Pràctica 1 Optimització

Implementació de l'algorisme del simplex

Sergi Guimerà Roig
Curs 2022/2023 Quadrimestre de primavera

Participants del grup:

Sergi Guimerà Roig 47124320B

Conjunt de dades:

nº 15

Descripció implementació

Per familiaritat amb el llenguatge python ha guanyat a l'hora de decidir amb quin llenguatge implementar el codi. El mòdul NumPy ha estat un factor clau, ja que et permet fer de tot en l'àmbit matricial sense esforç.

La idea de la implementació és crear un objecte per guardar i manipular les matrius i altres informacions necessàries mitjançant mètodes. Una funció que serveixi per iterar sobre els passos amb l'ordre corresponent i que pari quan trobi solució o es sàpiga què és un problema no acotat o no factible.

Format fitxers input:

Per llegir els fitxers, a l'hora d'executar el codi és necessari posar un argument que és el path del fitxer on està la informació del problema en forma estàndard (el problema ha d'estar en forma estàndard abans de passar-la al simplex).
Per exemple per l'alumne 15, el problema 1 seria:

```
PS C:\Users\User\OPT> py .\simplex.py .\datos_simplex_15_1.txt
```

L'estructura del fitxer ha de ser la següent:
una 'c=' a la primera línia. A la següent línia hi ha d'haver els valors dels costs de cada variable en ordre separats per espais. Posteriorment una salt de línia extra.
La següent línia 'A=' i després la matriu A, aquesta ha de tenir una fila en cada línia.

Finalment una línia amb 'b=' i la següent línia els termes independents de les equacions, també en ordre (el mateix que per A i c).

Per exemple el problema 1 de l'estudiant 15 és:

c=

-76	18	-62	-66	26	-30	4	90	80	-60	-1	1	-17	-40	0	0	0	0	0	0

A=

85	-71	-23	-40	42	12	-79	37	0	-1	-35	-29	79	28	0	0	0	0	0	0
10	27	-20	-82	60	-90	98	87	-86	-73	30	-21	57	65	0	0	0	0	0	0
56	-20	79	10	5	-51	12	92	-57	-61	99	-77	35	-94	0	0	0	0	0	0
77	56	92	54	-61	75	-28	-40	-27	37	85	-100	27	46	0	0	0	0	0	0
71	67	69	82	55	89	79	59	95	80	72	77	63	92	1	0	0	0	0	0
40	82	73	-72	60	-64	-6	19	-23	-48	22	-40	53	46	0	-1	0	0	0	0
89	95	73	97	-95	-54	8	-57	35	38	-71	-68	33	-47	0	0	-1	0	0	0
88	-31	45	-34	92	51	-17	-13	47	-35	44	-59	-47	18	0	0	0	-1	0	0
-12	85	47	-75	88	61	66	41	-20	71	84	74	-73	-46	0	0	0	0	-1	0
17	-31	86	21	-41	-70	84	-79	66	-95	85	35	-45	43	0	0	0	0	0	1

b=

5	62	28	293	1051	141	75	148	390	77

Creació fase I i II

Per tenir les variables relativament ordenades he fet una sola classe ‘Problem’ que serveix principalment per guardar les variables i després té mètodes per aplicar els diferents passos. Una funció externa a la classe s’encarrega d’executar els mètodes en l’ordre corresponent ‘simplex_solver’.

L’usuari que utilitza el simplex no cal que sàpiga de la fase I. És per això que el codi s’encarrega de crear la fase I per trobar una Solució Bàsica Factible (SBF) per començar la fase II. Una funció ‘SBF_init’ crea la instància fase 2, aquesta al ser creada crea una instància de la mateixa classe, però amb una variable fase que ara és 1, la resol i dels resultats es treu la informació necessària per començar la fase 2 sempre que sigui factible el problema (abans de passar la informació comprovem que sigui factible (z=0) i que no hi hagi cap variable artificial a la base). Un cop la informació està passada. La instància fase II és passada a la funció que executa l’algorisme.

