



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

ALGORITMO DE KARMARKAR

Pablo Valiña Villar

2018/2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Algoritmo de Karmarkar

Pablo Valiña Villar

Julio, 2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento: Estadística e Investigación Operativa
Título: Un algoritmo interior para problemas de programación lineal: El algoritmo de Karmarkar
Breve descripción do contido
<p>A pesar de su popularidad en la práctica, es bien sabido que, desde el punto de vista teórico, el método símplex es un algoritmo cuyo comportamiento en el peor caso es exponencial. El método de punto interior de Karmarkar es quizá la alternativa más frecuente al método símplex. Exhibe un comportamiento promedio comparable al del método símplex, pero además su comportamiento en el peor caso es polinomial. La estructura de este trabajo se centraría en los siguientes puntos:</p> <ul style="list-style-type: none">▪ Discusión de la complejidad computacional del método símplex.▪ Descripción detallada del algoritmo de Karmarkar.▪ Discusión de los aspectos teóricos que rodean al algoritmo de Karmakar: convergencia, generalidad y complejidad computacional.
Recomendacións
Outras observacións

Índice general

Resumen	VIII
Introducción	XI
1. Programación Lineal y El Método Simplex	1
1.1. Programación lineal	1
1.1.1. Formulación matemática	1
1.1.2. Supuestos subyacentes	2
1.1.3. Importancia de la programación lineal	3
1.1.4. Transformaciones de problemas de programación lineal	3
1.2. Método simplex	6
1.2.1. Introducción	6
1.2.2. Bases y Soluciones Básicas	7
1.2.3. El método de la M grande	11
1.2.4. La tabla del método simplex	12
1.2.5. Eficiencia del método	14
1.3. Dualidad	17
2. Complejidad y Cubos de Klee-Minty	21
2.1. Complejidad computacional	21
2.1.1. Clases de complejidad	24
2.1.2. Complejidad computacional del método simplex	25
2.2. Cubos de Klee-Minty	26
3. Algoritmo de Karmarkar. Definición	29
3.1. Conversión de un problema lineal general a la forma de Karmarkar	29
3.2. Algoritmo de Karmarkar	31
4. Análisis del Algoritmo de Karmarkar.	45

Resumen

La programación lineal constituye un importante campo de la optimización, debido a que casi cualquier problema práctico de la investigación operativa se puede plantearse como un problema de este tipo. En lo relacionado con la resolución de los problemas de programación lineal resaltamos el método símplex, como el más utilizado para resolver este tipo de problemas. Por múltiples razones es el método más popular hoy en día, entre ellas destaca su capacidad para resolver, en una cantidad de tiempo razonable, problemas de gestión importantes y complejos. Pero este método no es del todo eficaz en algunas ocasiones. Por este motivo surgió el algoritmo de Karmarkar, ya que se comporta de una manera más razonable en los casos donde el método símplex falla.

Abstract

Linear programming is an important field of optimization, because almost any practical problem of operations research can be posed as such a problem of this type. In regard the resolution of linear programming problems we highlight the simplex method, as the most used to solve this type of problems. For many reasons it is the most popular method nowadays, among them its ability to solve, in a reasonable amount of time, important and complex management problems. But this method is sometimes not entirely effective. This is why the Karmarkar algorithm was born, because it behaves in a more reasonable way in cases where the simplex method fails.

Introducción

La programación lineal se ocupa de la optimización (maximización o minimización) de una función lineal, mientras satisface un conjunto de restricciones. La programación lineal aparece durante la Segunda Guerra mundial para planificar los gastos de operaciones militares, con el fin de reducir los costos al ejército y aumentar el daño realizado al enemigo. Los considerados fundadores de la técnica son George Dantzig, un matemático estadounidense, quien es considerado el padre de la programación lineal, John Von Neumann, un matemático húngaro-estadounidense, que desarrolló la teoría de la dualidad y Leonid Kantoróvich, un matemático de origen ruso, que utiliza técnicas similares en la economía antes de Dantzig durante 1939, pero su trabajo permaneció desconocido hasta 1959. Por este motivo, la concepción de la clase general de problemas de programación lineal generalmente se le atribuye a Dantzig.

El nombre de programación lineal surge en 1947, cuando Dantzig trabajaba como asesor de las Fuerzas Aéreas de los Estados Unidos en el desarrollo de una herramienta para planificación de actividades armamentísticas, ya que cada una de estas planificaciones se denominaba “programación” (“program” en inglés).

En 1949 Dantzig publicó el método símplex para resolver problemas de programación lineal, y que desde entonces ha sido el algoritmo más utilizado para resolver este tipo de problemas, además, este algoritmo es considerado uno de los 10 algoritmos más importantes del siglo XX.

Aún así, existen ciertos tipos de problemas con una estructura concreta o con un número de variables y de inecuaciones verdaderamente grandes, donde dicho algoritmo tiene un peor rendimiento. En 1984, Narendra Karmarkar, un matemático indio establecido en Estados Unidos, diseñó una importante modificación del método del simplex, denominada el algoritmo de Karmarkar, que ha demostrado una eficiencia mayor en determinados casos.

Buena parte de este trabajo se dedicará a entender y analizar este algoritmo de Karmarkar.

Capítulo 1

Programación Lineal y El Método Simplex

La información expuesta en este capítulo se basa principalmente en los apuntes de González-Díaz (2017).

1.1. Programación lineal

1.1.1. Formulación matemática

La programación lineal es una parte importante de la investigación operativa, dedicada a la optimización (minimización o maximización) de funciones lineales, respetando siempre un conjunto de restricciones también lineales, de igualdad o desigualdad.

Vamos a ver ahora la formulación de dichos problemas. Los elementos principales para definir un problema general de programación matemática son los siguientes:

- **Variables de decisión**, $\mathbf{x} \in \mathbb{R}^n$: Representan las decisiones para las cuales buscamos una configuración de valores óptima.
- **Función objetivo**, $f : \mathbb{R}^n \rightarrow \mathbb{R}$: Esta es la función que queremos maximizar o minimizar y representa el coste o beneficio asociado a cada combinación de variables de decisión.
- **Restricciones**: Representan la viabilidad de los valores de las variables. Existen de dos tipos:
 - **Desigualdad**, $g_i(\mathbf{x}) \leq 0$, $i \in \{1, \dots, m\}$.
 - **Igualdad**, $h_j(\mathbf{x}) = 0$, $i \in \{1, \dots, l\}$.

Entonces para hacernos una idea lo más habitual en un problema de programación matemática sería lo siguiente:

$$\begin{aligned} &\text{Minimizar } f(\mathbf{x}) \\ &\text{Sujeto a } g_i(\mathbf{x}) \leq 0, \quad i \in \{1, \dots, s\} \\ &\quad h_j(\mathbf{x}) = 0, \quad j \in \{1, \dots, k\}. \end{aligned}$$

Dicho todo esto tenemos que tener claro que un problema de programación es un caso particular, como su propio nombre indica, un problema de programación matemática, ya que las funciones y las restricciones son todas lineales. Entonces para definir un problema de programación lineal necesitaremos:

- **Vector de costes**, $\mathbf{c} \in \mathbb{R}^n$.
- **Matriz de restricciones**, $\mathbf{A} \in \mathbb{R}^{m \times n}$.
- **Vector de lados derechos**, $\mathbf{b} \in \mathbb{R}^m$.

Además, las restricciones de la forma $\mathbf{x}_i \geq 0$ se conocen como restricciones de no negatividad. Una vez que tenemos identificados estos elementos podemos escribir de forma más compacta el problema de programación lineal anterior usando notación matricial:

$$\begin{aligned} &\text{Minimizar } \mathbf{c}\mathbf{x} \\ &\text{Sujeto a } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ &\quad \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

El conjunto de puntos $\{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ se conoce como región factible.

1.1.2. Supuestos subyacentes

Al modelar dichos problemas de programación lineal debemos ser conscientes de que esto conlleva implícitamente hacer una serie de supuestos, entre que podemos destacar los siguientes:

- **Proporcionalidad:** Dada una variable cualquiera x_j , su contribución al coste final es $c_j x_j$ y su contribución a la restricción i es $a_{ij} x_j$. Esto no permite modelar, por ejemplo, las siguientes situaciones:
 - **Economías de escala:** Situaciones en las que cuanto más compremos más barata nos saldrá cada unidad. La proporcionalidad “obliga” a que el coste por unidad sea el mismo independientemente del número de unidades que compremos.

- **Costes fijos:** Como por ejemplo el coste de una llamada telefónica, si x_j es la duración de la llamada, el coste asociado será de la forma $c_j x_j + f_j$ si $x_j > 0$ y 0 si $x_j = 0$.
- **Aditividad:** El coste total es la suma de los costes asociados a cada una de las variables, esto no permite, por ejemplo, considerar interacciones entre las variables, como la efectividad de un medicamento de que depende de la cantidad de alcohol que se tome.
- **Divisibilidad:** Las variables pueden tomar cualquier valor fraccionario. Esto no permitirá modelar problemas cuyas variables representen entidades indivisibles como por ejemplo personas, animales o coches.
- **Determinismo:** Se supone en estos problemas que los valores \mathbf{A} , \mathbf{b} y \mathbf{c} se conocen de manera totalmente precisa. Esto puede suponer problemas cuando estos parámetros representan por ejemplo precios y demandas, que en general están sujetos a cierta incertidumbre.

1.1.3. Importancia de la programación lineal

A pesar de todas estos supuestos, los modelos de programación lineal son los más usados hoy en día, especialmente por los siguientes motivos:

- **Facilidad de resolución:** Son problemas en la clase de complejidad P , para los cuales existen algoritmos polinomiales para resolverlos.
- **Aplicabilidad:** Muchos problemas de interés se pueden formular de esta forma como un problema de programación lineal, y además muchos otros pueden ser aproximados por dichos problemas.
- **Uso instrumental:** Muchos problemas de optimización de naturaleza no lineal, habitualmente las mejores técnicas para la resolución de este tipo de problemas son mediante sucesivas resoluciones de muchas “aproximaciones” lineales de dichos problemas.

1.1.4. Transformaciones de problemas de programación lineal

A continuación veremos algunas transformaciones que tienen bastante interés en la práctica. Otras se usan principalmente, porque las equivalencias resultantes son útiles a la hora de analizar teóricamente las propiedades de los problemas de programación lineal.

- **Función objetivo:** Todo problema de minimización puede transformarse en un problema de maximización y viceversa.

$$\max \sum_{j=1}^n c_j x_j = - \min \sum_{j=1}^n -c_j x_j.$$

Por tanto, cualquier problema de maximización (minimización) puede convertirse en uno de minimización (maximización) sin más que multiplicar los coeficientes por -1 . Una vez resuelto el nuevo problema, el óptimo de la función objetivo del problema original será el del nuevo problema pero multiplicado por -1 .

- **Restricciones:** En este caso hay varias manipulaciones posibles:
 - Si queremos cambiar una de estas restricciones “ \leq ” (“ \geq ”) por su opuesta “ \geq ” (“ \leq ”), simplemente tendremos que multiplicar por -1 todos los coeficientes de la restricción.
 - Si no queremos restricciones de desigualdad, podemos reemplazar cada restricción ($\sum_{j=1}^n a_{ij}x_j = b_i$) por dos restricciones de desigualdad ($\sum_{j=1}^n a_{ij}x_j \leq b_i$ y $\sum_{j=1}^n a_{ij}x_j \geq b_i$).
 - Si sólo queremos restricciones de igualdad, podemos reemplazar cada restricción de la forma $\sum_{j=1}^n a_{ij}x_j \geq b_i$ por $\sum_{j=1}^n a_{ij}x_j - x_i^s = b_i$ donde x_i^s es una nueva variable ($x_i^s \geq 0$). Dichas variables se conocen como variables de holgura asociadas a la restricción que han sido inducidas. Si la restricción fuese $\sum_{j=1}^n a_{ij}x_j \leq b_i$ bastaría con sumar la variable en vez de restarla.
- **No negatividad:** Si tengo una variable x_j que puede tomar valores positivos y negativos y quiero una representación en la que sólo haya variables no negativas, puedo reemplazar todas las apariciones de x_j en el modelo por $x'_j - x''_j$ donde $x'_j \geq 0$ y $x''_j \geq 0$. Estas variables auxiliares representarían, en cierto modo, la parte positiva y negativa de x_j . Si en el problema original quiero que x_j valga 7, puedo hacerlo con $x'_j = 7$ y $x''_j = 0$. Del mismo modo, si quiero que x_j valga -7 , puedo hacerlo con $x'_j = 0$ y $x''_j = -7$.

Cabe destacar dos formas en las que se nos puede presentar dichos problemas y que nos serán especialmente útiles:

- **Forma Estándar.** Todas las restricciones son de igualdad y todas las variables son no negativas.

$$\begin{array}{ll}
 \text{Minimizar} & \mathbf{c}\mathbf{x} \\
 \text{Sujeto a} & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{array}
 \qquad
 \begin{array}{ll}
 \text{Maximizar} & \mathbf{c}\mathbf{x} \\
 \text{Sujeto a} & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{array}$$

- **Forma Canónica.** En caso de que el problema sea de minimización, todas las restricciones son de la forma " \geq " y, si es de maximización, son de la forma " \leq ". En ambos casos, las variables han de ser no negativas.

$$\begin{array}{ll}
 \text{Minimizar} & \mathbf{c}\mathbf{x} \\
 \text{Sujeto a} & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{array}
 \qquad
 \begin{array}{ll}
 \text{Maximizar} & \mathbf{c}\mathbf{x} \\
 \text{Sujeto a} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{array}$$

Ejemplo 1.1. Vamos a ver uno de los ejemplos más importantes en lo que a la programación lineal se refiere, y es conocido como

El problema del transporte

La empresa americana Netflix se dedica principalmente al streaming de películas y series por Internet. La empresa dispone de una red de m servidores S_1, \dots, S_m desde los que tiene que enviar datos a cada uno de los n clientes C_1, \dots, C_n . La empresa desea realizar esta operación lo más eficientemente posible. Para ello, la empresa dispone de la tasa de transmisión entre cualquier servidor S_i y cada cliente C_j . En particular, t_{ij} denota el tiempo necesario para transmitir un megabyte entre i y j .

La empresa trabaja en intervalos de tiempo de longitud fija. Al final de cada intervalo la empresa conoce, para el siguiente intervalo, la capacidad de cada servidor, s_i , y la demanda de datos esperada de cada cliente, d_j . En base a esta información ha de decidir cómo distribuir la información entre servidores y clientes de tal manera que minimice el tiempo empleado por sus servidores.

Entonces vamos a identificar los datos que tenemos:

- **Variables de decisión:** Se corresponden con la cantidad de megabytes que enviamos entre cada servidor i y cada cliente j , x_{ij} .
- **Función objetivo:** Queremos minimizar el tiempo empleado por los servidores para enviar dicha información.

- **Restricciones:** Hemos de respetar la capacidad de cada uno de los servidores y satisfacer la demanda de cada uno de los clientes. Además, estaremos también sujetos a restricciones de no negatividad en nuestras variables.

Una vez dicho esto podemos expresar nuestro problema de la siguiente manera:

$$\begin{aligned}
 \text{Minimizar} \quad & \sum_{i=1}^m \sum_{j=1}^m t_{ij} x_{ij} \\
 \text{Sujeto a} \quad & \sum_{j=1}^m x_{ij} \leq s_i \quad i \in \{1, 2, \dots, m\} \\
 & \sum_{i=1}^m x_{ij} = d_j \quad j \in \{1, 2, \dots, n\} \\
 & x_{ij} \geq 0 \quad i \in \{1, 2, \dots, m\}, \quad j \in \{1, 2, \dots, n\}.
 \end{aligned}$$

1.2. Método símplex

1.2.1. Introducción

El método símplex es un algoritmo creado por George Dantzig y consiste en estudiar de modo sucesivo los distintos puntos extremos de la región factible. Dichos puntos serán muy importantes para la resolución, por lo tanto deberemos tener un gran control sobre ellos. Recordemos que un punto extremo es aquel que no se puede poner como un punto medio entre un par de puntos distintos. Es necesario recordar que el método símplex trabaja con problemas de forma estándar, pero cualquier problema de programación lineal se puede transformar en un problema de la siguiente forma, mediante las transformaciones vistas anteriormente.

$$\begin{aligned}
 \text{Minimizar} \quad & \mathbf{c}\mathbf{x} \\
 \text{Sujeto a} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{aligned} \tag{1.1}$$

Suponemos que \mathbf{A} es una matriz $m \times n$ de rango m y por lo tanto no hay restricciones redundantes. En este caso la región factible sería $S = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Es necesario conocer unas cuantas consideraciones relativas al método símplex:

- El esquema general del algoritmo parte de una solución factible. Cómo conseguir dicha solución es algo que analizaremos en la sección 1.2.3.
- El procedimiento de recálculo al final de cada iteración se conoce como pivotado y las variables de entrada y salida se conocen como elementos de pivote.

- El criterio de entrada consiste en que aquella variable para la cual, por cada unidad en que se incremente su valor, se obtenga una mayor mejora en la función objetivo.
- Un punto mucho más relevante para la eficiencia computacional del algoritmo tiene que ver con el cálculo de la inversa \mathbf{B}^{-1} . Debido a la complejidad de la operación por el método de Gauss-Jordan $O(m^3)$, el impacto de estas operaciones no será despreciable. Sin embargo, existe otro método mediante las denominadas tablas del simplex que hace que estas operaciones sean mucho menos costosas $O(m^2)$.
- En el caso de que existan varias soluciones el método simplex permite identificarlas todas.

A continuación presentamos un *algoritmo* con el que podremos resolver este tipo de problemas en unos sencillos pasos pero antes, para lograr la comprensión de este *algoritmo* será necesario definir unos conceptos previos.

