Primer proyecto, programación lineal

Sergio Arnaud 000159189
Saúl Alvarez 000158318

18 de marzo de 2018

1. Introducción

Un programa lineal en forma general es un problema de la forma:

Encontrar $\mathbf{x}$ tal que $c^T\mathbf{x}$ sea mínimo sujeto a $A\mathbf{x} \leq b$ y $x \geq 0$ (y encontrar el valor de $c^T\mathbf{x}$ en este caso). Tanto A, como b y c son constantes y conocidos.

Para este tipo de problemas existen tres casos:

- Puede existir al menos una solución que minimiza $c^T\mathbf{x}$
- Puede suceder que esta cantidad se pueda hacer arbitrariamente pequeña, y que por lo tanto el problema no esté acotado
- Puede suceder que no haya ninguna $\mathbf{x}$ que cumpla las restricciones. En este caso, se dice que el problema no tiene solución factible.

Para este proyecto, se garantiza que $b \geq 0$, así que las restricciones se cumplen con $\mathbf{x} = 0$, y por lo tanto, la tercera opción está descartada.

Para resolver programas lineales, nosotros implementamos el método del simplex, usando el paso de Bland. El formato de entrada y salida de el algoritmo es el solicitado en los lineamientos del proyecto.

2. Implementación

En general, se utilizan los paquetes Numpy y Scipy. para Instalarlos correr pip install numpy, pip install scipy o checar la documentación de python.

La implementación de simplex se encuentra en el Script mSimplexFaseII.py y la función a llamar es solve(A,b,c).

Los tests de el método Simplex se encuentran en el Script testFaseII.py, al ejecutar dicho script se realizan dos problemas realizados en clase y se compara la solución con la otorgada por el método linprog de la librería scipy de python. Posteriormente se realizan problemas aleatorios y se compara la solución de python con la solución obtenida por nuestro método.

Con respecto a el segundo problema, la función generaKleeMinty en el script generaKleeMinty.py obtiene A,b y c de el programa correspondiente y se debe ejecutar el script SimplexKleeMinty.py para obtener los resultados de la tabla.

Para obtener los resultados del tercer problema se debe ejecutar SimplexEmpirico.py. Éste script utiliza el packete "Bokeh"para realizar la gráfica por lo que se debe correr el siguiente comando en terminal para instalar el paquete: pip install bokeh.

3. Problema 2

El ejemplo de Klee-Minty

El segundo problema consistía en implementar el problema de Klee-Minty para $n = 3...10$ y reportar el número de iteraciones junto con el tiempo necesitado por la computadora para resolverlo.

El problema de Klee-Minty es un programa lineal en el que se busca encontrar:

$$\min \left\{ -\sum_{i=1}^{n} x_i \right\}$$

$$s.a. \quad 2\sum_{j=1}^{i-1} x_j + x_i \leq 2^i - 1$$

$$i \in \{2, \ldots, n\}$$

Tomando en cuenta, por supuesto, las restricciones de positividad:
$x_i \geq 0 \ \forall i \in \{1, \ldots, n\}$

Los resultados arrojados por nuestra implementación son los siguientes:

n	Número de iteraciones	Tiempo de CPU
3	7	0.00824284553527832
4	15	0.0016398429870605469
5	31	0.0031511783599853516
6	63	0.00633692741394043
7	127	0.013965129852294922
8	255	0.03334498405456543
9	511	0.06293106079101562
10	1023	0.12653493881225586

La relación entre tiempo de ejecución y número de iteraciones es complétamente esperada. Al duplicar el tamaño del problema, se duplica el número de iteraciones y en general, el tiempo de ejecución se duplica.

Lo más interesante de este resultado es que en todos los casos probados, el número de iteraciones es 2^n para n variables. Esto nos lleva a realizar una conjetura:

$\forall\, n \in \mathbb{N}$ el método Simplex con la regla de pivoteo de Bland realiza 2^n cambios de variables básicas

Esta conjetura es cierta y nos ayuda a determinar que la complejidad computacional del algoritmo del Simplex, al menos con la regla de Bland, es mayor o igual a 2^n.

Esto es algo que debe ser tomado en consideración al resolver un programa lineal de dimensiones *grandes* pues existe la posibilidad de que el número de iteraciones necesarias para resolver dicho problema sea exponencial.

Para analizar un poco más a fondo las implicaciones de el crecimiento exponencial tomemos el siguiente ejemplo. El problema de Klee-Minty con n=40 realiza 1099511627775 operaciones y de seguir la relación mostrada experimentalmente entre número de operaciones y tiempo de ejecución se esperaría un tiempo de ejecución de 6 años.

Sin embargo, el siguiente ejercicio mostrará que, en el caso promedio, dicho comportamiento es súmamente raro por lo que en general, no se espera un tiempo de ejecución tan alto.

4. Problema 3

Un estudio empírico de la complejidad computacional del método Simplex.