Aquesta funció no és més que un while True que executa els mètodes en ordre fins que alguns resultats el fan parar.

Pas 1

Aquest pas s’executa en la inicialització de l’objecte. En la fase 1 llegim A i b i creem totes les matrius i vectors i variables que utilitzarem (A, inversa de B, z, C, x). B és la matriu identitat al principi de la fase 1 i, per tant, la seva inversa també. Per altra part, C és un vector de 1 per les variables artificials i 0 per la resta. x pren els mateixos valors que el vector b que llegim de l’input.
Per la fase 2 agafem dels resultats de la fase 1 i llegim C de l’input.

Pas 2

Aquest pas consta de 2 mètodes:
cal_r que serveix per calcular els costos reduïts per les variables no bàsiques.
I check_r que serveix per comprovar si ja estem en l’òptim (en cas que siguin tots els components de r >= 0) i alhora calcula q (es queda el primer que és el d’índex més petit, ja que ordenem N).
Si ja estem en l’òptim retorna True que provocarà prints de la solució òptima i un break per acabar el while True.

Pas 3

Aquest pas també consta de 2 mètodes, cal_db i check_db:
cal_db serveix per calcular les DBF.

check_db mira si aquestes són >= 0 i per tant un problema no acotat, cosa que porta a avisar que no és acotat i un break per acabar el bucle.

Pas 4

Aquí he fet un mètode cal_o que serveix per calcular ϑ^* i p. Per totes les variables bàsiques mirem si dB(i) < 0, si és així calculem la seva ϑ corresponent. Si aquesta és menor que la resta actualitzem ϑ^* i p, si és igual mirem si l'índex corresponent és més petit i actualitzem (regla de Bland). Si no no actualitzem.

Pas 5

En aquest pas hi ha 3 mètodes públics. Un per actualitzar les variables (x i z).
Un per actualitzar els conjunts de variables $\mathfrak{B}$ i N.
I una per actualitzar la inversa.

Actualització inversa

Està implementat la actualització de la inversa. Per això construim E (fem una identitat i canviem la columna p).
Multipliquem E per la antigua inversa i ja tenim la nova inversa.

Resultats dades assignades

Les dades assignades (nº 15) tenen 4 problemes.

Problema 1:

