FACULTAT D'INFORMÀTICA DE BARCELONA

FIB UPC
OPTIMITZACIÓ

Pràctica 1: Implementació del Simplex Primal

Alumnes :

Granja Bayot, JORDI
DNI: 20560139W

Jerez Cubero, ALBERTO
DNI: 47407429J

Tutors :

Linares, MARI PAZ
Fonseca, PAU

March 18, 2024

Contents

1 Introducció

El present informe constitueix la justificació de la solució proposada pel problema plantejat. Aquest és, programar l'algoritme del símplex primal, per tal de resoldre un conjunt de problemes d'optimització lineal.
Entre els objectius de la pràctica, destaca la comprensió del funcionament del mètode del símplex, així com de conceptes teòrics associats a la programació lineal. En el nostre cas, el llenguatge de programació escollit ha estat Python. Tot i no ser un llenguatge destinat a les matemàtiques, el seu mòdul *Numpy* permet realitzar càlculs numèrics de forma eficient.

2 Descripció de la implementació

2.1 Introducció al algorisme

L'algorisme Simplex és un mètode molt utilitzat per a resoldre problemes de programació lineal. Aquests problemes impliquen l'optimització d'una funció objectiu lineal subjecta a restriccions lineals d'igualtat i desigualtat. L'algorisme Simplex millora iterativament la solució passant d'una solució factible a una altra al llarg de les cantonades de la regió factible fins a aconseguir la solució òptima. Es tracta d'una eina fonamental en recerca operativa, que es pot fer servir en intel·ligència artificial, per exmeple, per automatitzar processos d'optimització en sistemes de presa de decisions.

2.2 Algorisme

En aquest apartat es presenta un petit resum de l'algorisme del Simplex primal, per tal de guiar l'explicació de la seva implementació en Python.

1. Inicialització amb una SBF del (PL).
2. Comprovació d'optimalitat i selecció de la variable d'entrada.
   - 2.1 Càlcul costos reduïts:
   $$r' = C'_N - C'_B B^{-1} A_N \tag{1}$$
   - 2.2 Si $r' \geq [0] \implies$ SBF actual és òptima. Altrament, seleccionar una variable no bàsica q amb $r'_q < 0$
3. Càlcul d'una DBF de descens.
   - 3.1 Càlcul direcció associada a la variable d'entrada x_q:
   $$d_B = -B^{-1} A_q \tag{2}$$
   - 3.2 Si $d'_B \geq [0] \implies$ (PL) no fitat.
4. Càlcul de la longitud de pas màxim i selecció variable de sortida.
   - 4.1 Càlcul de la longitud de pas màxim:
   $$\theta^* = \min_{\{i=1,\ldots,m | d_B(i)<0\}} \left\{ {}^{-x_{B(i)}}/{}_{d_{B(i)}} \right\} \tag{3}$$
   - 4.2 Variable bàsica de sortida: $B(p)$ associada a θ^*.
5. Actualitzacions i canvi de base.
6. Anar a 2.

2.3 Implementació

En el nostre cas, hem considerat oportú programar l'algorisme en una classe `Simplex`, que ens ofereix organització del codi i encapsulació de funcionalitats: inicialització de les dades, resolució del problema, comprovació d'optimalitat, etc. La instància d'aquesta classe es crea amb els coeficients de cost 'c', la matriu de coeficients de les variables per cada restricció 'A', i les constants de les desigualtats 'b'.

De cara a trobar la SBF necessària per inicialitzar l'algorisme, la classe incorpora un argument 'phase1', que expressa si el problema associat a les dades correspon a la Fase I d'un altre problema. I és que al cap i la fi, la resolució és la mateixa; només haurem de controlar que les variables artificials no prenguin valor. D'altra banda, s'ofereix un argument 'debug_file', per tal d'indicar el nom del fitxer de text on es guardarà el registre de les iteracions i la solució.

Un cop creada la instància, només caldrà cridar el seu mètode `solve`, el qual començarà l'execució de l'algorisme. El comportament d'aquest mètode variarà lleugerament en funció de la fase del problema:

- **Si és Fase I:** Inicialitzem les variables de la base (i.e. les sintètiques creades) i les de la no-base; també la matriu bàsica i la seva inversa, que podem assegurar seran equivalents a Id_m.
- **Si no és Fase I:** Les variables de la base i la matriu bàsica inversa associada s'obtenen de cridar el mètode `phase_1`. Aquesta funció inicialitza tota la forma (c, A, b) del problema sintètic i crea una instància de la classe `Simplex`, amb l'argument 'phase1' a *True* i aquests nous paràmetres. Finalment, es resol el problema lineal, de manera que tindrem els resultats de la Fase I al problema original. Inicialitzem llavors la matriu bàsica resultant i les variables no bàsiques, ordenades ascendentment, ja que no té impacte en els resultats i ens permetrà aplicar la regla de Bland.

Tot seguit, inicialitzem c_B, c_N, A_N, x_B i z. Un cop representada la SBF inicial, aquestes variables ens permetran començar el procés iteratiu de l'algorisme.

La implementació s'estructura amb el bucle principal presentat a la secció 2.2, que es repetirà mentre el mètode `isoptimal` no avaluï a *True*. Aquesta funció calcula els costos reduïts segons l'Equació 1. Si tots són positius o zero, avaluarà a True; altrament, agafarem com a variable d'entrada 'q' la primera variable amb cost reduït negatiu. Considerant que les variables no bàsiques mantindran un ordre ascendent, estarem aplicant la regla de Bland.

