbol{x}_3$	2	–8	1/3
$\boldsymbol{x}_4$	0	0	1
$\boldsymbol{x}_5$	0	1	0

La solución final es $x_1 = 4$, $x_2 = \frac{10}{3}$ con $Z = 40$, la cual se muestra como el punto C en la figura.

Al expresar la ecuación (2.3-15) en términos de las variables no-básicas iniciales, la ecuación resultante es:

$$\frac{8}{3}x_1 + x_2 - 24x_5 = 14$$

Esta ecuación se muestra en la figura como una línea punteada. La ecuación de corte excluye la solución óptima simplex (punto A) y reduce el área de factibilidad del problema original, pero sin excluir ninguno de los puntos factibles originales (representados por las líneas verticales gruesas).



2.3.4 Convergencia del algoritmo

La demostración que este algoritmo converge es muy similar al del algoritmo dual fraccionario para problemas enteros por el hecho de que al usar el método simplex dual la columna cero decrece lexicográficamente con cada pivote[2].

2.4 El algoritmo dual completamente entero

Se describirá otro algoritmo para resolver problemas de programación entera mediante cortes de plano que también fue desarrollado por Ralph E. Gomory *(Gomory, All-integer integer programming algorithm, 1960)*. Se le llama algoritmo completamente entero porque, si se comienza con una tabla que inicialmente consiste de valores enteros, todas las tablas subsiguientes se mantendrán con valores enteros a través de todo el cálculo.

---

[2] Se puede hallar una demostración exhaustiva en Hu (1970), p. 259.

Una dificultad con los planos de corte fraccionarios es que pueden ocurrir errores de redondeo en la computadora, lo cual puede hacer que una solución entera óptima sea excluida, y que una solución subóptima se considere óptima. Por ejemplo, la solución al problema al final de una iteración podría tener valores enteros para todas las variables menos una, que tendría un valor de 7,999999. Dado que no es un entero, el algoritmo continuaría y cortaría este punto de la región factible. Si el valor se debió a un error de redondeo y en realidad debería haber sido 8, entonces cualquier solución entera subsiguiente sería subóptima.

El problema del error de redondeo fue reconocido y eliminado por Gomory cuando desarrolló este algoritmo de plano de corte entero. El método es, en realidad, una extensión del método simplex dual con la diferencia que la fila pivote se genera en cada iteración, de tal modo, que los coeficientes son todos enteros y el pivote es –1, lo cual garantiza que todas las tablas siguientes también tendrán valores enteros.

A diferencia del método fraccionario en donde se optimiza la tabla, se aplica un corte, se re-optimiza la tabla, se aplica otro corte, y así sucesivamente hasta obtener una tabla entera óptima, este algoritmo genera cortes que se aplican desde la primera iteración.

El algoritmo se inicia con una tabla dualmente factible como se da en el método simplex dual para problemas de programación lineal. Si todos los valores de la columna a_{i0} $(i = 1, 2, \dots, n+m)$ son enteros no negativos, entonces el problema está resuelto. Si una fila en particular tiene $a_{i0} < 0$, entonces se deriva una nueva ecuación que se añade al final de la tabla como una fila pivote.

Usando la notación explicada en la sección 2.1.3 y la ecuación (2.1-1), sea el problema entero:

$$x^t = A^t(-x_n^t) \qquad (2.4\text{-}1)$$

Se asume que en la tabla inicial (t = 0), todos los a_{ij} son enteros y todas las columnas $\propto_j$ $(j = 1, \ldots, n)$ son lexicográficamente positivos. Entonces todas las columnas se mantendrán lexicográficamente positivas a través de todo el cálculo.

2.4.1 Procedimiento

Paso 1: Comience con una matriz A^0 que sea completamente entera y con factibilidad dual.

Paso 2: Seleccione una fila con $a_{i0} < 0,\ \ i > 0$. Esta será la fila generadora. Si todos los $a_{i0} \geq 0$, entonces el problema está resuelto.

Paso 3: Escoja un $\lambda > 0$, según la regla que se explicará en la siguiente sección, y añada la siguiente fila al final de la tabla, como fila pivote.

$$x_k = \left[\frac{a_{i0}}{\lambda}\right] + \sum \left[\frac{a_{ij}}{\lambda}\right](-x_j)$$

Paso 4: Ejecute un paso simplex dual y regrese al paso 2.

2.4.2 Generación del corte

En este algoritmo, la selección de λ es clave para la generación del corte.

