Programación Lineal: Método Simplex

Optimización.

Gustavo A. Bula

Universidad Nacional de Colombia

March 5, 2024

Table of contents

1. Método Simplex

2. Método Simplex Matricial

Objetivos de sesión

- Programación Lineal: Método Simplex

Modelo de Programación Lineal

Minimizar $\quad z = c_1 x_x + c_2 x_2 + \cdots + c_n x_n$

Sujeto a:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \\
x_1, x_2, \ldots, x_n &\geq 0
\end{aligned}
$$

Modelo de Programación Lineal

Representación Matricial

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

donde

$$
\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
$$

$$
A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

$\mathbf{x} \in \mathbb{R}^{n\times 1}$, $\mathbf{b} \in \mathbb{R}^{m\times 1}$, $A \in \mathbb{R}^{m\times n}$ y $\mathbf{0} \in \mathbb{R}^{n\times 1}$

Solución del Problema P.L.

Se requiere que la matriz del problema esté en su forma aumentada. El problema puede ser descrito como sigue:
Maximizar z en:

$$\begin{bmatrix} 1 & -\mathbf{c}^T & 0 \\ 0 & \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{x}, \mathbf{x}_s \geq 0$$

donde x son las variables desde la forma estándar, x_s son las variables de holgura introducidas en el proceso de aumentación, c contiene los coeficientes de optimización, describe el sistema de ecuaciones contraídas, y z es la variable a ser maximizada.

BSF y un Método Ingenuo

1. Una solución óptima está localizada en un vertice
2. Un vértice es una Solución Básica Factible (BFS por las iniciales en inglés (Basic Feasible Solution)

Un algoritmo ingenuo

- Generar todas las soluciones básicas factibles
  - Seleccionar m variables básicas y llevar a cabo eliminación de Gauss
  - Probar que es factible.
- Seleccionar la solución básica factible (BFS) de menor costo (en un problema de minimización).

¿Cuántas soluciones básicas?

$$\binom{n}{m} = \frac{n!}{m!(m-n)!}$$

Esquema del algoritmo simplex

Objetivo: resolver un problema de programación lineal

1. Una solución óptima se ubica en un vértice.
2. Un vértice es una solución básica factible (BFS).
3. Se puede ir de una BFS a una BFS vecina.
4. Es posible detectar si un BFS es óptimo.
5. De cualquiera BFS, se puede pasar a una BFS con mejor(menor) costo.

Algoritmo Simplex

- Algoritmo de búsqueda local
  - pasar de una BFS a otra BFS
  - garantizado que se puede encontrar el óptimo global (gracias a la convexidad)
- Idea clave: cómo pasar de una BFS a otra BFS.

De una BFS a otra BFS

Cómo pasar a otra BFS:

- seleccionar una variable no básica con un coeficiente negativo: **variable de entrada**
- introducir esta variable en la base eliminando una variable básica: **variable de salida**
- realizar eliminación gaussiana

Movimiento local: intercambiar variables básicas y no básicas

¿Cómo escoger la variable de salida?

Debe mantenerse la **factibilidad**

$$l = \arg\min_{i : a_{ie} < 0} \frac{b_i}{-a_{ie}}$$

El movimiento Local

- Moviéndose de una BFS a otra BFS
  - seleccionar la variable de entrada x_e. variable no básica con coeficientes negativos en el lado derecho (RHS por las iniciales en inglés *right-hand side*)
  - seleccionar la variable saliente x_l para mantener la factibilidad

$$l = \arg\min_{i: a_{ie} < 0} \frac{b_i}{-a_{ie}}$$

  - Aplicar eliminación Gaussiana. Eliminar x_e de RHS
- Esta operación se llama pivotar en programación lineal. $pivotear(l, e)$

Detectar si una Solución es Óptima

Una BFS es óptima si su función objetivo, después de haber eliminado todas las variables básicas, es de la forma:

$$c_0 + c_1 x_1 + \ldots + c_n x_n$$

con

$$c_i \geq 0 \; 1 \leq i \leq n$$

De cualquiera BFS, se puede pasar a una BFS con mejor(menor) costo.

