Série 1 : Autour de l'algorithme de Gale-Shapley

Exercice 1: Algorithme de Gale-Shapley

On considère quatre femmes (A, B, C, D) et quatre hommes (a, b, c, d). La liste de préférences des femmes est :

A	c	b	d	a
B	b	a	c	d
C	b	d	a	c
D	c	a	d	b

et celle des hommes :

a	A	B	D	C
b	C	A	D	B
c	C	B	D	A
d	B	A	C	D

.

1. Appliquer l'algorithme de Gale-Shapley "côté homme", puis "côté femme".

2. Existe-t-il d'autres mariages stables ?

3. Quelqu'un vous propose un procédé qui consiste à choisir des divorces successifs : on commence par un couplage quelconque, p.ex. (Aa, Bb, Cc, Dd). S'il s'agit d'un couplage instable, comme c'est le cas dans le couplage choisi dans l'exemple, on détermine un couple formant une instabilité, ici A et b qui se préfèrent mutuellement, et on fait divorcer Aa et Bb. Ensuite, on remarie A avec b et B avec a. On obtient ainsi un nouveau couplage (Ab, Ba, Cc, Dd). On continue de la même manière jusqu'à l'obtention d'un couplage stable.

- Donner une séquence d'étapes conduisant à un couplage stable pour l'exemple ci-dessus.
- Le procédé qui consiste à choisir des divorces successifs de la façon indiquée plus haut conduit nécessairement à un couplage stable. *Vrai ou faux ?* Si c'est vrai le prouver, sinon donner un contre-exemple.

Exercice 2:

1. Dans chaque instance du problème du mariage stable, il existe un couplage stable contenant un couple (m, w) tel que m est le premier dans la liste de préférence de w et w est la première dans la liste de préférence de m. *Vrai ou faux*? Si c'est vrai donner une preuve, sinon donner un contre-exemple.

2. Considérons une instance du problème du mariage stable dans laquelle il existe un homme m et une femme w tels que m est le premier dans la liste de préférence de w et w est la première dans la liste de préférence de m. Dans ce cas, dans tout mariage stable S, le couple (m, w) appartient à S. *Vrai ou faux*?

3. Dans tout mariage stable, au moins une femme ou un homme est marié avec son meilleur partenaire (le premier de sa liste de préférence). *Vrai ou faux ?*

Exercice 3: Nombre de mariages stables

L'algorithme de Gale-Shapley garantit l'existence de mariage stable. On se demande ici s'il peut y avoir beaucoup de mariages stables.

1. S'il y a n hommes et n femmes, combien y a-t-il de mariages (couplages parfaits) possibles ?

2. Construire une instance avec 2 hommes et 2 femmes avec 2 mariages stables. Soit k le nombre de mariages stables d'une instance avec n hommes et n femmes. Construire une instance avec $2n$ hommes et $2n$ femmes où il y a (au moins) k^2 mariages stables. En déduire que le nombre de mariages stables peut être exponentiel en n.

Exercice 4: Problème des hôpitaux

Montrer la validité de l'extension vue en cours ('côté interne') de l'algorithme de Gale-Shapley au problème des hôpitaux.

Pour rappel : nous avons n internes, m hôpitaux, chaque hôpital ayant une capacité C_i, avec $\sum_{i=1}^{m} C_i = n$ (autant de places que de candidats).

On montrera :
— que l'algorithme termine ;
— que chaque interne est affecté ;
— que l'affectation obtenue est stable.

Rappel de l'algorithme :

1. Initialiser chaque interne/hôpital comme libre.
2. Tant que (il existe un interne libre qui n'a pas proposé à tous les hôpitaux)

```
Choisir un tel interne m
w = 1er hôpital dans la liste de m auquel m n'a pas encore proposé
si (w n'est pas complet)
    affecter m à w
sinon
    soit m' l'interne affecté à w 'le moins préféré' par w
    si (w préfère m à m')
        affecter m à w, et considérer m' comme libre
    sinon
        w rejette m
```

Série 2 : Efficacité, équité et programmation linéaire

Exercice 1: Résolution par les graphes

On considère les préférences suivantes.

A	a	d	c	b	e
B	c	b	e	d	a
C	a	b	c	d	e
D	c	d	b	e	a
E	e	c	a	b	d

a	A	C	D	E	B
b	C	B	A	D	E
c	E	D	C	B	A
d	B	D	A	E	C
e	E	C	A	B	D

.