1.2.2. Bases y Soluciones Básicas

El método simplex trabaja con problemas en forma estándar como el expuesto en la ecuación (1.1). Supongamos que \mathbf{A} es una matriz $m \times n$ de rango m , es decir, todas las filas de la matriz son linealmente independientes, lo que implica que no hay restricciones redundantes. Por otro lado, al tener el problema formulado en forma estándar, en toda solución factible habrá al menos m restricciones saturadas provenientes de $\mathbf{Ax} = \mathbf{b}$. Dado un punto extremo x de S (Cabe recordar que S sería la región básica factible, $S = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$), necesariamente habrá $(n - m)$ restricciones de no negatividad que se saturarán en x , para así tener un total de $m + (n - m) = n$ restricciones saturadas. Por tanto, tenemos el siguiente resultado :

Proposición 1.2. *Consideremos un problema de programación lineal en forma estándar con región factible donde $S = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ tiene rango m . Entonces, en todo punto extremo x de S hay al menos $(n - m)$ variables que toman el valor 0.*

Este resultado es la clave de la definición del concepto de solución básica. Consideremos el sistema $\mathbf{Ax} = \mathbf{b}$ y $\mathbf{x} \geq 0$ donde $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $\mathbf{b} \in \mathbb{R}^m$ son tales que $\text{rango}(\mathbf{A}) = \text{rango}(\mathbf{A}, \mathbf{b}) = m$. La matriz \mathbf{A} puede verse como $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$, donde $\mathbf{B} \in \mathbb{R}^{m \times m}$ y $\mathbf{N} \in \mathbb{R}^{m \times (n-m)}$. Supongamos además que \mathbf{B} tiene rango m y por tanto es invertible. Entonces, la solución $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$ al sistema $\mathbf{Ax} = \mathbf{b}$ y $\mathbf{x} \geq 0$ dada por

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \text{ y } \mathbf{x}_N = 0$$

se denomina *solución básica*. Además, si $\mathbf{x}_B \geq 0$, se llama *solución básica factible*.

Resumen del Método Simplex

Los pasos a seguir para resolver un problema serán los siguientes

▪ INICIALIZACIÓN

Partimos de una base \mathbf{B} que tiene asociada una solución básica factible $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$, con $x_B = \mathbf{B}^{-1}\mathbf{b}$ y $x_N = 0$. Además, $z_B = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b}$.

▪ PASO 1 (Criterio de entrada)

Calculamos los valores $z_j - c_j$ para todas las variables no básicas $j \in J$, donde $z_j = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A}_j^c = \mathbf{c}_B\mathbf{y}^{j*}$. Sea $k \in J$ el menor índice tal que $z_k - c_k = \max_{j \in J}\{z_j - c_j\}$, tenemos las siguientes posibilidades:

- Si $z_k - c_k \leq 0$, FIN. La solución básica factible $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$ es óptima.
- Si $z_k - c_k \geq 0$, ir al PASO 2 con k como variable de entrada.

▪ PASO 2 (Criterio de salida)

Calculamos $\mathbf{y}^k = \mathbf{B}^{-1}\mathbf{A}_k^c$. Tenemos las siguientes posibilidades:

- Si $\mathbf{y}^k \leq \mathbf{0}$, Fin. El problema no tiene óptimo finito. Podemos movernos indefinidamente a lo largo del rayo

$$R = \{(\mathbf{x}_B, \mathbf{x}_N) + (-\mathbf{y}^k, \mathbf{e}^k)x_k : x_k \geq 0\},$$

cuyos puntos tienen todas las componentes no negativas, por lo que serán soluciones factibles. A medida que $x_k \rightarrow +\infty$, tenemos que $z_B - (z_k - c_k)x_k$ tiende a $-\infty$.

- Si $\mathbf{y}^k \not\leq \mathbf{0}$ entonces \mathbf{A}_k^c entra en la base y x_k pasa a ser una variable básica. Abandona la base la variable i , aquella con el índice más bajo de entre las que primero se anulan en el rayo R :

$$\frac{(x_B)_i}{y_i^k} = \min\left\{\frac{(x_B)_l}{y_l^k} : y_l^k > 0\right\}$$

Recalcular \mathbf{B} , $\mathbf{x} = (x_B, x_N)$, z_B y los vectores \mathbf{y}^j . Volver al PASO 1.

Vamos a ver un ejemplo sencillo de este método expuesto anteriormente siguiendo las pautas anteriores.

Ejemplo 1.3. Para terminar este apartado vamos a presentar un ejemplo de la aplicación

del método símplex.

$$\begin{aligned} \text{Minimizar} \quad & -4x_1 - 2x_2 + 6x_3 \\ \text{Sujeto a} \quad & -x_1 + x_2 + 2x_3 \leq 8 \\ & +6x_1 + x_2 + 7x_3 \leq 6 \\ & -5x_1 + \quad + 6x_3 \leq 1 \\ & \mathbf{x} \geq 0. \end{aligned}$$

Lo primero que debemos tener en mente es pasar este problema a forma estándar:

$$\begin{aligned} \text{Minimizar} \quad & -4x_1 - 2x_2 + 6x_3 \\ \text{Sujeto a} \quad & -x_1 + x_2 + 2x_3 + x_4^s = 8 \\ & +6x_1 + x_2 + 7x_3 + x_5^s = 6 \\ & -5x_1 + \quad + 6x_3 + x_6^s = 1 \\ & \mathbf{x} \geq 0. \end{aligned}$$

Como vemos le hemos añadido tres variables de holgura para obtener la forma estándar, como todos los lados derechos son no negativos, esto dará lugar a una base natural, $\mathbf{B} = (A_4^c A_5^s A_6^c)$, obviamente es la matriz identidad y por tanto su inversa también lo es, entonces, la solución básica asociada son todos sus lados derechos, como hemos dicho antes, $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} = (8, 6, 1)$ y $\mathbf{x}_N = (0, 0, 0)$, que cumple las condiciones de no negatividad y por lo tanto es factible. Empezaremos a aplicar el algoritmo con esta base inicial.

- **Inicialización.** $\mathbf{B} = (A_4^c A_5^s A_6^c)$, $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$ con $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} = (8, 6, 1)$ y $\mathbf{x}_N = (0, 0, 0)$. Por último, $z_B = \mathbf{c}_B \mathbf{x}_B = (0, 0, 0)(8, 6, 1) = 0$.
- **Iteración 1. Paso 1 (Criterio de Entrada).** Calculamos los $z_j - c_j$:
 - $\mathbf{B}^{-1}\mathbf{A}_1^c = \mathbf{y}^1 = (-1, 6, -5)$. Por tanto $z_1 - c_1 = \mathbf{c}_B \mathbf{y}^1 - c_1 = 0 - (-4) = 4 > 0$.
 - $\mathbf{B}^{-1}\mathbf{A}_2^c = \mathbf{y}^2 = (1, 1, 0)$. Por tanto $z_2 - c_2 = \mathbf{c}_B \mathbf{y}^2 - c_2 = 0 - (-2) = 2 > 0$.
 - $\mathbf{B}^{-1}\mathbf{A}_3^c = \mathbf{y}^3 = (2, 7, 6)$. Por tanto $z_3 - c_3 = \mathbf{c}_B \mathbf{y}^3 - c_3 = 0 - (6) = -6 < 0$.

El máximo se alcanza para $j = 1$ y es mayor que cero. Por tanto, tenemos que x_1 es la candidata a entrar en la base.

- **Iteración 1. Paso 2 (Criterio de salida)** El vector \mathbf{y}^1 es $(-1, 6, -5)$, que sólo tiene una componente positiva, asociada con la variable x_5^s que saldrá de la base. La nueva base será $\mathbf{B} = (A_4^c A_1^c A_6^c)$

$$\mathbf{B} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 6 & 0 \\ 0 & -5 & 1 \end{pmatrix} \quad y \quad \mathbf{B}^{-1} = \begin{pmatrix} 1 & 0.17 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0.83 & 1 \end{pmatrix}.$$

Con lo que $x_B = B^{-1}b = (9, 1, 6)$, $x_N = (0, 0, 0)$ y $z_B = c_B x_B = (0, -4, 0)(9, 1, 6) = -4$.

■ **Iteración 2. Paso 1 (Criterio de entrada).** Calculamos los $z_j - c_j$:

- $B^{-1}A_2^c = y^2 = (1.17, 0.17, 0.83)$. Por tanto $z_2 - c_2 = c_B y^2 - c_2 = -0.67 - (-2) = 1.33 > 0$.
- $B^{-1}A_3^c = y^3 = (3.17, 1.17, 11.83)$. Por tanto $z_3 - c_3 = c_B y^3 - c_3 = -4.67 - (6) = -10.67 < 0$.
- $B^{-1}A_5^c = y^5 = (0.17, 0.17, 0.83)$. Por tanto $z_5 - c_5 = c_B y^5 - c_5 = -0.67 - (0) = -0.67 < 0$.

El máximo se alcanza para $j = 2$ y es el único mayor que cero. Por tanto, tenemos que x_2 es la candidata a entrar en la base.

■ **Iteración 2. Paso 2 (Criterio de Salida).** El vector y^2 es $(1.17, 0.17, 0.83)$, en el que tenemos que $\min\{\frac{9}{1.17}, \frac{1}{0.17}, \frac{6}{0.83}\} = \frac{1}{0.17}$, asociada con la variable x_1 que saldrá de la base. La nueva base será $B = (A_4^c A_2^c A_6^c)$.

$$B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad y \quad B^{-1} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Con lo que $x_B = B^{-1}b = (2, 6, 1)$, $x_N = (0, 0, 0)$ y $z_B = c_B x_B = (0, -2, 0)(2, 6, 1) = -12$.

■ **Iteración 3. Paso 1 (Criterio de entrada).** Calculamos los $z_j - c_j$:

- $B^{-1}A_1^c = y^1 = (-7, 6, -5)$. Por tanto $z_1 - c_1 = c_B y^1 - c_1 = -12 - (-4) = -8 < 0$.
- $B^{-1}A_3^c = y^3 = (-5, 7, 6)$. Por tanto $z_3 - c_3 = c_B y^3 - c_3 = -14 - (6) = -20 < 0$.
- $B^{-1}A_5^c = y^5 = (-1, 1, 0)$. Por tanto $z_5 - c_5 = c_B y^5 - c_5 = -2 - (0) = -2 < 0$.

El máximo se alcanza para $j = 5$ y es menor que cero. Por tanto, estamos ante una solución óptima.

La solución óptima, expresada únicamente con las variables del problema original, viene dada por $(0, 6, 0)$ y su coste asociado es -12 .

1.2.3. El método de la M grande

Es uno de los métodos más utilizados para la construcción de una solución inicial del método símplex. La idea del método de la M grande consiste en añadir variables artificiales en tantas restricciones como sea necesario, para que la nueva matriz de restricciones contenga a la matriz identidad como submatriz, lo que nos pone nuevamente en las condiciones del ejemplo resuelto anteriormente. Pero debemos tener en cuenta un detalle muy importante, puesto que, estas variables artificiales desempeñan un papel muy distinto al de las variables de holgura. Las variables artificiales aumentan el conjunto de soluciones factibles del problema original. Pero existe una manera de subsanar este pequeño inconveniente y es asumiendo que el coste de dichas variables viene dado por una constante M que sea tan grande como nosotros deseemos.

Ejemplo 1.4. Supongamos que tenemos el siguiente problema de programación lineal:

$$\begin{aligned} \text{Minimizar} \quad & 3x_1 + 3x_2 - 3x_3 + 4x_4 \\ \text{Sujeto a} \quad & -x_1 + x_2 + 6x_3 + \geq 0 \\ & +5x_1 + 2x_2 + 2x_3 - x_4 \geq 7 \\ & -x_1 + 3x_2 - x_3 + 2x_4 \leq 7 \\ & \mathbf{x} \geq 0. \end{aligned}$$

Al pasarlo a forma estándar obtendremos:

$$\begin{aligned} \text{Minimizar} \quad & +3x_1 + 3x_2 - 3x_3 + 4x_4 \\ \text{Sujeto a} \quad & -x_1 + x_2 + 6x_3 - x_5^s = 0 \\ & +5x_1 + 2x_2 + 2x_3 - x_4 - x_6^s = 7 \\ & -x_1 + 3x_2 - x_3 + 2x_4 + x_7^s = 7 \\ & \mathbf{x} \geq 0. \end{aligned}$$

Vemos que la última columna se corresponde con el vector $(0, 0, 1)$, pero todavía nos faltan las columnas $(1, 0, 0)$ y $(0, 1, 0)$ para tener que la matriz identidad es una submatriz de la matriz de restricciones. Además, en general no podemos multiplicar por -1 ninguna de las restricciones, pues con eso pasaríamos a tener componentes negativas en el vector de lados derechos. En este ejemplo concreto, como el lado derecho de la primera restricción es cero, sí que podríamos multiplicar por -1 dicha restricción para obtener la columna $(1, 0, 0)$. En cualquier caso, en este ejemplo presentaremos como haríamos en general, como si no tuviésemos que $b_1 = 0$. En dicho caso tendríamos que añadir dos variables artificiales, obteniendo:

$$\begin{aligned}
\text{Minimizar} \quad & 3x_1 + 3x_2 - 3x_3 + 4x_4 + \quad + \quad + Mx_8^a + Mx_9^a \\
\text{Sujeto a} \quad & -x_1 + x_2 + 6x_3 + \quad - x_5^s + \quad + x_8^a = 0 \\
& 5x_1 + 2x_2 + 2x_3 - x_4 \quad - x_6^s + \quad + x_9^a = 7 \\
& -x_1 + 3x_2 - x_3 + 2x_4 + \quad + x_7^s + \quad = 7 \\
& \mathbf{x} \geq 0.
\end{aligned}$$

Obteniendo un problema en el que la matriz de restricciones tiene una submatriz identidad, con lo que podremos aplicar el método símplex tomando como base inicial $\mathbf{B} = (\mathbf{A}_7^c, \mathbf{A}_8^c, \mathbf{A}_9^c)$ con solución asociada $x_B = (7, 0, 7)$ y $x_N = 0$, que es factible en el problema ampliado con las variables artificiales. Ahora habría que iterar el método símplex y ver si en la solución óptima todas las variables artificiales se han hecho cero o no para decidir si se ha encontrado una solución óptima del problema original o éste no tenía soluciones factibles. No presentamos aquí la resolución de este problema, donde lo resolvemos con la ayuda de las tablas del símplex del Ejemplo 1.6. En dicho apartado resolveremos un caso práctico de estas tablas, si quisiéramos una explicación más teórica podríamos consultar los apuntes de González-Díaz (2017).

1.2.4. La tabla del método símplex

La idea de la *tabla del método símplex* consiste en guardar en una tabla toda la información importante de cada iteración que llevemos a cabo. En el método símplex tenemos que estar calculando la matriz \mathbf{B}^{-1} en cada una de las iteraciones, lo cual lleva un gran trabajo computacional. Esta matriz es muy importante para la resolución de los sistemas $\mathbf{B}\mathbf{x}_B = \mathbf{b}$ y, especialmente, $\mathbf{B}\mathbf{y}^j = \mathbf{A}_j^c$, puesto que los vectores \mathbf{y}^j deben ser actualizados iteración a iteración. Para realizar un esquema ilustrativo de como podemos llegar a obtener dicha tabla, vamos suponer que estamos trabajando con una base \mathbf{B} y que las m columnas de \mathbf{B} se corresponden con las primeras m variables. Recordemos que \mathbf{N} denota la matriz formada por las columnas no básicas y usaremos $I_{m \times m}$ para denotar la submatriz identidad. Además será cómodo denotar $\mathbf{B}^{-1}\mathbf{b}$ como $\bar{\mathbf{b}}$. Entonces obtendremos la siguiente tabla:

		\mathbf{x}_B	\mathbf{x}_N	\bar{z}
$z_j - c_j$		$\mathbf{0}$	$\mathbf{c}_B \mathbf{B}^{-1} \mathbf{N} - \mathbf{c}_N$	$\mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}$
\mathbf{x}_B	\mathbf{c}_B	$I_{m \times m}$	$\mathbf{B}^{-1} \mathbf{N}$	$\mathbf{B}^{-1} \mathbf{b}$

Es necesario aclarar que $\mathbf{c}_B \mathbf{B}^{-1} \mathbf{B} - \mathbf{c}_B = \mathbf{0}$ y que $\mathbf{B}^{-1} \mathbf{B} = I_{m \times m}$. En forma un poco menos compacta tenemos:

		x_{B_1}	\cdots	x_{B_r}	\cdots	x_{B_m}	\cdots	x_j	\cdots	x_k	\cdots	\bar{z}
$z_j - c_j$		0	\cdots	0	\cdots	0	\cdots	$z_j - c_j$	\cdots	$z_k - c_k$	\cdots	$\mathbf{c}_B \bar{\mathbf{b}}$
\mathbf{x}_{B_1}	\mathbf{c}_{B_1}	1	\cdots	0	\cdots	0	\cdots	y_1^j	\cdots	y_1^k	\cdots	$\bar{\mathbf{b}}_1$
\vdots	\vdots	\vdots	\ddots	\vdots	\cdots	\vdots	\cdots	\vdots	\cdots	\vdots	\cdots	\vdots
\mathbf{x}_{B_r}	\mathbf{c}_{B_r}	0	\cdots	1	\cdots	0	\cdots	y_r^j	\cdots	y_r^k	\cdots	$\bar{\mathbf{b}}_r$
\vdots	\vdots	\vdots	\cdots	\vdots	\ddots	\vdots	\cdots	\vdots	\cdots	\vdots	\cdots	\vdots
\mathbf{x}_{B_m}	\mathbf{c}_{B_m}	0	\cdots	0	\cdots	1	\cdots	y_m^j	\cdots	y_m^k	\cdots	$\bar{\mathbf{b}}_m$

Supongamos entonces que decidimos que la variable k debe entrar en la base y la variable r debe salir de la misma. A continuación describimos las operaciones que habría que realizar sobre la tabla actual para obtener la tabla asociada a la nueva base y poder así iterar el algoritmo. Básicamente, lo que tenemos que hacer es:

- Actualizar los costes básicos (\mathbf{c}_B), variables básicas (\mathbf{x}_B) y sus valores ($\bar{\mathbf{b}}$).
- Actualizar los costes reducidos ($z_j - c_j$).
- Actualizar las columnas \mathbf{y}^j .

