

Langages recommandés : Java ou Python

1 Prise en main de `lp_solve`

Dans tout le TP, nous allons utiliser le programme `lp_solve`, qui est un solveur de programmation linéaires. La documentation se trouve « [ICI](#) »

Voilà un exemple de programme linéaire sous le format de base de `lp_solve` :

```
max: 2 x1 + x2;  
x1 + 2 x2 <= 20;  
3.2 x1 + x2 <= 15;
```

Pour faire tourner le solveur, il suffit de taper la commande :

```
$ lp_solve fichier.lp
```

où `fichier.lp` contient l'exemple ci-dessus. Le résultat est affiché comme ceci :

```
Value of objective function: 12.77777778  
  
Actual values of the variables:  
x1                      1.85185  
x2                      9.07407
```

Si on veut se restreindre à des `x1` et `x2` entiers il suffit d'ajouter la ligne suivante :

```
int x1, x2;
```

► **Exercice 1** ◀ Vérifiez que tout fonctionne correctement sur l'exemple ci-dessus. Quelles sont les valeurs optimales de `x1` et `x2` si on se restreint à des nombres entiers ?

► **Exercice 2** ◀ Vérifiez que vous trouvez bien les mêmes solutions à l'exercice 1 de la feuille de TD 1 avec le solveur que celles que l'on a déterminées graphiquement.

2 Un problème générique

On considère un problème plus général où l'on dispose de m types de ressources $R_1, \dots, R_m$ pour fabriquer n types de produits $P_1, \dots, P_n$. Chaque produit j rapporte un bénéfice de c_j pour chaque unité vendu.

Les différentes statistiques associées aux produits sont données dans un fichier texte formaté de la façon suivante :

```
3 2
350 750 600
P1 10 30 10 150
P2 10 10 20 100
```

- La première ligne indique le nombre de ressources m et de produits n .
- La deuxième ligne indique les limites sur les m ressources (ex: au plus 350 unités de la première ressource).
- Les n lignes qui suivent décrivent les produits. La première colonne est le nom du produit (ex: P2). Ensuite, suivent m colonnes avec les besoins en ressources (ex: 10 10 20). La dernière colonne est le bénéfice réalisé par unité de ce produit qui est vendue (ex: 100).

On peut fabriquer des fractions d'unité de produits, et on considère que tout sera vendu.

► **Exercice 3** ◀ En utilisant le fichier que vous pouvez récupérer « [ICI](#) », écrivez un programme qui prend en entrée un tel fichier et écrit sur un autre fichier de sortie le programme linéaire associé.

- ▷ Faites tourner le solveur, quel est le bénéfice optimal ?
- ▷ Combien de produits différents faut-il fabriquer ?
- ▷ Combien de temps a mis le solveur pour calculer la solution optimale ?

► **Exercice 4** ◀ Rajoutez la contrainte qu'on ne peut plus faire de fractions d'un produit, il faut en produire un nombre entier. Répondez avec cette contrainte aux questions de l'exercice précédent.

► **Exercice 5** ◀ Essayez avec le fichier (beaucoup) plus volumineux que vous pouvez trouver « [ICI](#) ».

3 Un problème de découpe

► **Exercice 6** ◀ Reprendre l'exercice de la feuille de TD 2 sur la découpe de barres de métal et trouvez la solution optimale avec le solveur.

► **Exercice 7** ◀ On considère maintenant que les tiges de base font 5,0 m et qu'on veut les découper en barres de longueurs 2,0 m, 1,2 m, 1,0 m et 0,5 m. Ecrire un programme qui génère tous les découpages maximaux possibles. Adaptez le pour générer le programme linéaire correspondant à une commande qui minimise le nombre de tiges utilisées pour produire 60 barres de 2,0 m, 100 de 1,2 m, 150 de 1,0 m et 350 de 0,5 m.

4 Le marchand de glace

► **Exercice 8** ◀ Reprendre le problème du marchand de glace vu en cours pour les demandes suivantes :

jan	350
fev	320
mar	440
avr	630
mai	630
jun	550
jui	680
aou	660
sep	350
oct	420
nov	380
dec	620

Visualiser dans un diagramme les différents plans de production optimaux (les x_i pour $i = 1, \dots, 12$) pour les coûts suivants :

- (a) coût de stockage : 0 €/tonne et mois, coût de réajustement : 100 €/tonne et mois
- (b) coût de stockage : 20 €/tonne et mois, coût de réajustement : 50 €/tonne et mois
- (c) coût de stockage : 50 €/tonne et mois, coût de réajustement : 20 €/tonne et mois
- (d) coût de stockage : 100 €/tonne et mois, coût de réajustement : 0 €/tonne et mois