Pogramación lineal
Algoritmo Símplex

Santiago Blasco Arnaiz

Noviembre 2018

Índice

1. Introducción

La programación lineal es un método para optimizar un sistema matemático representado por una función lineal y restringido por un conjunto de inecuaciones, estas forman una región factible en la que está definida la función objetivo. El objetivo de los algoritmos de programación lineal es encontrar el punto de dicha región factible dónde se optimiza la función objetivo.
El método más utilizado para realizar a cabo esta función es conocido como método Símplex.

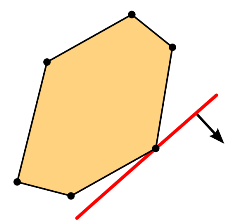

Figura 1: Región factible

2. Algoritmo Símplex

El algoritmo Símplex fue creado por George Bernard Dantzig, físico y matemático estadounidense, en el año 1947 para resolver problemas de progrmación lineal que involucraran más de dos variables lo que transforma la región factible en la que se encuentra el punto solución en un politopo como el mostrado en la figura 2.

2.1. Problema del algoritmo

El algoritmo Símplex examina las máximos existentes, que son los vertices del politopo anteriormente nombrado y va recorriendolos en el sentido que indica la función objetivo hasta encontrar el óptimo.
Para trabajar con este algoritmo debemos pasar las restricciones dadas por las inecuaciones a forma estándar para asegurar que el programa lineal tenga un óptimo en la región factible, el paso a forma estándar consiste en transformar las desigualdades en igualdades valiendonos de variables básicas, tambien llamadas variables de holgura o slack. Una vez hecha esta transformación procedemos a construir la tabla Símplex con la que obtendremos el óptimo del problema si este es alcanzable.

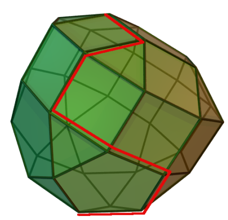

Figura 2: Recorrido de los vértices de un politopo.

Dicho optimo puede no ser alcanzable si los lados del politopo no son finitos, ya que el algoritmo se basa en visitar vértices si un lado en al menos una dirección es infinito no posee vertice porque nunca alcanza otro punto extremo, por tanto el programa lineal no tiene solución.

2.2. Solución mediante programación lineal

Para explicar el procedimiento del algoritmo Símplex nos valdremos del ejemplo de programación lineal mostrado en la figura 3 cuyo objetivo es máximizar. El primer paso como ya he dicho es escribir en forma normal las inecuaciones, para ello se añaden las variables de holgura, estas formaran una matriz identiad en la tabla Símplex como se ve en la figura 4. Realmente el valor de la variable de holgura depende de la condición de desigualdad si es $\leq$ tendrá valor 1 en dicha matriz, si es $\geq$ tendrá valor -1, pero en esta implementación como explicaré en el apartado sobre el planteamiento de los problemas sólo trabajo con maxmizaciones por tanto transformo todas las desigualdades de $\geq$ en desigualdades $\leq$ para simplificar la implementación.

```
Función a maximizar
Z= 20000 X + 20000 Y + 20000 Z + 20000 W

Inecuaciones
2X + 1Y + 1Z + 2W <= 24
2X + 2Y + 1Z <= 20
2Z + 2W <= 20
4W <= 16
```

Figura 3: Problema ejemplo

Continuamos rellenando la tabla con los coeficientes de las ecuaciones y en base a estos calcularemos la contribución total de cada variable solución (Zj), que se obitne multiplicando el valor del coeficiente de cada variable por el valor que poseen en la solución), al restarsela al coeficiente que tiene cada una de las variables en la función objetivo (Cj) obtendremos un índice de mejora o utilidades percibida (Cj-Zj), como estamos inicializando la tabla obtenemos una solución básica inicial que no es valida ya que únicamente contiene variables básicas, para mejorar esta solución debemos realizar iteraciones en la tabla, en cada iteración se realizan los siguientes pasos:

	Cj		20000.00	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00	
Cb	Variable solución	Solución	X	Y	Z	W ⇧	S0	S1	S2	S3	Divisiones
0.00	S0	24.00	2.00	1.00	1.00	2.00	1.00	0.00	0.00	0.00	12.00
0.00	S1	20.00	2.00	2.00	1.00	0.00	0.00	1.00	0.00	0.00	NA
0.00	S2	20.00	0.00	0.00	2.00	2.00	0.00	0.00	1.00	0.00	10.00
0.00	S3 ⇦	16.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	1.00	4.00
	Zj		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	Cj-Zj	0	20000.00	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00	

Figura 4: Tabla inicial de Símplex

- **Decidir la variable que entra.**
  Esta decisión se toma según los valores de la última fila de la tabla, la Cj-Zj. Para saber qué variable entra seleccionamos la variable de la columna que mayor Cj-Zj tenga, en este caso valen todas 20.000 así que nos quedamos con la última, la de W, aunque siempre que coincidan podemos coger cualquiera. A esta columna la llamaremos columna pivote. Seleccionamos la mayor porque esta será la que mayor rendimiento aporte.