El procedimiento en general se conoce como *pivotado*, pues todos los cálculos pivotan alrededor del elemento y_r^k de la tabla, correspondiente a la fila r y a la columna k , y consiste en lo siguiente:

Coefficiente y_l^j . La actualización de la tabla para obtener $\mathbf{B}^{-1}\mathbf{N}$:

- Si $l = r$, y_r^j se actualiza a $\frac{y_l^j}{y_r^k}$.
- Si $l \neq r$, y_r^j se actualiza a $y_l^j - \frac{y_r^j \cdot y_l^k}{y_r^k}$.

Coefficiente \bar{b}_l . La actualización es análoga a la de los coeficientes y_l^j :

- Si $l = r$, \bar{b}_l se actualiza a $\frac{\bar{b}_l}{y_r^k}$.
- Si $l \neq r$, \bar{b}_l se actualiza a $\bar{b}_l - \frac{\bar{b}_r \cdot y_l^k}{y_r^k}$.

Costes reducidos $z_j - c_j$. De nuevo es análogo, se actualiza $(z_j - c_j) - \frac{y_r^j \cdot (z_k - c_k)}{y_r^k}$.

Función objetivo \bar{z} . Otra vez se actualiza de la misma manera $\bar{z} - \frac{\bar{b}_r \cdot (z_k - c_k)}{y_r^k}$.

Es importante destacar una característica importante de la tabla del simplex. Supongamos que queremos resolver un problema de programación lineal con la tabla del simplex y para ello introducimos las variables artificiales necesarias para partir de una base inicial que se corresponde con la identidad $I_{m \times m}$. Entonces, dado que en cada iteración las

columnas de la tabla del símplex representan las columnas de la matriz $\mathbf{B}^{-1}\mathbf{A}$, tendremos que la submatriz formada por las columnas de la base inicial se corresponderán con $\mathbf{B}^{-1}\mathbf{I}_{m \times m} = \mathbf{B}^{-1}$. Es decir, en todas las iteraciones estamos calculando implícitamente la matriz \mathbf{B}^{-1} .

1.2.5. Eficiencia del método

El funcionamiento del método símplex consiste en saltar de solución básica factible en solución básica factible hasta encontrar el óptimo. No es difícil construir ejemplos en los que toda solución básica es factible y en los que el método símplex tiene que recorrerlas todas para alcanzar el óptimo. Por tanto, el número de operaciones del método símplex dependerá del número de soluciones básicas o, equivalentemente, del número de bases. Para el método símplex uno puede construir ejemplos patológicos en los que el algoritmo recorre todos y cada uno de los puntos extremos de la región factible. Esto implica que, según el análisis del peor caso, el método símplex es un algoritmo exponencial. Como es el caso de los cubos de Klee-Minty que veremos con más detalle en el capítulo 2.

Ejemplos

Ejemplo 1.5. Empezamos esta sección resolviendo de nuevo el ejemplo 1.3, pero esta vez mediante las tablas del símplex. El problema a resolver era

$$\begin{aligned} \text{Minimizar} \quad & -4x_1 - 2x_2 + 6x_3 \\ \text{Sujeto a} \quad & -x_1 + x_2 + 2x_3 \leq 8 \\ & +6x_1 + x_2 + 7x_3 \leq 6 \\ & -5x_1 + \quad + 6x_3 \leq 1 \\ & \mathbf{x} \geq 0. \end{aligned}$$

Tras introducir las variables de holgura como ya hicimos en el ejemplo 1.3 llegamos a la siguiente tabla inicial del símplex:

Tabla 1 (entra x_1 , sale x_5^s)

		x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	\bar{z}
$z_j - c_j$		4	2	-6	0	0	0	0
	c_B							x_B
x_4^s	0	-1	1	2	1	0	0	8
x_5^s	0	6	1	7	0	1	0	6
x_6^s	0	-5	0	6	0	0	1	1

Tabla 2 (entra x_2 , sale x_1)

		x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	\bar{z}
$z_j - c_j$		0	1.33	-10.67	0	0.67	0	-4
	c_B							x_B
x_4^s	0	0	1.17	3.17	1	0.17	0	9
x_1	-4	1	0.17	1.17	0	0.17	0	1
x_6^s	0	0	0.83	11.83	0	0.83	1	6

Tabla 3 (óptima)

		x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	\bar{z}
$z_j - c_j$		8	0	-20	0	0	0	-12
	c_B							x_B
x_4^s	0	-7	0	-5	1	0	0	2
x_2	-2	6	1	7	0	1	0	6
x_6^s	0	-5	0	0	0	1	1	1

Ejemplo 1.6. La tabla del método símplex

Supongamos que tenemos el siguiente problema de programación lineal,

$$\begin{aligned}
 \text{Minimizar} \quad & 3x_1 + 3x_2 - 3x_3 + 4x_4 + \quad + \quad + Mx_8^a + Mx_9^a \\
 \text{Sujeto a} \quad & -x_1 + x_2 + 6x_3 + \quad - x_5^s + \quad + x_8^a = 0 \\
 & 5x_1 + 2x_2 + 2x_3 - x_4 - x_6^s + \quad + x_9^a = 7 \\
 & -x_1 + 3x_2 - x_3 + 2x_4 + \quad + x_7^s + \quad = 7 \\
 & \mathbf{x} \geq 0.
 \end{aligned}$$

Tabla 1

(entra x_1 , sale x_9^a)

		x_1	x_2	x_3	x_4	x_5^s	x_6^s	x_7^s	x_8^s	x_9^s	\bar{z}
$z_j - c_j$		$4M$	$3M$	$-4M$	$-M$	$-M$	$-M$				$7M$
	c_B										x_B
x_7^s	0	-1	3	-1	2	0	0	1	0	0	7
x_8^a	M	-1	1	-6	0	-1	0	0	1	0	0
x_9^a	M	5	2	2	-1	0	-1	0	0	1	7

Tabla 2

(entra x_2 , sale x_8^a)

		x_1	x_2	x_3	x_4	x_5^s	x_6^s	x_7^s	x_8^s	x_9^s	\bar{z}
$z_j - c_j$			$1.4M$	$-5.6M$	$-0.2M$	$-M$	$-0.2M$			$-0.8M$	$1.4M$
	c_B										x_B
x_7^s	0	0	3.4	-0.6	1.8	0	-0.2	1	0	0.2	8.4
x_8^a	M	0	1.4	-5.6	-0.2	-1	-0.2	0	1	0.2	1.4
x_1	3	1	0.4	0.4	-0.2	0	-0.2	0	0	0.2	1.4

Tabla 3

(entra x_4 , sale x_7^s)

		x_1	x_2	x_3	x_4	x_5^s	x_6^s	x_7^s	x_8^s	x_9^s	\bar{z}
$z_j - c_j$									$-M$	$-M$	
	c_B										x_B
x_7^s	0	0	0	13	2.29	2.43	0.29	1	-2.43	-0.29	5
x_2	3	0	1	-4	-0.14	-0.71	-0.14	0	0.71	0.14	1
x_1	3	1	0	2	-0.14	0.29	-0.14	0	0.29	0.14	1

Tabla 4

(óptima)

		x_1	x_2	x_3	x_4	x_5^s	x_6^s	x_7^s	x_8^s	x_9^s	\bar{z}
$z_j - c_j$									$-M$	M	
	c_B										x_B
x_4	-4	0	0	5.69	1	1.06	0.13	0.44	-1.06	-0.13	2.19
x_2	3	0	1	-3.19	0	-0.56	-0.12	0.06	0.56	-0.12	1.31
x_1	3	1	0	2.81	0	0.44	-0.12	0.06	-0.44	0.12	1.31

1.3. Dualidad

A cualquier problema de programación lineal le podemos asociar un nuevo problema de programación lineal, conocido como problema dual. Vamos a comenzar exponiendo la formulación dual de un problema de programación lineal en forma estándar:

Problema primal (P)		Problema dual (D)	
Minimizar	$c\mathbf{x}$	Maximizar	$b\boldsymbol{\pi}$
Sujeto a	$\mathbf{Ax} = \mathbf{b}$	Sujeto a	$\mathbf{A}^t\boldsymbol{\pi} \leq \mathbf{c}$
	$\mathbf{x} \geq \mathbf{0}$.		$\boldsymbol{\pi}$ sin restr.

y ahora en forma canónica:

Problema primal (P)		Problema dual(D)	
Minimizar	$c\mathbf{x}$	Maximizar	$b\boldsymbol{\pi}$
Sujeto a	$\mathbf{Ax} \geq \mathbf{b}$	Sujeto a	$\mathbf{A}^t\boldsymbol{\pi} \leq \mathbf{c}$
	$\mathbf{x} \geq \mathbf{0}$.		$\boldsymbol{\pi} \geq \mathbf{0}$.

Después de haber visto estas dos formulaciones tanto del problema primal como del problema dual podremos sacar unas cuantas conclusiones:

- Si el primal es un problema de minimización, entonces el dual lo es de maximización y viceversa.
- En el problema primal trabajamos con la matriz \mathbf{A} , mientras que en el dual lo hacemos con la matriz traspuesta \mathbf{A}^t . Es decir, en vez de \mathbf{Ax} tendremos $\boldsymbol{\pi}\mathbf{A} = \mathbf{A}^t\boldsymbol{\pi}$.
- Cada variable del primal se corresponde con una restricción del dual.
- Cada restricción del primal se corresponde con una variable del dual.
- El vector \mathbf{b} el cual es un vector de lados derechos en el primal pasar a ser un vector de costes en el dual.
- El vector de costes del primal \mathbf{c} es el vector de lados derechos del dual.
- Las variables no restringidas pasa a restricciones de igualdad y las variables no negativas pasan a restricciones de menor o igual.

- Restricciones de igualdad pasan a variables no restringidas y restricciones de mayor o igual pasan a variables no negativas.

Al ver todas estas relaciones entre los dos problemas podemos elaborar un Cuadro 1.1 que nos facilite la obtención del problema dual a partir de su primal.

Una posible interpretación del dual podría ser la siguiente. En general un problema primal en forma canónica se puede ver como un problema de minimizar costes sujetos a una producción mínima. En este caso, el problema dual consiste en maximizar beneficios sujetos a unos precios máximos.

Primal		Dual	
f. objetivo	minimizar	maximizar	f. objetivo
restricciones	$\geq b_i$	≥ 0	variables
	$\leq b_i$	≤ 0	
	$= b_i$	no restr.	
variables	≥ 0	$\leq c_j$	restricciones
	≤ 0	$\geq c_j$	
	no restr.	$= c_j$	

Cuadro 1.1: Transformando un problema en su dual.

Ejemplo 1.7. Vamos a considerar el siguiente problema de programación lineal y para eso vamos a tomar el un problema donde podamos observar la mayoría de las transformaciones para obtener su respectivo problema dual.

$$\begin{aligned}
 &\text{Minimizar} && -4x_1 - 6x_2 + 6x_3 + 4x_4 \\
 &\text{Sujeto a} && -x_1 + \quad + 4x_3 - 5x_4 \leq 4 \\
 &&& -2x_1 - 4x_2 - 8x_3 - 8x_4 \geq 6 \\
 &&& +2x_1 + 4x_2 + 8x_3 + 3x_4 = 7 \\
 &&& \mathbf{x} \geq 0.
 \end{aligned}$$

Para dicho problema, y apoyandonos en el Cuadro 1.1, podremos obtener su dual

$$\begin{array}{ll}
 \text{Maximizar} & + 4x_1 + 6x_2 + 7x_3 \\
 \text{Sujeto a} & - \pi_1 - 2\pi_2 + 2\pi_3 \leq -4 \\
 & + \quad - 4\pi_2 + 4\pi_3 \leq -6 \\
 & + 4\pi_1 - 8\pi_2 + 8\pi_3 \leq 6 \\
 & - 5\pi_1 - 8\pi_2 + 3\pi_3 \leq 4 \\
 & \pi_1 \leq 0 \\
 & \pi_2 \geq 0 \\
 & \pi_3 \text{ sin restr.}
 \end{array}$$

Para acabar de comprender este problema debemos tener en cuenta que la matriz del problema primal es \mathbf{A} , mientras que la matriz del dual es \mathbf{A}^t . Observamos la diferencia calculándolas,

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 4 & -5 \\ -2 & -4 & -8 & -8 \\ 2 & 4 & 8 & 3 \end{pmatrix} \quad y \quad \mathbf{A}^t = \begin{pmatrix} -1 & -2 & 2 \\ 0 & -4 & 4 \\ 4 & -8 & 8 \\ -5 & -8 & 3 \end{pmatrix}.$$

Proposición 1.8. *El dual del dual coincide con el primal.*

Demostración. La demostración se realiza calculando el dual dos veces consecutivamente y observando que obtenemos un resultado que coincide con el problema inicial. \square

Dualidad débil y dualidad fuerte

Para presentar este apartado vamos a comenzar por enunciar la propiedad conocida como *dualidad débil* en la siguiente proposición.

Proposición 1.9 (Dualidad débil). *Dado un par de problemas de duales (P) y (D), la función objetivo asociada a cualquier solución factible del problema de minimización es mayor o igual que la función objetivo asociada a cualquier problema de maximización.*

Demostración. Tomando como referencia un problema primal y otro dual en forma canónica y que \mathbf{x} y $\boldsymbol{\pi}$ son soluciones factibles del primal y del dual, respectivamente. $\mathbf{A}^t\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{A} \geq \mathbf{c}$ y $\boldsymbol{\pi} \geq 0$. De estas afirmaciones podemos obtener la siguiente expresión:

$$\mathbf{c}\mathbf{x} \geq (\boldsymbol{\pi}\mathbf{A})\mathbf{x} = \boldsymbol{\pi}(\mathbf{A}\mathbf{x}) \geq \boldsymbol{\pi}\mathbf{b}.$$

\square

A partir de esta proposición podemos deducir dos propiedades que son muy importantes a la hora de desarrollar algoritmos. La primera de ellas es que las soluciones factibles del problema de minimización siempre nos darán cotas superiores para el problema de maximización y, análogamente, las soluciones factibles del problema de maximización siempre nos darán cotas inferiores para el problema de minimización. Además, la propiedad de dualidad débil nos permite obtener los dos siguientes corolarios.

Corolario 1.10. *Si \mathbf{x} y $\boldsymbol{\pi}$ son soluciones factibles del primal y del dual tales que $\mathbf{c}\mathbf{x} = \boldsymbol{\pi}\mathbf{b}$, entonces \mathbf{x} y $\boldsymbol{\pi}$ son soluciones óptimas de sus respectivos problemas.*

Corolario 1.11. *Dado un par de problemas duales (P) y (D), si la función objetivo de uno de ellos es no acotada, entonces el otro no tiene soluciones factibles.*

Para terminar esta sección vamos a enunciar la propiedad de dualidad fuerte, cuya demostración no veremos ya que es bastante laboriosa y no tiene demasiado interés con el tema que tratamos en esta memoria, puede verse en González-Díaz (2017).

Proposición 1.12 (Dualidad fuerte). *Dado un par de problemas duales (P) y (D), si uno de ellos tiene una solución óptima, entonces también el otro tiene solución óptima y los valores óptimos de la función objetivo coinciden. En particular, si $\bar{\mathbf{x}}$ y $\bar{\boldsymbol{\pi}}$ son soluciones óptimas de (P) y (D), respectivamente, tenemos que*

$$\mathbf{c}\bar{\mathbf{x}} = \bar{\boldsymbol{\pi}}\mathbf{b}.$$

Cabe destacar que a diferencia de la propiedad de dualidad débil, la propiedad de dualidad fuerte no se tiene que cumplir necesariamente para problemas no lineales. Por último vamos a presentar un resultado que nos permite calcular la solución óptima del dual con una base óptima del primal.

Proposición 1.13. *Supongamos que el problema primal viene dado en forma canónica y que tiene a $\bar{\mathbf{x}}$ como solución óptima con base asociada \mathbf{B} . Entonces,*

$$\bar{\boldsymbol{\pi}} = \mathbf{c}_B \mathbf{B}^{-1}.$$

es una solución óptima del problema dual.

Capítulo 2

Introducción a la Complejidad Computacional y Cubos de Klee-Minty

La información que redactamos en este capítulo esta basada fundamentalmente en el libro Bazaraa et al. (2011) y en los apuntes de programación lineal y entera de González-Díaz (2017).

