Modelos de Optimización
Laboratorio 4

Osmany Pérez Rodríguez
Enrique Martínez González
Carmen Irene Cabrera Rodríguez
Grupo C412 Equipo 9

1 Métodos numéricos para problemas de optimización no lineales

Solucionar problemas de optimización que involucran funciones no lineales (ya sea la función objetivo, o las restricciones a las que está sujeto el problema) es un proceso complicado dado que, en la mayoría de los casos, requiere la solución de sistemas complejos de ecuaciones no lineales. De hecho, en la práctica, hallar el mínimo o máximo global de una función no-lineal cualquiera es un problema abierto en matemáticas. Los métodos existentes sólo obtienen mínimo locales.

Entre los métodos numéricos que existen con este propósito se pueden mencionar:

- Método de máximo descenso
- Método de Newton
- Métodos de Quasi-Newton(BFGS, DFP, Gradiente Conjugado)
- Método de región de confianza
- Método de penalidad
- Método de barrera
- Método de secuenciación cuadrática

En particular, los métodos que se tomaron en consideración para la solución de los problemas orientados fueron los últimos tres, puesto que estos son métodos para problemas con restricciones, como es el caso; y los otros métodos son para problemas irrestrictos. Se verá a continuación un breve análisis de estos tres algoritmos.

Suponga que se tiene el problema de programación no lineal:

$$\begin{aligned} & min f(x) \\ & s.a : g_j(x) \leqslant 0, j = 1, \ldots, m \\ & \quad h_i(x) = 0, i = 1, \ldots, n \end{aligned}$$

Con $f, g, h \in C^1(\Omega)$; considerando a Ω el conjunto de soluciones factibles; es decir, $\Omega = \{x \in \mathbb{R} : h_i(x) = 0, g_j(x) \leqslant 0\}$

1.1 Método de Penalidad

La idea de este método es minimizar f sobre $\mathbb{R}^n$, aplicándole una penalización a los puntos que no están en Ω.

Se define la función de penalización cuadrática como:

$$Q(x, c) = f(x) + c \sum_{i=1}^{n} h_i^2(x) + c \sum_{j=1}^{m} g_j^+(x)^2$$

Donde c es el *parámetro de penalización* y $g_j^+ = max\{0, g_j(x)\}$. Se considera la secuencia $\{c_k\}$ tal que $c_k \to \infty$ cuando $k \to \infty$; a medida que aumente c, la violación de las restricciones será castigada con mayor severidad. Luego basta hallar el minimizador x_k de $Q(x, c_k)$ para cada k utilizando algún método de optimización irrestricto. Cualquier punto límite de la sucesión de puntos $\{x_k\}_{k\in\mathbb{N}}$ obtenidos con el método de penalidad es solución del problema $minf(x), x \in \Omega$.

Se debe tener en cuenta a la hora de aplicar este método, que si se aumenta el valor de c_k demasiado rápido, puede causar inestabilidad numérica y divergencia de los métodos de optimización; por ello, se debe comenzar con un c pequeño y aumentarlo de forma gradual.

Este método resulta de gran utilidad, puesto que es aplicable con generalidad a problemas de optimización con restricciones, ya sean de igualdad o desigualdad. Además, el punto inicial se puede seleccionar de forma arbitraria, no necesita pertenecer al conjunto de soluciones factibles. No obstante, dado que el método itera por la región no factible del problema, tanto el costo como las funciones de restricción pueden estar indefinidos. Además, si el proceso de iteración termina de forma prematura (por haber alcanzado el másimo número de iteraciones) el punto obtenido finalmente puede no ser factible y por tanto, no sería solución del problema.

1.1.1 Implementación

Dadas las funciones f, h_i, g_j, el punto inicial $x_0 \in \mathbb{R}^n$ y $c_0 \in \mathbb{R}, \alpha \in \mathbb{R}_{>1}, \varepsilon, k_{max}$ se presenta el siguiente pseudocódigo para el algoritmo:

1. Definir $Q(x, c_k)$
2. $x_{k+1} = minQ(x, c_k)$ utilizando método de optimización irrestricta que asume como punto inicial x_k
3. Si $x_{k+1} \in \Omega \Rightarrow$ PARAR
4. Si no:

$$\begin{aligned} c_{k+1} &= \alpha c_k \\ k &= k + 1 \end{aligned}$$

5. Si $||x_k - x_{k+1}|| > \varepsilon$ y $k \leqslant k_max$ repetir el proceso desde el punto 1. De lo contrario PARAR.

El código que contiene la implementación de este método se puede acceder a través del siguiente enlace.

1.2 Método de Barrera

A diferencia del algoritmo anterior, la idea del método de barrera es aproximarse a los puntos de la frontera de Ω desde su interior. Si se consigue una función B tal que :

- B es continua
- $B(x) \geqslant 0$
- $B(x) \to \infty$ cuando x se acerca a la frontera de Ω

Entonces se puede resolver el problema $minf(x) + \mu B(x)$ con métodos de optimización irrestricta. entre las funciones de barrera más comunes está $B(x) = \sum_i -\frac{1}{g_i(x)}$, que fue la utilizada en la implementación del método. De este modo, se transforma el problema en minimizar la función

$$R(x, \mu) = f(x) + \mu B(x)$$

Análogo al método de penalidad, cualquier punto límite de la sucesión de puntos $\{x_k\}_{k\in\mathbb{N}}$ generada por este método es solución del problema.

Note que dado que el algoritmo se mueve por los puntos interiores de Ω, este método es aplicable a problemas con restricciones dadas por desigualdades, ya que en particular, es necesario que $int\Omega \neq \varnothing$. En particular, este método es aplicable a los ejercicios que se resolverán a continuación, puesto que las restricciones de los mismos están dadas por desigualdades.