Els passos que s'aniran repetint són els següents:

1. **Crida al mètode get_direction**. Calcularà la direcció bàsica factible de descens segons l'Equació 2. Si aquesta direcció es positiva o zero, pararem l'execució, ja que es tracta d'un problema no fitat.

2. **Crida al mètode get_theta**. Calcularà la longitud de pas màxim, segons l'Equació 3, i determinarà la variable de sortida associada. Per a realitzar això, iterem sobre tots els elements negatius del vector de direcció, calculem el quocient amb el valor de la variable corresponent, i anem actualitzant segons si la *theta* és menor que l'anterior. Un cop tenim la longitud de pas, seleccionem la seva variable associada com la variable d'entrada 'p'. En cas d'empat en el valor de la theta, seleccionarem la d'índex més baix segons la regla de Bland.

3. **Crida al mètode update_values**. Actualitzarà els següents valors: la inversa de la nova matriu bàsica B (amb un mètode auxiliar `update_inverse`), A_N, les variables bàsiques i no-bàsiques, a l'igual que els seus coeficients i, finalment, els valors de les variables bàsiques, així com el de la funció objectiu resultant.

Un cop sortim del bucle amb l'avaluació a *True* de la funció `isoptimal`, si no ens trobem en la Fase I, tenim garantida la solució òptima. En canvi, si hi estem, haurem de parar més compte.

En primer lloc, si la z no és equivalent a zero (amb certa tolerància a causa de la precisió de còmput), el problema no és factible, ja que les variables artificials prenen valor a l'òptim. D'altra banda, si la z val 0 i no tenim variables artificials a la base, hem acabat. No obstant, si això últim no es dona, comprovat amb el mètode `clean_phase1`, voldrà dir que tenim una solució degenerada. Per a solucionar això, iterarem d'una forma lleugerament diferent que abans. Un cop calculem els costos reduïts, entrarà aquella variable que tingui el cost reduït igual a zero amb el menor índex (regla de Bland). D'aquesta manera ens mantindrem en l'òptim, però ens traurem les variables artificials de sobre. Tot seguit, retornarem les variables bàsiques i la inversa de la matriu bàsica, per inicialitzar el (PL) original.

2.4 Manipulació de l'entrada

En cas que les dades del problema vinguin en el mateix format que els arxius de l'enunciat de la pràctica, cal executar la funció `parse` del fitxer `Parser.py`, amb el *path* del fitxer com a argument. Aquesta retorna les dades 'c', 'A', 'b' en forma de *Numpy arrays*, amb els quals podrem inicialitzar la classe `Simplex` i resoldre el (PL) associat. Específicament, la funció ve descrita de la següent manera:

- **Pre-condició:** el *path* és vàlid per a un fitxer llegible. Els paràmetres del problema s'escriuen en les següents línies del patró '<paràmetre>=.', tenint com a paràmetres c, A, b. Només s'han d'escriure nombres.
- **Post-condició :** Crea a partir de les dades del *path* un vector amb coeficients de cost 'c', un altre amb el costat dret de les restriccions 'b', i una matriu amb coeficients de variables per a cada restricció 'A'.

2.5 Exemple d'ús

```
from Simplex import Simplex
from Parser import parse

c, A, b = parse('Datos/Datos5_1.txt')
sim = Simplex(c, A, b, debug_file = 'Debug_Datos5_1')
i_B, x_B = sim.solve()
```

Aquesta execució retornarà les variables de la SBF òptima, així com els seus valors. Per a veure tota la informació del flux d'execució i resultats, mirar el fitxer generat amb el nom introduït a l'argument *debug_file*.

2.6 Gestió de casos especials

La identificació dels casos especials, que realment classifiquen el (PL), es fa mitjançant la funcionalitat *raise Exception* de Python, informant a l'usuari del tipus del problema a partir d'un missatge.

- **No factible**: L'òptim de la Fase I requereix donar valor a les variables sintètiques, és a dir, z no és equivalent a 0. El missatge associat és: "El problema no és factible".
- **No fitat**: tots els elements de la direcció bàsica factible són positius o zero. El missatge associat és: "El problema no està fitat, sempre pot millorar".
- **Resoluble**: altrament.

2.7 Consideració del rendiment

En quant al rendiment, tot i que no ha sigut la principal prioritat en la implementació (s'ha tingut en compte de totes maneres), podem intentar observar sobre quins temps d'execució oscil·la l'obtenció de resultats respecte la dimensionalitat del problema (només el nombre de variables, les restriccions seràn sempre 10). Realitzarem 100 iteracions per conjunt de dades i n'extraurem la mitjana, també realitzarem la mitjana entre conjunts de dades de la mateixa mida de variables. La columna nombre de variables **NO** inclou les variables de folguera. Els resultats han sigut de la següent forma:

Nombre de variables	Temps d'execució(ms)
3	7.992
5	8.442
10	9.543
14	11.797
20	16.512
50	18.68

Table 1: Taula de temps d'execució respecte dimensionalitat.