2.1. Complejidad computacional

En este capítulo se analizarán algunos temas fundamentales sobre la eficiencia de la complejidad computacional, es decir, algoritmos polinómico-temporales capaces de resolver problemas de programación lineal de una manera eficiente, especialmente la complejidad computacional del algoritmo símplex, ya que presenta un crecimiento exponencial en esfuerzo en ciertas clases de problemas cuando crece el tamaño del problema. Por fortuna existen algunos resultados teóricos de caso promedio que ayudan a resolver la discrepancia entre los resultados teóricos del peor de los casos y la eficiencia práctica observada del algoritmo símplex. Sin embargo, existen algoritmos teóricamente eficientes para problemas de programación lineal que muestran un crecimiento en el esfuerzo de solución que es polinómico en el tamaño del problema. Existe un algoritmo bastante conocido que cae en esta categoría, el de algoritmo de Karmarkar del cuál hablaremos en el siguiente capítulo. Supongamos el siguiente problema de optimización de programación lineal

$$\text{Minimizar}\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

en donde \mathbf{A} es una matriz de $m \times n$, y $m, n \geq 2$. Aquí y a lo largo de todo este capítulo se supone que todos los datos son enteros, quizá transformados a esa forma a partir de datos racionales. Por un caso de este problema se entiende un elemento particular de esta clase de problemas con valores o datos paramétricos específicos. Así, la especificación de los valores para $m, n, \mathbf{c}, \mathbf{A}$, y \mathbf{b} define un caso de un problema de programación lineal. El tamaño de un caso de este problema se representa por los parámetros (m, n, L) , donde L es el número de bits binarios requeridos para registrar todos los datos del problema, y se denomina longitud de entrada de un caso del problema. Obsérvese que el número de bits binarios requeridos para representar un entero positivo Δ es $\lceil \log(1 + \Delta) \rceil$, en donde $\lceil \cdot \rceil$ denota el valor redondeado y $\log(\cdot)$ representa el logaritmo en base 10. Por ejemplo, a fin de representar cualquier entero $\Delta \in (2^r, 2^{r+1})$ para un entero $r \geq 1$, se requieren $(r + 1)$ bits binarios. Si Δ es negativo o positivo, es posible usar un bit adicional para registrar su signo, y en consecuencia es posible representar Δ usando $1 + \lceil \log(1 + |\Delta|) \rceil$ bits. Por consiguiente, los datos del programa lineal anterior se pueden registrar mediante un *esquema de codificación binaria* usando el siguiente número de bits:

$$L = [1 + \log(1 + m)] + [1 + \log(1 + n)] + \sum_j \{1 + \log(1 + |c_j|)\} \\ + \sum_i \sum_j \{1 + \log(1 + |a_{ij}|)\} + \sum_i \{1 + \log(1 + |b_j|)\}.$$

Consideremos un algoritmo que resuelva un problema de programación lineal en un número finito de pasos. Para analizar su complejidad computacional, es necesario determinar una cota superior del esfuerzo requerido para resolver cualquier caso de este problema. Este esfuerzo puede calcularse en términos del número de operaciones elementales como sumas, multiplicaciones y comparaciones requeridas para resolver el problema como una función del tamaño de dicho problema. Afortunadamente, un conteo exacto del número total de operaciones es innecesario. Sólo se requiere determinar una función $g(m, n, L)$ en términos del tamaño del problema, de modo que para alguna constante $\delta \geq 0$ suficientemente grande, el número total de operaciones elementales requeridas por el algoritmo para resolver el problema no sea mayor que $\delta g(m, n, L)$. Cuando la función $g(m, n, L)$ es un polinomio en m, n y L , se dice que el algoritmo es de *complejidad polinómica*. Por ejemplo, si un algoritmo implica realmente un máximo de $6m^2n + 15mn + 12m$ operaciones elementales para resolver cualquier caso de alguna clase de problemas, en donde el tamaño del problema esta determinado por m y n , entonces podemos decir simplemente que es de complejidad $O(m^2n)$. Así, basta averiguar que el crecimiento en esfuerzo para el algoritmo es dictado de manera dominante por la función $g(m, n) = m^2n$, ya que es la función que crece más rápido. Como sólo es necesario determinar una cota superior del esfuerzo de solución, entonces al

definir el tamaño de un caso de un problema no es necesario calcular L , es decir, si L se toma como alguna cota inferior del número real de bits requeridos para registrar los datos y presentar un crecimiento polinómico en esfuerzo como una función monótona creciente de esta cantidad, entonces lo anterior también debe implicar una cota polinómica del esfuerzo en términos del tamaño real del problema. Así, por ejemplo L puede tomarse como la siguiente cantidad en el análisis de problemas de programación lineal.

$$L = 1 + \log(m) \log(n) + \sum_j \{1 + \log(1 + |c_j|)\} \quad (2.1)$$

$$+ \sum_i \sum_j \{1 + \log(1 + |a_{ij}|)\} + \sum_i \{1 + \log(1 + |b_i|)\}.$$

La obsesión con un algoritmo polinómico-temporal puede apreciarse bien al comparar el crecimiento n^2 contra 2^n . Sea n el que representa el tamaño del problema, y sean A_1 y A_2 dos algoritmos para resolver este problema. Además supongamos que el algoritmo A_1 es de complejidad $O(n^2)$ y el algoritmo A_2 es de complejidad $O(2^n)$. Lo anterior significa que existen constantes δ_1 y δ_2 tales que los algoritmos están acotados por las expresiones $\delta_1 n^2$ y $\delta_2 2^n$ respectivamente. Por consiguiente para un problema de tamaño $n = 50$, suponiendo que $\delta_1 = \delta_2 = 1$, aunque el algoritmo A_1 efectuará no más de 2500 operaciones, el número de operaciones requeridas por el algoritmo A_2 puede llegar hasta 1.1259×10^{15} . Por lo que con este criterio de evaluación, primero se prefieren polinomios de orden bajo. Es obvio de que el algoritmo polinómico-temporal de complejidad $O(n^{50})$ no es de mayor valor práctico que el algoritmo A_2 . Segundo, sería deseable que las magnitudes de los coeficientes del polinomio fuesen relativamente pequeñas. Si $\delta_1 = 2^{50}$ y $\delta_2 = 1$, entonces aunque el esfuerzo con el algoritmo A_2 aumenta considerablemente más rápido que con un incremento en n que con el algoritmo A_1 , desde un punto de vista práctico el algoritmo A_1 es inmanipulable para cualquier valor de $n \geq 1$. Sin embargo, el algoritmo A_2 es manejable por lo menos para valores suficientemente pequeños de n . Merece la pena observar el rol desempeñado por el esquema de codificación binaria en la definición de un algoritmo polinómico-temporal. Supongamos que para resolver problemas de programación lineal se ha propuesto un algoritmo de complejidad $O(mn\Delta)$, en donde Δ es el coeficiente mayor valor absoluto de \mathbf{c} , \mathbf{A} y \mathbf{b} . Entonces ese algoritmo no es de complejidad polinómica. De hecho, al definir $L_1 = \log(\Delta)$ para representar el tamaño del problema, se observa que al algoritmo es de complejidad exponencial de orden $O(mn2^{L_1})$. Supongamos que se ha decidido medir el tamaño del problema mediante un esquema en el que se utiliza un pulsación para cada unidad de datos, llamado esquema de *codificación por pulsación*. Para almacenar el entero $\Delta > 0$, se requieren Δ pulsaciones, de modo que el tamaño del problema estará determinado por la tripleta (m, n, Δ) y no por $(m, n, \log \Delta)$. En este caso, el esfuerzo

algorítmico está efectivamente acotado superiormente por un polinomio (m, n, Δ) .

2.1.1. Clases de complejidad

A medida que el campo de investigación de operaciones continuó su desarrollo después de la aparición de la programación lineal, se introdujo una rica colección de varios tipos de problemas. En consecuencia surgió la necesidad de poder clasificar y comparar formalmente problemas y algoritmos en cuanto a su tratabilidad computacional.

A la hora de trabajar con algoritmos tenemos que tener claro que debemos de saber como de bueno es un algoritmo dado. Si es efectivo resolviendo problemas que es para lo que ha sido diseñado, y para resolver estas preguntas es donde surge la teoría de la complejidad computacional. La finalidad de acotar esa cantidad de operaciones, que el algoritmo necesitará en función de los datos necesarios para especificar el problema. Diremos que una clase de problemas de optimización es fácil si se puede desarrollar un algoritmo que resuelva cualquier problema en la clase en tiempo polinomial. Los algoritmos en tiempo polinomial se llaman algoritmos eficientes. Hay muchos problemas de programación que no se sabe si son fáciles o no. Es decir, no se sabe si existe algún algoritmo eficiente que los resuelva.

Clases P , NP , NP -completo, NP -duro

Ahora vamos a definir las distintas clases de complejidad (existen otras clases pero estas son las importantes). Diremos que un problema pertenece a la clase de complejidad P si existe un algoritmo que resuelve cualquier tipo de problema en tiempo polinomial. Diremos que un problema pertenece a la complejidad NP si existe un algoritmo que puede verificar cualquier solución a un ejemplo en dicha clase en tiempo polinomial. La gran pregunta de la teoría de complejidad computacional es si las dos clases que acabamos de definir coinciden. En el caso de que fueran iguales, tendría una gran repercusión, ya que existirían algoritmos para la factorización de números primos lo que que comprometería seriamente muchos protocolos de seguridad. En el estudio de las distintas clases de complejidad resulta de especial utilidad transformar unos problemas en otros. Supongamos que un problema P_1 es tal que cualquier ejemplo en dicha clase se puede transformar en tiempo polinomial en un ejemplo de la clase P_2 . En este caso, se dice que P_1 se reduce a P_2 en tiempo polinomial. Si la clase P_2 pertenece a P , entonces también la clase P_1 pertenece a P . Dado un elemento de P_1 lo podremos resolver en tiempo polinomial sin más que transformarlo previamente (en tiempo polinomial) en un elemento de P_2 . Este enfoque permite definir la clase de problemas NP -completos como los más difíciles de entre todos los de NP . Un problema P_1 es NP -completo si cualquier problema en NP se puede reducir a P_1 en tiempo

polinomial. Por tanto, dado un algoritmo para el problema P_1 , puede ser adaptado para resolver en un tiempo extra polinomial cualquier otro problema.

2.1.2. Complejidad computacional del método símplex

Cuando se presentó por primera vez el algoritmo símplex, la reacción intuitiva de la comunidad de investigadores fue que este algoritmo no iba a demostrar ser muy eficiente. Por su naturaleza, el método símplex se desliza sobre las aristas de un poliedro y no intenta penetrar en el interior de éste o pasar a caras de dimensión más alta. Sin embargo, los investigadores se sorprendieron agradablemente cuando en la práctica observaron que este método operaba de manera excelente. Para casi todos los problemas prácticos, empíricamente se ha observado que este método se lleva a cabo aproximadamente en $\frac{3}{2}$ iteraciones, y raras veces en más de $3m$ iteraciones, en donde la matriz de coeficientes $\mathbf{A} \in m \times n$. Sin embargo, queda el hecho de que el algoritmo está efectivamente atrapado en el aspecto potencialmente combinatorio de los $\binom{n}{m}$ vértices que el algoritmo puede recorrer posiblemente. Como

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} = \binom{n}{m} \binom{n-1}{m-1} \binom{n-2}{m-2} \cdots \left[\frac{n-(m-1)}{m-(m-1)} \right] > \left(\frac{n}{m} \right)^m,$$

que es por lo menos 2^m siempre que $n \geq 2m$, entonces el potencial de un orden de esfuerzo exponencial para algunos problemas es bastante plausible.

El temor de un esfuerzo exponencial con el método símplex se confirmó cuando produjeron una clase de problemas definidos por $m = n$ restricciones de igualdad en $2n$ variables no negativas, para las cuales el algoritmo símplex requiere $2^n - 1$ iteraciones, recorriendo todos los vértices del problema. Esta clase de problemas resulta de la definición de la región factible como una distorsión adecuada del hipercubo n -dimensional en E^n , que tiene 2^n vértices. Matemáticamente, el problema en E^n puede plantearse como sigue, en donde ϵ es algún número racional en el intervalo $(0, \frac{1}{2})$.

$$\begin{array}{ll} \text{Minimizar} & x_n \\ \text{Sujeto a} & 0 \leq x_1 \leq 1 \\ & \epsilon x_{j-1} \leq x_j \leq 1 - \epsilon x_{j-1} \quad \text{para } j = 2, \dots, n \\ & x_j \geq 0 \quad \quad \quad j = 1, \dots, n. \end{array}$$

Una vez que este problema se ha transformado de manera adecuada, se demuestra que, al introducir la variable no básica con el mayor coste reducido y empezando en el origen, el

algoritmo símplex recorre $2^n - 1$ aristas de este politopo y recorre todos los 2^n vértices. Al fin al plantear el problema en la forma deseada, considere una transformación lineal

$$y_1 = x_1, y_j = (x_j - \epsilon x_{j-1})/\epsilon^{j-1} \quad \text{para } j = 2, \dots, n.$$

Aplicando esta transformación y haciendo $\theta = 1/\epsilon$, se obtiene el siguiente problema equivalente.

$$\begin{aligned} \text{Maximizar} \quad & \sum_{j=1}^n y_j \\ \text{Sujeto a} \quad & y_1 \leq 1 \\ & y_{j-1} + 2 \sum_{k=1}^{j-1} y_k \leq \theta^{j-1} \quad \text{para } j = 2, \dots, n \\ & y_1, \dots, y_n \geq 0. \end{aligned} \tag{2.2}$$

Por conveniencia, s_1, \dots, s_n se definen como las variables de holgura asociadas con las n restricciones anteriores. Obsérvese que la solución básica con variables básicas $(s_1, \dots, s_{n-1}, y_n)$ es factible, produciendo $y = (0, \dots, 0, \theta^{n-1})$ con valor de la función objetivo θ^{n-1} . Lo anterior corresponde a la solución $x = (0, \dots, 0, 1)$ del problema original. Observando la restricción final y la función objetivo en el problema original, lo anterior corresponde claramente a la solución óptima de este problema.

2.2. Cubos de Klee-Minty

La exposición que aquí comienza se basa en los apuntes González-Díaz (2017) y en la Wikipedia (https://en.wikipedia.org/wiki/Klee%E2%80%93Minty_cube). Los cubos de Klee-Minty que llevan el nombre de Victor Klee y George J. Minty, porque fueron sus descubridores. Klee y Minty demostraron que el algoritmo símplex tiene un peor rendimiento en el peor caso cuando se inicializa en una esquina, es decir, el algoritmo recorra todos los puntos extremos antes de alcanzar el único óptimo (en el que $x_n = 1$ y todas las demás componentes son 0). Como dichos cubos tienen 2^n puntos extremos, es evidente que no es un algoritmo de tiempo polinomial, sino que muestra un comportamiento exponencial, puesto que, el algoritmo símplex atraviesa los bordes entre vértices hasta que alcanza un vértice óptimo. En el caso que se muestra, el algoritmo símplex toma cinco pasos. Sin embargo, el algoritmo símplex visita cada vértice en el peor caso de un problema cuya región factible es el cubo Klee-Minty, por lo que el número de pasos aumenta exponencialmente con la dimensión del problema. El cubo Klee-Minty se especificó originalmente con un sistema parametrizado de desigualdades lineales, con la dimensión como parámetro. Cuando la dimensión es dos, el “cubo” es un cuadrado aplastado. Cuando la dimensión es

tres, el “cubo” es un cubo aplastado. Para la resolución de dichos problemas los mejores algoritmos que se conocen requieren, en el peor caso, un número de iteraciones del orden de $O(2p^{\sqrt{m}})$, donde m es, como siempre, el número de restricciones del problema, y p el tipo de politopo de Klee-Minty que tenemos (varía segundo la cantidad de puntos extremos que tenemos). Sin embargo, este tipo de preguntas, aún siendo interesantes desde el punto de vista teórico, no tienen gran importancia práctica. Lo que es importante mencionar que sí se conocen algoritmos polinomiales para resolver problemas de programación lineal, entre los que destaca el algoritmo de punto interior de Karmarkar, que analizaremos a continuación.

Para tener una mejor percepción vamos a volver a la ecuación (2.2) y vamos a intentar plantearnosla para el caso $n = 2$ y $n = 3$.

A continuación se considerará el caso $n = 2$. En este caso, el problema es de la forma

$$\begin{aligned} \text{Maximizar} \quad & y_1 + y_2 \\ \text{Sujeto a} \quad & y_1 \leq 1 \\ & y_2 + 2y_1 \leq \theta \\ & y_1, y_2 \geq 0. \end{aligned}$$

En la figura 2.1(a) se muestra la región factible usando $\theta = 3$. Obsérvese que para el origen, las variables básicas son (s_1, s_2) . Por tanto, ambos valores $z_j - c_j$ para y_1 y y_2 son -1 . Si tomamos como variable de entrada a y_1 se llega a la solución de punto extremo adyacente* con base (y_1, s_2) . Continuando y adaptandonos a las restricciones, seguiremos la trayectoria del simplex A, B, C, D en la figura 2.1(a). Al hacer lo anterior, se recorren las 2^2 soluciones básicas factibles con variables básicas respectivas $(s_1, s_2), (y_1, s_2), (y_1, y_2)$ y finalmente (s_1, y_2) , es decir, el peor caso posible. Nos quedaría comprobar que $z_j - c_j = -1$ para la variable de cada iteración. Cabe destacar que si hubiéramos elegido como variable de entrada y_2 el problema se habría resuelto en un sola iteración. A continuación se considerará el caso $n = 3$. En este caso, el problema es de la siguiente forma

$$\begin{aligned} \text{Maximizar} \quad & y_1 + y_2 + y_3 \\ \text{Sujeto a} \quad & y_1 \leq 1 \\ & y_2 + 2y_1 \leq \theta \\ & y_3 + 2y_1 + 2y_2 \leq \theta^2 \\ & y_1, y_2, y_3 \geq 0. \end{aligned}$$

*Se denomina adyacente ya que anteriormente obtenemos un vértice que no es óptimo

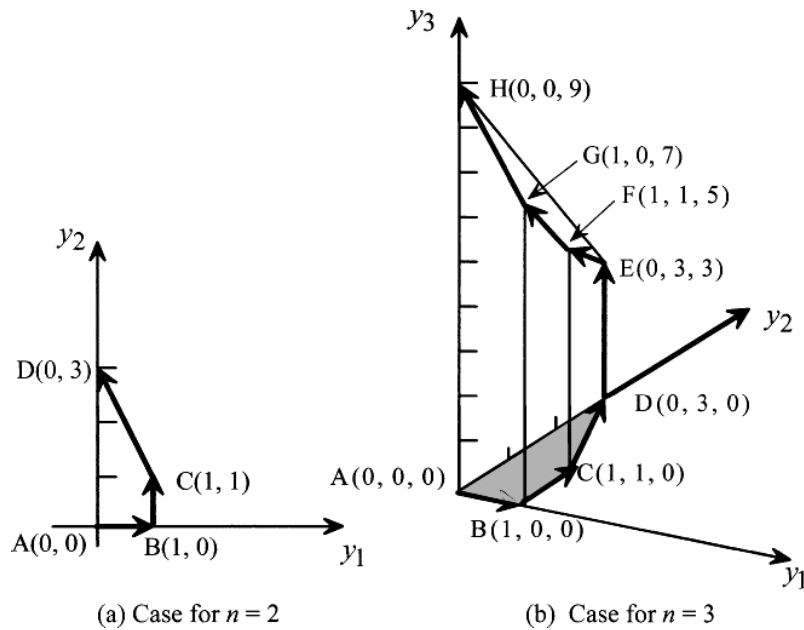


Figura 2.1: Ilustración de los politopos tipo Klee-Minty para $n=2$ y 3 .