1. Tracer le graphe permettant de savoir s'il existe un mariage où chaque personne obtient son premier choix. Est-ce possible ?

2. Tracer le graphe permettant de savoir s'il existe un mariage où chaque personne obtient un de ses deux premiers choix. Est-ce possible ?

3. Quelle est la valeur d'une solution optimale selon l'équité ?

Exercice 2: Efficacité

On considère les préférences suivantes.

A	a	b	c
B	b	a	c
C	b	c	a

a	C	A	B
b	A	B	C
c	A	C	B

.

1. Représenter le graphe (où les arêtes sont valuées) modélisant le problème de la recherche d'une solution optimale selon l'efficacité.

2. Quelle est selon-vous la valeur d'une solution optimale selon ce critère ?

Exercice 3: Efficacité et stabilité

Construire une instance admettant une solution d'efficacité maximale qui ne soit pas stable.

Exercice 4: PL et résolution graphique

Une entreprise fabrique deux types de ceintures A et B. Le type A est de meilleure qualité que le type B.

Le bénéfice net est 2 euros pour le type A et 1,5 euros pour le type B.

Le temps de fabrication pour le type A est 2 fois le temps de fabrication pour le type B ; si toutes les ceintures étaient du type B, l'entreprise pourrait en fabriquer 1000 par jour.

L'approvisionnement en cuir est suffisant pour 800 ceintures par jour (type A ou B).

Enfin, 400 boucles de type A et 700 boucles de type B sont disponibles chaque jour.

Le but est de maximiser le bénéfice total de l'entreprise.

1. Formuler le problème en un programme linéaire.
2. Résoudre le PL graphiquement.

Exercice 5: Encore un PL

Une entreprise a la faculté de fabriquer, sur une machine donnée pouvant fonctionner 40 heures par semaine, trois produits différents P1, P2 et P3.

L'article P1 laisse un profit net de 8 euros, l'article P2 de 24 euros, et l'article P3 de 6 euros.

Les rendements de la machine sont pour les trois produits P1 P2 et P3 respectivement 50, 25 et 75 articles par heure.

On sait, d'autre part, grâce à une étude de marché, que les possibilités de vente ne dépassent pas 1000 objets P1, 400 objets P2 et 1500 objets P3 par semaine.

On se pose le problème de répartir la capacité de production entre les trois produits de manière à maximiser le profit.

Formuler le problème comme un programme linéaire et le résoudre graphiquement.

Exercice 6: Couplages parfaits

Nous avons vu que la recherche d'une solution d'équité au moins k pour le problème du mariage revenait à déterminer l'existence d'un couplage parfait dans un graphe biparti. On s'intéresse ici à trouver une condition (idéalement nécessaire et suffisante) sur l'existence d'un couplage parfait dans un graphe biparti (avec autant de sommets à gauche et à droite).

1. Supposons que deux hommes h_1, h_2 n'aient, à eux deux, qu'un seul voisin f_1. Peut-il exister un couplage parfait ? Plus généralement, étant donné un sous-ensemble S d'hommes, on note $N(S)$ l'ensemble de leurs voisins. Donner une condition nécessaire sur S et $N(S)$ pour l'existence d'un couplage parfait.

2. Cette condition vous paraît-elle suffisante ?

3. Qu'en serait-il pour le problème des colocataires ?

Série 3 : Véracité et manipulation

Exercice 1: Enchère et véracité

Un ensemble $\{1, 2, \cdots, n\}$ de n personnes participe à une vente aux enchères d'un objet. Chaque personne i a en tête une valeur u_i de l'objet (le prix maximum qu'elle serait à payer pour l'avoir). Lors de la vente, chaque personne annonce (de manière simultanée) un prix - 'bid' - b_i. Un mécanisme d'enchère consiste, étant donné les b_i, à attribuer l'objet à une personne, et à donner le prix p que cette personne va payer l'objet.

Il est intéressant d'avoir un mécanisme incitant les participant à révéler leur vraie évaluation de l'objet, c'est-à-dire à déclarer comme *bid* $b_i = u_i$; cela évite les comportements stratégiques. Dans ce cadre, un mécanisme est dit à véracité garantie si une personne n'a jamais intérêt à déclarer un *bid* $b_i \neq u_i$, quels que soient les prix annoncés par les autres personnes.

Si une personne ne reçoit pas l'objet, son utilité est 0. Si elle reçoit l'objet en payant un prix p, son utilité est $u_i - p$. Chaque personne cherche bien sûr à maximiser son utilité.