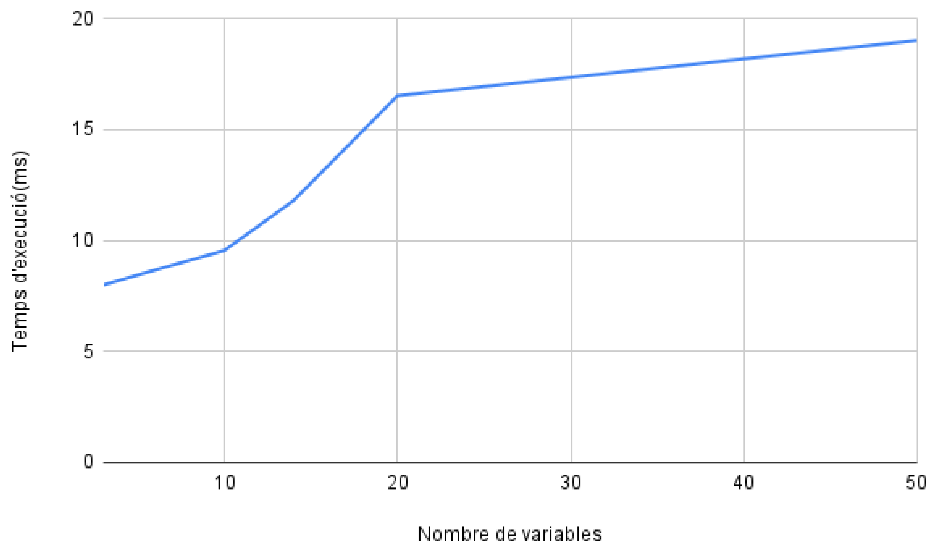

Figure 1: Nombre de variables contra temps d'execució.

Amb aquest experiment ràpid, podem observar que trobem la solució força ràpida, en mili-segons, i que escala amb una ordre de magnitud $O(n)$, és a dir, lineal.

2.8 Proves i validació

Les proves i validació de la nostra implementació ha constat de dues fases:

1. **Comprovació de passos intermedis:** Les instàncies de la classe `Simplex` accepten com a argument un nom pel fitxer de sortida de l'execució, on es guardaran tots els detalls del flux d'execució. Així doncs, tant pels problemes amb correcció donats a l'aula, com per els problemes amb resultat dels dos membres del grup, hem revistat que totes les passes del algorisme fessin correctament la seva funció.

2. **Comprovació de resultats finals:** El primer problema per individu conté la solució del problema. Així doncs, hem provat la nostra implementació sobre aquests primers problemes del conjunt de dades.

D'altra banda, també s'ha provat problemes alternatius amb objectius específics. Per exemple, un problema no fitat per provar el funcionament de les nostres excepcions.

3 Solució obtinguda per als problemes assignats

Per acabar, s'adjunta la solució obtinguda pels problemes assignats als DNI's 47407429J i 20560139W, que es corresponen amb els conjunts de dades nº5 i nº23, respectivament.

3.1 Conjunt de dades 5

3.1.1 Problema 1

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 3, q = 0, B(p) = 0, theta* = 1.05, z = 2774.8
Iteration 2: p = 7, q = 0, B(p) = 1, theta* = 0.5796, z = 2534.3344
Iteration 3: p = 8, q = 0, B(p) = 2, theta* = 2.7817, z = 1880.7927
Iteration 4: p = 6, q = 0, B(p) = 3, theta* = 0.1521, z = 1750.9617
Iteration 5: p = 2, q = 1, B(p) = 5, theta* = 1.5123, z = 1238.437
Iteration 6: p = 0, q = 1, B(p) = 6, theta* = 2.2376, z = 566.3974
Iteration 7: p = 4, q = 3, B(p) = 9, theta* = 0.0523, z = 562.7734
Iteration 8: p = 2, q = 1, B(p) = 7, theta* = 0.6888, z = 450.6957
Iteration 9: p = 5, q = 3, B(p) = 10, theta* = 0.0003, z = 450.5515
Iteration 10: p = 4, q = 1, B(p) = 5, theta* = 1.3254, z = 270.4462
Iteration 11: p = 2, q = 1, B(p) = 8, theta* = 2.0918, z = 207.9635
Iteration 12: p = 9, q = 3, B(p) = 11, theta* = 0.5397, z = 100.0837
Iteration 13: p = 4, q = 0, B(p) = 4, theta* = 0.4858, z = 51.1916
Iteration 14: p = 5, q = 1, B(p) = 7, theta* = 0.9519, z = 30.0405
Iteration 15: p = 1, q = 3, B(p) = 12, theta* = 0.1726, z = 0.0
Basic feasible solution found, iteration 15
Phase II
Iteration 1: p = 2, q = 0, B(p) = 5, theta* = 1.3444, z = -202.5737
Iteration 2: p = 5, q = 1, B(p) = 9, theta* = 1.5258, z = -391.3039
Iteration 3: p = 9, q = 2, B(p) = 10, theta* = 0.1114, z = -399.0948
Iteration 4: p = 9, q = 4, B(p) = 14, theta* = 61.7522, z = -470.6913
Iteration 5: p = 0, q = 0, B(p) = 7, theta* = 0.2207, z = -522.2405
Iteration 6: p = 0, q = 1, B(p) = 8, theta* = 0.1683, z = -530.6747
Iteration 7: p = 4, q = 2, B(p) = 10, theta* = 0.4071, z = -545.5424
Iteration 8: p = 5, q = 6, B(p) = 16, theta* = 20.1888, z = -577.6367
Iteration 9: p = 4, q = 0, B(p) = 4, theta* = 0.1074, z = -638.2692
Iteration 10: p = 4, q = 6, B(p) = 15, theta* = 221.1492, z = -665.7822
Iteration 11: p = 8, q = 1, B(p) = 6, theta* = 0.0082, z = -666.0968
Iteration 12: p = 9, q = 7, B(p) = 17, theta* = 42.1154, z = -670.5369
Iteration 13: p = 0, q = 0, B(p) = 2, theta* = 0.7894, z = -673.1988
Iteration 14: p = 4, q = 8, B(p) = 18, theta* = 56.0922, z = -715.1134
Iteration 15: p = 8, q = 1, B(p) = 7, theta* = 0.1174, z = -721.1665
Iteration 16: p = 8, q = 2, B(p) = 8, theta* = 0.1483, z = -729.8938
Optimal solution found, iteration 16, z = -729.8938415358061
End of primal simplex