En la figura 2.1(b) se muestra la región factible cuando $\theta = 3$ como antes. Ahora se hará la siguiente observación. Primero, observe que con $y_3 = 0$, los puntos extremos en el espacio (y_1, y_2) obtenidos para el caso $n = 2$ siguen siendo puntos extremos en este caso; s_3 es la tercera variable básica. Los conjuntos correspondientes de variables básicas son (s_1, s_2, s_3) , (y_1, s_2, s_3) , (y_1, y_2, s_3) . Segundo, observe que al desplazarse desde cada una de estas bases a la siguiente, el valor $z_j - c_j$ para la nueva variable no básica y_3 también permanece en -1 a lo largo del proceso, ya que la última componente del vector multiplicador símplex $\mathbf{c}_b \mathbf{B}^{-1}$ es cero para todas estas bases \mathbf{B} . Por tanto, la trayectoria A, B, C, D sigue siendo una trayectoria símplex. Por otro lado, y_3 permanece dentro hasta el final. Tercero, en el vértice D es necesario introducir y_3 y como s_3 y y_3 no pueden, ambas, ser básicas, entonces s_3 debe salir de la base. Para las iteraciones restantes, y_3 sigue siendo básica. Aplicando la última ecuación para resolver y_3 en términos de y_1, y_2 y s_3 y eliminando y_3 del problema se obtiene el mismo problema que para el caso $n = 2$, excepto que los coeficientes de y_1 y y_2 ahora son -1 . En otras palabras es como si el problema de $n = 2$ hubiera pasado a ser un problema de minimización al final de su solución. Por consiguiente al mantener este criterio de entrada, la trayectoria símplex simplemente se invierte a través de las bases para el caso $n = 2$, en donde y_3 es la variable básica adicional a lo largo de todo el proceso. A continuación se recorren los vértices E, F, G y H. Por consiguiente, en el proceso de resolución de este problema se han recorrido $2^3 - 1$ aristas y 2^3 puntos extremos.

Capítulo 3

Algoritmo de Karmarkar. Definición

Para realizar el análisis del algoritmo de Karmarkar nos basaremos principalmente en el libro Bazaraa et al. (2011) . Empezaremos por ver la transformación de un problema lineal a la forma de Karmarkar.

3.1. Conversión de un problema lineal general a la forma de Karmarkar

Para poder utilizar el algoritmo de Karmarkar, primero necesitaremos conocer como obtener un problema con la forma de Karmarkar a partir de un problema de programación lineal general.

Consideramos un problema general de programación lineal

$$\text{Minimizar}\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \quad (3.1)$$

en donde \mathbf{A} es una matriz de $m \times n$ con rango m y todos los datos son enteros. A continuación se demostrará como convertir este problema en la forma del problema lineal (3.5) requerido por Karmarkar, a la vez que se cumplen las hipótesis (A1) y (A2). (Las veremos en profundidad en la siguiente sección).

Primero comenzaremos por regularizar el problema (3.1), es decir, añadiendo una restricción de acotamiento $\sum_{j=1}^n x_j \leq Q$. Aquí Q puede tomarse como alguna cota entera conocida (preferentemente pequeña) sobre el sumatorio de las variables, en el peor de los casos es posible tomar $Q = 2^L$, en donde L está dada por la ecuación 2.1. Si esta restricción es compatible en condiciones de optimalidad con el valor objetivo de magnitud $-2^{O(L)}$, entonces puede deducirse que el problema (3.1) es no acotado.

Al añadir esta restricción junto con una variable de holgura x_{n+1} , la ecuación (3.1) puede

volver a escribirse como

$$\text{Minimizar}\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{1}\mathbf{x} + x_{n+1} = Q, \mathbf{x} \geq \mathbf{0}, x_{n+1} \geq 0\}.$$

En este momento es posible homogeneizar las restricciones $\mathbf{A}\mathbf{x} = \mathbf{b}$ reemplazando equivalentemente el lado derecho \mathbf{b} con $\mathbf{b}(\mathbf{1}\mathbf{x} + x_{n+1})/Q$, observando la restricción de igualdad final anterior. Sin embargo, con esto se arruinará cualquier estructura en \mathbf{A} , ya que \mathbf{b} suele ser densa. Alternativamente, se introducirá una variable ficticia x_{n+2} junto con la restricción $x_{n+2} = 1$ a fin de escribir de manera equivalente las restricciones $\mathbf{A}\mathbf{x} - \mathbf{b}x_{n+2} = 0$, $x_{n+2} = 1$, $\mathbf{1}\mathbf{x} + x_{n+1} + x_{n+2} = Q + 1$, más las restricciones de no negatividad. Usando la restricción de igualdad final para homogeneizar la restricción $x_{n+2} = 1$, el problema puede escribirse de manera equivalente como

$$\begin{aligned} \text{Minimizar}\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} - \mathbf{b}x_{n+2} = \mathbf{0}, \mathbf{1}\mathbf{x} + x_{n+1} - Qx_{n+2} = 0, \\ \mathbf{1}\mathbf{x} + x_{n+1} + x_{n+2} = (Q + 1), \mathbf{x} \geq \mathbf{0}, x_{n+2} \geq 0\}. \end{aligned}$$

A continuación se aplicará la transformación $x_j = (Q + 1)y_j$, $j = 1, \dots, n + 2$, a fin de obtener la unidad en el lado derecho de la restricción de igualdad final y así deducir el siguiente problema equivalente.

$$\begin{aligned} \text{Minimizar}\{\mathbf{c}\mathbf{y} : \mathbf{A}\mathbf{y} - \mathbf{b}y_{n+2} = \mathbf{0}, \mathbf{1}\mathbf{y} + y_{n+1} - Qy_{n+2} = 0, \\ \mathbf{1}\mathbf{y} + y_{n+1} + y_{n+2} = 1, \mathbf{y} \geq \mathbf{0}, y_{n+1} \geq 0, y_{n+2} \geq 0\}. \end{aligned} \quad (3.2)$$

Las restricciones en la ecuación (3.1) son ahora de la forma del problema (3.5). Al fin de satisfacer la hipótesis (A1), en la ecuación (3.1) se introducirá una variable artificial y_{n+3} con coeficiente objetivo M y coeficientes de las restricciones tales que la solución $(y_1, \dots, y_{n+3}) = [1/(n+3), \dots, 1/(n+3)]$ sea factible. Para que ocurra lo anterior, la suma de los coeficientes de las restricciones homogéneas resultantes debe ser igual a cero y el coeficiente de y_{n+3} en la restricción de igualdad final debe ser la unidad. Observe que existe un valor de M de magnitud $2^{O(L)}$ que asegura que la variable artificial y_{n+3} será igual a cero en condiciones de optimalidad. Por tanto la longitud de entrada de la ecuación (3.1) con la variable artificial sigue siendo $O(L)$. Así, este problema artificial está dado de la siguiente manera

$$\begin{aligned} \text{Minimizar} \quad & \mathbf{c}\mathbf{y} + M y_{n+3} \\ \text{Sujeto a} \quad & \mathbf{A}\mathbf{y} - \mathbf{b}y_{n+2} - [\mathbf{A}\mathbf{1}^t - \mathbf{b}]y_{n+3} = \mathbf{0} \\ & \mathbf{1}\mathbf{y} + y_{n+1} - Qy_{n+2} - (n + 1 - Q)y_{n+3} = 0 \\ & \mathbf{1}\mathbf{y} + y_{n+1} + y_{n+2} + y_{n+3} = 1 \\ & y_j \geq 0, j = 1, \dots, n + 3. \end{aligned} \quad (3.3)$$

El problema (3.3) es de la forma (3.5) en $(n+3)$ variables, en donde $[1/(n+3), \dots, 1/(n+3)]$ es una solución factible, Además, su tamaño esta polinómicamente relacionado con el del problema (3.1), es decir, su longitud de entrada es de $O = (L)$.

Por último, se considera la hipótesis (A2), vamos a presentar una manera sencilla aunque computacionalmente inconveniente de satisfacer dicha hipótesis en la deducción anterior. Por dualidad de la programación lineal, vista anteriormente en la sección 1.3 y en concreto por la proposición 1.9 se observa que las condiciones de optimalidad para el problema (3.1) requieren de una solución, en caso de existir, del sistema

$$\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \boldsymbol{\pi}\mathbf{b} \leq \mathbf{cx} \rightarrow \boldsymbol{\pi}\mathbf{A} \leq \mathbf{c}, \quad y \quad \mathbf{cx} = \boldsymbol{\pi}\mathbf{b}. \quad (3.4)$$

Al escribir tales restricciones en forma estándar del tipo $\mathbf{A}'\mathbf{x}' = \mathbf{b}'$ y $\mathbf{x}' \geq \mathbf{0}$, las transformaciones anteriores pueden usarse para obtener un problema del tipo (3.3), en donde la función objetivo es simplemente la minimización de la variable artificial y_{n+3} . Si el problema original tiene un óptimo, entonces este problema (3.3) tiene un valor óptimo igual a cero. Además, su solución lleva a soluciones óptimas del problema original (3.1). En caso contrario, el valor objetivo del problema (3.3) es positivo en condiciones de optimalidad. Este hecho será revelado por la incapacidad del algoritmo para hacer que el valor objetivo llegue por abajo de 2^{-l} dentro de la cota polinómica 10^{nl} del número de iteraciones, como se establecerá en la siguiente sección, en donde $l \equiv O(L)$ representa la longitud de entrada del problema transformado (3.3). En este caso, el problema original es no factible o no acotado, y esto puede resolverse probando de manera semejante la factibilidad.

3.2. Algoritmo de Karmarkar

El método símplex, habiendo experimentado considerables cambios, refinamientos y sofisticaciones, no tuvo competencia seria sino hasta 1984, cuando Narendra Karmarkar propuso un nuevo algoritmo polinómico-temporal para resolver problemas de programación lineal. Este algoritmo estudia problemas de programación lineal de la forma

$$\begin{aligned} \text{Minimizar} \quad & \mathbf{cx} \\ \text{Sujeto a} \quad & \mathbf{Ax} = \mathbf{0} \\ & \mathbf{1x} = 1 \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (3.5)$$

en donde \mathbf{A} es una matriz de $m \times n$ con rango m , $n \geq 2$, \mathbf{c} y \mathbf{A} están integradas exclusivamente de enteros, $\mathbf{1}$ es un vector renglón de n unos y se cumplen las siguientes hipótesis (A1) y (A2)

(A1) El punto $x_0 = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)$ es factible en el problema (3.5).

(A2) El valor óptimo del objetivo del problema (3.5) es cero.

A primera vista, la forma del problema (3.5) y las hipótesis acompañantes (A1) y (A2) pueden parecer excesivamente restrictivas. Sin embargo, como hemos demostrado anteriormente, cualquier problema general de programación lineal puede plantearse de esta forma mediante el empleo de variables artificiales, una restricción artificial de acotamiento y redefiniciones de las variables, en caso de ser necesario. Además, se demostrará que el análisis en estas condiciones puede extenderse fácilmente para resolver polinómicamente el problema general de programación lineal. Se procederá con la obtención de un algoritmo polinómico-temporal para resolver el problema (3.5) asumiendo las hipótesis (A1) y (A2). Observe que con estas hipótesis el problema (3.5) es factible y acotado, y por tanto tiene una solución óptima.

Empezando con $x_0 = (1/n, \dots, 1/n)$ y $k = 0$, el algoritmo ejecuta el siguiente paso iterativo, dada cualquiera solución factible $x_k > \mathbf{0}$. Se define la matriz diagonal $D_k = \text{diag}\{x_{k1}, \dots, x_{kn}\}$, en donde $x_k = (x_{k1}, \dots, x_{kn})^t$, y se considera la siguiente transformación de variables.

$$y = \frac{D_k^{-1}x}{\mathbf{1}D_k^{-1}x},$$

es decir,

$$y_i = \frac{(x_i/x_{ki})}{\sum_{j=1}^n (x_j/x_{kj})} \quad \text{para } i = 1, \dots, n. \quad (3.6)$$

Obsérvese que la región factible en el problema (3.5) está descrita por la intersección de un subespacio lineal de dimensión $(n-m)$ definido por el conjunto homogéneo de desigualdades $Ax = 0$ con el simplex $S_x = \{x : \mathbf{1}x = 1, x \geq 0\}$ de dimensión $(n-1)$. La intersección es alguna región de dimensión $(n-m-1)$. En la figura 3.1 se ilustra una situación $n = 3$ dimensiones con una sola restricción de igualdad homogénea.

Bajo la transformación (3.6), cualquier punto de S_x se transforma en un punto en el simplex $S_y = \{y : \mathbf{1}y = 1, y \geq 0\}$ de dimensión $(n-1)$. En particular, el punto actual x_k se transforma en el punto $y_0 = (1/n, \dots, 1/n)$, el cero del simplex S_y . Obsérvese que si debiera usarse una la transformación afín $y' = D_k^{-1}x$, que es simplemente una operación de reescalado, entonces un punto $x \in S_x$ se transformaría, por ejemplo, en el punto y' que se muestra en la figura 3.2. Si ese punto y' se proyecta sobre S_y a lo largo del rayo definido por el origen y y' , entonces se obtendría el punto y dado por la transformación (3.6), como se muestra en la figura 3.2. Lo anterior supone dividir y' entre la suma de sus componentes como en la transformación (3.6), de modo que la suma de las componentes

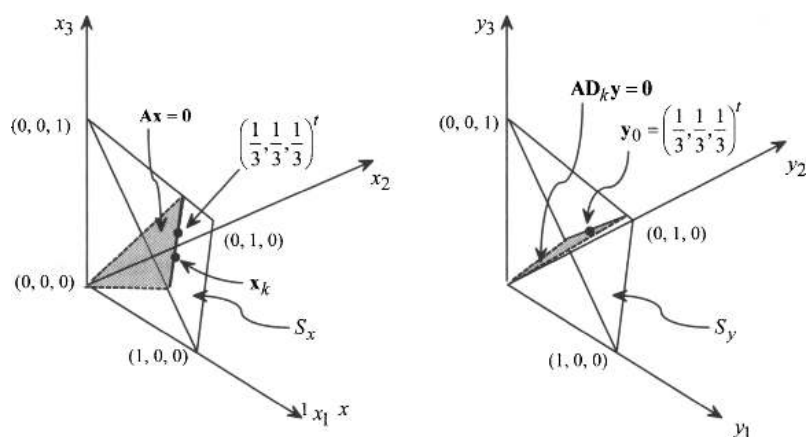


Figura 3.1: Transformación proyectiva de la región factible.

de \mathbf{y} sea igual a la unidad. Ésta es la razón por la cual la transformación (3.6) se denomina *transformación proyectiva*. La ejecución de esta operación en la región factible en el espacio \mathbf{x} da por resultado la región factible en el espacio \mathbf{y} , como se ilustra en la figura 3.1.

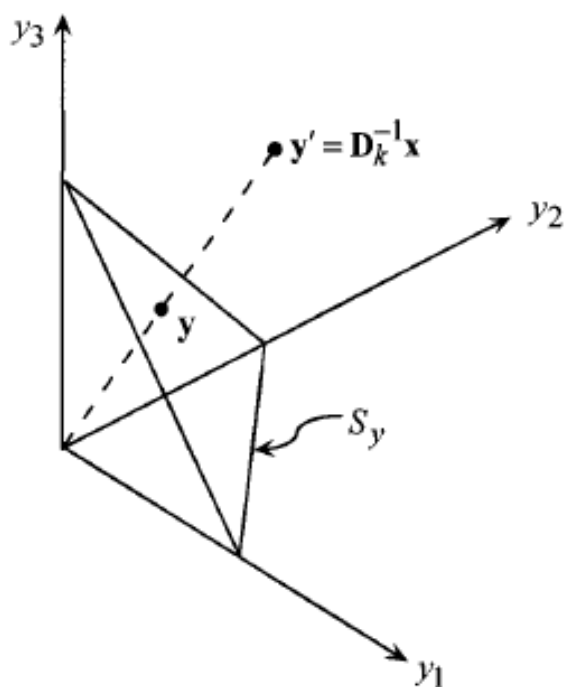


Figura 3.2: Transformación proyectiva.