Recordemos que [x] denota el mayor número entero menor o igual a x. Si y es cualquier número real (positivo o negativo), y λ es cualquier número positivo, entonces se puede escribir

$$y = \left[\frac{y}{\lambda}\right] \lambda + r_y \tag{2.4-2}$$

donde $0 \leq r_y \leq 1$ (r_y es el residuo de y dividido por λ). En particular, nos interesa,

$$1 = \left[\frac{1}{\lambda}\right] \lambda + r \tag{2.4-3 }$$

Si $\lambda > 1$, entonces $\left[\frac{1}{\lambda}\right] = 0$ y $r = 1$. (2.4-4)

$$\text{Si } \lambda = 1, \text{ entonces } \left[\frac{1}{\lambda}\right] = 1 \text{ y } r = 0. \tag{2.4-5}$$

Se procederá a derivar una desigualdad que tiene que ser satisfecha por toda solución entera. Considere una ecuación en la t-ésima tabla (omitiendo el subíndice de fila) con $a_0 < 0$:

$$x = a_0 + \sum a_j(-x_j^t), \tag{2.4-6}$$

donde x es cualquier componente de $\boldsymbol{x}$ y las x_j^t son las variables no básicas actuales. Se puede expresar x, a_0 y a_j en términos de λ por sustitución en la ecuación ((2.4-2), donde

$$x = x \times 1 = x\left\{\left[\frac{1}{\lambda}\right]\lambda + r\right\} \tag{2.4-7}$$

$$a_j = \left[\frac{a_j}{\lambda}\right]\lambda + r_j \ \ (j = 0,1,2 \ldots, n) \tag{2.4-8}$$

Al sustituir las ecuaciones (2.4-7) y (2.4-8) en la ecuación (2.4-6) y transfiriendo términos se obtiene

$$\sum_{j=1}^{n} r_j\, x_j^t + rx = r_0 + \lambda\left[\left[\frac{a_0}{\lambda}\right] + \sum_{j=1}^{n}\left[\frac{a_j}{\lambda}\right](-x_j^t) + \left[\frac{1}{\lambda}\right](-x)\right] \tag{2.4-9}$$

Dado que $r_j \geq 0$ y $r \geq 0$, y se requiere que tanto x como x_j sean no negativos, el lado izquierdo de la ecuación (2.4-9) siempre será no negativo. Considérese ahora los términos encerrados en paréntesis en el lado derecho de la ecuación (2.4-9). Cada uno de los términos es un entero o se requiere que sea entero. Por lo tanto, todo en el paréntesis es un entero. Se denotará dicho contenido con la variable de holgura x_k, es decir:

$$x_k = \left[\left[\frac{a_0}{\lambda}\right] + \sum_{j=1}^{n}\left[\frac{a_j}{\lambda}\right](-x_j^t) + \left[\frac{1}{\lambda}\right](-x)\right] \tag{2.4-10}$$

Sin embargo, x_k no sólo es un entero, sino un entero no negativo, pues si fuese un entero negativo como –1, –2, …, etc., al multiplicarlo por λ ($\lambda > r_0$) haría que el lado derecho de la ecuación (7) fuese negativo, mientras que el lado derecho es no negativo, como ya se vio.

Consideremos ahora los casos $\lambda = 1$ y $\lambda > 1$.

- Para $\lambda = 1$, $\left[\frac{a_j}{\lambda}\right] = a_j$ y $\left[\frac{1}{\lambda}\right] = 1$. Se puede entonces sustituir x en la ecuación (2.4-10) con la ecuación (2.4-6):

$$x_k = [a_0] + \sum [a_j]\left(-x_j^t\right) - \left\{a_0 + \sum a_j(-x_j^t)\right\} = -f_0 + \sum f_j x_j^t \qquad (2.4\text{-}11)$$

Nótese que esto no es más que la ecuación (2.2-7) para el corte Gomory derivado en la sección 2.2.

- Para $\lambda > 1$, $\left[\frac{1}{\lambda}\right] = 0$ y la ecuación (2.4-10) queda así:

$$x_k = \left[\frac{a_0}{\lambda}\right] + \sum_{j=1}^{n} \left[\frac{a_j}{\lambda}\right]\left(-x_j^t\right) \qquad (2.4\text{-}12)$$

La ecuación (2.4-12) tiene que ser satisfecha por toda solución entera factible de la ecuación (2.4-1). Si $a_0 < 0$ en la ecuación (2.4-6), entonces $\left[\frac{a_0}{\lambda}\right] < 0$ en la ecuación (2.4-12). Por lo tanto, la ecuación (2.4-12) puede usarse como la fila pivote en el método simplex dual. En particular, para cualquier $a_0 < 0$ siempre se puede escoger un λ lo suficientemente grande para que el pivote en (2.4-12) sea –1. Con esto se logra que la tabla siempre se mantenga con números enteros.

Como escoger λ

Se describirá a continuación la regla para escoger λ en el paso 3 del algoritmo. Con el fin de acelerar la convergencia del algoritmo, se escoge λ con base en los siguientes criterios:

1) Mantener la tabla dualmente factible y con números enteros, (que se logra con un pivote = –1).
2) Lograr factibilidad primal en algún momento.
3) Lograr el mayor cambio en el valor de la función objetivo.