Nous supposerons pour simplifier que tous les prix annoncés sont différents.

1. On considère le mécanisme de l'enchère au meilleur prix : la personne i^* ayant proposé le *bid* le plus élevé remporte l'enchère : elle achète l'objet au *bid* annoncé $p = b_{i^*}$.

Le mécanisme est-il à véracité garantie ?

2. On considère maintenant le mécanisme de l'enchère au deuxième prix (enchère de Vickrey) : la personne i^* ayant proposé le prix le plus élevé remporte l'enchère, mais elle achète maintenant l'objet au prix correspondant au 2ème *bid* annoncé le plus élevé. Ce mécanisme est-il à véracité garantie ?

Exercice 2: Une stratégie pour un interne

Vous êtes interne et vous êtes classé(e) premier(ère) dans l'hôpital H. L'algorithme de Gale-Shapley côté hôpitaux est utilisé. Vous avez vu en cours que cet algorithme n'est pas à véracité garantie. H n'est pas votre premier choix mais de peur de perdre le bénéfice de cette place, vous décidez de placer H en premier, de manière à être sûr(e) d'avoir ce poste. Bon ou mauvais calcul ?

Exercice 3: Manipulons les élections !

1. On considère le mécanisme d'élection à la présidence de la république française. Ce mécanisme est-il à véracité garantie ? *Note : dans le cas d'élection, on parle de (non) manipulabilité.*

2. Pouvez-vous proposer un règle électorale qui ne soit pas manipulable ?

Exercice 4: Véracité et efficacité

On considère les préférences suivantes :

A	b	c	a	d
B	a	b	c	d
C	c	b	d	a
D	d	a	b	c

a	A	B	C	D
b	B	A	D	C
c	C	B	D	A
d	D	A	B	C

.

1. Quelle est la solution maximisant l'efficacité totale ?

2. Vous êtes l'agent a et avez connaissance du tableau précédent : que faites-vous ?

3. Qu'en conclut-on quant à la véracité de la procédure ?

Série 4 : Annales d'examens

Exercice 1: (1ere session 2017-2018) **Algorithme de Boston**

On considère une instance du problème du mariage stable avec n hommes $h_1, \ldots, h_n$, n femmes $f_1, \ldots, f_n$, et pour chaque personne sa liste de préférence sur les personnes du sexe opposé.

On considère l'algorithme d'affectation de Boston, qui fonctionne comme ceci :

1. Chaque homme fait une proposition à la femme qu'il préfère ; les n propositions sont simultanées ;
2. Une femme peut recevoir 0, 1 ou plusieurs propositions. Si elle en reçoit (une ou plusieurs), elle accepte la meilleure pour elle, et cela de manière définitive. Cela forme un certain nombre de couples.
3. Les personnes en couples sont retirées de l'instance, et on itère cela tant que tout le monde n'est pas en couple.

On considère l'instance suivante :

A	d	c	a	b
B	d	b	c	a
C	a	d	b	c
D	a	c	b	d

a	B	A	D	C
b	A	C	B	D
c	C	A	D	B
d	B	C	D	A

.

1. Appliquer l'algorithme de Gale-Shapley côté hommes. Pour les propositions on prendra toujours les hommes dans l'ordre alphabétique. On donnera la séquence des propositions. Pour chaque proposition on précisera si elle est acceptée ou non, ainsi que l'ensemble des couples formés.

2. Appliquer l'algorithme de Boston sur l'instance. On donnera à chaque itération l'ensemble des propositions et des couples formés. La solution fournie est-elle stable (sur cet exemple) ?

3. On s'intéresse à savoir si l'algorithme de Boston est à véracité garantie.

1. On considère une instance avec 3 hommes A, B, C et trois femmes a, b, c. Les trois hommes ont les mêmes préférences : $a > b > c$. Les 3 femmes ont les mêmes préférences : $A > C > B$. Montrer à l'aide de cet exemple qu'un homme peut avoir intérêt à mentir sur ses préférences.
2. Une femme peut-elle avoir intérêt à mentir, ou l'algorithme de Boston est-il à véracité garantie côté femmes ? Vous justifierez votre réponse (preuve ou contre-exemple).

4. Montrer que l'algorithme de Boston ne donne pas toujours une solution stable. On pourra prendre un exemple avec 3 hommes et 3 femmes.