Optimal solution:

vb = [ 2 12  5  0 18 16  3  1  8 17]
xb = [  1.29    2.004   2.064   2.313  76.751 138.887   1.817   3.812   0.148
 128.446]
z = -729.8938415358061
r = [1.3773e+02 7.9890e+01 7.4330e+01 9.7620e+01 2.0563e+02 1.0643e+02
 1.4260e+01 5.5000e-01 1.8000e-01 2.8000e-01]
```

Figure 2: Flux d'execució problema nº1 del conjunt de dades nº5.

3.1.2 Problema 2

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 8, q = 1, B(p) = 1, theta* = 0.5676, z = 1865.0541
Iteration 2: p = 0, q = 0, B(p) = 0, theta* = 0.3998, z = 1571.8482
Iteration 3: p = 2, q = 0, B(p) = 2, theta* = 0.3363, z = 1420.418
Iteration 4: p = 2, q = 0, B(p) = 3, theta* = 0.3435, z = 1354.1592
Iteration 5: p = 5, q = 2, B(p) = 5, theta* = 0.5662, z = 1204.8913
Iteration 6: p = 9, q = 1, B(p) = 4, theta* = 0.3007, z = 1186.3568
Iteration 7: p = 8, q = 0, B(p) = 2, theta* = 0.3219, z = 1174.3391
Iteration 8: p = 3, q = 1, B(p) = 6, theta* = 0.1327, z = 1149.9755
Iteration 9: p = 4, q = 1, B(p) = 7, theta* = 1.6529, z = 862.315
Iteration 10: p = 7, q = 0, B(p) = 1, theta* = 1.2888, z = 707.0867
Iteration 11: p = 6, q = 0, B(p) = 8, theta* = 1.2808, z = 474.4383
Iteration 12: p = 7, q = 0, B(p) = 9, theta* = 0.1145, z = 461.8696
Iteration 13: p = 3, q = 1, B(p) = 10, theta* = 2.6376, z = 143.8394
Iteration 14: p = 2, q = 2, B(p) = 11, theta* = 0.2471, z = 44.9828
Iteration 15: p = 1, q = 2, B(p) = 6, theta* = 0.595, z = -0.0
Basic feasible solution found, iteration 15
Phase II
Iteration 1: p = 6, q = 1, B(p) = 3, theta* = 0.11, z = 73.8632
Iteration 2: p = 0, q = 2, B(p) = 12, theta* = 0.6208, z = 9.9958
Iteration 3: p = 3, q = 3, B(p) = 13, theta* = 0.359, z = -27.4536
Iteration 4: p = 6, q = 6, B(p) = 16, theta* = 106.3801, z = -52.4082
Iteration 5: p = 0, q = 7, B(p) = 17, theta* = 7.4641, z = -58.5341
Iteration 6: p = 6, q = 2, B(p) = 3, theta* = 0.2854, z = -122.2781
Iteration 7: p = 4, q = 2, B(p) = 8, theta* = 1.132, z = -285.3858
Iteration 8: p = 6, q = 1, B(p) = 1, theta* = 0.3511, z = -296.4562
Iteration 9: p = 6, q = 3, B(p) = 10, theta* = 0.2015, z = -313.3263
Iteration 10: p = 7, q = 4, B(p) = 12, theta* = 0.2037, z = -358.9074
Iteration 11: p = 7, q = 9, B(p) = 19, theta* = 85.4842, z = -431.0743
Iteration 12: p = 1, q = 2, B(p) = 3, theta* = 0.136, z = -461.4369
Iteration 13: p = 6, q = 1, B(p) = 1, theta* = 0.622, z = -468.9447
Iteration 14: p = 6, q = 3, B(p) = 9, theta* = 0.6044, z = -483.2634
Iteration 15: p = 8, q = 7, B(p) = 15, theta* = 76.7276, z = -545.8224
Iteration 16: p = 1, q = 5, B(p) = 10, theta* = 0.6411, z = -568.1461
Optimal solution found, iteration 16, z = -568.1461036220002
End of primal simplex

Optimal solution:

vb = [17 10 11 13  8  5  9 19 15  4]
xb = [7.82356e+02 6.41000e-01 3.55300e+00 1.06900e+00 2.30900e+00 2.18200e+00
 7.82000e-01 3.95317e+02 1.37627e+02 2.90700e+00]
z = -568.1461036220002
r = [2.8782e+02 4.5970e+01 1.0764e+02 4.1180e+01 1.2159e+02 1.1855e+02
 1.5087e+02 7.1000e-01 8.0000e-02 7.2000e-01]
```

Figure 3: Flux d'execució problema nº2 del conjunt de dades nº5.