El problema 1 venia amb un òptim trobat, el qual té nombres naturals (0 inclòs) per tots els valors de les variables. La implementació realitzada en python troba un òptim alternatiu on les variables prenen valors reals. A causa de problemes amb la precisió dels floats el print final dels valors de les variables i els costs reduïts es fa amb una precisió de 5 decimals.

```
[simplexP] Fase 1 Iteració 1:  q = 1, B(p) = 21, theta*= 0.05882, z = 2239.35294
[simplexP] Fase 1 Iteració 2:  q = 2, B(p) = 27, theta*= 0.41198, z = 1953.36321
[simplexP] Fase 1 Iteració 3:  q = 3, B(p) = 23, theta*= 0.17353, z = 1907.55073
[simplexP] Fase 1 Iteració 4:  q = 5, B(p) = 22, theta*= 0.63726, z = 1576.42323
[simplexP] Fase 1 Iteració 5:  q = 4, B(p) = 26, theta*= 0.20677, z = 1533.91610
[simplexP] Fase 1 Iteració 6:  q = 7, B(p) = 3, theta*= 0.03032, z = 1401.90442
[simplexP] Fase 1 Iteració 7:  q = 6, B(p) = 28, theta*= 0.04480, z = 1367.51395
[simplexP] Fase 1 Iteració 8:  q = 3, B(p) = 4, theta*= 0.28973, z = 1254.78848
[simplexP] Fase 1 Iteració 9:  q = 8, B(p) = 3, theta*= 0.91431, z = 1009.80281
[simplexP] Fase 1 Iteració 10:  q = 10, B(p) = 29, theta*= 0.12390, z = 950.80010
[simplexP] Fase 1 Iteració 11:  q = 9, B(p) = 6, theta*= 4.06678, z = 842.46261
[simplexP] Fase 1 Iteració 12:  q = 12, B(p) = 10, theta*= 0.94607, z = 646.58729
[simplexP] Fase 1 Iteració 13:  q = 3, B(p) = 30, theta*= 0.33006, z = 622.95966
[simplexP] Fase 1 Iteració 14:  q = 11, B(p) = 3, theta*= 0.34523, z = 604.21618
[simplexP] Fase 1 Iteració 15:  q = 13, B(p) = 25, theta*= 0.21926, z = 532.71116
[simplexP] Fase 1 Iteració 16:  q = 4, B(p) = 8, theta*= 3.04178, z = 292.61210
[simplexP] Fase 1 Iteració 17:  q = 6, B(p) = 24, theta*= 1.86208, z = 0.00000
FASE II
[simplexP] Fase 2 Iteració 1:  q = 3, B(p) = 11, theta*= 1.94964, z = -171.35783
[simplexP] Fase 2 Iteració 2:  q = 14, B(p) = 9, theta*= 0.37271, z = -313.73374
[simplexP] Fase 2 Iteració 3:  q = 10, B(p) = 2, theta*= 1.14626, z = -374.60358
[simplexP] Fase 2 Iteració 4:  q = 11, B(p) = 6, theta*= 0.98453, z = -412.21163
[simplexP] Fase 2 Iteració 5:  q = 17, B(p) = 11, theta*= 206.44641, z = -478.97281
[simplexP] Fase 2 Iteració 6:  q = 19, B(p) = 4, theta*= 265.84225, z = -519.62444
[simplexP] Fase 2 Iteració 7:  q = 18, B(p) = 13, theta*= 102.70911, z = -526.27498



Solució òptima trobada:



z* = -526.2749772644347
Aquest valor s'aconsegueix amb:
conjunt variables bàsiques: [1, 5, 7, 17, 18, 19, 10, 14, 12, 3]
x1* = 2.62789
x2* = 0.00000
x3* = 2.23195
x4* = 0.00000
x5* = 1.63234
x6* = 0.00000
x7* = 2.68819
x8* = 0.00000
x9* = 0.00000
x10* = 3.83998
x11* = 0.00000
x12* = 0.95639
x13* = 0.00000
x14* = 0.29815
x15* = 0.00000
x16* = 0.00000
x17* = 255.11946
x18* = 102.70911
x19* = 334.13007
x20* = 0.00000
r:   18.03262   20.85927   64.91003   78.71002   138.26511   61.40582   10.25419   0.41991   0.27735   0.14066

END
```