Exercice 2: *(1ère session 2015-2016)* **Stabilité**

1. Vrai/faux Indiquez si la proposition suivante est vraie ou fausse. Vous prouverez votre réponse.

Dans un mariage stable, il est impossible qu'il y ait un couple (h, f) tel que f soit en dernière position dans les préférences de h, et h en dernière position dans les préférences de f.

Exercice 3: *(2ème session 2017-2018)* programme linéaire et stabilité

1. On considère le problème du mariage stable avec 3 hommes et 3 femmes et les préférences suivantes.

h_1	f_3	f_2	f_1
h_2	f_2	f_3	f_1
h_3	f_3	f_2	f_1

f_1	h_2	h_3	h_1
f_2	h_3	h_1	h_2
f_3	h_2	h_3	h_1

Montrer, en donnant une paire instable, que la solution $(h_1-f_1), (h_2-f_2), (h_3-f_3)$ n'est pas stable.

Dans la suite, on considère une instance du mariage stable avec un ensemble $H = \{h_1, \ldots, h_n\}$ d'hommes et un ensemble $F = \{f_1 \ldots, f_n\}$ de femmes. On note x_{ij} une variable valant 1 si h_i et f_j sont en couple, 0 sinon ($i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$).

2. Rappeler les contraintes (vues en cours) exprimant le fait qu'une solution correspond à un mariage : toute personne est dans un et un seul couple (et il y a donc n couples).

3. Pour un homme h_i et une femme f_j, notons $P_i(j)$ l'ensemble des femmes que h_i préfère (strictement) à f_j, et $Q_j(i)$ l'ensemble des hommes que f_j préfère (strictement) à h_i.

On considère, pour un certain homme h_i et une certaine femme f_j, la contrainte suivante :

$$x_{ij} + \sum_{k:f_k \in P_i(j)} x_{ik} + \sum_{t:h_t \in Q_j(i)} x_{tj} \geq 1$$

Considérons la paire instable (h_i, f_j) de la question 1 : que valent $P_i(j)$ et $Q_j(i)$? Ecrire alors la contrainte précédente pour cette paire instable. La contrainte est-elle vérifiée par la solution de la question 1 ?

Expliquer de manière générale ce que représente la contrainte. On prendra soin d'argumenter précisément la réponse.

4. Exprimer le problème de la recherche d'un mariage stable d'utilité totale maximale (parmi les mariages stables) comme un programme linéaire sur les variables binaires x_{ij} (on notera w_{ij} l'utilité du couple (h_i, f_j)). Combien y a-t-il de contraintes ?

Exercice 4: *(1ère session 2015-2016)* Véracité

C'est l'anniversaire de votre ami Roberto. Vous donnez rendez-vous à ses autres amis pour discuter du cadeau. La première question est de se mettre d'accord sur le prix du cadeau. Chaque personne i a un prix idéal en tête v_i. Une personne est contente si le prix effectif p est proche de v_i : elle cherche plus précisément à minimiser $|v_i - p|$.

Vous cherchez un mécanisme pour décider du prix du cadeau.

1. Quelqu'un propose le mécanisme suivant : chaque personne donne un prix p_i, et le prix du cadeau p sera la moyenne des p_i. Ce mécanisme est-il à véracité garantie ? Vous prouverez votre réponse.

2. On considère maintenant l'élément médian : supposons pour simplifier que vous êtes un nombre impair d'amis $n = 2k + 1$. Etant donné les n prix p_i, le mécanisme donne comme prix du cadeau p le prix médian. Ce mécanisme est-il à véracité garantie ? Vous prouverez votre réponse.

Rappel : L'élément médian d'un ensemble de $2k + 1$ entiers est le $(k + 1)$ième lorsqu'on les classe par valeur croissante. Ainsi, l'élément médian de $\{3, 5, 6, 14, 17\}$ est 6.

Aide : pour une personne particulière, considérant les $2k$ prix annoncés par les autres, on pourra étudier le cas où le prix idéal de la personne serait le prix médian, puis le cas où il ne le serait pas.

TME, semaines 1 à 4

1 Problème et affectation

Brillant étudiant, vous prévoyez d'entrer l'année prochaine dans le master d'informatique de Sorbonne Université. Ce master comporte 9 parcours : ANDROIDE, BIM, DAC, IMA, RES, SAR, SESI, SFPN, STL. Le recrutement dans chacun des parcours se fait selon les dossiers, et selon les vœux exprimés par les étudiants.

Tout ce qui suit est bien entendu totalement fictif !