3.1.3 Problema 3

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 7, q = 1, B(p) = 1, theta* = 0.7536, z = 1417.942
Iteration 2: p = 9, q = 1, B(p) = 2, theta* = 0.6358, z = 1189.6915
Iteration 3: p = 8, q = 1, B(p) = 3, theta* = 0.056, z = 1172.102
Iteration 4: p = 7, q = 1, B(p) = 4, theta* = 0.7697, z = 1139.4979
Iteration 5: p = 0, q = 4, B(p) = 7, theta* = 0.3235, z = 1133.2621
Iteration 6: p = 8, q = 3, B(p) = 6, theta* = 0.0033, z = 1133.0974
Iteration 7: p = 7, q = 4, B(p) = 8, theta* = 0.8523, z = 998.6249
Iteration 8: p = 8, q = 0, B(p) = 0, theta* = 0.0842, z = 998.0277
Iteration 9: p = 0, q = 1, B(p) = 3, theta* = 0.0208, z = 997.8683
Iteration 10: p = 0, q = 5, B(p) = 9, theta* = 0.0363, z = 993.7757
Iteration 11: p = 8, q = 4, B(p) = 6, theta* = 0.0982, z = 990.2647
Iteration 12: p = 8, q = 8, B(p) = 12, theta* = 0.0915, z = 985.4976
Iteration 13: p = 8, q = 9, B(p) = 13, theta* = 0.0772, z = 983.7
Iteration 14: p = 4, q = 10, B(p) = 14, theta* = 281.9113, z = 701.7887
Iteration 15: p = 8, q = 9, B(p) = 12, theta* = 0.0915, z = 700.8269
Iteration 16: p = 5, q = 10, B(p) = 15, theta* = 323.4877, z = 377.3393
Iteration 17: p = 8, q = 0, B(p) = 0, theta* = 0.0708, z = 364.2173
Iteration 18: p = 8, q = 9, B(p) = 13, theta* = 0.0772, z = 360.3842
Iteration 19: p = 0, q = 7, B(p) = 10, theta* = 0.1053, z = 359.0092
Iteration 20: p = 6, q = 10, B(p) = 16, theta* = 46.7446, z = 312.2647
Iteration 21: p = 0, q = 2, B(p) = 3, theta* = 0.0446, z = 304.0085
Iteration 22: p = 0, q = 10, B(p) = 17, theta* = 6.7149, z = 303.7583
Iteration 23: p = 0, q = 11, B(p) = 18, theta* = 4.9796, z = 303.6269
The problem is not feasible, artificial variables take value at optimal
```

Figure 4: Flux d'execució problema nº3 del conjunt de dades nº5.

3.1.4 Problema 4

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 2, q = 0, B(p) = 0, theta* = 6.1979, z = 3608.2604
Iteration 2: p = 0, q = 0, B(p) = 1, theta* = 4.1842, z = 1572.9471
Iteration 3: p = 8, q = 0, B(p) = 2, theta* = 1.7367, z = 1332.5278
Iteration 4: p = 3, q = 3, B(p) = 6, theta* = 1.437, z = 1169.4565
Iteration 5: p = 8, q = 1, B(p) = 4, theta* = 1.5858, z = 1135.9466
Iteration 6: p = 7, q = 3, B(p) = 7, theta* = 0.8917, z = 654.3748
Iteration 7: p = 9, q = 2, B(p) = 5, theta* = 0.8875, z = 407.3099
Iteration 8: p = 8, q = 1, B(p) = 3, theta* = 0.1799, z = 406.4122
Iteration 9: p = 8, q = 2, B(p) = 8, theta* = 0.1125, z = 394.5681
Iteration 10: p = 6, q = 3, B(p) = 9, theta* = 0.0602, z = 384.6508
Iteration 11: p = 0, q = 3, B(p) = 10, theta* = 0.2808, z = 139.5382
Iteration 12: p = 4, q = 1, B(p) = 2, theta* = 0.4179, z = 119.1931
Iteration 13: p = 4, q = 2, B(p) = 4, theta* = 0.5503, z = 85.7166
Iteration 14: p = 5, q = 2, B(p) = 3, theta* = 0.7571, z = 31.2347
Iteration 15: p = 0, q = 2, B(p) = 11, theta* = 1.275, z = 11.0829
Iteration 16: p = 7, q = 3, B(p) = 12, theta* = 1.9352, z = 10.587
Iteration 17: p = 1, q = 4, B(p) = 13, theta* = 0.162, z = -0.0
Basic feasible solution found, iteration 17
Phase II
Iteration 1: p = 7, q = 2, B(p) = 7, theta* = 1.7494, z = -510.6691
Iteration 2: p = 0, q = 2, B(p) = 10, theta* = 1.0378, z = -597.8754
Iteration 3: p = 9, q = 3, B(p) = 12, theta* = 2.5543, z = -733.9261
Iteration 4: p = 2, q = 3, B(p) = 11, theta* = 0.451, z = -740.0198
Iteration 5: p = 7, q = 4, B(p) = 14, theta* = 17.3185, z = -751.9429
Iteration 6: p = 8, q = 0, B(p) = 0, theta* = 0.3174, z = -779.899
Iteration 7: p = 2, q = 3, B(p) = 7, theta* = 0.9705, z = -792.6855
Iteration 8: p = 2, q = 5, B(p) = 15, theta* = 43.532, z = -800.7861
Iteration 9: p = 3, q = 8, B(p) = 18, theta* = 52.5538, z = -882.2341
Iteration 10: p = 0, q = 2, B(p) = 5, theta* = 2.2815, z = -982.6691
Iteration 11: p = 9, q = 7, B(p) = 16, theta* = 13.8209, z = -983.1301
Iteration 12: p = 9, q = 8, B(p) = 17, theta* = 12.8884, z = -997.8924
Iteration 13: p = 1, q = 5, B(p) = 10, theta* = 1.2251, z = -1088.0069
Iteration 14: p = 2, q = 9, B(p) = 19, theta* = 109.4773, z = -1160.7605
Iteration 15: p = 0, q = 4, B(p) = 8, theta* = 0.0222, z = -1162.4251
Iteration 16: p = 0, q = 7, B(p) = 13, theta* = 0.007, z = -1162.5085
Iteration 17: p = 0, q = 10, B(p) = 20, theta* = 0.4676, z = -1162.9657
Iteration 18: p = 2, q = 9, B(p) = 15, theta* = 130.2497, z = -1313.6899
Iteration 19: p = 1, q = 5, B(p) = 8, theta* = 3.9478, z = -1648.4308
Iteration 20: p = 5, q = 1, B(p) = 2, theta* = 1.1831, z = -1700.2189
Iteration 21: p = 5, q = 10, B(p) = 19, theta* = 30.5379, z = -1707.8286
Iteration 22: p = 8, q = 11, B(p) = 21, theta* = 84.1183, z = -1982.9993
Iteration 23: p = 4, q = 3, B(p) = 3, theta* = 5.3757, z = -2184.6905
Iteration 24: p = 4, q = 11, B(p) = 16, theta* = 154.2811, z = -2360.8
Iteration 25: p = 6, q = 12, B(p) = 22, theta* = 1.0485353683789312e+19, z = -1.21630102731956e+19
The Problem is not bounded, it always can improve
```

