

Tarea II implementación del método Simplex

Joel Chacón Castillo

Abstract—El método simplex es utilizado para resolver problemas de programación lineal formulados en su forma estándar, este método es convertido en un problema de combinatoria ya que se desea encontrar el punto óptimo básico factible que corresponde a un vértice del polítopo. Aunque existen variantes de este algoritmo, en este trabajo se presenta una forma simple e intuitiva que a largo plazo puede ser optimizado bajo con ciertas estrategias que conciernen al tema de métodos numéricos.

I. INTRODUCTION

PROGRAMACIÓN lineal tiene un número de funciones objetivo y restricciones lineales, las cuales pueden estar conformadas tanto por igualdades como desigualdades. El conjunto factible es un polítopo, el cual es un conjunto convexo conectado con caras poligonales planas [1]. Los programas lineales usualmente son establecidos y analizados en la siguiente forma estándar:

La forma estándar de un problema de programación lineal es:

$$\min \quad c^T \vec{x}, \quad \text{sujeito a:} \quad A\vec{x} = \vec{b}, \quad \vec{x} \geq 0 \quad (1)$$

La forma dual equivalente es representado de la forma:

$$\max \quad \vec{b}^T \vec{\pi}, \quad \text{sujeito a:} \quad A^T \vec{\pi} \leq \vec{c} \quad (2)$$

A. Condiciones de optimalidad

$$\begin{aligned} \mathcal{L}(x, \pi, s) &= c^T x - \pi^T (Ax - b) - s^T x \\ \text{sujeito a} \\ A^T \pi + s &= c \\ Ax &= b \\ x &\geq 0 \\ s &\geq 0 \\ s_i x_i &= 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (3)$$

donde π es el coeficiente de Lagrange.

II. MÉTODO/ALGORITMO

A. Conversión de la forma Primal a la Dual

Para representar la forma dual desde la forma primal se realiza el proceso: Se puede encontrar la forma dual desde la forma primal realizando varias sustituciones en el lagrangiano, esto es:

$$\begin{aligned} \mathcal{L}(x, \pi, s) &= c^T x - \pi^T (Ax - b) - s^T x \\ &= (c^T - A^T \pi - s)x + \pi^T b \end{aligned} \quad (4)$$

Por lo tanto la función Dual se obtiene si:

$$q(\pi, s) = \inf_x \mathcal{L}(x, \pi, s) = (c^T - A^T \pi - s)x + \pi^T b \quad (5)$$

Algorithm 1 Simplex

```

1: Dados  $B_i, N_i, x_B = B^{-1}b \geq 0, x_N = 0$ 
2:  $max\_ite = mCn$  (Combinación)
3: while  $max\_ite > i$  do
4:   Resolver  $B^T \pi = c_B$  para  $\pi$ .
5:   Calcular  $s_N = c_N - N^T \pi$ .
6:   if  $s_N \geq 0$  then
7:     Salir, óptimo encontrado.
8:   Seleccionar  $q \in N$  con  $s_q < 0$  siendo el índice que entrará al conjunto básico.

9:   Resolver  $Bt = A_q$  para  $t$ .
10:  if  $t \leq 0$  then
11:    Salir, el problema no está limitado.
12:  Calcular  $x_q^+ = \min_{i|t_i > 0} (x_B)_i / t_i$ , definir el índice  $p$  como el índice que
    corresponde al mínimo valor logrado.
13:  Actualizar  $x_B^+ = x_B - tx_q^+, x_N^+ = (0, \dots, 0, x_q^+, 0, \dots, 0)$ 
14:  Cambiar el conjunto básico de índices  $B_i$  agregando a  $q$  y en el lugar de  $p$ 
15:   $i = i + 1$ 

```

Entonces se tiene su cota inferior al considerar primer condición $(c^T - A^T \pi - s) = 0$ por lo tanto como resultado se tiene $q(\pi, s) = \pi^T b$.