Procedimiento para escoger λ:

Sea la fila generadora

$$x = a_0 + \sum a_j(-x_j) \tag{2.4-13}$$

y que la fila derivada sea

$$x_k = \left[\frac{a_0}{\lambda}\right] + \sum \left[\frac{a_j}{\lambda}\right](-x_j) \tag{2.4-14}$$

Para cualquier $a_j < 0$, siempre se puede escoger λ lo suficientemente grande de modo que $\left[\frac{a_0}{\lambda}\right] = -1$. Según el método simplex dual lexicográfico de escoger la columna pivote, se quiere que, si $\propto_p$ es la columna pivote,

$$\left|\frac{-1}{\left[\frac{a_p}{\lambda}\right]} \propto_p\right| = \min \left|\frac{-1}{\left[\frac{a_j}{\lambda}\right]} \alpha_j\right| \qquad (\alpha_j < 0). \tag{2.4-15}$$

Puesto que $\left[\frac{a_p}{\lambda}\right] = -1$; $\left[\frac{a_j}{\lambda}\right]$ es un entero negativo, –1, –2, …, digamos $-u_j$:

$$\frac{\propto_p}{1} \prec^L \frac{\propto_j}{u_j} \prec^L \frac{\propto_j}{1}$$

de modo que $\propto_p$ tiene que ser la columna más pequeña lexicográficamente. Esto significa que entre todas las posibles columnas pivote (con $a_j < 0$) la columna pivote siempre será la columna lexicográficamente más pequeña indistintamente de cuál λ se escoja.

Nótese que si hay dos valores λ_1 y λ_2 que producen $\left[\frac{a_p}{\lambda}\right] = -1$, se escoge el valor más pequeño porque la cantidad por la cual $\propto_0$ decrecerá es

$$\propto_0^{t+1} = \propto_0^t + \left[\frac{a_{i0}}{\lambda_i}\right] \propto_p^t \qquad (\lambda_i = \lambda_1, \lambda_2).$$

Es decir, mientras más pequeño sea λ, más decrecerá lexicográficamente la columna 0. Por lo tanto, la regla es:

PASO 1. Sea k la fila fuente.

PASO 2. Sea $\propto_p$ la columna lexicográficamente más pequeña con $a_{kj} < 0$.

PASO 3. Para cada $a_{kj} < 0$, sea u_j, el mayor entero tal que $\propto_p \prec^L \frac{\propto_j}{u_j}$.

PASO 4. Sea $\left[\frac{a_{kj}}{\lambda_j}\right] = u_j$ o $\lambda_j = \frac{a_{kj}}{u_j}$. Este es el λ_j que hará que $\propto_p \prec^L \frac{\alpha_j}{\left[\frac{-a_{kj}}{\lambda_j}\right]}$.

PASO 5. Sea $\lambda = max_j \, \lambda_j$ para $a_{kj} < 0$.

Este método de selección de λ es para hacer que el pivote sea –1, que la tabla se mantenga dualmente factible y a la vez lograr el mayor decremento lexicográfico en la columna 0.

Una característica interesante del algoritmo completamente entero es que su validez no depende de que los coeficientes de a_{ij} sean enteros[3].

2.4.3 Ejemplo.

Se considera el siguiente problema de programación entera, que se resolverá atendiendo al procedimiento descrito:

$$\max x_0 = -10x_1 - 14x_2 - 21x_3$$

Sujeto a

$$2x_1 + 2x_2 + 7x_3 \geq 14$$

$$8x_1 + 11x_2 + 9x_3 \geq 12$$

$$9x_1 + 6x_2 + 3x_3 \geq 10$$

---

[3] Una demostración de eso puede hallarse en Hu, 1970, p. 251.

$$x_1, x_2, x_3 \geq 0 \text{ entero}$$

Lo cual genera la siguiente tabla inicial:

t_0	1	$-x_1$	$-x_2$	$-x_3$	
x_0	1	10	14	21	
x_1	0	–1	0	0	
x_2	0	0	–1	0	
x_3	0	0	0	–1	
x_4	–14	–2	–2	–7	← Fila generadora
x_5	–12	–8	–11	–9	
x_6	–10	–9	–6	–3	
x_7	–4	–1	–1	–2	$\lambda = \frac{7}{2}$

$$\mu_1 = 1, \qquad 10 \leq \frac{14}{\mu_2}, \quad \mu_2 = \left\lfloor\frac{14}{10}\right\rfloor = 1, \quad 10 \leq \frac{21}{\mu_3}, \quad \mu_3 = \left\lfloor\frac{21}{10}\right\rfloor = 2$$

$$\lambda_1 = \frac{2}{\mu_1} = 2, \quad \lambda_2 = \frac{2}{\mu_2} = 2, \quad \lambda_3 = \frac{7}{\mu_3} = \frac{7}{2}$$

$$\lambda = \max(2, 2, \frac{7}{2}) = \frac{7}{2}$$

El corte entonces es

$$x_7 = \left\lfloor\frac{-14}{\lambda}\right\rfloor + \left\lfloor\frac{-2}{\lambda}\right\rfloor(-x_1) + \left\lfloor\frac{-2}{\lambda}\right\rfloor(-x_2) + \left\lfloor\frac{-7}{\lambda}\right\rfloor(-x_3),$$

