Curso de Optimización, 2022

Instituto de Matemática y Estadística (IMERL)

**Práctico 6: Programación lineal y método Simplex**

**Ejercicio 6.1.** Resolución de un LP con SIMPLEX, conociendo una base factible

Considere el problema  siendo *A* una matriz *m*x*n* con rango *m<n* , .

Suponga dado un subconjunto *IB* de m índices incluido en {1,...,n}, para el cual se cumple que , siendo *B=A(:,IB)*.

Escriba un programa **prog61** para hallar el óptimo de *(P).* Use el esquema iterativo que se adjunta, en que los multiplicadores *lam* corresponden a las restricciones *g(x)=b-Ax=0*, cr=(crB,crN*)* son los costos reducidos, y *Atj* es la columna *j* de la matriz .

Hay que programar las partes que están indicadas en amarillo.

**function [x,status,lam,cr,IBopt]=prog71(c,A,b,IB)**

**[m,n]=size(A);**

**IN=(1:n); IN(IB)=[];**

**iter=0;**

**Niter=m\*n;**

**status=0; % no encontro el optimo**

**nofin=1;**

**while iter<Niter & nofin,**

 **iter=iter+1;**

*cálculo de lam para que crB=0*

 *cálculo de crN*

 **if all(crN>=0),**

 **nofin=0; status=1; % se halló el óptimo**

 **x =zeros(n,1); x(IB)=**

 **else,**

*hallo un índice j sacar de IN*

 *cálculo de Atj*

 **I1=find(Atj>0);**

 **if ~isempty(I1),**

*hallo un índice s para sacar de IB*

*actualizo IB y IN*

 **else,**

 **nofin=0;**

 **status=2; % problema no acotado inferiormente**

 **x=NaN;**

 **end**

 **end**

**end**

Pruebe el programa con los datos:

*c*=[2 1 4 3 5 –1]’,

*b*=[4 9 7]’,

*A*=[3 1 5 2 –2 1; 1 4 1 2 0 3; 3 1 1 5 1 1],

*IB*=[2 4 6]

**Ejercicio 6.2.** Resolución de un LP en 2 fases

1. Escriba un programa **prog62.m** para resolver el problema (P) del ejercicio anterior, sin conocer una base inicial factible. Para eso, organice 2 etapas:

***Fase 1*:** Use **prog61** para resolver un problema artificial, a partir de cuyo resultado encuentre una base inicial factible, o decida que no existen puntos factibles (o sea que (P) tiene dominio vacío).

***Fase 2*:** Use **prog61** para hallar el óptimo de (P) o determinar que el problema no está acotado inferiormente.

1. Compare el programa con los códigos de programación lineal de Matlab/Octave (LP, LINPROG, GLPK). Estudie los tiempos de ejecución y los resultados obtenidos para algunos problemas de distinto tamaño. Pruebe el caso de restricciones no factibles.

Para sortear los datos se pueden usar las funciones **randn** y **rand** de Matlab/Octave.

1. Amplíe **prog61** para que en el caso no está acotado inferiormente determine una dirección de decrecimiento infinito, que estará incluida entre los argumentos de salida.
2. **Opcional:** Estudie variantes que puedan mejorar el tiempo de cálculo en problemas grandes, como por ejemplo alternativas a la inversión de la matriz B en cada iteración.
3. **Opcional:** Use el solver de Matlab/Octave con la variante de algoritmo de puntos interiores.