Supongamos que, en la BFS,

- $b_1 > 0, \ldots, b_m > 0$
- existe una variable entrante e con $c_e < 0$
- existe una variable saliente l

▷ entonces el movimiento pivote (e, l) está mejorando

Programa Lineal en Forma Estándar

- ¿En caso de maximización?
- ¿Que sucede si una variable puede tomar valores negativos?
- ¿Que sucede si existen restricciones en igualdad?
- ¿Que pasa si una variable puede tomar valores enteros?
- ¿Y si hay restricciones no lineales?

Programa Lineal en Forma Estándar

¿En caso de maximización?

Usar *minimizar* $-(c_1x_1 + \ldots + c_nx_n)$

¿Que sucede si una variable puede tomar valores negativos?

Reemplazar x_i por $x_i^+ - x_i^-$

¿Que sucede si existen restricciones en igualdad?

Usar dos desigualdades

Programa Lineal en Forma Estándar

¿Que pasa si una variable puede tomar valores enteros?

No es un problema de programación lineal, es programación lineal entera mixta.

¿Y si hay restricciones no lineales?

Es llamado un problema no lineal

Programa Lineal en Forma Estándar

Para convertir un PL en su forma estándar, cada una de las desigualdades debe ser reemplazada por una restricción de igualdad:

- Para convertir una restricción (a_i) de desigualdad $\leq$ en una restricción de igualdad se define una variable de holgura (*slack* s_i, $s_i \geq$), que es la cantidad de recurso que no es utilizado en la restricción i.
- Para convertir una restricción (a_i) de desigualdad $\leq$ en una restricción de igualdad se define una variable de exceso (*excess* e_i, $e_i \geq$), que es la cantidad de recurso que no es utilizado en la restricción i.

Modelo de Programación Lineal: forma canónica

Para el caso de la forma canónica de maximización:

- La función objetivo debe ser de maximización
- Las variables de decisión no negativas.
- Las restricciones son del tipo $\leq$.

$$
\begin{array}{ll}
\text{Maximizar} & z = c_1 x_x + c_2 x_2 + \cdots + c_n x_n \\
\text{Sujeto a:} & \\
& a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\
& x_1, x_2, \ldots, x_n \geq 0
\end{array}
$$

Modelo de Programación Lineal: forma canónica

Para el caso de la forma canónica de minimización:

- La función objetivo es minimizada.
- Las restricciones son de tipo $\geq$.
- Las variables de decisión son no negativas.

$$\begin{array}{ll} \text{Minimizar} & z = c_1x_x + c_2x_2 + \cdots + c_nx_n \\ \text{Sujeto a:} & \end{array}$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\geq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\geq b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \end{aligned}$$

Encontrando una **BFS** inicial

Una solución básica de $Ax = b$ se obtiene igualando $n - m$ variables (**variables no básicas**) a 0 y resolviendo los valores de las m variables restantes (**variables básicas**). Esto supone que establecer las $n - m$ variables iguales a 0 produce valores únicos para las m variables restantes o, de manera equivalente, las columnas para las m variables restantes son linealmente independientes.

Cualquier solución básica de la forma estándar $AX = b$ en la que todas las variables sean **no negativas** es una solución básica factible (BFS).

Encontrando una **BFS** inicial

¿Qué sucede si tenemos un problema de la forma?

$$
\begin{aligned}
\text{Minimizar} \quad & z = c_1 x_x + c_2 x_2 + \cdots + c_n x_n \\
\text{Sujeto a:} \quad & \\
& a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
& x_1, x_2, \ldots, x_n \geq 0
\end{aligned}
$$

Introducir variables artificiales y_i

Encontrando una **BFS** inicial

$$
\begin{array}{ll}
\text{Minimizar} & z = c_1x_x + c_2x_2 + \cdots + c_nx_n \\
\text{Sujeto a:} &
\end{array}
$$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m &= b_m \\
x_1, x_2, \ldots, x_n &\geq 0
\end{aligned}
$$

Se tiene una forma sencilla de encontrar una BFS pero es una solución a otro problema. Primero encontrar si existe una BFS y luego encontrar la BFS óptima.

Encontrando una **BFS** inicial

$$
\begin{array}{ll}
\text{Minimizar} & z = y_1 + y_2 + \ldots + y_m \\
\text{Sujeto a:} & \\
& a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 = b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m = b_m \\
& x_1, x_2, \ldots, x_n \geq 0
\end{array}
$$

Primero encontrar si existe una BFS: si todos los y_i son ceros tenemos una BSF sin los y_i

¿Cómo obtener una solución básica factible?