$$x_7 = -4 - 1(-x_1) - 1\,(-x_2) - 2\,(-x_3) \geq 0$$

Esta ecuación se añade al final de la tabla. Pivotando y repitiendo el procedimiento se tienen las siguientes tablas:

t_1	1	$-s_1$	$-x_2$	$-x_3$	
x_0	–40	10	4	1	
x_1	4	–1	1	2	
x_2	0	0	–1	0	
x_3	0	0	0	–1	
x_4	–6	–2	0	–3	← Fila generadora

x_5	20	–8	–3	7	
x_6	26	–9	3	15	
x_8	–2	–1	0	–1	$\lambda = 3$

t_2	1	$-s_1$	$-x_2$	$-s_2$	
x_0	–42	9	4	1	
x_1	0	–3	1	2	
x_2	0	0	–1	0	
x_3	2	1	0	–1	
x_4	0	1	0	–3	
x_5	6	–15	–3	7	
x_6	–4	–24	3	15	← Fila generadora
x_9	–1	–1	0	0	$\lambda = 24$

t_3	1	$-s_3$	$-x_2$	$-s_2$	
x_0	–51	9	4	1	
x_1	3	–3	1	2	
x_2	0	0	–1	0	
x_3	1	1	0	–1	
x_4	–1	1	0	–3	← Fila generadora
x_5	21	–15	–3	7	
x_6	20	–24	3	15	
x_{10}	–1	0	0	–1	$\lambda = 3$

t_4	1	$-s_3$	$-x_2$	$-s_4$
x_0	–52	9	4	1
x_1	1	–3	1	2
x_2	0	0	–1	0
x_3	2	1	0	–1
x_4	2	1	0	–3
x_5	14	–15	–3	7
x_6	5	–24	3	15

La solución óptima, que aparece en la tabla 4, es $x_0 = -52, x_1 = 1,\ x_2 = 0,\ x_3 = 2.$

2.4.4 Convergencia del algoritmo

Al igual que con el método dual fraccionario, esta demostración también depende del supuesto que hay un límite inferior M para la función objetivo x_0. Dado que todos los pasos de pivote son del método simplex dual, es cierto que

$$\propto_0^t \succ^L \propto_0^{t+1}.$$

Puesto que todos los números permanecen enteros, si a_{00} decrece, debe hacerlo por una cantidad entera y caerá debajo de M después de un número finito de pasos. Si el algoritmo no es finito, entonces a_{00} debe permanecer fijo para todos los ciclos después de un valor t mayor que, digamos t, donde $t > t_0$.

Considere ahora el primer componente a_{10} del vector α_0, éste también debe decrecer por una cantidad entera, si decrece estrictamente. Si a_{10} decrece por debajo de cero, entonces será seleccionada como la fila generadora. Si a_{1j} > 0 para toda j, entonces eso demuestra que el problema no es factible. Si a_{1j}< 0 para algún j, entonces la fila derivada es

$$x_k = \left[\frac{a_{10}}{\lambda}\right] + \sum_{j=1}^{n} \left[\frac{a_{1j}}{\lambda}\right] (-x_j)$$

Esta fila añadida tiene $\left[\frac{a_{10}}{\lambda}\right] < 0$ y $\left[\frac{a_{1j}}{\lambda}\right] = -1$ para por lo menos una j. (La regla para seleccionar λ lo garantiza). Pivotar sobre esta fila incrementará estrictamente a_{10}. Dado que $\propto_0^t \succ^L \propto_0^{t+1}$, esto implica que a_{00}^{t+1} debe decrecer, lo cual contradice nuestro supuesto que a_{00} es fijo. Por lo tanto, si el algoritmo no es finito, a_{1j} debe permanecer fijo para todos los ciclos después de un valor t mayor que, digamos t_1, donde $t_1 > t_0$.

El mismo argumento puede aplicarse al segundo componente, al tercero, y así sucesivamente. Así que después de un número finito de pasos, todos los a_{i0} $(i = 1, \dots, n + m)$ serán enteros no negativos.

Capítulo 3

Comparación de los tres métodos de corte

En esta tesis se han examinado tres algoritmos de corte introducidos por R. E. Gomory: el algoritmo fraccionario para problemas enteros, el algoritmo fraccionario para problemas mixtos y el algoritmo completamente entero. En este capítulo se desea destacar sus similitudes y diferencias, así como sus limitaciones.

3.1 Similitudes

Al igual que todos los métodos de corte, los tres métodos producen nuevas restricciones, llamados "cortes", que se añaden a las originales para eventualmente producir un problema lineal cuya solución óptima entera corresponde a un vértice con componentes completamente enteros. Los tres métodos utilizan el método dual simplex y utilizan el corte como la fila pivote.

3.2 Diferencias

La diferencia principal entre los algoritmos de corte estriba en la manera de generar las desigualdades que producen los cortes y la manera de aplicarlos.