Problema 2

En aquest cas també trobem una solució no degenerada.

```
[simplexP] Fase 1 Iteració 1:  q = 3, B(p) = 23, theta*= 0.52778, z = 2165.88889
[simplexP] Fase 1 Iteració 2:  q = 4, B(p) = 22, theta*= 0.46174, z = 2070.18051
[simplexP] Fase 1 Iteració 3:  q = 5, B(p) = 3, theta*= 0.31839, z = 2056.79650
[simplexP] Fase 1 Iteració 4:  q = 2, B(p) = 28, theta*= 0.09418, z = 2054.68790
[simplexP] Fase 1 Iteració 5:  q = 6, B(p) = 2, theta*= 0.09043, z = 2044.91512
[simplexP] Fase 1 Iteració 6:  q = 7, B(p) = 24, theta*= 2.23702, z = 1327.28065
[simplexP] Fase 1 Iteració 7:  q = 1, B(p) = 27, theta*= 0.52040, z = 990.78759
[simplexP] Fase 1 Iteració 8:  q = 3, B(p) = 21, theta*= 0.12963, z = 966.51885
[simplexP] Fase 1 Iteració 9:  q = 2, B(p) = 26, theta*= 0.03933, z = 953.80727
[simplexP] Fase 1 Iteració 10:  q = 8, B(p) = 6, theta*= 0.45378, z = 941.30339
[simplexP] Fase 1 Iteració 11:  q = 10, B(p) = 30, theta*= 0.91510, z = 592.85672
[simplexP] Fase 1 Iteració 12:  q = 9, B(p) = 25, theta*= 0.51835, z = 332.47811
[simplexP] Fase 1 Iteració 13:  q = 11, B(p) = 7, theta*= 0.90244, z = 182.02465
[simplexP] Fase 1 Iteració 14:  q = 6, B(p) = 3, theta*= 2.16633, z = 102.81605
[simplexP] Fase 1 Iteració 15:  q = 12, B(p) = 5, theta*= 0.84384, z = 61.49664
[simplexP] Fase 1 Iteració 16:  q = 3, B(p) = 2, theta*= 0.33722, z = 20.60269
[simplexP] Fase 1 Iteració 17:  q = 13, B(p) = 29, theta*= 0.15668, z = -0.00000
FASE II
[simplexP] Fase 2 Iteració 1:  q = 2, B(p) = 4, theta*= 2.18898, z = -49.95853
[simplexP] Fase 2 Iteració 2:  q = 7, B(p) = 8, theta*= 2.05618, z = -386.35498
[simplexP] Fase 2 Iteració 3:  q = 4, B(p) = 2, theta*= 0.27182, z = -473.52298
[simplexP] Fase 2 Iteració 4:  q = 16, B(p) = 11, theta*= 79.12015, z = -495.73513
[simplexP] Fase 2 Iteració 5:  q = 18, B(p) = 4, theta*= 160.95026, z = -528.27862
[simplexP] Fase 2 Iteració 6:  q = 17, B(p) = 1, theta*= 134.01328, z = -584.45271



Solució òptima trobada:



z* = -584.4527054170535
Aquest valor s'aconsegueix amb:
conjunt variables bàsiques: [6, 18, 12, 16, 9, 3, 17, 7, 13, 10]
x1* = 0.00000
x2* = 0.00000
x3* = 2.10249
x4* = 0.00000
x5* = 0.00000
x6* = 3.50909
x7* = 0.25129
x8* = 0.00000
x9* = 0.48742
x10* = 0.15970
x11* = 0.00000
x12* = 1.00686
x13* = 5.57569
x14* = 0.00000
x15* = 0.00000
x16* = 122.03949
x17* = 134.01328
x18* = 538.08355
x19* = 0.00000
x20* = 0.00000
r:   51.61730   26.16759   32.12158   168.13083   179.70437   116.43330   30.68920   0.42961   0.05440   1.10452

END
```

Problema 3

A diferència dels altres dos problemes anteriors aquest no és factible.

```
[simplexP] Fase 1 Iteració 1:  q = 2, B(p) = 28, theta*= 0.64706, z = 1379.41176
[simplexP] Fase 1 Iteració 2:  q = 3, B(p) = 25, theta*= 0.52202, z = 1092.76986
[simplexP] Fase 1 Iteració 3:  q = 1, B(p) = 30, theta*= 0.08890, z = 1063.57726
[simplexP] Fase 1 Iteració 4:  q = 4, B(p) = 2, theta*= 0.34679, z = 1012.40099
[simplexP] Fase 1 Iteració 5:  q = 5, B(p) = 1, theta*= 0.21498, z = 979.04016
[simplexP] Fase 1 Iteració 6:  q = 9, B(p) = 27, theta*= 0.34585, z = 931.16378
[simplexP] Fase 1 Iteració 7:  q = 8, B(p) = 5, theta*= 0.08569, z = 915.66185
[simplexP] Fase 1 Iteració 8:  q = 11, B(p) = 4, theta*= 0.18925, z = 883.60252
[simplexP] Fase 1 Iteració 9:  q = 16, B(p) = 26, theta*= 270.95983, z = 612.64269
[simplexP] Fase 1 Iteració 10:  q = 5, B(p) = 3, theta*= 0.21226, z = 600.09240
[simplexP] Fase 1 Iteració 11:  q = 15, B(p) = 29, theta*= 59.44516, z = 562.76168
[simplexP] Fase 1 Iteració 12:  q = 3, B(p) = 5, theta*= 0.18655, z = 515.20808
PROBLEMA LINEAL NO FACTIBLE
```