A fin de describir algebraicamente esta región factible en el espacio \mathbf{y} , considere la

transformación proyectiva inversa que se obtiene al despejar \mathbf{x} en S_x a partir de la transformación (3.6). Con lo anterior se obtiene $\mathbf{x} = (\mathbf{D}_y \mathbf{y})(\mathbf{1D}_k^{-1} \mathbf{x})$. Como $\mathbf{1x} = 1$, se tiene $(\mathbf{1D}_k \mathbf{y})(\mathbf{1D}_k^{-1} \mathbf{x}) = 1$. Por tanto, $(\mathbf{1D}_k^{-1} \mathbf{x}) = 1/(\mathbf{1D}_k \mathbf{y})$. Al sustituir lo anterior en la expresión previa para \mathbf{x} se obtiene

$$\mathbf{x} = \frac{\mathbf{D}_k \mathbf{y}}{\mathbf{1D}_k \mathbf{y}}. \quad (3.7)$$

Por consiguiente, las transformaciones (3.6) y (3.7) mandan puntos de simplex S_x sobre puntos de simplex S_y y viceversa.

Bajo las transformaciones (3.6) y (3.7), el problema (3.5) se transforma en el siguiente problema en el espacio \mathbf{y} sobre un subconjunto de simplex S_y .

$$\text{Minimizar} \left\{ \frac{\mathbf{cD}_k \mathbf{y}}{\mathbf{1D}_k \mathbf{y}} : \mathbf{AD}_k = 0, \mathbf{1y} = 1, \mathbf{y} \geq 0 \right\}. \quad (3.8)$$

Obsérvese que aunque las restricciones permanecen lineales debido a que $\mathbf{Ax} = \mathbf{0}$ es homogénea, la función objetivo se ha convertido en un cociente de funciones lineales. Este tipo de problema se denomina problema de *programación lineal fraccional*. Sin embargo, debido a la hipótesis (A2), el valor óptimo del objetivo en la ecuación (3.8) es cero. Así, es posible minimizar equivalentemente el numerador del problema, ya que el denominador es positivo, acotado y se encuentra lejos del cero para todo $\mathbf{y} \in S_y$. Lo anterior lleva al siguiente problema

$$\text{Minimizar} \{ \bar{\mathbf{c}} \mathbf{y} : \mathbf{P} \mathbf{y} = \mathbf{P}_0, \mathbf{y} \geq 0 \}, \quad (3.9)$$

en donde

$$\mathbf{P} = \begin{bmatrix} \mathbf{AD}_k \\ \mathbf{1} \end{bmatrix}, \quad \bar{\mathbf{c}} = \mathbf{cD}_k \quad \mathbf{y} \quad \mathbf{P}_0 = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}.$$

En vez de resolver el problema (3.9), lo que es equivalente a resolver el problema original, se optimizará una restricción particular más sencilla de este problema. Así se llegará a una nueva solución factible. Al repetir de manera iterada el paso será posible obtener polinómicamente una solución óptima del problema original.

Para definir esta restricción, considere la esfera o la bola de dimensión n dada por $B(\mathbf{y}_0, r)$ con centro $\mathbf{y}_0 = (1/n, \dots, 1/n)$ y con radio idóneo r de modo que la intersección de esta bola con la restricción simplex $\{\mathbf{y} : \mathbf{1y} = 1\}$ es una bola de dimensión $(n-1)$ con el mismo centro y el mismo radio, inscrita en S_y (ver la figura 3.3). Obsérvese que este radio r es la distancia del centro $(1/n, \dots, 1/n)$ del simplex al centro de una de sus facetas. Sin embargo, las facetas de S_y son sencillamente de una dimensión menor. Por tanto, al considerar la faceta en que la componente y_1 es cero, por ejemplo, se obtiene r como la distancia de $(1/n, \dots, 1/n)$ a $(0, 1/(n-1), \dots, 1/(n-1))$. Con lo anterior se obtiene $r = 1/\sqrt{n(n-1)}$.

Considere una restricción de la ecuación 3.9 en la que se busca minimizar $\bar{c}\mathbf{y}$ sujeta a $\mathbf{P}\mathbf{y} = \mathbf{P}_0$, $\mathbf{y} \geq \mathbf{0}$, y $\mathbf{y} \in B(\mathbf{y}_0, \alpha r)$, en donde $0 < \alpha < 1$ es una constante. La elección específica de α se abordará más tarde. Observe que la bola inscrita se ha reducido por un factor α , por lo que su radio es αr . Además, dejando factible esta restricción, los iterandos permanecen estrictamente positivos. Ya que $\mathbf{y} \geq \mathbf{0}$ se implica por la intersección de la bola y la restricción simplex $\{\mathbf{y} : \mathbf{1}\mathbf{y} = 1\}$, esta restricción es equivalente al problema

$$\text{Minimizar}\{\bar{c}\mathbf{y} : \mathbf{P}\mathbf{y} = \mathbf{P}_0, (\mathbf{y} - \mathbf{y}_0)^t(\mathbf{y} - \mathbf{y}_0) \leq \alpha^2 r^2\}, \quad (3.10)$$

en donde $\{\mathbf{y} : (\mathbf{y} - \mathbf{y}_0)^t(\mathbf{y} - \mathbf{y}_0) \leq \alpha^2 r^2\}$ describe la bola $B(\mathbf{y}_0, \alpha r)$. En la figura 3.3 se muestra la región factible sobre el simplex bidimensional.

Obsérvese que el sistema $\mathbf{P}\mathbf{y} = \mathbf{P}_0$ define un subespacio afín de dimensión $(n - m - 1)$ que pasa por el centro de la bola. Por tanto la región factible en el problema (3.10) es una bola de dimensión $(n - m - 1)$ centrada en \mathbf{y}_0 . En la figura 3.4 se ilustra una situación una situación para $n = 4$, $m = 1$ sobre un simplex de dimensión $n - 1 = 3$. Resulta bastante intuitivo que la solución óptima del problema (3.10) se obtiene proyectando el gradiente negativo de la función objetivo $-\bar{c}^t$ centrado en \mathbf{y}_0 , sobre el espacio nulo o la superficie de la restricción de $\mathbf{P}\mathbf{y} = \mathbf{P}_0$ y moviéndose desde \mathbf{y}_0 a lo largo de esta dirección proyectada hacia la frontera de la bola $B(\mathbf{y}_0, \alpha r)$. Este proceso de solución es una consecuencia del hecho de que se está optimizando una función lineal sobre una bola, y que actualmente se está situando en el centro de la bola.

Al denotar por \mathbf{c}_p la proyección del vector gradiente $-\bar{c}^t$ y por \mathbf{y}_{nuevo} el óptimo del problema (3.10), se obtiene

$$\mathbf{y}_{nuevo} = \mathbf{y}_0 - \alpha r \frac{\mathbf{c}_p}{\|\mathbf{c}_p\|}. \quad (3.11)$$

Si $\mathbf{c}_p = \mathbf{0}$ entonces cualquier solución es factible es óptima.

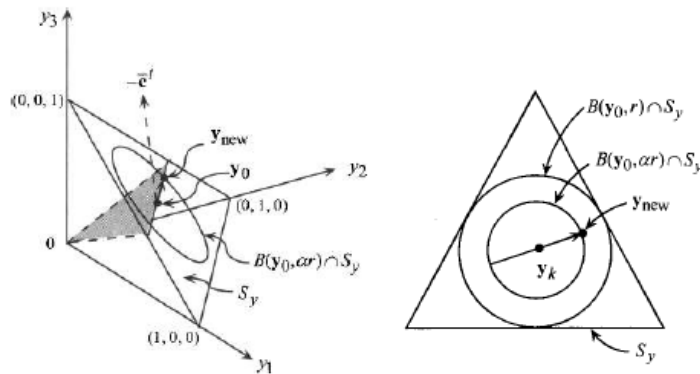


Figura 3.3: Paso principal del algoritmo de karmarkar.

Para calcular \mathbf{c}_p , observese que como \mathbf{c}_p está en la superficie de la restricción de $\mathbf{P}\mathbf{y} = \mathbf{P}_0$, entonces el vector $(\bar{\mathbf{c}}^t - \mathbf{c}_p)$ está contenido en el espacio generado por los gradientes a $\mathbf{P}\mathbf{y} = \mathbf{P}_0$. Así, existe un vector $\bar{\mathbf{w}}$ tal que $\mathbf{P}^t\bar{\mathbf{w}} = \bar{\mathbf{c}}^t - \mathbf{c}_p$. Al multiplicar ambos lados por \mathbf{P} , se obtiene $\mathbf{P}\mathbf{P}^t\bar{\mathbf{w}} = \mathbf{P}\bar{\mathbf{c}}^t$, ya que por definición se tiene $\mathbf{P}\mathbf{c}_p = 0$. Debido a que \mathbf{A} es de rango completo por renglones y $\mathbf{x}_k > 0$, entonces la matriz cuadrada $\mathbf{P}\mathbf{P}^t$ es invertible, con lo que se obtiene $\bar{\mathbf{w}} = (\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}\bar{\mathbf{c}}^t$. Por consiguiente, se tiene $\mathbf{c}_p = \bar{\mathbf{c}}^t - \mathbf{P}^t\bar{\mathbf{w}}$ dado por

$$\mathbf{c}_p = [\mathbf{I} - \mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}]\bar{\mathbf{c}}^t. \quad (3.12)$$

Dado \mathbf{y}_{nuevo} en la ecuación (3.11), el vector \mathbf{x}_{k+1} revisado correspondiente en el espacio \mathbf{x} se obtiene mediante la transformación inversa (3.7) como

$$\mathbf{x}_{k+1} = \frac{\mathbf{D}_k\mathbf{y}_{nuevo}}{\mathbf{1D}_k\mathbf{y}_{nuevo}}. \quad (3.13)$$

Observese que $\mathbf{y}_{nuevo} > 0$, ya que está en el interior de la esfera de dimensión $(n-1)$ inscrita en S_y . Por tanto, en la ecuación (3.13) se tiene que $\mathbf{x}_{k+1} > 0$. Con lo anterior se completa una iteración. El paso puede repetirse después de incrementar k en una unidad. El valor objetivo en el problema (3.5) de la sucesión de iterandos así generados tiende al valor objetivo óptimo de cero en el límite cuando $k \rightarrow \infty$, aunque no necesariamente de manera monótona. Por tanto, para fines prácticos, este proceso puede terminarse cuando el valor objetivo se aproxima lo suficiente a cero.

Para fines teóricos, considere la siguiente cota inferior L de la longitud de entrada del problema (3.5):

$$L = [1 + \log(1 + |c_{j \text{ máx}}|) + \log(|\det_{\text{máx}}|)], \quad (3.14)$$

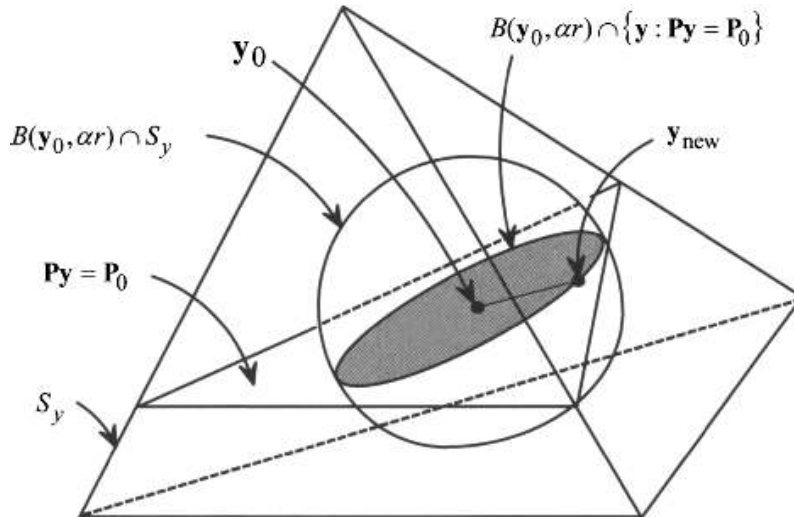


Figura 3.4: Ilustración del algoritmo de karmarkar.

en donde $|c_j \text{ máx}|$ es el mayor valor numérico de cualquier coeficiente de costo c_j y $|det_{\text{máx}}|$ es el mayor valor numérico del determinante de cualquier base del problema (3.5). Con base en una propiedad de los determinantes se concluye que L es, en efecto, una cota inferior de la longitud de entrada del problema (3.5), ya que $\log(|det_{\text{máx}}|) \leq \log(1 + m) + \sum_i \sum_j \log(1 + |a_{ij}|)$. Más adelante veremos que al usar el valor de $\alpha = (n - 1)/3n$, el algoritmo produce una solución con valor objetivo menor que 2^{-L} en un lapso de $10nL$ iteraciones, con un esfuerzo total de complejidad polinómica igual a $O(n^{3.5}L)$. Una vez que ocurre esto, por medio del siguiente proceso polinómico es posible obtener una solución óptima del *punto extremo* exacta del problema (3.5).

Empezando con el iterando final \mathbf{x}_k obtenido con valor objetivo $\mathbf{c}\mathbf{x}_k < 2^{-L}$, este procedimiento encuentra una solución de punto extremo con un valor objetivo por lo menos tan aceptable aplicando el siguiente método, denominado *esquema de purificación*. Si n restricciones linealmente independientes son conectantes en \mathbf{x}_k , entonces \mathbf{x}_k ya es una solución básica factible. En caso contrario, existe una dirección $\mathbf{d} \neq 0$ que está en el espacio nula de las restricciones conectantes; es decir, que satisface el sistema homogéneo de igualdades correspondientes a las restricciones conectantes. El método mueve ahora el iterando actual a lo largo de la dirección \mathbf{d} si $\mathbf{c}\mathbf{d} < 0$ y a lo largo de la dirección $-\mathbf{d}$ en caso contrario, hasta que alguna restricción bloquea cualquier movimiento adicional por medio de consideraciones de factibilidad. Lo anterior debe suceder, ya que la región factible es acotada. Observe que en la nueva solución el valor objetivo no es mayor que $\mathbf{c}\mathbf{x}_k < 2^{-L}$, y que por lo menos una restricción linealmente independiente adicional es conectante. Procediendo de esta manera es posible obtener una solución básica factible $\bar{\mathbf{x}}$ del problema (3.5) con valor objetivo estrictamente menor que 2^{-L} .

Obsérvese que este proceso implica cuando mucho $n - m(m + 1)$ de tales pasos, ya que empieza con las $(m + 1)$ restricciones de igualdad linealmente independientes conectantes en el problema (3.5) y luego en cada paso añade a este conjunto por lo menos una restricción conectante linealmente independiente adicional. Además, observe que cada paso es de complejidad polinómica. Así, $\bar{\mathbf{x}}$ se determina a partir del iterando final \mathbf{x}_k en tiempo polinómico. A continuación se demostrará que esta solución de punto extremo resultante $\bar{\mathbf{x}}$ es en efecto óptima para el problema (3.5). Debido a lo anterior, el proceso se denomina *rutina de redondeo óptimo*. Como todos los datos son enteros, cualquier solución básica factible del problema (3.5) con base \mathbf{B} tiene un valor objetivo " $\mathbf{c}_b\mathbf{B}^{-1}\mathbf{b}$ " dado por N/D , en donde N y D son enteros, y en donde $D = |det\mathbf{B}|$. Sin embargo, $D < 2^L$. Así, por la hipótesis (A2), cualquier solución de punto extremo no óptima del problema (3.5) debe tener un valor objetivo positivo $N/D \geq 1/D > 2^{-L}$. Como $\mathbf{c}\bar{\mathbf{x}} < 2^{-L}$, entonces se debe tener que $\bar{\mathbf{x}}$ es óptimo para el problema (3.5) con $\mathbf{c}\bar{\mathbf{x}} = 0$.

Resumen del Algoritmo de Karmarkar

Los pasos a seguir para resolver un problema serán los siguientes

INICIALIZACIÓN

Se calculan $r = 1/\sqrt{n(n-1)}$, $L = [1 + \log(1 + |c_j \text{ máx}|) + \log(|\det_{\text{máx}}|)]$. El cálculo directo de L es difícil y en la práctica se puede evitar usando otros criterios.

Una forma de estimar L es

$$L' = 1 + \log(1 + |c_j \text{ máx}|) + \log(1 + m) + \log(|\det_{\text{máx}}|) \quad \text{donde } L \leq L'$$

A continuación se elige $\alpha = (n-1)/3n$. Sea $\mathbf{x}_0 = (1/n, \dots, 1/n)$ y se hace $K = 0$.

PASO PRINCIPAL

Si $\mathbf{c}\mathbf{x}_k < 2^{-L}$, se aplica la rutina de redondeo óptimo para determinar una solución óptima y se detiene. (Prácticamente, como 2^{-L} puede ser muy pequeño, el proceso podría terminar cuando $\mathbf{c}\mathbf{x}_k$ es menos que alguna otra tolerancia deseada).

En caso contrario, se define

$$\mathbf{D}_k = \text{diag}\{x_{k1}, \dots, x_{kn}\}, \quad \mathbf{y}_0 = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)^t$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{A}\mathbf{D}_k \\ \mathbf{1} \end{bmatrix} \quad \text{y} \quad \bar{\mathbf{c}} = \mathbf{c}\mathbf{D}_k,$$

y se calcula

$$\mathbf{y}_{\text{nuevo}} = \mathbf{y}_0 - \alpha r \frac{\mathbf{c}_p}{\|\mathbf{c}_p\|} \quad \text{donde } \mathbf{c}_p = [\mathbf{I} - \mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}]\bar{\mathbf{c}}^t.$$

Por tanto se obtiene $\mathbf{x}_{k+1} = (\mathbf{D}_k\mathbf{y}_{\text{nuevo}})/(\mathbf{1}\mathbf{D}_k\mathbf{y}_{\text{nuevo}})$. K se incrementa en una unidad y se repite el paso principal.