En el corte fraccionario, primero se llega a una solución óptima relajada, se añade la ecuación de corte y luego se continúa con el método simplex dual. En el algoritmo entero, los cortes se añaden desde el principio y el método simplex dual se usa durante todo el procedimiento.

En el método entero, el elemento pivote siempre es –1. Esto conlleva la ventaja de eliminar las dificultades de redondeo en los cálculos cuando se usa una computadora.

En el método fraccionario se alcanza la solución cuando todas las variables de decisión restringidas a enteros llegan, efectivamente, a ser enteros (si hay solución). En el método dual completamente entero, se llega a la solución sólo cuando todas sean no-negativas.

3.3 Limitaciones

Los métodos de corte tienden a ser ineficientes porque el número de cortes a agregar, aunque finito, suele ser bastante grande. Por lo que, hoy día, la mayoría de los solucionadores de programación entera comerciales utilizan las técnicas de plano de corte en combinación con otros métodos como el de Ramificación y Acotación (Malón, 2020, pág. v).

Si bien el algoritmo dual completamente entero elimina el problema de redondeo, su convergencia puede mostrar un comportamiento errático dependiendo de las características de cada problema. Véase por ejemplo (Baugh, Ibaraki, & Muroga, 1968). Otra desventaja de éste y todos los algoritmos duales enteros es que, hasta que se encuentra la solución óptima, no se conoce ninguna solución factible.

En la siguiente tabla se presenta una comparación de los tres métodos analizados en este trabajo, desde el punto de vista de sus procedimientos.

Comparación de los tres métodos de corte de Gomory		
Algoritmo fraccionario para problemas enteros	**Algoritmo fraccionario para problemas mixtos**	**Algoritmo completamente entero**
Sujeto a errores de redondeo.		Elimina los problemas de redondeo.
a_{ij} tienen que ser enteros en la tabla inicial.		a_{ij} pueden ser números reales en la tabla inicial. (a_{0j} tienen que ser enteros).
a_{ij}^t no tienen que ser enteros en tablas sucesivas.		Si a_{ij} son enteros en la tabla inicial, entonces se mantienen enteros.

<table>
<tr><td colspan="2">Se puede usar el método simplex primal o dual para obtener el valor óptimo y luego se usa el método simplex dual.</td><td>El método simplex dual se usa a través de todo el cálculo.</td></tr>
<tr><td colspan="2">Un a_{i0} no entero convierte la i-ésima fila en una fila generadora.</td><td>Un a_{i0} negativo convierte a la i-ésima fila en una fila generadora. ($i \neq 0$)</td></tr>
<tr><td>La fila pivote es la ecuación:
$$x_k = -f_0 + \sum f_j x_j$$</td><td>La fila pivote es la ecuación:
$$x_k = -f_0 + \sum_{j \in N^+} a_j x_j + \frac{f_0}{f_0 - 1} \sum_{j \in N^-} a_j x_j$$</td><td>La fila pivote es la ecuación:
$$x_k = \left[\frac{a_0}{\lambda}\right] + \sum \left[\frac{a_j}{\lambda}\right](-x_j^t).$$
$a_0 < 0, \quad a_j \neq 0$</td></tr>
<tr><td>$\lambda = 1$</td><td>No aplica</td><td>$\lambda > 1$</td></tr>
<tr><td colspan="2">La columna pivote p se determina por:
$$\frac{1}{f_p}\alpha_k \prec^L \frac{1}{f_j}\alpha_j$$</td><td>La columna pivote p es siempre la columna lexicográficamente más pequeña con $a_p < 0$, y se determina antes de generar el corte.</td></tr>
<tr><td colspan="2">Se pueden añadir muchas desigualdades a la vez y luego aplicar el método simplex dual.</td><td>Las desigualdades se añaden sólo una a la vez a fin de mantener la tabla completamente entera.</td></tr>
</table>

Capítulo 4

Ejemplo de aplicación

La empresa Panameñísima de Acero[4] quiere participar en una licitación para la compra de barras de acero corrugado que se usarán en las columnas de la línea 3 del Metro hacia Arraiján-La Chorrera. La empresa puede ofrecer barras de acero grado 60 y grado 40. A fin de cumplir con sus compromisos existentes, la empresa debe sujetarse a las restricciones de disponibilidad de material y de horas-hombre.

Figura 1. Crédito: Metro de Panamá

Por tonelada	Grado 60	Grado 40
Precio de venta (miles)	B/.10	B/.8
Costo de producción (miles)	B/.4	B/.3
Uso de materia prima	3	1
Hora-hombre	1	2

La empresa tiene 11 toneladas de materia prima disponible inmediatamente para el proyecto. Cada tonelada de acero grado 60 requiere 3 unidades de materia prima, mientras que el de 40, sólo una.

La máquina que produce el acero grado 40 requiere 2 horas-hombre por tonelada. Mientras que la de 60, sólo requiere la mitad por ser de compra más reciente. Debido a compromisos previos, sólo se disponen de 5 horas-hombre para este proyecto.