- **Decidir la variable que sale.**
  Para esta decisión se tiene en cuenta la columna de la variable que acaba-

mos de seleccionar y los valores de restricción (columna Solución), calculamos la división de cada valor de Solución ente su correspondiente de la columna de la variable que entra, la variable que salga será la que se encuentre en la fila del menor valor positivo de esas divisiones, en este caso la variable básica S3. Esto es así porque de seleccionar una distinta de la menor estaríamos quebrantando alguna de las restricciones. A la fila escogida la llamaremos fila pivote.

- **Cálculo de la fila pivote**
  Habiendo ya cambiado ambas variables procedemos a dividir la fila pivote (Solución y todos los coeficientes de las ecuaciones) por el elemento ubicado en la intersección de la fila y columna pivote, en este caso 4.

- **Cálculo del resto de filas**
  Para calcular el resto de filas debemos coger en cada una el elemento que se encuentra en la columna pivote y multiplicarlo por la fila pivote que acabamos de calcular, la nueva fila será el resultado de restarle a la antigua el resultado del cálculo que acabamos de describir.

- **Calculo de las utilidades percibidas por cada variable**
  Este paso es volver a calcular la fila Cj-Zj con los nuevos valores.

Tras realizar estos pasos la tabla Símplex queda como se muestra en la figura 5 la variable que entra es la Z y la que sale la S2, continua iterando porque existen elementos mayores que 0 en la fila Cj-Zj.

	Cj		20000.00	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00	
Cb	Variable solución	Solución	X	Y	Z	W	S0	S1	S2	S3	Divisiones
0.00	S0	16.00	2.00	1.00	1.00	0.00	1.00	0.00	0.00	-0.50	16.00
0.00	S1	20.00	2.00	2.00	1.00	0.00	0.00	1.00	0.00	0.00	20.00
0.00	S2	12.00	0.00	0.00	2.00	0.00	0.00	0.00	1.00	-0.50	6.00
20000.00	W	4.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.25	NA
	Zj		0.00	0.00	0.00	20000.00	0.00	0.00	0.00	5000.00	
	Cj-Zj	80000.0	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00	-5000.00	

Figura 5: Tabla Símplex tras una iteración

Todavía queda iterar otra vez porque como vemos existen elementos mayores que 0 en la fila Cj-Zj, la variable que entra es la Y y la que sale es S1.

	Cj		20000.00	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00	
Cb	Variable solución	Solución	X	Y	Z	W	S0	S1	S2	S3	Divisiones
0.00	S0	10.00	2.00	1.00	0.00	0.00	1.00	0.00	-0.50	-0.25	10.00
0.00	S1	14.00	2.00	2.00	0.00	0.00	0.00	1.00	-0.50	0.25	7.00
20000.00	Z	6.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	-0.25	NA
20000.00	W	4.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.25	NA
	Zj		0.00	0.00	20000.00	20000.00	0.00	0.00	10000.00	0.00	
	Cj-Zj	200000.0	20000.00	20000.00	0.00	0.00	0.00	0.00	-10000.00	0.00	

Figura 6: Tabla Símplex tras dos iteraciones

Se deja de iterar cuando todos los elementos de la fila Cj-Zj son menores o iguales a 0, esto indica que la solución es óptima, no se puede obtener una mejor

	Cj		20000.00	20000.00	20000.00	20000.00	0.00	0.00	0.00	0.00
Cb	Variable solución	Solución	X	Y	Z	W	S0	S1	S2	S3
0.00	S0	3.00	1.00	0.00	0.00	0.00	1.00	-0.50	-0.25	-0.38
20000.00	Y ⇦	7.00	1.00	1.00	0.00	0.00	0.00	0.50	-0.25	0.12
20000.00	Z ⇦	6.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	-0.25
20000.00	W ⇦	4.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.25
	Zj	⇩	20000.00	20000.00	20000.00	20000.00	0.00	10000.00	5000.00	2500.00
Fin	Cj-Zj	340000.0	0.00	0.00	0.00	0.00	0.00	-10000.00	-5000.00	-2500.00

Figura 7: Tabla Símplex tras tres iteraciones

contribución de las variables.