RUTINA DE REDONDEO ÓPTIMO

Empezando con \mathbf{x}_k , se determina una solución de punto extremo $\bar{\mathbf{x}}$ para el problema (3.5) con $\mathbf{c}\bar{\mathbf{x}} \leq \mathbf{c}\mathbf{x}_k < 2^{-L}$ usando el *esquema de purificación* previo. Se termina con $\bar{\mathbf{x}}$ como una solución óptima del problema 3.5.

Ejemplo 3.1. Consideramos en siguiente problema de programación lineal

$$\begin{aligned} \text{Minimizar } & x_2 \\ \text{Sujeto a } & x_1 + x_2 - 2x_3 = 0 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3, \geq 0 \end{aligned}$$

Aquí $n = 3$ y $m = 2$. La región factible de dimensión $n - m - 1 = 1$ se ilustra en la figura 3.5. Es evidente que el punto $\mathbf{x}_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^t$ es factible y la solución $\left(\frac{2}{3}, 0, \frac{1}{3}\right)$ es óptima con valor objetivo igual a cero. Por tanto, se cumplen las hipótesis (A1) y (A2). También se tiene $r = 1\sqrt{6}$ y $\alpha = \frac{2}{9}$.

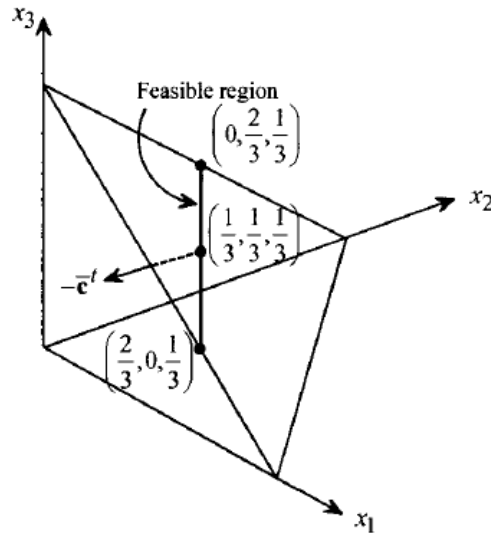


Figura 3.5: Ilustración del ejemplo 3.1.

Iteración 1

Empezando con \mathbf{x}_0 y $k = 0$, se define $\mathbf{D}_0 = \text{diag}\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. Como $\mathbf{1D}_k^{-1}\mathbf{x} = \sum 3x_i = 3$, entonces la transformación proyectiva (3.7) proporciona $\mathbf{y} \equiv \mathbf{x}$. Por tanto, el problema coincide con el de la figura 3.5. El punto actual en el espacio es, $\mathbf{y}_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^t$.

Luego se calcula $\bar{\mathbf{c}} = \mathbf{cD}_k = (0, \frac{1}{3}, 0)$. A fin de usar la ecuación (3.12), se calcula usando $\mathbf{A} = [1, 1, -2]$ y $\mathbf{P} = \begin{bmatrix} \mathbf{AD}_k \\ \mathbf{1} \end{bmatrix}$.

$$\mathbf{P} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & -\frac{2}{3} \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{P}\mathbf{P}^t = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & 3 \end{bmatrix}, \quad (\mathbf{P}\mathbf{P}^t)^{-1} = \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix},$$

$$(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & -1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}, \quad (\mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}\mathbf{P}^t = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Por la ecuación 3.12 se obtiene

$$\mathbf{c}_p = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{3} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} \\ \frac{1}{6} \\ 0 \end{bmatrix}, \quad \|\mathbf{c}_p\| = \sqrt{\frac{1}{6^2} + \frac{1}{6^2} + 0} = \frac{\sqrt{2}}{6}.$$

Observese que en el caso de la dirección $-\mathbf{c}_p$ va de $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$ hacia la solución óptima $(\frac{2}{3}, 0, \frac{1}{3})^t$. Aplicando la ecuación (3.11), con lo anterior se obtiene

$$\mathbf{y}_{\text{nuevo}} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} - \left(\frac{2}{9}\right) \left(\frac{1}{\sqrt{6}}\right) \left(\frac{6}{\sqrt{2}}\right) \begin{bmatrix} -\frac{1}{6} \\ \frac{1}{6} \\ 0 \end{bmatrix} = \begin{bmatrix} 0.397484 \\ 0.269183 \\ 0.333333 \end{bmatrix}.$$

Al transformar en el espacio \mathbf{x} mediante la ecuación (3.13), se obtiene $\mathbf{x}_1 = \mathbf{y}_{\text{nuevo}}$ para esta primera iteración. El valor actual es $\mathbf{c}\mathbf{x}_1 = 0.269183$.

Iteración 2

Ahora se tiene que $K = 1$. Se define

$$\mathbf{D}_1 = \text{diag}\{0.397484, 0.269183, 0.333333\}, \quad \mathbf{c} = \bar{\mathbf{c}}\mathbf{D}_k = (0, 0.269183, 0),$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{A}\mathbf{D}_k \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} 0.397484 & 0.269183 & -0.666666 \\ 1 & 1 & 1 \end{bmatrix}.$$

Con la base en la ecuación (3.12) se calcula

$$\mathbf{c}_p = [\mathbf{I} - \mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}]\bar{\mathbf{c}}^t = \begin{bmatrix} -0.1324029 \\ 0.1505548 \\ -0.0181517 \end{bmatrix} \quad y \quad \|\mathbf{c}_p\| = 0.2013121.$$

Al sustituir en la ecuación (3.11) se obtiene, usando $\frac{\alpha r}{\|\mathbf{c}_p\|} = 0.4506524$, que

$$\mathbf{y}_{\text{nuevo}} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} (-0.4506524) \begin{bmatrix} -0.1324029 \\ 0.1505548 \\ -0.0181517 \end{bmatrix} = \begin{bmatrix} 0.3930009 \\ 0.2654855 \\ 0.3415134 \end{bmatrix}.$$

Por la ecuación (3.13) se obtiene

$$\mathbf{D}_1 \mathbf{y}_{\text{nueva}} = \begin{bmatrix} 0.1562112 \\ 0.0714641 \\ 0.1138378 \end{bmatrix}, \quad \mathbf{1D}_1 \mathbf{y}_{\text{nueva}} = 0.3415131,$$

$$\mathbf{x}_2 = \frac{\mathbf{D}_1 \mathbf{y}_{\text{nueva}}}{\mathbf{1D}_1 \mathbf{y}_{\text{nueva}}} = \begin{bmatrix} 0.457409 \\ 0.209258 \\ 0.333333 \end{bmatrix}.$$

Con lo anterior se completa la segunda iteración. El valor de la función objetivo actual es $\mathbf{c}\mathbf{x}_2 = 0.209258$.

Resulta evidente que los iterando \mathbf{x}_k que están generando son del tipo $(\frac{1}{3} + \theta, \frac{1}{3} - \theta, \frac{1}{3})^t$, en donde θ crece gradualmente. De hecho, al efectuar los cálculos anteriores es posible comprobar que θ tiende a $\frac{1}{3}$ en el límite cuando $k \rightarrow \infty$. Obsérvese que $L = [1 + \log(2) + \log(3)] = 4$, y así $2^{-L} = 0.0625$ para este problema. Entonces, teóricamente es posible detener los cálculos cuando $\mathbf{c}\mathbf{x}_k < 0.0625$. Lo anterior sucede para $k = 6$, a partir de lo cual se obtiene $\mathbf{x}_6 = (0.606509, 0.060158)^t$, con $\mathbf{c}\mathbf{x}_6 = 0.060158$. Por último, esta solución se redondea óptimamente. Observe que sólo las dos restricciones de igualdad (linealmente independientes) en el problema son conectantes en \mathbf{x}_6 . Por tanto, es necesario determinar primero un solución $\mathbf{d} = (d_1, d_2, d_3) \neq 0$ del sistema homogéneo $d_1 + d_2 - 2d_3 = 0$ y $d_1 + d_2 + d_3 = 0$. Al despejar d_1 y d_3 en términos de d_2 , se obtiene $d_1 = -d_2$ y $d_3 = 0$. Tomando arbitrariamente $d_2 = 1$ se obtiene la dirección $\mathbf{d} = (-1, 1, 0)^t$. Como $\mathbf{c}\mathbf{d} = 1 > 0$, el movimiento se hace en la dirección $-\mathbf{d} = (1, -1, 0)^t$. La restricción $\mathbf{x}_2 \geq 0$ bloquea el movimiento a lo largo de esta dirección después de un longitud de paso de 0.060158 y se obtiene la tercera restricción linealmente independiente requerida. La nueva solución $\bar{\mathbf{x}} = (0.666667, 0, 0.333333)^t$ es así una solución de punto de extremo, que además es óptima.

Ejemplo 3.2. Consideramos ahora el siguiente problema de programación lineal de la forma (3.5):

$$\begin{aligned} \text{Minimizar} \quad & -x_1 - 2x_2 + 4x_5 \\ \text{Sujeto a} \quad & x_1 - x_2 + 2x_3 + \quad - x_5 = 0 \\ & x_1 - x_2 + \quad + x_4 - 4x_5 = 0 \\ & x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Obsérvese que aquí la solución $\mathbf{x}_0 = (\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})^t$ es factible, el valor objetivo es no negativo para todas las soluciones factibles, mientras que es igual a cero para la solución

factible $\mathbf{x}^* = \left(0, \frac{2}{5}, \frac{2}{5}, 0, \frac{1}{5}\right)^t$. Por tanto, \mathbf{x}^* es óptima con valor objetivo cero, y así, se cumplen las hipótesis requeridas (A1) y (A2).

Ahora se tiene $n = 5$, por lo que $r = 1/\sqrt{20}$ y $\alpha = \frac{4}{15}$. Siguiendo con la sugerencia anterior de utilizar $\log(1 + m) + \sum_i \sum_j \log(1 + \|\mathbf{a}_{ij}\|)$ en vez de tener que calcular $\log(|\det_{\max}|)$, se tiene $L = [1 + \log(5) + \log(3) + \{4\log(2) + 3\log(3) + \log(5)\}] = 16$, de modo que $2^{-L} = 0.00001526$. Se empieza con la solución positiva factible conocida \mathbf{x}_0 con valor objetivo 0.2.

Iteración 1

Para la primera iteración se calculan las siguiente cantidades

$$\mathbf{D}_0 = \text{diag}\{0.2, 0.2, 0.2, 0.2\}, \quad \mathbf{y}_0 = (0.2, 0.2, 0.2, 0.2, 0.2)^t,$$

$$\bar{\mathbf{c}} = (-0.2, -0.4, 0, 0, 0.8),$$

$$\mathbf{P} = \begin{bmatrix} 0.2 & -0.2 & 0.4 & 0 & -0.4 \\ 0.2 & 0.4 & 0 & 0.2 & -0.8 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$[\mathbf{I} - \mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}] = \begin{bmatrix} 0.6947 & -0.1474 & 0.3754 & -0.2175 & 0.0456 \\ -0.1474 & 0.2737 & 0.2211 & -0.3579 & 0.0105 \\ -0.3754 & 0.2211 & 0.2854 & -0.1181 & -0.0129 \\ -0.2175 & -0.3579 & -0.1181 & 0.7415 & -0.0480 \\ 0.0456 & 0.0105 & -0.0129 & -0.0480 & 0.0047 \end{bmatrix},$$

$$\mathbf{c}_p = (-0.0435, -0.0716, -0.0236, 0.1483, -0.0096)^t, \quad \|\mathbf{c}_p\| = 0.1722,$$

$$\mathbf{y}_{nueva} = \mathbf{y}_0 - \alpha r \frac{\mathbf{c}_p}{\|\mathbf{c}_p\|} = (0.2151, 0.2248, 0.2082, 0.1487, 0.2033)^t,$$

$$\mathbf{x}_1 = \frac{\mathbf{D}_0 \mathbf{y}_{nueva}}{\mathbf{1} \mathbf{D}_0 \mathbf{y}_{nueva}} \equiv \mathbf{y}_{nueva},$$

y $\mathbf{c}\mathbf{x}_1 = 0.1487$. Lo anterior se repite ahora con \mathbf{x}_1 , con lo que llega la sucesión

$$\mathbf{x}_2 = (0.2268, 0.2446, 0.2149, 0.1078, 0.2059)^t \quad \text{con } \mathbf{c}\mathbf{x}_2 = 0.1078,$$

$$\mathbf{x}_3 = (0.2355, 0.2596, 0.2200, 0.0769, 0.2079)^t \quad \text{con } \mathbf{c}\mathbf{x}_3 = 0.0769,$$

hasta obtener

$$\mathbf{x}_{26} = (0.2569681, 0.2972037, 0.2329655, 0.0000150, 0.2128477)^t,$$

$$\text{con } \mathbf{c}\mathbf{x}_{26} = 0.000015,$$

donde se cumpliría que $\mathbf{c}\mathbf{x}_{26} < 2^{-L} = 0.00001526$. Para efectuar el redondeo óptimo, se requiere determinar una solución diferente de cero del sistema homogéneo $d_1 - d_2 + 2d_3 - 2d_5 = 0$, $d_1 + 2d_2 + d_4 - 4d_5 = 0$ y $d_1 + d_2 + d_3 + d_4 + d_5 = 0$. Al escribir d_1, d_2 y d_3 en términos de d_4 y d_5 , se obtiene $d_1 = d_4 + 20d_5$, $d_2 = -d_4 - 8d_5$ y $d_3 = -d_4 - 13d_5$. Haciendo arbitrariamente $d_4 = 1$ y $d_5 = 0$ se obtiene la dirección $\mathbf{d} = (1, -1, -1, 1, 0)^t$. Como $\mathbf{c}\mathbf{d} = 1 > 0$, el movimiento es a lo largo de la dirección $-\mathbf{d}$. Este movimiento es bloqueado por la restricción $\mathbf{x}_4 \geq 0$ a una longitud de paso 0.000015 con lo que se llega a una nueva solución $\bar{\mathbf{x}} = (0.2569531, 0.2972187, 0.2329805, 0, 0.2128477)^t$. Observe que $\mathbf{c}\bar{\mathbf{x}} = 0$, por lo que incidentalmente ya se ha llegado a una solución óptima, aunque no de punto de extremo. Continuando, ahora se tiene cuatro restricciones linealmente independientes conectantes. Al añadir la restricción $d_4 = 0$ a este sistema homogéneo, se obtiene $d_1 = 20d_5$, $d_2 = -8d_5$ y $d_3 = -13d_5$, en términos de d_5 . Haciendo arbitrariamente $d_5 = 1$ se obtiene la dirección $\mathbf{d} = (20, -8, -13, 0, 1)^t$. Como $\mathbf{c}\mathbf{d} = 0$ nos podemos mover a lo largo de la dirección \mathbf{d} . La restricción $\mathbf{x}_3 \geq 0$ bloque el movimiento a una longitud de paso 0.017921576, con lo que se llega a la solución revisada $\bar{\mathbf{x}} = (0.6153846, 0.1538461, 0, 0.2307692)^t$. Como en esta nueva solución $\bar{\mathbf{x}}$ hay cinco restricciones linealmente independientes conectantes, entonces se trata de una solución óptima de punto extremo. Hay que tener en cuenta que si en el paso anterior el movimiento se hubiera efectuado a lo largo de la dirección $-\mathbf{d}$, entonces se llegaría a la solución $(0, 0.4, 0.4, 0, 0.2)^t$, que es la otra de las dos soluciones óptimas de punto extremo alternativas de este problema.

Capítulo 4

Análisis del Algoritmo de Karmarkar.

Para realizar este análisis tomamos como referencia un capítulo del Bazaraa et al. (2011) en el cuál analiza en profundidad el funcionamiento del dicho algoritmo.

En este capítulo vamos a hablar sobre el análisis de convergencia y complejidad del algoritmo de karmarkar aplicado al al problema (3.5) bajo las hipótesis (A1) y (A2).