Figure 5: Flux d'execució problema nº4 del conjunt de dades nº5.

3.2 Conjunt de dades 23

3.2.1 Problema 1

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 2, q = 1, B(p) = 1, theta* = 1.4474, z = 3146.9474
Iteration 2: p = 7, q = 0, B(p) = 0, theta* = 1.4915, z = 2687.9723
Iteration 3: p = 3, q = 0, B(p) = 2, theta* = 2.9225, z = 1726.2046
Iteration 4: p = 4, q = 0, B(p) = 3, theta* = 1.5592, z = 1316.8887
Iteration 5: p = 8, q = 1, B(p) = 5, theta* = 0.2585, z = 1283.465
Iteration 6: p = 6, q = 1, B(p) = 6, theta* = 0.7508, z = 1202.1715
Iteration 7: p = 9, q = 1, B(p) = 7, theta* = 1.0993, z = 798.0856
Iteration 8: p = 1, q = 0, B(p) = 4, theta* = 1.5407, z = 764.1308
Iteration 9: p = 0, q = 0, B(p) = 8, theta* = 1.8092, z = 306.8655
Iteration 10: p = 6, q = 1, B(p) = 10, theta* = 2.7429, z = 176.8456
Iteration 11: p = 5, q = 2, B(p) = 11, theta* = 1.5585, z = 0.0
Basic feasible solution found, iteration 11
Phase II
Iteration 1: p = 2, q = 0, B(p) = 6, theta* = 0.9281, z = -305.0395
Iteration 2: p = 3, q = 1, B(p) = 9, theta* = 0.2944, z = -344.3181
Iteration 3: p = 2, q = 0, B(p) = 1, theta* = 0.1236, z = -395.392
Iteration 4: p = 2, q = 3, B(p) = 13, theta* = 0.241, z = -450.0207
Iteration 5: p = 6, q = 3, B(p) = 12, theta* = 0.3346, z = -564.2863
Iteration 6: p = 8, q = 0, B(p) = 1, theta* = 0.3982, z = -582.0955
Iteration 7: p = 4, q = 0, B(p) = 2, theta* = 0.1626, z = -587.9065
Iteration 8: p = 8, q = 4, B(p) = 14, theta* = 85.6394, z = -590.0079
Iteration 9: p = 7, q = 6, B(p) = 16, theta* = 52.6316, z = -620.2935
Iteration 10: p = 4, q = 2, B(p) = 3, theta* = 0.0058, z = -621.0639
Iteration 11: p = 6, q = 6, B(p) = 15, theta* = 273.5141, z = -625.1733
Iteration 12: p = 4, q = 8, B(p) = 18, theta* = 12.8614, z = -626.7939
Iteration 13: p = 2, q = 4, B(p) = 5, theta* = 0.3716, z = -638.4376
Iteration 14: p = 3, q = 6, B(p) = 12, theta* = 0.3153, z = -645.7406
Iteration 15: p = 5, q = 9, B(p) = 19, theta* = 103.3635, z = -670.6294
Iteration 16: p = 7, q = 0, B(p) = 0, theta* = 1.1185, z = -672.7913
Iteration 17: p = 6, q = 7, B(p) = 13, theta* = 1.2881, z = -702.9399
Iteration 18: p = 7, q = 4, B(p) = 9, theta* = 0.0092, z = -702.9866
Iteration 19: p = 4, q = 7, B(p) = 15, theta* = 118.5324, z = -703.6484
Optimal solution found, iteration 19, z = -703.6483711990912
End of primal simplex