[4] El nombre de la empresa y las cantidades son ficticios.

Sólo se permiten unidades enteras de toneladas en la producción. Aunque la competencia es fuerte, debido a que el proyecto requiere grandes cantidades de acero, se puede asumir razonablemente que toda la producción se venderá a los precios indicados en la tabla. Se desea hallar la combinación de producción de barras de acero grado 60 y 40 que maximice la ganancia.

4.1 Planteamiento del problema

El proceso se inicia con la definición de las variables de decisión, la función objetivo y las restricciones.

4.1.1 Las variables de decisión

Se usará la variable x_1 para indicar el número de toneladas producidas de acero grado 60, y x_2, para las de grado 40.

4.1.2 La función objetivo

El objetivo es maximizar la ganancia neta al suplir la demanda que está definida por el precio de venta menos los costos por tonelada producida. Es decir:

$$\max\ (10-4)x_1 + (8-3)x_2$$

que resulta:

$$\max\ 6x_1 + 5x_2$$

4.1.3 Las restricciones

Hay que suplir la demanda dentro las restricciones impuestas por la disponibilidad de materia prima y personal:

$$3x_1 + x_2 \le 11 \qquad \text{restricción de materia prima}$$

$$x_1 + 2x_2 \le 5 \qquad \text{restricción de horas-hombre.}$$

Se completa el modelo con las condiciones de no negatividad para x_1 y x_2.

El problema, entonces, queda planteado así:

$$\max \; 6x_1 + 5x_2$$

sujeto a:

$$3x_1 + x_2 \leq 11$$

$$x_1 + 2x_2 \leq 5$$

$$x_1, x_2 \geq 0 \text{ y entero}$$

4.2 La solución

Usando los valores del problema planteado, y agregando las variables de holgura x_3 y x_4, la tabla inicial, para resolver el problema por el método dual fraccionario para problemas enteros, queda así:

t_0	1	$-x_1$	$-x_2$
x_0	0	−6	−5
x_1	0	−1	0
x_2	0	0	−1
x_3	11	3	1
x_4	5	1	2

Paso 1: Resolver el problema lineal asociado usando el método simplex. Aplicando el método simplex construimos las tablas t_1 y t_2. Los valores sombreados representan el pivote.

t_1	1	$-x_3$	$-x_2$
x_0	22	2	−3
x_1	11/3	1/3	1/3
x_2	0	0	−1
x_3	0	−1	0
x_4	4/3	−1/3	5/3

t_2	1	$-x_3$	$-x_4$
x_0	122/5	7/5	9/5
x_1	17/5	2/5	−1/5
x_2	4/5	−1/5	3/5
x_3	0	−1	0
x_4	0	0	−1

El valor óptimo para la solución relajada es $Z = x_{00} = \frac{122}{5}$, $x_1 = \frac{17}{5}$ y $x_2 = \frac{4}{5}$.

Paso 2: Como la solución no es entera, se aplica un corte de Gomory. Se selecciona una fila con a_{i0} no entero como la fila generadora. Arbitrariamente se escoge la ecuación correspondiente a x_2 porque tiene la mayor parte fraccionaria.

La ecuación correspondiente a x_2 es:

$$x_2 = \frac{4}{5} - \frac{1}{5}(-x_3) + \frac{3}{5}(-x_4)$$

Por la ecuación (2.2-2):

$$f_0 = \frac{4}{5}$$

$$f_3 = \frac{4}{5}$$

$$f_4 = \frac{3}{5}$$

Sustituyendo en la ecuación (2.2-7) se tiene el corte de Gomory:

$$\boldsymbol{x_5} = -\frac{4}{5} - \frac{4}{5}(-\boldsymbol{x_3}) - \frac{3}{5}(-\boldsymbol{x_4}) \tag{4.2-1}$$

Esta restricción se añade al final de la tabla $\boldsymbol{t_2}$, dando origen a la tabla 3:

t_3	1	$-x_3$	$-x_4$	
x_0	122/5	7/5	9/5	
x_1	17/5	2/5	−1/5	
x_2	4/5	−1/5	3/5	Fila generadora
x_3	0	−1	0	
x_4	0	0	−1	
x_5	−4/5	4/5	3/5	