1. Allez sur le site du master, à l'adresse `www-master.ufr-info-p6.jussieu.fr/`, où vous pouvez trouver une description des différents parcours. Puis rendez vous à la page du sondage dont l'adresse est indiquée dans le fichier lien.txt sur le Moodle de l'UE (dossier "Fichiers TME 1 à 4") et renseignez votre ordre de préférence sur les parcours du master [1]. Attention à bien mettre une et une seule fois chaque parcours !

2. Téléchargez les deux fichiers tests TestPrefSpe.txt et TestPrefEtu.txt.

Fichier TestPrefSpe.txt : il correspond aux préférences des parcours sur un groupe fictif de 11 étudiants. La première ligne contient le nombre d'étudiants, la deuxième les capacités d'accueil des parcours (2 pour ANDROIDE,1 pour BIM,...), ensuite une ligne par parcours donnant les préférences sur les étudiants (0 ANDROIDE 7 ... signifie que l'étudiant 7 est classé premier par le parcours numéroté 0 ANDROIDE). Les parcours sont numérotés de 0 à 8.

Fichier TestPrefEtu.txt : il correspond à des préférences d'étudiants. La première ligne contient le nombre d'étudiants, puis ensuite une ligne par étudiant donnant ses préférences sur les parcours (0 Etu0 5 ... signifie que l'étudiant numéro 0, nommé Etu0, préfère SAR (parcours numéro 5), puis ...).

Note : deux fichiers contenant quelques instructions utiles en Python vous sont fournis, lisez la section 4, notamment si vous n'avez jamais utilisé Python.

3. Ecrivez une fonction lisant les fichiers des préférences des étudiants et des préférences des parcours, et créant deux matrices de préférences correspondantes. *Vous pourrez représenter une matrice simplement par une liste de listes.*

Codez l'extension de l'algorithme de Gale-Shapley aux problèmes des hôpitaux "côté étudiants", et appliquez l'algorithme sur les deux fichiers tests.

4. Proposez, puis codez, une adaptation de Gale-Shapley "côté parcours". Appliquez l'algorithme sur les deux fichiers tests.

5. (facultatif) Ecrivez une méthode prenant en entrée une affectation (et les matrices de préférences), et renvoyant la liste des paires instables. Vérifiez que les affectations obtenues dans les questions précédentes sont stables.

6. *Application aux préférences du groupe d'étudiants de l'UE.*

Chacun des parcours a produit un ordre de préférence (fictif [2] !) sur le groupe d'étudiants de l'UE, cf fichier PrefSpe.txt.

Téléchargez le fichier, ainsi que le fichier PrefEtu.txt correspondant aux préférences que vous avez exprimées.

7. Appliquez l'algorithme de Gale-Shapley côté étudiant, puis côté parcours. Dans quel parcours êtes-vous, dans chacun des deux cas ? Comparez plus généralement les affectations obtenues.

1. Si vous tenez à garder cela secret, mettez n'importe quoi.
2. Ces préférences ont été générées aléatoirement.

2 Evolution du temps de calcul

8. Ecrivez deux méthodes prenant en paramètre un nombre n d'étudiants :
- l'une générant un tableau des préférences de ces n étudiants sur les 9 parcours du master (préférences aléatoires),
- l'autre générant un tableau des préférences des 9 parcours du master sur les n étudiants. Les préférences seront aléatoires. Pensez à définir les capacités d'accueil des parcours (la somme devant faire n). On pourra générer des capacités de manière déterministe (et par exemple à peu près équilibrées entre les parcours).

9. Mesurez le temps de calcul de votre algorithme de Gale-Shapley pour différentes valeurs de n. Vous ferez plusieurs tests pour chaque valeur de n pour avoir une valeur significative). On pourra par exemple faire varier n de 200 à 2000 par pas de 200, et faire 10 tests pour chaque valeur de n. On pourra tracer une courbe représentant le temps de calcul (moyen) en fonction de n.

10. Quelle complexité obtient-on ? Est-ce cohérent avec la complexité théorique ?

11. (facultatif) Faites de même avec le nombre d'itérations. Combien fait-on d'itérations en moyenne ? Est-ce cohérent avec une analyse théorique de l'algorithme ?

3 Equité et PL(NE)

On souhaite maintenant trouver une solution (pas forcément stable) qui maximise l'utilité minimale des étudiants. Autrement dit, on veut trouver le plus petit k tel qu'il existe une affectation où tout étudiant a un de ses k premiers choix (utilité au moins $m - k$ pour chaque étudiant).