Optimal solution:

vb = [ 8  4  5 12 15 19 13  9 14  7]
xb = [  0.704    2.831    2.179    1.588 118.532 338.095    1.921    2.68    64.914
   1.346]
z = -703.6483711990912
r = [ 27.    44.19  95.04  23.81 153.78 171.37  21.61   0.22   0.62   0.  ]
```

Figure 6: Flux d'execució problema nº1 del conjunt de dades nº23.

3.2.2 Problema 2

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 5, q = 0, B(p) = 0, theta* = 0.1875, z = 2220.875
Iteration 2: p = 5, q = 0, B(p) = 1, theta* = 0.2368, z = 2163.4211
Iteration 3: p = 9, q = 1, B(p) = 2, theta* = 0.6432, z = 1708.9481
Iteration 4: p = 1, q = 2, B(p) = 4, theta* = 0.2776, z = 1552.3095
Iteration 5: p = 2, q = 0, B(p) = 0, theta* = 0.045, z = 1523.6652
Iteration 6: p = 2, q = 0, B(p) = 3, theta* = 0.0638, z = 1506.1728
Iteration 7: p = 0, q = 1, B(p) = 5, theta* = 0.0927, z = 1461.4878
Iteration 8: p = 1, q = 1, B(p) = 6, theta* = 0.3464, z = 1449.5348
Iteration 9: p = 9, q = 2, B(p) = 7, theta* = 1.03, z = 1206.268
Iteration 10: p = 6, q = 0, B(p) = 0, theta* = 1.2702, z = 1190.737
Iteration 11: p = 2, q = 1, B(p) = 4, theta* = 0.0433, z = 1170.8518
Iteration 12: p = 2, q = 2, B(p) = 8, theta* = 0.1143, z = 1127.9621
Iteration 13: p = 3, q = 3, B(p) = 9, theta* = 0.2086, z = 679.582
Iteration 14: p = 8, q = 0, B(p) = 2, theta* = 0.6034, z = 579.654
Iteration 15: p = 6, q = 0, B(p) = 3, theta* = 1.5701, z = 517.8279
Iteration 16: p = 0, q = 2, B(p) = 10, theta* = 0.4045, z = 472.062
Iteration 17: p = 7, q = 3, B(p) = 11, theta* = 0.6072, z = 229.786
Iteration 18: p = 6, q = 0, B(p) = 0, theta* = 0.8504, z = 164.7824
Iteration 19: p = 9, q = 1, B(p) = 4, theta* = 1.1223, z = 154.7089
Iteration 20: p = 4, q = 1, B(p) = 5, theta* = 0.769, z = -0.0
Basic feasible solution found, iteration 20
Phase II
Iteration 1: p = 5, q = 2, B(p) = 12, theta* = 0.1661, z = -288.0729
Iteration 2: p = 1, q = 3, B(p) = 13, theta* = 0.8602, z = -430.3936
Iteration 3: p = 4, q = 5, B(p) = 15, theta* = 146.8548, z = -499.7322
Iteration 4: p = 2, q = 1, B(p) = 3, theta* = 0.7181, z = -515.3011
Iteration 5: p = 6, q = 6, B(p) = 16, theta* = 314.9534, z = -543.2262
Iteration 6: p = 5, q = 9, B(p) = 19, theta* = 39.9918, z = -596.5232
Iteration 7: p = 0, q = 0, B(p) = 0, theta* = 0.0949, z = -610.4347
Iteration 8: p = 2, q = 4, B(p) = 8, theta* = 1.2391, z = -667.1686
Optimal solution found, iteration 8, z = -667.1685963568806
End of primal simplex

Optimal solution:

vb = [ 0 13  8  9 15 19 16 11  2  4]
xb = [2.95300e+00 2.60000e-02 1.23900e+00 1.36400e+00 8.24010e+01 1.69453e+02
 2.81444e+02 1.62300e+00 3.18600e+00 4.97000e+00]
z = -667.1685963568806
r = [5.1440e+01 1.6990e+01 1.9120e+01 1.3530e+02 1.2921e+02 1.0447e+02
 7.0120e+01 4.2000e-01 3.0000e-02 7.0000e-02]
```

Figure 7: Flux d'execució problema nº2 del conjunt de dades nº23.

3.2.3 Problema 3

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 8, q = 0, B(p) = 0, theta* = 0.0563, z = 1276.6761
Iteration 2: p = 8, q = 0, B(p) = 1, theta* = 0.0556, z = 1266.4444
Iteration 3: p = 9, q = 1, B(p) = 2, theta* = 0.8, z = 1139.0667
Iteration 4: p = 8, q = 5, B(p) = 7, theta* = 0.0473, z = 1136.1339
Iteration 5: p = 9, q = 7, B(p) = 9, theta* = 1.2778, z = 1093.6536
Iteration 6: p = 4, q = 12, B(p) = 14, theta* = 185.8301, z = 907.8235
Iteration 7: p = 6, q = 13, B(p) = 16, theta* = 177.7647, z = 730.0588
Iteration 8: p = 8, q = 6, B(p) = 6, theta* = 0.0667, z = 727.3333
Iteration 9: p = 7, q = 13, B(p) = 17, theta* = 31.9037, z = 695.4296
Iteration 10: p = 8, q = 6, B(p) = 7, theta* = 0.0784, z = 694.2614
Iteration 11: p = 5, q = 13, B(p) = 18, theta* = 4.0, z = 682.0
Iteration 12: p = 8, q = 5, B(p) = 5, theta* = 0.0, z = 682.0
Iteration 13: p = 8, q = 9, B(p) = 11, theta* = 0.0, z = 682.0
The problem is not feasible, artificial variables take value at optimal
```

Figure 8: Flux d'execució problema nº3 del conjunt de dades nº23.

3.2.4 Problema 4