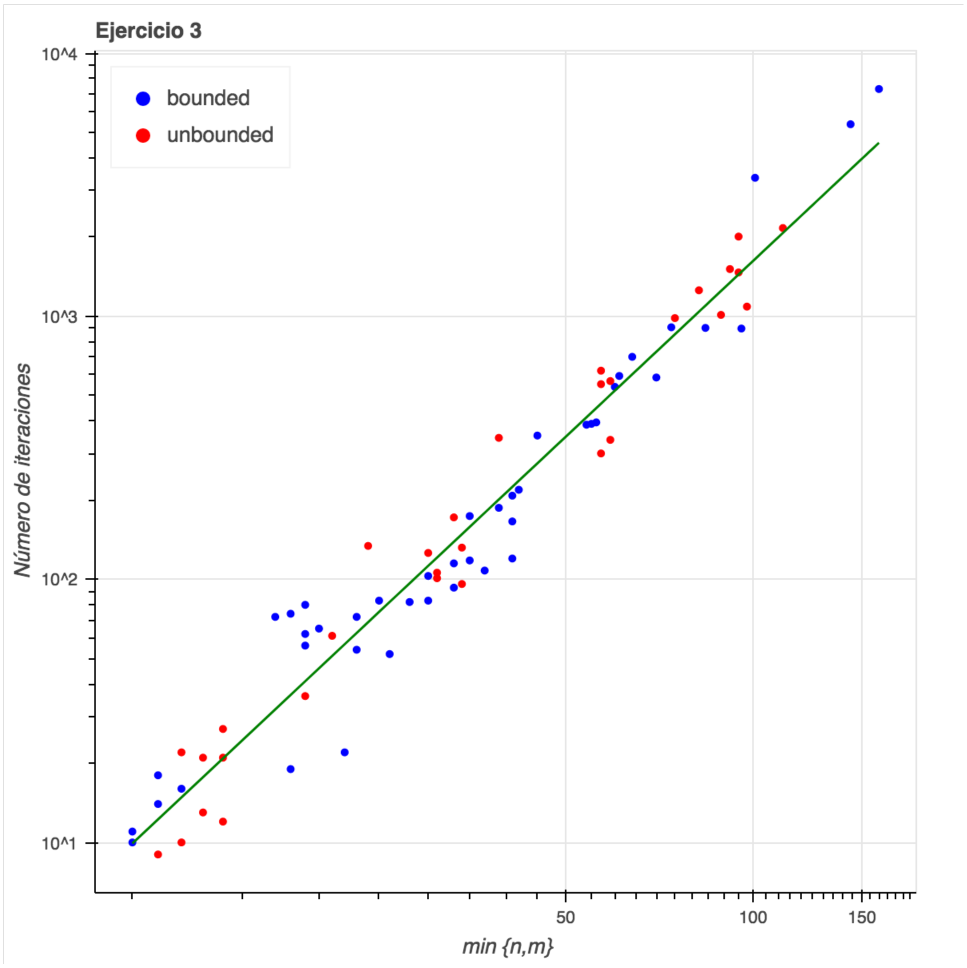
Para realizar un estudio empírico de la complejidad computacional del método Simplex resolveremos 75 problemas de optimización lineal de la forma:

$$\min \left\{c^T x\right\}$$
$$Ax <= b$$
$$x_i \geq 0 \ \forall i \in \{1, \ldots, n\}$$

Donde A,b,c tienen dimensiones y entradas creadas aleatoriamente de la siguiente manera:

1. m,n son generadas por la siguiente expresión : $10e^{\log 20r}$ donde r es un número uniformemente distribuido en el intervalo cero 1
2. Las entradas de A,b,c son generadas por la siguiente expresión $100r$ donde r es una instancia de una distribución normal(0,1)

Al correr nuestra implementación de simplex con 75 problemas de optimización generada de dicha manera, almacenando los valores de n,m, el número de iteraciones y si el problema era o no acotado y realizar la gráfica log,log con eje X la dimensión $min\{m, n\}$ y en el eje Y el número de iteraciones para cada caso obtuvimos la siguiente gráfica



Es bastante clara la relación lineal (en la gráfica log-log) entre $min\{m, n\}$ y el número de iteraciones.

Más aún, se realizó el ajuste de una recta a dichos datos donde se determinó que la mejor solución al problema de mínimos cuadrados con tales datos está dada por la recta con pendiente $m = 2.2117694336401894$ y ordenada $b = 1.215677503056605$

De esta forma, tenemos:

$$log(y) = mlog(x) + b$$

Entonces,

$$10^{log(y)} = 10^{mlog(x)+b}$$

De forma que,

$$y = 10^b x^m$$

Y por lo tanto

$$O(y) = O(10^b x^m) = O(x^m)$$

Con esto, concluimos que el número de iteraciones tiene una complejidad aproximada de $O(min\{m,n\}^{2.2117..})$, es decir, en general el número de iteraciones del algoritmo simplex es cuadrático con respecto al mínimo entre m y n.

Esto es algo súmamente positivo ya que, pese a que en el peor caso podemos esperar 2^n iteraciones, en el caso promedio no será asi pues en el caso general se realizan n^2 operaciones.