Problema 4

L'últim problema és no acotat. és a dir que hi ha una direcció on z disminueix i per on podem anar infinitament i continuem complint les restriccions del problema.

```
[simplexP] Fase 1 Iteració 1:  q = 1, B(p) = 28, theta*= 8.50000, z = 3473.50000
[simplexP] Fase 1 Iteració 2:  q = 2, B(p) = 33, theta*= 7.10314, z = 2355.48652
[simplexP] Fase 1 Iteració 3:  q = 3, B(p) = 31, theta*= 1.90670, z = 1649.14095
[simplexP] Fase 1 Iteració 4:  q = 4, B(p) = 30, theta*= 0.21264, z = 1621.56995
[simplexP] Fase 1 Iteració 5:  q = 5, B(p) = 34, theta*= 0.81790, z = 1173.88193
[simplexP] Fase 1 Iteració 6:  q = 6, B(p) = 27, theta*= 0.89373, z = 654.67753
[simplexP] Fase 1 Iteració 7:  q = 8, B(p) = 4, theta*= 0.18384, z = 604.84359
[simplexP] Fase 1 Iteració 8:  q = 9, B(p) = 25, theta*= 0.13091, z = 570.12250
[simplexP] Fase 1 Iteració 9:  q = 7, B(p) = 6, theta*= 2.01046, z = 443.93850
[simplexP] Fase 1 Iteració 10:  q = 11, B(p) = 32, theta*= 1.21972, z = 72.82263
[simplexP] Fase 1 Iteració 11:  q = 4, B(p) = 29, theta*= 0.14821, z = 55.53821
[simplexP] Fase 1 Iteració 12:  q = 10, B(p) = 4, theta*= 0.22232, z = 52.91232
[simplexP] Fase 1 Iteració 13:  q = 13, B(p) = 26, theta*= 0.39455, z = 0.00000
FASE II
[simplexP] Fase 2 Iteració 1:  q = 12, B(p) = 5, theta*= 0.83814, z = -911.88810
[simplexP] Fase 2 Iteració 2:  q = 4, B(p) = 1, theta*= 0.03040, z = -913.80508
[simplexP] Fase 2 Iteració 3:  q = 5, B(p) = 7, theta*= 0.12635, z = -921.62569
[simplexP] Fase 2 Iteració 4:  q = 15, B(p) = 5, theta*= 8.77314, z = -923.24205
[simplexP] Fase 2 Iteració 5:  q = 1, B(p) = 4, theta*= 0.00759, z = -923.30111
[simplexP] Fase 2 Iteració 6:  q = 17, B(p) = 1, theta*= 1.16120, z = -923.99070
[simplexP] Fase 2 Iteració 7:  q = 5, B(p) = 15, theta*= 0.39035, z = -957.86829
[simplexP] Fase 2 Iteració 8:  q = 1, B(p) = 8, theta*= 0.58679, z = -1086.76135
[simplexP] Fase 2 Iteració 9:  q = 18, B(p) = 13, theta*= 20.01580, z = -1100.65962
[simplexP] Fase 2 Iteració 10:  q = 16, B(p) = 3, theta*= 79.82815, z = -1114.31570
[simplexP] Fase 2 Iteració 11:  q = 19, B(p) = 9, theta*= 302.28646, z = -1227.57247
[simplexP] Fase 2 Iteració 12:  q = 15, B(p) = 12, theta*= 50.18060, z = -1239.41182
[simplexP] Fase 2 Iteració 13:  q = 20, B(p) = 15, theta*= 85.62297, z = -1295.90497
[simplexP] Fase 2 Iteració 14:  q = 21, B(p) = 20, theta*= 212.87947, z = -1322.86236
[simplexP] Fase 2 Iteració 15:  q = 3, B(p) = 10, theta*= 0.66017, z = -1331.20003
[simplexP] Fase 2 Iteració 16:  q = 9, B(p) = 3, theta*= 0.41615, z = -1335.19447
[simplexP] Fase 2 Iteració 17:  q = 22, B(p) = 9, theta*= 99.44135, z = -1456.10054
[simplexP] Fase 2 Iteració 18:  q = 4, B(p) = 21, theta*= 5.43618, z = -1546.78835
[simplexP] Fase 2 Iteració 19:  q = 8, B(p) = 1, theta*= 0.12637, z = -1561.37307
[simplexP] Fase 2 Iteració 20:  q = 12, B(p) = 8, theta*= 0.12590, z = -1570.82628
[simplexP] Fase 2 Iteració 21:  q = 15, B(p) = 2, theta*= 30.10972, z = -1642.68550
[simplexP] Fase 2 Iteració 22:  q = 1, B(p) = 12, theta*= 0.17246, z = -1644.52959
[simplexP] Fase 2 Iteració 23:  q = 10, B(p) = 4, theta*= 2.27871, z = -1759.54145
[simplexP] Fase 2 Iteració 24:  q = 12, B(p) = 1, theta*= 2.38288, z = -1763.44440
[simplexP] Fase 2 Iteració 25:  q = 20, B(p) = 12, theta*= 170.36853, z = -2142.17732
[simplexP] Fase 2 Iteració 26:  q = 9, B(p) = 10, theta*= 8.39309, z = -2378.28220
[simplexP] Fase 2 Iteració 27:  q = 21, B(p) = 9, theta*= 633.87500, z = -3235.82813
[simplexP] Fase 2 Iteració 28:  q = 13, B(p) = 15, theta*= 6.22638, z = -3284.44738
[simplexP] Fase 2 Iteració 29:  q = 24, B(p) = 5, theta*= 61.35043, z = -3555.83419
[simplexP] Fase 2 Iteració 30:  q = 15, B(p) = 13, theta*= 8189.00000, z = -51332.00000
PROBLEMA NO ACOTAT
END
```

