MA37A Optimización. Semestre 2007-2
Profesor: Alejandro Jofré **Auxiliar:** Sebastián Court, Julio Deride, Oscar Peredo.

Tarea #2
Optimización con Restricciones

31 de Octubre del 2007

Fecha de Entrega: 19 de Noviembre del 2007

1. Objetivos

El objetivo de esta tarea es implementar un método para resolver un problema de optimización con restricciones en $\mathbb{R}^n$. En particular, se implementará el método de Zoutendijk descrito en [1], que pertenece a la clase de los métodos de direcciones de descenso factibles.

El método de Zoutendijk construye una dirección de descenso a partir de un punto x_k y luego realiza una búsqueda lineal a lo largo de esa dirección, para obtener un nuevo punto x_{k+1}. La dirección de descenso d_k es la solución de un problema de programación lineal que depende del gradiente de la función objetivo y la región factible activa.

El problema modelo que se desea resolver es el siguiente:

$$(P): \begin{cases} \min_{x\in\mathbb{R}^n} & f(x) \\ & Ax \leq b \\ & Ex = e \end{cases}$$

donde $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^n, E \in \mathbb{R}^{l\times n}$ y $e \in \mathbb{R}^l$.

Antes de explicar el algoritmo, connviene explicar la siguiente notación:

$$A^T = [A_1^T, A_2^T],\ b^T = [b_1^T, b_2^T]$$

cuando se evaluan las restricciones $Ax \leq b$ en un punto x factible, algunas de las filas pueden estar activas o no activas, es decir, $A_{i,\cdot}x = b_i$ ó $A_{i,\cdot}x < b_i$, en este caso y siempre que ocurra esto, se denotaran las filas activas de A como A_1 y las no activas como A_2, con respecto al punto factible x.

El algoritmo es el siguiente:

Encontrar un punto factible x_1, es decir, $Ax_1 \leq b, Ex_1 = e$.
$k \leftarrow 1$.
repeat
  Dado x_k, descomponer A^T en $[A_1^T, A_2^T]$ y b^T en $[b_1^T, b_2^T]$ como se explica en el enunciado, de manera que se $A_1 x_k = b_1$ y $A_2 x_k < b_2$.
  Resolver el siguiente problema lineal:

$$(D_k) : \begin{cases} \min_{d \in \mathbb{R}^n} & \nabla f(x_k)^T d \\ & A_1 d \leq 0 \\ & Ed = 0 \\ & |d_i| \leq 1 \qquad \forall i = 1, ..., n \end{cases}$$

  Llamemos d_k a la solución de este problema.
  if $\nabla f(x_k)^T d_k = 0$ **then**
    PARAR. x_k es un punto KKT del problema (P).
  else
    Calcular $\lambda_k = \text{argmin}\{f(x_k + \lambda d_k) : 0 \leq \lambda \leq \lambda_{\text{máx}}\}$, donde $\lambda_{\text{máx}}$ se calcula de la siguiente manera:

$$\lambda_{\text{máx}} = \begin{cases} \text{mín} \left\{ \dfrac{\hat{b}_i}{\hat{d}_i} : \hat{d}_i > 0 \right\} & \hat{d} \not\leq 0 \\ \infty & \hat{d} \leq 0 \end{cases}$$

    donde $\hat{b} = b_2 - A_2 x_k$ y $\hat{d} = A_2 d_k$.
    $x_{k+1} = x_k + \lambda_k d_k$
    $k \leftarrow k + 1$
  end if
until Se cumpla el criterio de parada

La implementación del método debe realizarse en ambiente MATLAB, desarrollando todas las funciones que estime necesarias.

IMPORTANTE: Puede serle útil investigar acerca de la función `linprog` incluida en el toolbox de optimización de MATLAB (viene por defecto en todas las intalaciones). Esta función resuelve problemas de programación lineal con restricciones de igualdad y desigualdad.