Al fin de demostrar la convergencia del algoritmo y establecer su complejidad polinómica, se emplean dos constructos. El primero es considerado una *relajación* del problema (3.9), que complementa su restricción (3.10), pero los resultados de este método no son muy satisfactorios, por lo que nos centraremos en el segundo método ya que nos asegura convergencia polinómico-temporal. Dicho método consiste en el uso de una novedosa función denominada *función potencial*, que mide el avance del algoritmo en un manera bastante ingeniosa y ayuda a establecer su complejidad polinómica, y se define como

$$f(x) = \sum_{j=1}^n \ln \left[\frac{\mathbf{c}\mathbf{x}}{x_j} \right] = n \ln(\mathbf{c}\mathbf{x}) - \sum_{j=1}^n \ln(x_j), \quad (4.1)$$

en donde $\ln(\Delta)$ denota el logaritmo Neperiano de base e . Obsérvese que aunque la transformaciones proyectivas (3.6) y (3.7) no preservan la linealidad de una función según es vista por los objetivos de las ecuaciones (3.5) y (3.8), sí preserva razones de funciones lineales. De hecho, bajo las ecuaciones (3.6) y (3.7) $\mathbf{c}\mathbf{x}/x_j = \mathbf{c}\mathbf{D}_k\mathbf{y}/x_{kj}y_j \equiv \bar{\mathbf{c}}\mathbf{y}/x_{kj}y_j$, de modo que la función potencia(4.1) se transforma como

$$F(y) \equiv f \left[\frac{\mathbf{D}_k\mathbf{y}}{\mathbf{1D}_k\mathbf{y}} \right] = \sum_{j=1}^n \ln \left[\frac{\bar{\mathbf{c}}\mathbf{y}}{x_{kj}y_j} \right] = n \ln(\bar{\mathbf{c}}\mathbf{y}) - \sum_{j=1}^n \ln(y_j) - \sum_{j=1}^n \ln(x_{kj}). \quad (4.2)$$

A continuación mediremos la disminución en el valor de la función potencial (4.1) o, de manera equivalente, en la función (4.2) en el espacio \mathbf{y} en la iteración \mathbf{k} . Usando la función

potencial (4.2), se obtiene

$$F(\mathbf{y}_{\text{nuevo}}) - F(\mathbf{y}_0) = n \ln \left[\frac{\bar{\mathbf{c}}\mathbf{y}_{\text{nuevo}}}{\bar{\mathbf{c}}\mathbf{y}_0} \right] - \sum_{j=1}^n \ln[ny_{\text{nuevo}j}],$$

ya que $y_0 j = 1/n$ para $j = 1, \dots, n$. Sin embargo, podemos acotar de la siguiente manera el $\ln[\bar{\mathbf{c}}\mathbf{y}/x_{kj}y_j]$ gracias a la ecuación (4.10)(lo analizaremos más adelante).

$$\ln \left[\frac{\bar{\mathbf{c}}\mathbf{y}}{x_{kj}y_j} \right] \leq \ln \left[1 - \frac{\alpha}{(n-1)} \right] \leq -\frac{\alpha}{(n-1)} \quad (4.3)$$

y por una propiedad de la función $\ln(\Delta)$ obtenemos

$$F(\mathbf{y}_{\text{nuevo}}) - F(\mathbf{y}_0) \leq -\frac{n\alpha}{(n-1)} - \sum_{j=1}^n \ln[ny_{(\text{nuevo})j}]. \quad (4.4)$$

Por consiguiente, si $-\sum_{j=1}^n \ln[ny_{(\text{nuevo})j}]$ no se incrementa demasiado en la relación con $n\alpha/(n-1)$, entonces se obtiene una caída suficiente en la función potencial. Lo anterior permite garantizar los resultados de convergencia deseados. De hecho al definir $\bar{\alpha} = n\alpha/(n-1)$ y elegir $0 < \alpha < 1$ suficientemente pequeño de modo que $\sqrt{(n-1)/n\bar{\alpha}} = \sqrt{n/(n-1)}\alpha < 1$, se demostrará que

$$\|n\mathbf{y} - \mathbf{1}\| \leq \sqrt{\frac{n-1}{n}}\bar{\alpha} < 1, \quad \mathbf{1}\mathbf{y} = 1, \quad \mathbf{y} > \mathbf{0},$$

implica que

$$0 \leq -\sum_{j=1}^n \ln[ny_j] \leq \frac{\bar{\alpha}^2}{2(1-\bar{\alpha})^2}. \quad (4.5)$$

Observe que $\mathbf{y}_{\text{nuevo}}$ satisface las condiciones en la ecuación (4.5), ya que $\mathbf{1}\mathbf{y}_{\text{nuevo}} = 1$, $\mathbf{y}_{\text{nuevo}} > \mathbf{0}$ y por las ecuaciones (3.11) y (3.12), $\|n\mathbf{y}_{\text{nuevo}} - \mathbf{1}\| = n\alpha r = \sqrt{(n-1)/n\bar{\alpha}}$. Usando la ecuación (4.5) en la ecuación (4.4), obtenemos

$$F(\mathbf{y}_{\text{nuevo}}) - F(\mathbf{y}_0) \leq -\bar{\alpha} + \frac{\bar{\alpha}^2}{2(1-\bar{\alpha})^2} \leq -\frac{1}{5} \quad \text{cuando } \bar{\alpha} \equiv \frac{n\alpha}{(n-1)} = \frac{1}{3}. \quad (4.6)$$

Por consiguiente, cuando $\bar{\alpha} = \frac{1}{3}$ la función $F(\cdot)$, y en consecuencia la función potencial $f(\cdot)$, disminuye $\frac{1}{5}$ en cada iteración. Lo anterior significa que al cabo de k iteraciones, $f(\mathbf{x}_k) - f(\mathbf{x}_0) \leq -k/5$. Sin embargo,

$$f(\mathbf{x}_k) - f(\mathbf{x}_0) = n \ln \left[\frac{\mathbf{c}\mathbf{x}_k}{\mathbf{c}\mathbf{x}_0} \right] - \sum_{j=1}^n \ln(nx_{kj}), \quad \text{ya que } \mathbf{x}_0 = \left(\frac{1}{n}, \dots, \frac{1}{n} \right).$$

De este modo obtenemos

$$n \ln \left[\frac{\mathbf{c}\mathbf{x}_k}{\mathbf{c}\mathbf{x}_0} \right] \leq \sum_{j=1}^n \ln(nx_{kj}) - \frac{k}{5}$$

Pero usando el hecho de que la media geométrica $[\prod_{j=1}^n (x_{kj})]^{1/n}$ no es mayor que la media aritmética $\mathbf{1}\mathbf{x}_k/n$ y que $\mathbf{1}\mathbf{x}_k = 1$, se tiene

$$\sum_{j=1}^n \ln(nx_{kj}) = n \ln \left[n \left(\prod_{j=1}^n x_{kj} \right)^{1/n} \right] \leq n \ln[\mathbf{1}\mathbf{x}_k] = 0.$$

En consecuencia, con base en la desigualdad previa, la expresión anterior significa

$$\ln \left[\frac{\mathbf{c}\mathbf{x}_k}{\mathbf{c}\mathbf{x}_0} \right] \leq -\frac{k}{5n} \quad \text{para } k = 0, 1, 2, \dots \quad (4.7)$$

Así, aunque el valor de la función objetivo pueda crecer de una iteración a otra, a medida que proceden las iteraciones se mantiene una disminución total suficiente a partir del valor objetivo original en $K = 0$. De hecho, por la ecuación (4.7), se tendrá que $\mathbf{c}\mathbf{x}_k < 2^L$ cuando $\mathbf{c}\mathbf{x}_k \leq (\mathbf{c}\mathbf{x}_0)e^{-k/5n} < 2^{-L}$. Por consiguiente, cuando $k = 10nL$, se tendrá

$$\mathbf{c}\mathbf{x}_k \leq (\mathbf{c}\mathbf{x}_0) \exp^{-k/5n} = \left(\sum_{j=1}^n \frac{c_j}{n} \right) (e^{-2L}) < (2^L)(2^{-L}) = 2^{-L}.$$

Además, cada iteración requiere no más de $O(n^3)$ cálculos, por lo que el algoritmo es de complejidad polinómica $O(n^4L)$. Lo que es más poderoso, obsérvese que

$$\mathbf{P}\mathbf{P}^t = \begin{bmatrix} \mathbf{A}\mathbf{D}_k^2\mathbf{A}^t & \mathbf{0} \\ \mathbf{0} & n \end{bmatrix},$$

y que el único cambio en esta matriz de una iteración a otra es en los elementos de matriz diagonal \mathbf{D}_k . De manera correspondiente, karmarkar demuestra cómo una ligera modificación del algoritmo, basada en actualizar en vez de volver a calcular una inversa idónea en la operación de proyección de gradiente, puede efectuarse con un esfuerzo de $O(n^{2.5})$ por iteración. Lo anterior da por resultado una complejidad polinómica reducida total de $O(n^{3.5}L)$ para el algoritmo modificado.

Para terminar de demostrar la complejidad polinómica de este algoritmo, nos quedará por demostrar la deducción de las ecuaciones (4.3) y (4.5). Vamos a comenzar por la ecuación (4.3) para esto debemos considerar el problema obtenido al añadir la restricción redundante $\mathbf{y} \in B(\mathbf{y}_0, R)$ al problema (3.9), aunque eliminando las condiciones de no negatividad, cabe destacar que $R = \sqrt{(n-1)n}$. Al realizar esta demostración observaremos muy brevemente el método de relajación citado anteriormente.

$$\text{Minimizar} \{ \bar{\mathbf{c}}\mathbf{y} : \mathbf{P}\mathbf{y} = \mathbf{P}_0, (\mathbf{y} - \mathbf{y}_0)^t(\mathbf{y} - \mathbf{y}_0) \leq R^2 \}. \quad (4.8)$$

De nuevo, la región factible en la ecuación (4.8) es una bola de dimensión $(n - m - 1)$ con centro en \mathbf{y}_0 , definida por la intersección de $B(\mathbf{y}_0, R)$ y el subespacio afín $\{\mathbf{y} : \mathbf{P}\mathbf{y} = \mathbf{P}_0\}$

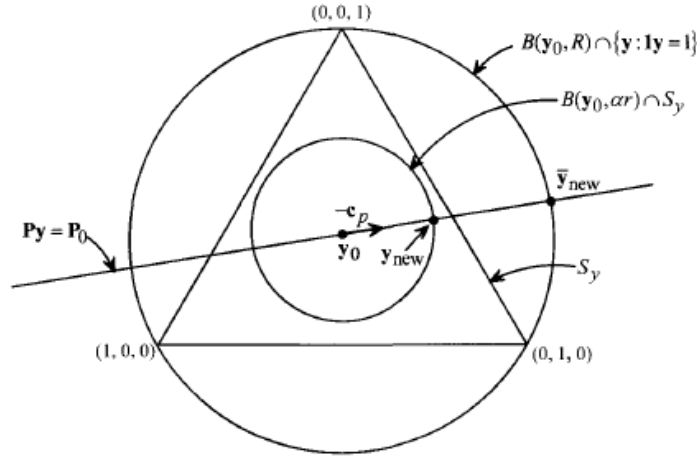


Figura 4.1: Relajación sobre la bola circunscrita.

de dimensión $(n - m - 1)$. En la figura 4.1 se muestra esta intersección para el caso en que $n = 3$ y $m = 1$. Por tanto, la solución óptima $\bar{\mathbf{y}}_{\text{nuevo}}$ del problema (4.8) se obtiene de manera semejante a como se hizo para la ecuación (3.11) y esta dada por

$$\mathbf{y}_{\text{nuevo}} = \mathbf{y}_0 - \frac{R\mathbf{c}_p}{\|\mathbf{c}_p\|}, \quad (4.9)$$

en donde \mathbf{c}_p se definió en la ecuación (3.12). Ahora es posible obtener una estimación del avance realizado con respecto a la función objetivo $\bar{\mathbf{c}}\mathbf{y}$ en el espacio \mathbf{y} como sigue. Antes de nada, primero tenemos que observar que por las ecuaciones (3.11) y (3.12) que en efecto, $\mathbf{c}_p \neq 0$, entonces $\bar{\mathbf{c}}\mathbf{y}_{\text{nuevo}} < \bar{\mathbf{c}}\mathbf{y}_0$, ya que $\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}_{\text{nuevo}}) = (\alpha r \bar{\mathbf{c}}\mathbf{c}'_p \|\mathbf{c}_p\|) = \alpha r \|\mathbf{c}_p\| > 0$. La última igualdad aquí se concluye por el hecho de que $\bar{\mathbf{c}}^t = \bar{\mathbf{c}}_p + \mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}\bar{\mathbf{c}}^t$ de la ecuación (3.12), y entonces $\bar{\mathbf{c}}\mathbf{c}_p = \|\mathbf{c}_p\|^2 + \bar{\mathbf{c}}\mathbf{P}^t(\mathbf{P}\mathbf{P}^t)^{-1}\mathbf{P}\mathbf{c}_p = \|\mathbf{c}_p\|^2$, ya que por definición $\mathbf{P}\mathbf{c}_p = 0$. De hecho denotando por \mathbf{y}^* a una solución óptima del problema (3.9), en virtud de que el problema (3.10) es una restricción de este problema y el problema (4.8) es una relajación de este problema, se tiene

$$\mathbf{c}\mathbf{y}_{\text{nuevo}} \leq \bar{\mathbf{c}}\mathbf{y}^* \leq \bar{\mathbf{c}}\mathbf{y}_{\text{nuevo}} < \bar{\mathbf{c}}\mathbf{y}_0.$$

Ademas, usando la expresión anterior junto con las ecuaciones (4.9) y (3.12), se obtiene

$$0 < \bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}_{\text{nuevo}}) \leq \bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}^*) \leq \bar{\mathbf{c}}(\mathbf{y}_0 - \bar{\mathbf{y}}_{\text{nuevo}}) = \frac{R}{\|\mathbf{c}_p\|} \bar{\mathbf{c}}\mathbf{c}_p = \frac{R}{\alpha r} \bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}_{\text{nuevo}})$$

Por tanto, lo anterior establece que $\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}^*) \leq (R/\alpha r)\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}_{\text{nuevo}}) = (R/\alpha r)[\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}^*) - \bar{\mathbf{c}}(\mathbf{y}_{\text{nuevo}} - \mathbf{y}^*)]$. Expresando $\bar{\mathbf{c}}(\mathbf{y}_{\text{nuevo}} - \mathbf{y}^*)$ en términos de $\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}^*) > 0$, se obtiene

$$\frac{\bar{\mathbf{c}}(\mathbf{y}_{\text{nuevo}} - \mathbf{y}^*)}{\bar{\mathbf{c}}(\mathbf{y}_0 - \mathbf{y}^*)} \leq 1 - \frac{\alpha r}{R} \equiv 1 - \frac{\alpha}{(n-1)}.$$

Observe que con la hipótesis (A2) se tiene que $\bar{\mathbf{c}}\mathbf{y}^* \equiv 0$, de modo que

$$\frac{\bar{\mathbf{c}}\mathbf{y}_{\text{nuevo}}}{\bar{\mathbf{c}}\mathbf{y}_0} \leq 1 - \frac{\alpha}{(n-1)}. \quad (4.10)$$

Finalmente para finalizar el capítulo nos queda deducir la ecuación (4.5). Vamos a considerar el siguiente problema de optimización, que permite acotar $-\sum_{j=1}^n \ln[ny_j]$ sujeta a las condiciones de la ecuación (4.5):

$$\text{Maximizar} \left\{ -\sum_{j=1}^n \ln[ny_j] : (n - \mathbf{y} - \mathbf{1})^t (n\mathbf{y} - \mathbf{1}) \leq \left(\frac{n-1}{n}\right) \bar{\alpha}^2, \mathbf{1}\mathbf{y} = 1, \mathbf{y} \geq \mathbf{0} \right\}. \quad (4.11)$$

En la primera restricción, obsérvese que como su lado derecho es menor que la unidad, entonces debe tenerse $\mathbf{y} > \mathbf{0}$ para la factibilidad del problema (4.11). Debido a que la función objetivo es equivalente a minimizar el producto $[y_1, y_2, \dots, y_n]$, es posible demostrar que en condiciones de optimalidad del problema (4.11), la primera restricción es conectante. Además, es posible demostrar que $1 \leq q \leq (n-1)$ componentes de una solución óptima son las mismas y que son iguales a un valor común U , por ejemplo, mientras las $(n-q)$ componentes restantes son iguales a algún valor común V , en donde $0 < U1/n < V < 1$. Como deben cumplirse $\mathbf{1}\mathbf{y} = 1$ y $(n\mathbf{y} - \mathbf{1})(n\mathbf{y} - \mathbf{1})^t = [(n-1)/n]\bar{\alpha}^2$, se obtiene $qU + (n-q)V = 1$ y $q(nU-1)^2 + (n-q)(nV-1)^2 = [(n-1)/n]\bar{\alpha}^2$. Con lo anterior se obtiene

$$nU = (1 - \theta_1), \quad nV = (1 + \theta_2),$$

en donde

$$\theta_1 = \sqrt{\frac{(1-1)(n-q)\bar{\alpha}}{q} \frac{\bar{\alpha}}{n}} \quad y \quad \theta_2 \equiv \frac{q\theta_1}{n-q}. \quad (4.12)$$

Observe que la solución óptima es positiva, como se requiere. Por tanto, para algún $1 \leq q \leq (n-1)$ se obtiene que el valor del objetivo en la ecuación (4.11) satisface

$$-\sum_{j=1}^n \ln[ny_j] \leq -q \ln[1 - \theta_1] - (n-q) \ln[1 + \theta_2]. \quad (4.13)$$

Con base en el conocido teorema de segundo orden de Taylor, para cualquier $0 < \theta < 1$ se obtiene que $\ln[1 - \theta] = -\theta - \theta^2/2\Delta^2$, en donde $(1 - \theta) < \Delta < 1$. A su vez, esto significa que $-\ln[1 - \theta] \leq \theta + \theta^2/2(1 - \theta)^2$. De manera semejante, $-\ln[1 + \theta] \leq -\theta + \theta^2/2$ para cualquier $0 < \theta_i < 1$ para $i = 1, 2$ y usando la ecuación (4.12), se obtiene

$$\begin{aligned} -\sum_{j=1}^n \ln[ny_j] &\leq -q \left[-\theta_1 + \frac{\theta_1^2}{2(1-\theta_1)^2} \right] + (n-q) \left[-\theta_2 + \frac{\theta_2^2}{2} \right] \\ &= \frac{q\theta_1^2}{2(1-\theta_1)^2} \left[\frac{n - q\theta_1(2 - \theta_1)}{n - q} \right] \end{aligned}$$

$$\begin{aligned} &= \frac{(n-1)}{n} \frac{\bar{\alpha}^2}{2(1-\theta_1)^2} \left[1 - \frac{q\theta_1(2-\theta_1)}{n} \right] \\ &\leq \frac{\bar{\alpha}^2}{2(1-\theta_1)^2} \leq \frac{\bar{\alpha}^2}{2(1-\bar{\alpha})^2}, \end{aligned}$$

ya que $0 < \theta_1 < \bar{\alpha} < 1$. Así se establece la ecuación (4.5) y se completa la deducción.

Bibliografía

BAZARAA, M. S., J. J. JARVIS, AND H. D. SHERALI (2011): *Linear programming and network flows*, John Wiley & Sons.

GONZÁLEZ-DÍAZ, J. (2017): *Programación Lineal y Entera*, Máster en Técnicas Estadísticas, apuntes de la asignatura de Programación Lineal y Entera.