B. Implementación del Algoritmo Simplex

Este método itera en los puntos básicos factibles que corresponden al polítopo factible. Los puntos básicos factibles corresponden a un vértice del polítopo, por lo tanto de una forma greedy en cada iteración se visita cada vértice adyacente hasta encontrar el punto factible óptimo. En caso de que el problema no esté limitado el procedimiento no terminará, por lo tanto es recomendable asignar un número máximo de iteraciones. En este procedimiento se dividen el espacio que corresponde al problema, donde se considera la base B y sus índices B_i donde cada uno indica la i -ésima columna de la matriz de restricciones A .

El complemento de B_i es denotado como N_i esto es $N_i = \{1, 2, \dots, b\} \setminus B_i$. También se particionan los vectores x, s, c en

$$\begin{aligned} x_{B_i} &= [x_i]_{i \in B_i} & x_{N_i} &= [x_i]_{i \in N_i} \\ s_{B_i} &= [s_i]_{i \in B_i} & s_{N_i} &= [s_i]_{i \in N_i} \\ c_{B_i} &= [c_i]_{i \in B_i} & c_{N_i} &= [c_i]_{i \in N_i} \end{aligned} \quad (6)$$

Dadas las condiciones se tiene que $Ax = Bx_b + Nx_n = b$. Las variables primales x para este simplejo iterado se definen de la forma $x_B = B^{-1}b, x_N = 0$. Se pueden encontrar los elementos restantes π y s_N asignando $s_B = 0$ obteniendo $B^T \pi = c_B$ y $N^T \pi + s_N = c_N$. Desde que B es cuadrada y no singular, se obtiene $\pi = B^{-T} c_B$ y se obtiene $s_N = c_N - N^T \pi = c_N - (B^{-1} N)^T c_B$.

En este método se deben tener en cuenta tres aspectos importantes:

- Se puede mantener una factorización LU o QR de la matriz básica B con el fin de resolver π y t , ya que si $LU = B$, entonces primero se puede resolver $L\hat{t} = A_q$ y después $U\hat{t} = \hat{t}$

- Se puede implementar algún método más sofisticado para elegir el índice q como el componente negativo de s_N , por lo regular existen varios componentes negativos.
- Es importante manejar los pasos degenerados, en donde $x_q^+ = 0$.

El método simplex requiere un punto básico factible y el conjunto básico de índices $B_i \subset \{1, 2, \dots, n\}$ con $B_i = m$ donde:

- La matriz B es no singular y de dimensión $m \times m$.
- $x_B \geq 0$ y $x_N = 0$, donde N_i es el complemento de B_i .

Para encontrar esta información se puede implementar el procedimiento *Phase-I/Phase-II*. Este procedimiento consiste de dos fases, en la fase I se construye un problema auxiliar, este es distinto a la forma estándar, este problema es resuelto por medio del método de simplex. el problema de la fase I es construida de tal forma que es sencillo calcular los puntos iniciales básicos factibles. Por lo tanto la solución de la fase I proporciona un conjunto básico factible para la fase II.

Posteriormente, en la fase II se resuelve un problema muy similar a la forma estándar, iniciado desde la solución obtenida en la fase I. Por lo tanto la solución del problema se puede obtener de la solución obtenida en la fase II.

Particularmente, en la fase I se introducen las variables artificiales z , y se re-define la función objetivo como la suma de las variables artificiales.

$$\min \quad e^T \vec{z}, \quad \text{sujeto a:} \quad A\vec{x} + E\vec{z} = \vec{b}, \quad (\vec{x}, \vec{z}) \geq 0 \quad (7)$$

Específicamente, el problema de la fase I es:

donde $z \in \mathbb{R}$, $e = (1, 1, \dots, 1)^T$, y E es la matriz diagonal donde los elementos diagonales son $E_{jj} = +1$ si $b_j \geq 0$, $E_{jj} = -1$ si $b_j < 0$. Se observa que el punto (x, z) es definido por

$$x = 0, z_j = |b_j|, j = 1, 2, \dots, m \quad (8)$$

En cualquier punto factible de (7), las variables artificiales z representan las cantidades por las cuales se violan las restricciones $Ax = b$ por los componentes de x , y la función objetivo esta comprendida simplemente por la suma de estas violaciones, por lo tanto minimizando esta suma se está forzando que x se convierta factible para el problema original (estándar).