El algoritmo termina, todos los elementos de Cj-Zj son menores o iguales a 0, como podemos ver en la figura 7 la ganancia es de 340.000 y las variables solución son Y,Z y W con los valores que tienen a su lado. (Las variables básicas no se tienen en cuenta ya que su valor es 0, no contribuyen al resultado)

2.3. Planteamiento de los problemas

La programación lineal se encarga tanto de maximizar como de minimizar funciones, como ya he comentado antes para facilitar la implementación del algoritmo, transformaré aquellos problemas que pidan minimizar un función sujeta a unas restricciones en un problema de maximización equivalente. Esto es posible gracias a la teoría de la dualidad que dice que todo problema de programación lineal, llamado primal, lleva asociado otro que se conoce como dual, con el resultado de este segundo se pude obtener el del primero.

Transformando los problemas de minimización en problemas de maximización evito tener que cambiar los criterios para escoger las variables que entran y salen en cada iteración de la tabla así como la condición de fin, además al transformarlas en un problema de maximizacion trabajo con restricciones que llevan igualdades del tipo $\leq$ lo cual facilita más aún la implementación no teniendo que alterar el valor de las variables de holgura, será siempre positivo.

En nuestro caso para obtener el problema de maximización dual primero hacemos que todas las desigualdades del problema de minimización sean del tipo $\geq$, a continuación proyectamos el problema en una matriz como vemos

$$MTec = \begin{bmatrix} -1 & -1 & 0 & 0 & -120 \\ 0 & 0 & -1 & -1 & -250 \\ 1 & 0 & 1 & 0 & 200 \\ 0 & 1 & 0 & 1 & 150 \\ 425 & 525 & 475 & 500 & 0 \end{bmatrix}$$

a esta matriz se la llama matriz tecnológica y para obtener el problema dual deberemos invertirla y hacer el proceso inverso de proyección, como puede verse en la figura 8 los terminos independientes de las restricciones pasan a ser los coeficientes de la función objetivo dual. Al transformar el problema todas las desigualdades se invierten, he ahi el motivo por el que cambiar todas al mismo tipo antes de transformar. En la figura 8 también podemos ver como primero cambiamos el signo de las inecuaciones y a continuación realizamos la transformación al problema de maximización dual.

Función a minimizar
Z= 425 X + 525 Y + 475 Z + 500 W

Inecuaciones
X + Y <= 120
Z+W<= 250
X+Z>=200
Y+W>=150

Función a minimizar
Z= 425 X + 525 Y + 475 Z + 500 W

Inecuaciones
-X - Y >= -120
-Z-W>= -250
X+Z>=200
Y+W>=150

Función a maximizar
Z= -120X-250Y+200Z+150W

Inecuaciones
-X+Z<= 425
-X+W<= 525
-Y+Z<= 475
-Y+W<= 500

Figura 8: Trnasición de problema primal al dual

El procedimiento para la tabla Símplex es el mismo ya que estamos maximizando, pero estamos obteniendo la solución para el problema dual, para conseguir la solución del primal debemos fijarnos en los valores Cj-Zj de las variables básicas, como vemos en la figura 9 estos valores son negativos, debemos quedarnos con el módulo de ellos y asignar su valor a la variable correspondiente del problema primal.

	Cj		-120.00	-250.00	200.00	150.00	0.00	0.00	0.00	0.00
Cb	Variable solución	Solución	X	Y	Z	W	S0	S1	S2	S3
200.00	Z	475.00	0.00	-1.00	1.00	0.00	0.00	0.00	1.00	0.00
0.00	S1	75.00	0.00	0.00	0.00	0.00	-1.00	1.00	1.00	-1.00
-120.00	X	50.00	1.00	-1.00	0.00	0.00	-1.00	0.00	1.00	0.00
150.00	W	500.00	0.00	-1.00	0.00	1.00	0.00	0.00	0.00	1.00
	Zj		-120.00	-230.00	200.00	150.00	120.00	0.00	80.00	150.00
Fin	Cj-Zj	164000.0	0.00	-20.00	0.00	0.00	-120.00	0.00	-80.00	-150.00