Paso 3: Luego de añadir esta restricción se continua con el método simplex dual lexicográfico explicado en la sección 2.1.4. Con base en las ecuaciones (2.1-2) y (2.1-3) se escoge $\boldsymbol{x}_5$ como la variable que sale de la base y $\boldsymbol{x}_3$como la que entra. La tabla se actualiza aplicando las ecuaciones (2.1-4), (2.1-5) y (2.1-6) para las columnas 0, 1 y 2 de la tabla 4, respectivamente:

$$\propto_0^4 = \propto_0^3 - \propto_1^3$$

$$\propto_1^4 = -\frac{5}{4}\propto_1^3$$

$$\propto_2^4 = \propto_2^3 - \frac{3}{4}\propto_1^3$$

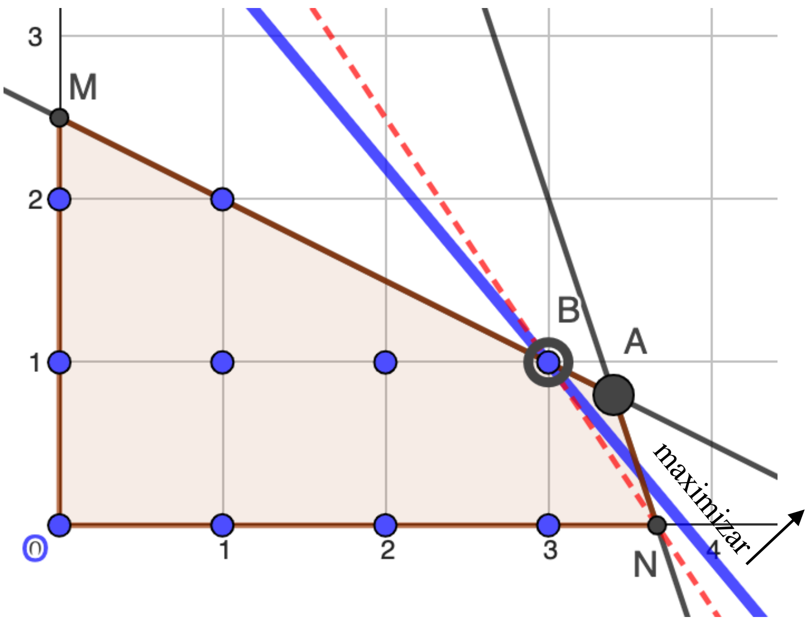
lo que da como resultado la tabla final:

$\boldsymbol{t_4}$	1	$-\boldsymbol{x}_5$	$-\boldsymbol{x}_4$
$\boldsymbol{x_0}$	23	7/4	3/4
$\boldsymbol{x_1}$	3	1/2	−1/2
$\boldsymbol{x_2}$	1	−1/4	3/4
$\boldsymbol{x_3}$	1	−5/4	3/4
$\boldsymbol{x_4}$	0	0	−1
$\boldsymbol{x_5}$	0	−1	0

La solución óptima entera es **Z=** $\boldsymbol{x_{00} = 23,\ x_1 = 3}$ **y** $\boldsymbol{x_2 = 1}$ que significa que se deben producir tres toneladas de barras de acero grado 60 y una de grado 40, con una ganancia máxima de B/.23,000.

Solución Gráfica

Como el problema sólo tiene dos variables de decisión se puede apreciar la solución gráficamente. La línea más gruesa representa la función objetivo. El área sombreada es la región factible para el problema lineal asociado y los puntos dentro de ella son las soluciones enteras factibles del problema original. La línea punteada corresponde al corte de Gomory, ecuación ((4.2-1). El punto A es la solución óptima relajada y el punto B, la solución óptima entera del problema.



Conclusiones

El objetivo de la disciplina denominada Investigación de Operaciones se enfoca a los problemas relacionados con el mejoramiento de la eficiencia de una organización. La naturaleza de la organización no es limitante, por lo que ha sido aplicada de manera extensa en una gran diversidad de áreas: producción, transporte, construcción, salud, planeación, fuerzas armadas, servicios públicos, por mencionar unas cuantas de ellas. En muchas de estas áreas se requiere que las variables de decisión involucradas tengan valores enteros. De allí la importancia y el auge de la Programación Entera.

Desde su introducción en 1958 por Gomory, los planos de corte han jugado un papel fundamental en la teoría de la programación entera.

A pesar de este importantísimo avance teórico, los corte de Gomory han sido considerados ineficientes debido a que se requieren muchos cortes para llegar a una solución. Por eso, hoy día los solucionadores comerciales lo utilizan en combinación con otros métodos para llegar a una solución de una manera más rápida y eficiente.

Pese a esta limitación, el uso de desigualdades lineales válidas, como planos de corte, es una herramienta poderosa y bien establecida en la programación entera. Por eso, en lugar de abandonarlos en la obsolescencia, su estudio continuo ha proporcionado perspectivas nuevas sobre la programación entera que han dado origen a algoritmos de corte más eficientes, que utilizan sus bases teóricas como principios fundamentales. De allí que los métodos de plano de corte seguirán siendo temas de discusión relevantes en el estudio de la solución de problemas enteros.

Recomendaciones

Así como la solución al problema relajado, por el algoritmo simplex, representa el "inicio" para el corte de Gomory en la búsqueda de la solución de un problema entero, así también espero que los resultados de este trabajo sean el inicio de proyectos futuros en el tema de la programación entera, pues, con respecto a los métodos de corte de plano, aún queda mucho por investigar.

Una de las debilidades de los métodos de corte es su tiempo de convergencia. Este tema, de mejorar su eficiencia, sigue siendo objeto de mucho estudio.