En general se aplica el método simplex a (7) con el punto inicial (8). Si se obtiene una solución óptima para la cual la función objetivo $e^T z$ es positiva, se concluye que el problema original no es factible. De otra forma el simplex identifica un punto factible para el problema lineal de la segunda fase

$$\min \quad \vec{c}^T \vec{x}, \quad \text{sujeto a:} \quad A\vec{x} + \vec{z} = \vec{b}, \quad \vec{x} \geq 0, \quad 0 \geq z \geq 0 \quad (9)$$

Este problema es equivalente a la forma estándar, ya que cualquier solución debería tener $z = 0$, es importante mantener las variables artificiales z en la fase II. Además, para evitar iteraciones adicionales, es posible quitar variables artificiales en la fase II.

Es posible optimizar este procedimiento por medio de varias estrategias de álgebra lineal, esta implementación se puede considerar como la ingenua “naive”. Inclusive existen estrategias para seleccionar de forma más eficiente el índice que se ingresará en el conjunto básico de índices.

Algorithm 2 Algoritmo Simplex de dos fases

- 1: Construir la función objetivo que corresponde a la ecuación (7).
- 2: **Fase I**
- 3: Asignar $B_i = z_i$, es decir el conjunto base está conformado por los índices de las variables artificiales indicado en la ecuación 8.
- 4: Obtener x_B, x_N, B_i, N_i el algoritmo 1, donde A está conformada por $[A \quad z]$.
- 5: **Fase II**
- 6: Construir la función objetivo que corresponde a la ecuación (9).
- 7: Obtener x_B con el algoritmo 1, es importante usar la matriz A , los índices del conjunto básico B_i , el punto básico factible x_B que se obtuvieron en la fase I.

C. Análisis de rendimiento del método simplex

Para analizar el rendimiento del método Simplex se desarrolló un procedimiento para generar problemas de Programación Lineal (PL). Es importante considerar que la matriz A debe ser de rango completo $\text{rank}(A) = m$, para realizar esto existen distintas estrategias:

- Generar una matriz de forma aleatoria y reducir a su forma triangular con el método de Gauss-Jordan.
- Generar una matriz de forma aleatoria y obtener la base con el procedimiento de Gram Schmidt, posteriormente agregar las m columnas que sean múltiples de la base.
- Generar una matriz de forma aleatoria, verificar si es de rango completo en caso de que no lo sea volver a generar otra matriz.
- Generar un vector \vec{X} de forma aleatoria, construir una matriz de la forma $A = \vec{x}^T \vec{x}$, agregar columnas con factores múltiples a la matriz A .

Existen otros factores que pueden influir en este proceso, como es el caso en que la matriz está conformada por una gran cantidad de ceros que dependiendo del método para resolver el sistema de ecuaciones (para t y π) puede ser más rápido. Además también es importante observar el rango de los números aleatorios, ya que si el rango es elevado se pueden tener problemas de precisión.

Para relajar los problemas de precisión se consideran números enteros de forma aleatoria.

Los experimentos de rendimiento consisten en dos grupos:

- Generar aleatoriamente en su forma estándar para $n = 200 * i$ con $i \in \{1, 2, \dots, 50\}$ y $m = n/2$.
- Generar aleatoriamente en su forma estándar para $n = 2000$ y $m = 200 * i$ con $i \in \{1, 2, \dots, 10\}$

El procedimiento para generar la matriz de forma aleatoria corresponde a la última opción de la mencionadas anteriormente donde se genera un vector con números aleatorios y posteriormente se multiplican. El experimento se ejecutó en el cluster “El insurgente”, una observación interesante es que la configuración de python automáticamente realiza la paralelización en los productos vectoriales, la forma en que se paraleliza esta parte es con memoria compartida. La máquina central donde se realizaron las pruebas consta de 24 procesadores (físicos 12 con hyperthreading), la memoria RAM disponible es de 126 Gigabytes. La versión de python es 2.7.12+. El número máximo de iteraciones asignado fue asignado como $\min(mCn, 1000000)$ ya que si el número de bases posibles asignadas es a lo más mCn , es decir el número de formas de elegir m elementos como base de n índices posibles, y aunque existe un número finito de iteraciones

en esta configuración el número de combinaciones pueden aumentar de forma exponencial.