Comentaris:

El codi està bastant comentat.
Dins el zip hi ha els fitxers input preparats.

Com no he trobat cap problema de degeneració en la fase 1 on es quedés una variable artificial dins les bàsiques no he implementat la permutació que serviria per 'arreglar' la SBF inicial sense utilitzar simplex dual.

Els òptims proposats dels problemes 1 són amb variables enteres. En canvi, tots els òptims trobats són amb variables reals.

Al llarg de la pràctica han passat certes qüestions amb la precisió que et donen els floats. Intentar representar un rang d'infinits nombres amb un nombre limitat de bits genera certes imprecisions. Per exemple:

```
>>> .15-.1
0.04999999999999999
```

En el nostre cas això ens afecta sobretot en els valors de z, ϑ, i les variables no bàsiques.

Un exemple és que a l'acabar la fase I, z = 2.2737367544323206e-13 quan totes les variables artificials són 0. I per tant: $\sum_{i \in Artificials} x_i = 0$

En el Problema 1 la variable x4 que és no bàsica té un valor diferent de 0. Que a més a més és negatiu, cosa que no pot ser perquè una de les restriccions sobre les variables és que siguin majors o iguals a 0.

```
z* = -526.2749772644347
Aquest valor s'aconsegueix amb:
conjunt variables bàsiques: [1, 5, 7, 17, 18, 19, 10, 14, 12, 3]
x1* = 2.6278886200084477
x2* = 0.0
x3* = 2.2319536105689552
x4* = 2.220446049250313e-16
```

Per culpa d'aquests errors, tal com s'ha dit prèviament, els prints finals s'han canviat a una precisió de 5 decimals. També quan acaba la fase I i hem de comprovar si z > 0 cosa que el converteix el problema en infactible, en lloc de comparar amb 0 comparem amb un ε (0.000001).
Un altre enfocament hauria sigut treballar sempre amb menys precisió, és a dir arrodonir a un cert nombre de decimals.