```
Start primal simplex with Bland's rule
Phase I
Iteration 1: p = 3, q = 0, B(p) = 0, theta* = 7.5714, z = 3617.0
Iteration 2: p = 5, q = 0, B(p) = 1, theta* = 4.1195, z = 2862.7631
Iteration 3: p = 2, q = 0, B(p) = 2, theta* = 1.7229, z = 2227.7595
Iteration 4: p = 1, q = 0, B(p) = 3, theta* = 2.8916, z = 1149.28
Iteration 5: p = 4, q = 0, B(p) = 4, theta* = 4.2088, z = 502.2688
Iteration 6: p = 8, q = 0, B(p) = 5, theta* = 0.3006, z = 445.1033
Iteration 7: p = 9, q = 1, B(p) = 7, theta* = 0.5078, z = 351.4147
Iteration 8: p = 0, q = 0, B(p) = 6, theta* = 1.6197, z = 254.7853
Iteration 9: p = 6, q = 0, B(p) = 8, theta* = 0.4704, z = 104.1207
Iteration 10: p = 7, q = 0, B(p) = 9, theta* = 0.1475, z = -0.0
Basic feasible solution found, iteration 10
Phase II
Iteration 1: p = 6, q = 0, B(p) = 10, theta* = 1.0966, z = -737.6521
Iteration 2: p = 6, q = 1, B(p) = 11, theta* = 0.8185, z = -806.2278
Iteration 3: p = 2, q = 2, B(p) = 12, theta* = 2.2747, z = -846.0562
Iteration 4: p = 8, q = 3, B(p) = 13, theta* = 0.74, z = -909.2216
Iteration 5: p = 2, q = 2, B(p) = 8, theta* = 0.8684, z = -923.9346
Iteration 6: p = 4, q = 4, B(p) = 14, theta* = 71.027, z = -942.6385
Iteration 7: p = 5, q = 6, B(p) = 16, theta* = 125.0284, z = -1052.0335
Iteration 8: p = 2, q = 1, B(p) = 2, theta* = 1.1627, z = -1109.0666
Iteration 9: p = 7, q = 6, B(p) = 15, theta* = 54.0992, z = -1184.2959
Iteration 10: p = 0, q = 3, B(p) = 8, theta* = 0.4447, z = -1187.906
Iteration 11: p = 2, q = 7, B(p) = 17, theta* = 42.0484, z = -1225.6153
Iteration 12: p = 0, q = 4, B(p) = 6, theta* = 0.3618, z = -1242.0215
Iteration 13: p = 9, q = 9, B(p) = 19, theta* = 101.0562, z = -1334.3487
Iteration 14: p = 7, q = 1, B(p) = 2, theta* = 1.367, z = -1391.7315
Iteration 15: p = 8, q = 0, B(p) = 1, theta* = 0.1209, z = -1404.151
Iteration 16: p = 8, q = 3, B(p) = 8, theta* = 0.2413, z = -1407.7238
Iteration 17: p = 0, q = 8, B(p) = 15, theta* = 24.1868, z = -1414.0936
Iteration 18: p = 1, q = 9, B(p) = 18, theta* = 285.141, z = -1421.4035
Iteration 19: p = 6, q = 10, B(p) = 20, theta* = 18.5738, z = -1429.5637
Iteration 20: p = 7, q = 1, B(p) = 3, theta* = 0.5072, z = -1612.7693
Iteration 21: p = 7, q = 12, B(p) = 22, theta* = 25.332, z = -1634.8253
Iteration 22: p = 0, q = 9, B(p) = 11, theta* = 7.8074, z = -1710.0189
Iteration 23: p = 8, q = 12, B(p) = 21, theta* = 1584.5946, z = -2299.2432
Iteration 24: p = 3, q = 4, B(p) = 5, theta* = 3.4987, z = -2462.0422
Iteration 25: p = 3, q = 8, B(p) = 9, theta* = 3.7994, z = -2858.957
Iteration 26: p = 3, q = 12, B(p) = 15, theta* = 331.5, z = -4231.5
The Problem is not bounded, it always can improve
```

Figure 9: Flux d'execució problema nº4 del conjunt de dades nº23.

4 Conclusió

La solució proposada per l'algorisme de Simplex sembla funcionar d'una forma robusta i concisa. A més a més, l'estructura esta organitzada i neta per a beneficiar-se de l'alt nivell d'abstracció de Python, que permet entendre el codi d'una forma senzilla i clara.

D'altra banda, l'experiment realitzat mostra l'augment de temps d'execució del programa envers el nombre de variables. En aquest hem pogut veure que tot i probablement no pot ser comparat amb les implementacions més ràpides d'aquest algorisme, té un temps d'execució de pocs mili-segons, amb una escalada per nombre de variables presumiblement lineal.

Finalment, les proves sobre el conjunt de dades associat a cada alumne (5 i 23) han acabat sense cap complicació, trobant solucions òptimes, solucions no acotades, i fins i tot no factibles.