Figura 9: Solución al problema de minimización

3. Pseudocódigo

```
función simplex (función, inecuaciones)
    for i = 0 hasta tamaño inecuaciones:
        variablesObejtivo = variables de la función y de holgura
        valoresObjetivo = valores de las variables de la función y de holgura
        variablesSolucion = variables de holgura
        valoresSolucion = termino independiente de las inecuaciones
        valoresVariables = valor de las variables de holgura (0)
        variables = valor de las variables en las inecuaciones
    for i = 0 hasta tamaño variablesObjetivo:
        control = valoresObjetivo– (su correspondiente de variables * valoresVariables)
    mientras todos los elementos de control no sean negativos hacer:
        a = número de la columna en la que se encuentra el mayor valor de control
        for i = 0 hasta tamaño variablesSolución:
            divisiones = elemento de valoresSolucion / correspondiente elemento de
            la columna a de variables
        b = número de la fila en la que se encuentra el menor de divisiones
        Se divide la fila b de variables entre el elemento [b][a] de variables
        for i = 0 hasta tamaño variables:
            Fila de variables se le resta la fila b multiplicada por el elemento a de la fila
            en cuestión
        for i = 0 hasta tamaño variablesObjetivo:
            Se vuelven a realizar los cálculos de control
```

Figura 10: Pseudocódigo del algoritmo

3.1. Análisis de coste computacional

El estudio del coste de este algoritmo va sujeto a ciertas condiciones ya que no todos los problemas de programación lineal tienen una solución, si no la tuviera tardaría un tiempo infinito en calcularla. Partiendo del hecho de que estudiamos el coste de encontrar una solución para un problema de programación lineal que la tiene y no buscamos que esta sea entera, ya que esto aumenta más aún la complejidad por tener que aproximar a una solución exacta con los cálculos que esto conlleva, nos encontramos con un orden de complejidad cuadrático O(n^2) en el caso promedio, este coste viene dado por el número de variables, para n variables en el problema repetiremos los calculos de la tabla un máximo de n veces y estos cálculos tienen una complejidad de O(n). Se itera máximo n veces porque el algoritmo no recorre todos los vértices del politopo, tan sólo recorre los vertices que mejoran respecto al anterior la solución y esto es gracias a que sigue la dirección de la función objetivo.

```
función simplex (función, inecuaciones)
   for i = 0 hasta tamaño inecuaciones:
         variablesObejtivo = variables de la función y de holgura
         valoresObjetivo = valores de las variables de la función y de holgura
         variablesSolucion = variables de holgura                                   O(n)
         valoresSolucion = termino independiente de las inecuaciones
         valoresVariables = valor de las variables de holgura (0)
         variables = valor de las variables en las inecuaciones
   for i = 0 hasta tamaño variablesObjetivo:
         control = valoresObjetivo– (su correspondiente de variables * valoresVariables)   O(n)
   mientras todos los elementos de control no sean negativos hacer:
         a = número de la columna en la que se encuentra el mayor valor de control
         for i = 0 hasta tamaño variablesSolución:
               divisiones = elemento de valoresSolucion / correspondiente elemento de   O(n)
               la columna a de variables
         b = número de la fila en la que se encuentra el menor de divisiones
         Se divide la fila b de variables entre el elemento [b][a] de variables          O(n²)
         for i = 0 hasta tamaño variables:
               Fila de variables se le resta la fila b multiplicada por el elemento a de la fila   O(n)
               en cuestión
         for i = 0 hasta tamaño variablesObjetivo:
               Se vuelven a realizar los cálculos de control   O(n)
```

Figura 11: Coste del algoritmo

3.2. Análisis de coste espacial

El cálculo del coste espacial se limita a conocer el tamaño de la tabla Símplex y este viene dado por el número de variables más alguna lista auxiliar para los cálculos que se realizan, el coste simplificado responde a O(n^2).

Referencias

[1] Facultad de Ingenieria-Universidad de la República - Uruguay https://www.fing.edu.uy/inco/cursos/io/archivos/teorico/todo.pdf

[2] G.Brassard and P. Bratley, *Fundamentos de algoritmia*

[3] Explicación algoritmo Símplex y ejemplificación https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigación-de-operaciones/método-simplex/

[4] Algoritmo Símplex, imágenes e información acerca de él. https://en.wikipedia.org/wiki/Simplex_algorithm

[5] Apuntes Programación Lineal, asignatura Algoritmos y Computación, Universidad de Valladolid, curso 2018/2019