2. Entregables

La implementación se debe realizar como una función de MATLAB llamada `zou`, en un archivo llamado `zou.m`. La función debe ejecutarse de la siguiente manera:

```
>>[x,f]=zou(epsilon,i,A,b,E,e)
```

Los parámetros de entrada de `zou` son:

- `epsilon`: Tolerancia aceptada.
- `i`: Índice de la función a minimizar. Puede utilizar la función `func_original.m` vista en el Laboratorio Presencial 1 si lo estima necesario (o modificarla).
- `A`: Matríz A asociada a las restricciones de desigualdad. En caso de que el problema (P) no tenga restricciones de desigualdad, se debe ingresar la matríz de ceros de $\mathbb{R}^{m \times n}$.

- `b`: Vector b asociado a las restricciones de desigualdad. En caso de que el problema (P) no tenga restricciones de desigualdad, se debe ingresar el vector de ceros de $\mathbb{R}^n$.
- `E`: Matríz E asociada a las restricciones de igualdad. En caso de que el problema (P) no tenga restricciones de igualdad, se debe ingresar la matríz de ceros de $\mathbb{R}^{l\times n}$.
- `e`: Vector e asociado a las restricciones de igualdad. En caso de que el problema (P) no tenga restricciones de igualdad, se debe ingresar el vector de ceros de $\mathbb{R}^l$.

Los parámetros de salida son:

- `x`: Punto candidato a ser óptimo (KKT).
- `f`: Valor de la función objetivo en el punto estacionario.

Puede utilizar otras funciones ubicadas en diferentes archivos de tipo `.m`. Todos los archivos deben entregarse agrupados en un ZIP.

Se considerará que siempre existe un punto factible en el problema, para que la función entregue un resultado (las pruebas se realizarán con problemas de región factible no vacía).

Además de los archivos con la implementación, se debe adjuntar un informe en formato de texto llamado `informe.txt` (**NO** en formato Microsoft Word) donde se detallen aspectos de la ejecución de su implementación, además de todos los detalles para realizar una corrección adecuada. El informe no debe tener mas de 3 páginas de largo.

3. Corrección

La tarea se puede realizar en grupos de 3 personas. Los integrantes deben indicarse en las primeras lineas del archivo `informe.txt`. En la corrección se utilizarán 5 problemas de base. Cada problema vale un punto. Se asignará un punto por la claridad del informe con las indicaciones de la ejecución y orden del código entregado. La nota será mínima en caso de que su tarea no pueda ser ejecutada. Puede probar su implementación en los siguientes ejemplos:

- $\min\{2x_1^2+2x_2^2-2x_1x_2-4x_1-6x_2 : x_1+x_2\leq 2, x_1+5x_2\leq 5, x_1\geq 0, x_2\geq 0\}$. Solución: $x=(35/31,24/31)$, $f=-7{,}16$.
- $\min\{x_1^3-2x_2^2+x_1-3 : x_1+x_2=0, x_1\in[0,2], x_2\in[0,2]\}$. Solución: $x=(0,0)$, $f=-3$.
- $\min\{x_1^2+x_1x_2+2x_2^2-6x_1-2x_2-12x_3 : x_1+x_2+x_3=2, -x_1+2x_2\leq 3, x_i\geq 0\}$. Solución: $x_1=x_2=0, x_3=2, f=-24$
- $\min\{2x_1^2-x_1x_2+3x_2^2-3x_1-2x_2 : -3\leq x_1\leq 0, -4\leq x_2\leq 1\}$. Solución: $x_1=0, x_2=1/3, f=-1/3$
- $\min\{-2x_1-6x_2+x_1^2-2x_1x_2+2x_2^2 : x_1+x_2\leq 2, -x_1+2x_2\leq 2, x_i\geq 0\}$. Solución: $x_1=4/5, x_2=6/5, f=-7,2$

Referencias

[1] M. Bazaraa, C. M. Shetty, *Nonlinear Programming. Theory and Algorithms*, J. Wiley & Sons, 1979.