Se puede observar en la figura 1 el comportamiento incremental del tiempo conforme aumenta el número de variables y restricciones, en teoría el límite superior en este caso es $O(n^3)$ ya que no se realiza el cálculo de la inversa $x_B = B^{-1}b$ y por lo tanto no son necesarios demasiados recursos del sistema. Aun que popularmente existen variantes del método Simplex como es el tradicional tableau [2], donde de forma continua se realizan operaciones entre filas y columna considerado como una variante del método de Gauss. La implementación que se presenta en este trabajo está basado en el libro de Jorge Nocedal, y puede ser requerir más operaciones ya que se resuelven dos sistemas de ecuaciones, la primera para calcular los valores π del lagrangiano y la segunda para el cálculo de la “Dirección” t . No obstante es importante notar que se puede implementar alguna factorización (LU , QR entre otros) teniendo un menor coste computacional, siendo al final equivalente al procedimiento Tableau, y en algunos casos (sistemas ralas o sparse) más eficiente. Como ya se mencionó el vector basico inicial x_B es generado en la fase I, y para esta primer fase el punto factible ya está definido.

D. Problema de asignación de ambulancias aéreas

El objetivo es determinar una signación de costo mínimo de helicópteros a los sitios que satisfacen la demanda proyectada para el siguiente periodo de tiempo.

1) *Declaración del problema:* El servicio de ambulancia tiene un conjunto de ubicaciones con un número de helicópteros en cada sitio. Cada sitio requerido es visitado por su respectivo helicóptero. Existe un costo en la reasignación de un helicóptero a un sitio distinto. Por lo tanto, el servicio proporcionado desea determinar un costo de asignación mínimo de helicópteros a los sitios para satisfacer la demanda en el siguiente periodo de asignación proyectado en cada sitio.

E. Formulación matemática

Dado un conjunto de ubicaciones, las distancias entre las ubicaciones, el número de helicópteros recientemente asignados a cada ubicación, la demanda proyectada para cada ubicación y el costo por kilómetro, El objetivo del problema de asignación de ambulancias es determinar el costo mínimo en la reasignación para satisfacer la demanda proyectada. El problema puede ser formulado como un problema de programación lineal porque la función objetivo y las restricciones son todas funciones lineales. Los pares de distancia son calculadas con la distancia Euclídea.

Asignar: L = el conjunto de ubicaciones.

Parámetros:

- x_i = la coorenada X para la ubicación de $i, \forall i \in L$.
- y_i = la coorenada Y para la ubicación de $i, \forall i \in L$.
- d_i = demanda proyectada para el siguiente periodo de la ubicación $i, \forall i \in L$.
- s_i = número de helicópteros recientemente asignados a la ubicación $i, \forall i \in L$.
- c = costo de transportar por kilómetro.

TABLE I
INFORMACIÓN DE LOS HELICÓPTEROS ASIGNADOS

	X - Coord	Y-Coord	# Assigned	Projected Demand
Location #1	36	20	6	7
Location #2	23	30	2	3
Location #3	23	56	3	2
Location #4	10	15	3	4
Location #5	5	5	4	2

- $dist_{ij}$ = Distancia Euclídea entre la ubicación i y la ubicación $j, \forall i \in L, \forall j \in L$.
- $dist_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$.

Variables: z_{ij} = número de helicópteros a ser desplazados de la ubicación i a la $j, \forall i \in L, \forall j \in L$.

Función objetivo:

$$\text{Minimizar} \sum_{(i,j) \in L \times L} dist_{ij} * c * z_{ij} \quad (10)$$

Restricciones: Restricciones de balance para cada ubicación i en el conjunto L .

$$\sum_{j \in L} z_{ij} + s_i = d_i, \forall i \in L \quad (11)$$

Como ejemplo, se considera un servicio de ambulancias aéreas con 5 ubicaciones en Midwest. La empresa tiene la necesidad de reubicar a los helicópteros basados en la demanda mensual proyectada para cada ubicación. El costo de reasignar un helicóptero del sitio i al sitio j es de 100 pesos por kilómetro. Se muestra la tabla de cada ubicación.