12. Ecrivez, pour un certain k, un PLNE permettant de savoir s'il existe une affectation où tout étudiant a un de ses k premiers choix.

13. Sur l'exemple, et pour $k = 3$, écrivez une méthode permettant de générer un fichier .lp correspondant au PLNE, cf section 5.

14. Résolvez le PLNE à l'aide de Gurobi (cf section 5). Existe-t-il une solution ?

15. Trouvez le plus petit k tel qu'il existe une solution. Décrivez une solution maximisant l'utilité minimale. Quelle était l'utilité minimale des solutions fournies par l'algorithme de Gale-Shapley (côté étudiants et côté parcours) ?

16. (facultatif) Ecrivez un PLNE maximisant l'utilité totale des étudiants, puis résolvez-le avec Gurobi. Quelle utilité moyenne obtient-on ?

17. (facultatif) Soit k^* le plus petit k trouvé à la question 15. Ecrire un PLNE maximisant l'utilité totale des étudiants parmi les solutions où chaque étudiant a un de ses k^* premiers choix.

18. (facultatif) Comparez les différentes solutions obtenues (GS côté étudiant, GS côté parcours, solutions des questions 15, 16 et 17) : stabilité, utilité moyenne, utilité minimale. Si vous avez fait la question 5, comparez aussi le nombre de paires instables dans chacune des solutions calculées.

4 Vous n'avez jamais utilisé Python ?

Pas d'inquiétude. Pour cette partie nous n'aurons besoin que de notions basiques d'algorithmique, et vous saurez facilement adapter la syntaxe (écriture de boucles, tests, écriture de fonctions,...) à ce langage proche d'un pseudo-code.

Afin de vous aider, vous pouvez télécharger les fichiers `main.py`, `exemple.py` et `test.txt`.

Lancez alors `Spyder`. Ouvrez et lisez les deux fichiers `.py` téléchargés précédemment.

Placez-vous sur le fichier `main.py`, puis cliquez sur le petit triangle (*Exécutez le fichier*). En bas à droite de la fenêtre apparaissent les affichages du code. Ce fichier fait appel aux fonctions définies dans le fichier `exemple.py` (commande `import exemple.py`). Vous trouverez dans ce fichier quelques-unes des commandes utiles pour votre projet. Pour le reste, je vous fais confiance !

5 Fichiers .lp et Gurobi

5.1 Fichier .lp

Le "format .lp" est un format standard d'écriture d'un programme linéaire.

En voici un exemple :

```
 Maximize
obj: x1 + 2 x2 + 3 x3 + x4
Subject To
c1: - x1 + x2 + x3 + 10 x4 <= 20
c2: x1 - 3 x2 + x3 <= 30
c3: x2 - 3.5 x4 = 0
Bounds
0 <= x1 <= 40
-2 <= x4 <= 3
Binary
x3 x4
End
```

Aide : pour des variables x_{ij} à deux indices, pensez à séparer les indices dans vos noms de variables (par exemple x_{2_3}). Attention également à bien mettre des espaces (entre coefficients, variables, ...) comme dans l'exemple ci-dessus.

5.2 Solveur

Nous allons utiliser le solveur Gurobi pour résoudre le(s) PLNE. Gurobi prend en entrée un fichier (par exemple au format lp). Avant de pouvoir l'utiliser directement, taper les 3 commandes ci-dessous :

```
export GUROBI_HOME="/opt/gurobi801/linux64"
export PATH="${PATH}:${GUROBI_HOME}/bin"
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${GUROBI_HOME}/lib"
```

Maintenant, en vous plaçant dans le répertoire contenant votre fichier `fichierpl.lp`, vous pouvez taper la commande :

```
/opt/gurobi801/linux64/bin/gurobi_cl ResultFile=affectation.sol fichierpl.lp
```

La solution trouvée est alors décrite dans le fichier `affectation.sol`.

Voici une solution alternative (ceci évite d'avoir a retaper les commandes ci-dessus à chaque session) :

- Allez à la racine de votre répertoire, et regardez si vous avez un fichier .bashrc (>ls -a car fichier caché).
- Si c'est le cas, ajoutez à la fin de votre fichier les 3 commandes :
  `export GUROBI_HOME="/opt/gurobi801/linux64"`

```
export PATH="${PATH}:${GUROBI_HOME}/bin"
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${GUROBI_HOME}/lib"
```

— Sinon, créer un fichier nommé .bashrc contenant ces trois commandes à la racine de votre répertoire.
— Déloguez-vous, reloguez-vous.