El propio Gomory sugirió dos técnicas para hacer cortes "más fuertes", en el sentido que remuevan una porción mayor del área factible sin remover puntos enteros factibles. El primero es llamado **corte t-Gomory** (Gomory, 1958). El segundo, llamado **corte de Gomory fuerte**, es un caso especial del corte de Gomory para problemas mixtos (Gomory, 1960).

Farwell (2006) sugiere un tercer método producto de la combinación de los dos propuestos por Gomory. Letchforda & Lodib (2002) también han sugerido otro algoritmo para fortalecer el corte de Gomory. Por su relevancia, recomiendo la profundización en el estudio de estas propuestas.

Gomory también reconoció el problema del redondeo en las computadoras y lo resolvió con el algoritmo completamente entero. Pero, en ocasiones, su uso ha dado resultados erráticos de convergencia, dependiendo del tamaño y tipo del problema (Baugh, Ibaraki, & Muroga, 1968).

Farwell (2006) ha sugerido que el problema también se resuelve usando aritmética exacta en lugar de aritmética de punto flotante. La aritmética exacta es una forma de realizar operaciones aritméticas con fracciones en lugar de usar representaciones aproximadas. Su desventaja es que hace que las operaciones con fracciones se vuelvan más complejos. Además, se corre el peligro de exceder la capacidad de almacenamiento de la computadora, dado que los valores de los numeradores y denominadores se deben almacenar por separado. Un trabajo de interés podría ser estudiar el orden de complejidad de uno y otro procedimiento.

Los métodos de plano de corte también son aplicables en la programación no lineal, es decir, problemas de optimización en los que la función objetivo o alguna de las restricciones corresponden a funciones no lineales (Boyd & Vandenberghe, 2003). La adaptación teórica de los métodos de plano de corte a este tipo de problemas es digna de estudios posteriores.

Bibliografía

Baugh, C. R., Ibaraki, T., & Muroga, S. (1968). *Results in Using Gomory's All-Integer Integer Algorithm to Design Optimum Logic Networks*. Recuperado el julio de 2023, de http://www.jstor.org/stable/169072

Boyd, S., & Vandenberghe, L. (18 de September de 2003). *Localization and Cutting-plane Methods*. Recuperado el julio de 2023, de web.standford.edu: https://web.stanford.edu/class/ee392o/localization-methods.pdf

Cornuéjols, G. (2005, febrero 2). *Revival of the Gomory Cuts in the 1990´s.* Retrieved from andrew.cmu.edu: https://www.andrew.cmu.edu/user/gc0v/webpub/gomory.pdf

Cornuéjols, G. (2010). *The Ongoing Story of Gomory Cuts*. Obtenido de https://www.math.uni-bielefeld.de/documenta/vol-ismp/37_cornuejols-gerard.pdf

Farwell, K. (2006). *Gomory Cutting Plane Algorithm Using Exact Arithmetic.* Recuperado el Julio de 2023, de https://homepages.rpi.edu/~mitchj/phdtheses/kris/krisfarwell.pdf

G. Srinivasan, Lecture 37, All Integer Dual Algorithm, Indian Institute of Technology, Madras, 2012. https://m.youtube.com/watch?v=vqk0a8y8w_U&feature=youtu.be

Gomory, R. E. (1958). *Outline of an Algorithm for Integer Solutions to Linear Programs. Bulletin of the American Mathematical Society, 64(5), 275–278.*

Gomory, R. E. (1960). *An Algorithm for the Mixed Integer Problem. Rand Report, RM-2597.* Santa Mónica, CA:: Rand Corporation.

Gomory, R. E. (29 de enero de 1960). *All-integer integer programming algorithm*. (researchgate.net, Productor) Recuperado el julio de 2023, de Research Gate: https://www.researchgate.net/publication/238987161_All-integer_integer_programming_algorithm

Hu, T.C., *Integer Programming and Network Flows*, Addison-Wesley Publishing Company, 1970

Kaufmann, A., & Henry-Labordere, A. (1977). *Integer and Mixed Programming: Theory and Applications.* Paris, Francia: Academic Press.

Letchforda, A. A., & Lodib, A. (2002). *Strengthening Chva ́tal-Gomory cuts and Gomory fractional cuts.* Operations Research Letters, 30(2):74–82,.

Malón, T. M. (2020). *Problemas de programación lineal entera. Aplicación a un problema de planificación de turnos hospitalarios*. Obtenido de https://zaguan.unizar.es/record/98113/files/TAZ-TFG-2020-2162.pdf

Mathur, K., & Salkin, H. M. (1989). *Foundations of Integer Programming.* Elsevier Science Publishing Co., Inc.

Operations Research: An Introduction : Integer Linear Programming. (s.f.). Recuperado el Julio de 2023, de brainkart.com: https://www.brainkart.com/article/Computational-Considerations-in-ILP_11245/

Wikipedia. (s.f.). Recuperado en julio de 2023, de Cutting-plane method: https://en.m.wikipedia.org/wiki/Cutting-plane_method