

## Tp Simplex

Résoudre un programme linéaire grâce à la méthode du  
simplex <https://gitedu.hesge.ch/math/2e-annee/tp-math>

Juliano Souza Luz, Thibault Capt

18.01.2023

## TP Simplex

### Choix du langage

Nous avons donc utilisé le Java afin de résoudre ce travail pratique. En effet, python est un langage qui est très lent à l'exécution, car celui-ci est un langage **interprété**. Nous pouvons ainsi essayer de gagner du temps d'exécution pour la résolution du problème.

### Astuces utilisées

Ensuite, dans la plupart des cas, nous avons utilisé des **array** à 1 ou 2 dimension(s) qui est bien plus rapide que les Listes Java. (<https://stackoverflow.com/questions/716597/array-or-list-in-java-which-is-faster>).

Nous avons aussi opté pour que l'utilisateur puisse choisir un mode débogage ou non, ce qui permet de supprimer presque tous les prints en mode non-débogage et gagner énormément de temps d'exécution.

### Analyse des performances

Nous avons reçu plusieurs fichiers à tester pour vérifier le bon fonctionnement de notre code (nous approfondirons les résultats à "null" dans les problèmes rencontrés) :

- input.txt
  - Valeur obj : -34.25
  - nombre de pivots : 2
  - Temps d'exécution : 17ms
- input\_test.txt
  - Valeur obj : null
  - nombre de pivots : null
  - Temps d'exécution : null
- input\_test2.txt
  - Valeur obj : null
  - nombre de pivots : null
  - Temps d'exécution : null
- inputNonAdmissible.txt
  - Valeur obj : -34.25

- nombre de pivots : 3
  - Temps d'exécution : 20ms
- flow\_1.txt
  - Valeur obj : 78768
  - nombre de pivots : 9
  - Temps d'exécution : 33ms
- network1.txt
  - Valeur obj : 78768
  - nombre de pivots : 9
  - Temps d'exécution : 30ms
- network2.txt
  - Valeur obj : 0
  - nombre de pivots : 10
  - Temps d'exécution : 55ms
- network3.txt
  - Valeur obj : 84077
  - nombre de pivots : 45
  - Temps d'exécution : 162ms
- network4.txt
  - Valeur obj : null
  - Nombre de pivots : null
  - Temps d'exécution : null
- network5.txt
  - Valeur obj : NaN
  - Nombre de pivots: 1255
  - Temps d'exécution : 26066ms

Le temps d'exécution est, dans l'ensemble, assez faible, bien qu'il soit encore possible de l'améliorer.

### **Problèmes rencontrés**

Nous avons rencontré plusieurs problèmes au cours de ce travail pratique.

Premièrement, lors de la phase 1, nous ne gérons pas tous les négatifs des membres de droite. En effet, nous passons en paramètre la ligne où nous trouvons un négatif et remplaçait cette ligne en question. Le problème étant que s'il y avait plusieurs négatifs dans les membres de droite, il y resterait.

Deuxièmement et n'est pas résolu, nous pouvons voir que nous avons des résultats à null. En effet, ce problème survient lorsque notre résultat du tableau auxiliaire de la phase 1 (avant le pivot) est négatif. Par exemple, prenons le fichier `input_test`, nous ne trouvons pas de résultat. Nous avons essayé de changer la valeur de notre epsilon (initialement à  $1E-7$ ) mais cela n'y change rien.

Troisièmement, le fichier `network5` affiche des NaN (not a number). Il y a donc des divisions par 0 quelque part pour ce fichier. Le problème est qu'il y a ~1400 pivots pour ce fichier et ainsi le débogage est bien trop long pour regarder pivot par pivot.