- Considere $Ax = b$
- Selecciones m columnas de A linealmente independientes, A_B

$$A_B x_b + A_N x_n = b$$

$$A_B x_b = b - A_N x_n$$

$$x_b = A_B^{-1} b - A_B^{-1} A_N x_n$$

$$A_B x_b = b' - A'_N x_n$$

- La solución es factible (BFS) si $b' \geq 0$ La matriz A_B es llamada una **base**

- Problema de programación lineal

$$minimizar \;\; cx$$

sujeto a:

$$Ax = b$$

- Una solución básica factible : Base B:

$$x_b = A_B^{-1}b - A_B^{-1}A_N x_n$$

- ¿Cuál es el costo para la base **B**?

$$cx = c_B x_B + c_N x_N$$

¿Cuál es el costo para la base **B**?

$$
\begin{aligned}
cx &= c_B x_B + c_N c_N \\
&= c_B(A_B^{-1}b - A_B^{-1}A_N x_n) + c_N x_N \\
&= c_B A_B^{-1}b - c_B A_B^{-1}A_N x_n + c_N x_N \\
&= c_B A_B^{-1}b + (c_N - c_B A_B^{-1}A_N)x_N \\
&= c_B A_B^{-1}b + (c_N - c_B A_B^{-1}A_N)x_N + (c_B - c_B A_B^{-1}A_B)x_B \\
&= c_B A_B^{-1}b + (c - c_B A_B^{-1}A)x
\end{aligned}
$$

Define $\Pi = c_B A_B^{-1}$ multiplicadores del método simplex

$$cx = \Pi b + (c - \Pi A)x$$

Probar que una base es óptima

$$cx = c_B A_B^{-1} b + (c - c_B A_B^{-1} A)x$$

$$cx = \Pi b + (c - \Pi A)x$$

Probar que una base es óptima es verificar que $c \geq \Pi A$, $(c - \Pi A)$ son no-negativos

Probar que una base es óptima

Sea $c* = c - \Pi A$ and $c_0* = \Pi b$. Dado que $c*$ son no negativos,

$$c \geq \Pi A$$

Si se considera cualquier solución factible y. Tendremos que esta satisface

$$Ay = b$$

Entonces,

$$cy \geq \Pi A y = \Pi b = c_0*$$

Algoritmo Simplex

- Mientras $\exists 1 \leq i \leq n : c_i < 0$ hacer:
    - Escoja e de tal forma que $\bar{c} < 0$ para *minimización*
    - $l = \arg\min\limits_{i : a_{ie} > 0} \frac{\bar{b}_i}{\bar{a}_{ie}}$
    - intercambie e (variable entrante) por l (variable saliente)

Prueba de optimalidad

Probar que una base es óptima para un problema de minimización es verificar que $c \geq \Pi A$. Los costos reducidos, $\bar{c} = (c - \Pi A)$, son no-negativos

$$\Pi = c_B A_B^{-1}$$

$$\bar{c} = c - c_B A_B^{-1} A$$

Prueba de factibiliad

seleccionar la variable saliente x_l para mantener la factibilidad

$$l = \arg\min_{i:a_{ie}>0} \frac{\bar{b}_i}{\bar{a}_{ie}}$$

donde

$$\bar{b} = A_B^{-1} b$$

y

$$\bar{a}_e = A_B^{-1} A_{Ne}$$

El Algoritmo Simplex

Para el caso de minimización

- Supongamos que durante la ejecución
  - $b_1, \ldots, b_m$ son siempre estrictamente positivos
  - la función objetivo está limitada por debajo
- entonces el algoritmo simplex termina con una solución óptima

Terminación del Algoritmo Simplex

Existen varios enfoques:

- Regla blanda: seleccione siempre la primera variable que entra lexicográficamente.
- Regla de pivoteo lexicográfica: romper los empates cuando se selecciona la variable que sale usando una regla lexicográfica:

$$l = \underset{i:a_{ie}<0}{arg - lex - min} \frac{b_i}{-a_{ie}}$$

- Métodos de perturbación