Otra ventaja que provee la utilización de este método, es que típicamente se soluciona usando métodos de Newton para problemas irrestrictos, que convergen rápidamente si se inicializa cerca de la solución.

1.2.1 Implementación

La implementación de este método es análoga a la del método de penalidad.

Dadas las funciones f, g_i, el punto inicial $x_0 \in \mathbb{R}^n$ y $\mu_0 \in \mathbb{R}, \alpha \in \mathbb{R}_{<1}, \varepsilon, k_{max}$ se presenta el siguiente pseudocódigo para el algoritmo:

1. Definir $R(x, \mu_k)$
2. $x_{k+1} = minR(x, \mu_k)$ utilizando método de optimización irrestricta que asume como punto inicial x_k
3. $$\begin{aligned} \mu_{k+1} &= \alpha\mu_k \\ k &= k + 1 \end{aligned}$$
4. Si $||x_k - x_{k+1}|| > \varepsilon$ y $k \leqslant k_max$ repetir el proceso desde el punto 1. De lo contrario PARAR.

Se puede acceder al código que contiene la implementación a través del siguiente enlace.

Ambos métodos anteriores requieren solucionar un subproblema de optimización irrestricto. En ambos casos se decidió utilizar el método BFGS, una variante de los métodos Quasi-Newton.

1.2.2 Método BFGS (Broyden-Fletcher-Goldfarb-Shanno)

El algoritmo de Newton tiene la ventaja de lograr la convergencia cuadrática si el punto inicial está cerca del óptimo; de lo contrario, la dirección puede no ser de descenso y diverger. Además, en cada iteración es necesario calcular la inversa de la matriz hessiana de f en x_k de manera exacta, lo cual es costoso computacionalmente. Por esta razón se proponen los métodos de QuasiNewton, que permiten construir iterativamente la matriz S_k de manera que se verifique las siguientes condiciones:

- Definida positiva: Si S_k es definida positiva $\Rightarrow S_{k+1}$ también lo es
- Aproximación de la matriz hessiana inversa de f: $x_k \to x^*, S_k \to [\nabla^2 f(x^*)]^{-1}$ para $k \to \infty$

La forma de construir estos métodos asegura:

- Convergencia global
- Rapidez de convergencia mayor que lineal
- Convergencia a un mínimo local

Puede encontrar una implementación del método en el siguiente enalce; no obstante, para la resolución de los ejercicios propuestos se utilizó la función *minimize* de la biblioteca de Python *Scipy.minimize*, pasando como argumento el solver *'BFGS'*. Se profundizará un poco en esta herramienta más adelante.

1.3 Método de Secuenciación Cuadrática

Los métodos de secuenciación cuadrática (*SQP - Sequential Quadratic Programming*) son altamente utilizados en la vida real, al ser capaces de manejar cualquier nicel de no linealidad, incluidas las restricciones. Sin embargo, su principal desventaja es que el método incorpora numerosas derivaciones, que probablemente se necesiten trabajar de forma analítica, lo que resulta un problema al trabajar con muchas variables en problemas muy grandes.

La idea de este método consiste en transformar el problema dado en los problemas cuadráticos $QP(x_k, \lambda_k, \mu_k)$ dados por:

$$\begin{aligned} min f(x_k) + \nabla f(x)^T d + \frac{1}{2} d^T [\nabla^2 L_k] d \\ s.a h_i(x_k) + \nabla h_i(x_k)^T d = 0; i = 1, \ldots, n \\ g_j(x_k) + \nabla g_j(x_k)^T d \leqslant 0; j01, \ldots, m \end{aligned}$$

donde

$$\nabla^2 L_k = \nabla^2 f(x_k) + \sum_{i=1}^{n} (\lambda_k)_i \nabla^2 h_i(x_k) + \sum_{j=1}^{m} (\mu_k)_j \nabla^2 g_j(x_k)$$

Dada la complejidad y el costo de este algoritmo es mejor utilizarlo en problemas de alta no linealidad en la función objetivo y sus restricciones; que no es el caso de los problemas propuestos. No obstante, se brinda una solución para los problemas con este método a través de la biblioteca *Scipy* de Python; utilizando como *solver* el método *'SLSQP'*.

1.4 SciPy.optimize-minimize

Scipy es un ecosistema basado en Python de código abierto destinado a software matemáticos, a la ciencia y la ingeniería. En particular, la biblioteca *optimize* de SciPy provee métodos para la minimización (o maximización) de funciones objetivos, sujetas o no a restricciones. Incluye solucionadores para problemas no lineales (con soporte para algoritmos de optimización local y global), programación lineal, para hallar raíces, entre otros.

El método utilizado para la solución de los problemas fue *minimize*; que permite pasar como parámetros:

- la función objetivo a ser minimizada
- x_0 el punto inicial
- *method*: tipo de solucionador. Presenta variados tipos de solucionadores, entre ellos: BFGS, que fue el utilizado en los métodos de penalidad y barrera para resolver el problema irrestricto, y SLSQP. Este últimmo minimiza la función dada utilizando el método *Sequential Least Squares Quadratic Programming*, una de las variaciones de los métodos de secuenciación cuadrática.
- la matriz jacobiana de la función objetivo
- la matriz hessiana de la función objetivo
- las restricciones
- *options* entre las cuales se puede setear, por ejemplo, la máxima cantidad de iteraciones que realizará el algoritmo.

Para más información con respecto a esta herramienta puede consultar la documentación a través del siguiente enlace