La solución al problema es: $z_{3,2} = 1, z_{5,1} = 1$, y $z_{5,4} = 1$ con un valor de la función objetivo de 7161.87. La distancia están como sigue: $dist_{3,2} = 26, dist_{5,1} = 34.438$, y $dist_{5,4} = 11.180$.

REFERENCES

- [1] J. Nocedal and S. J. Wright, *Sequential quadratic programming*. Springer, 2006.
- [2] R. G. Bland, “New finite pivoting rules for the simplex method,” *Mathematics of operations Research*, vol. 2, no. 2, pp. 103–107, 1977.

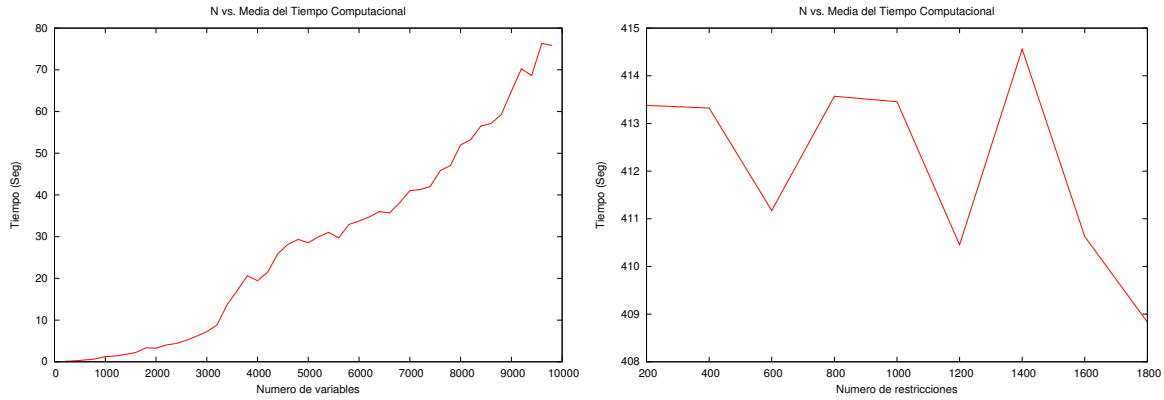


Fig. 1. En la izquierda la media contra tiempo con $n = 200 * i$ e $i \in \{1, 2, \dots, 50\}$ donde $m = n/2$ y en la derecha la media contra tiempo con $n = 2000$ donde $m = 200 * i$.

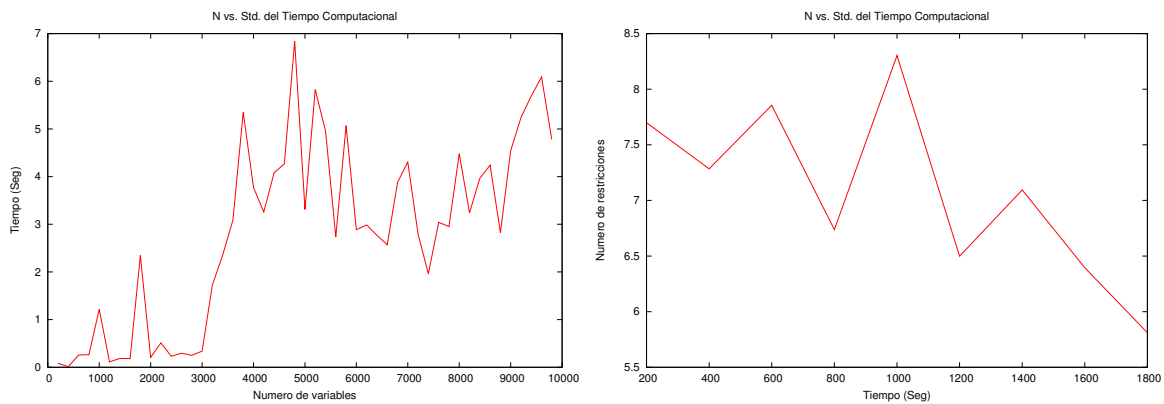


Fig. 2. En la izquierda la varianza contra tiempo con $n = 200 * i$ e $i \in \{1, 2, \dots, 50\}$ donde $m = n/2$ y en la derecha la varianza contra tiempo con $n = 2000$ donde $m = 200 * i$.