

# Théorie des graphes et Recherche Opérationnelle

STI 3A Année 2016-2017

Recherche Opérationnelle  
Solveurs de problèmes linéaires

## Préambule

Il existe un certain nombre de logiciels permettant de résoudre des programmes linéaires. Parmi eux, nous pouvons citer **Cplex**, développé par la société Ilog (racheté par IBM) <http://www-142.ibm.com/software/products/fr/fr/ibmilogcple/>. Ce logiciel (payant) est l'un des plus performants sur le marché. Il traite différents types de problèmes, comme les programmes linéaires et les programmes quadratiques, en continu, mais aussi en nombres entiers. Il permet de résoudre des problèmes de grande taille. Cette notion est relative car elle est reliée à la difficulté intrinsèque du problème et de l'instance traitée : un programme linéaire en nombres entiers nécessite des ressources importantes, car il utilise en général des techniques de type séparation/évaluation (Branch&Bound) pour trouver les solutions optimales.

Dans le domaine des logiciels libres, **lpsolve** est distribué sous licence **LGPL** (<http://lpsolve.sourceforge.net>). Il permet de résoudre des programmes linéaires de taille raisonnable. Il peut être utilisé pour effectuer une première évaluation d'un problème avant d'utiliser un produit commercial. Plusieurs versions sont disponibles sous tous les systèmes d'exploitation. Sous Windows, il existe une interface graphique simple. Nous allons mettre en évidence deux utilisations de ce logiciel :

1. une interface en ligne de commande ;
2. une API que l'on peut interfacer avec de nombreux langages (comme, par exemple, C, C++, Java, VB.Net)

Une documentation complète se trouve ici : <http://lpsolve.sourceforge.net/5.5/>

## 1 Utilisation de l'interface en ligne de commande

La commande à utiliser est **lp\_solve**. Elle prend en entrée un fichier représentant le problème étudié sous forme simple. Différents formats sont acceptés, mais le plus simple est le format **.lp**.

Un exemple simple de fichier est :

```
min: -x1 -2 x2 +0.1 x3 +3 x4;
r_1: +x1 +x2 <= 5;
r_2: +2 x1 -x2 >= 0;
r_3: -x1 +3 x2 >= 0;
r_4: +x3 +x4 >= 0.5;
      x3 >= 1.1;
```

```
int x3, x4;
```

Si ce texte est sauvé dans le fichier **probleme.lp**, le résultat de la commande :  
**lp\_solve < probleme.lp** est :

Value of objective function: -8.13333

Actual values of the variables:

x1	1.66667
x2	3.33333
x3	2
x4	0

1. En utilisant ce formalisme, reprendre les exemples du TD 2, en particulier, sur les gants de toilette et les serviettes.
2. Considérer le cas où on ne doit plus avoir que 80 serviettes de bain.

## 2 Utilisation de l'API lpsolve

Lorsque les problèmes sont de grande taille, il peut être fastidieux de créer un fichier spécifique correspondant à l'instance considérée. Il faut alors automatiser par programmation ce processus. Cette technique atteint ses limites pour résoudre certains types de problèmes.

En effet, il peut être judicieux de ne prendre en compte qu'une partie des contraintes. On peut montrer de manière théorique que si ce premier problème a une solution, alors c'est la solution du problème global. Si cela n'est pas le cas, on ajoute des contraintes supplémentaires qui affinent la définition du problème. D'autres techniques sont utilisées pour augmenter dynamiquement le problème. Lors de la résolution, le solveur peut utiliser la solution déjà calculée sur le système précédent ce qui donne un point de départ pour le calcul du simplexe.

Vous trouverez sur le serveur enseignement, les fichiers d'exemples servant de base pour ce TD.

1. Compiler le programme `Exemple.c` et faites le tourner.
2. Résoudre le problème suivant :

```
max: x1 + 2 x2 ;  
r_1: +x1 +x2 <= 5;
```

3. Ajouter la contrainte supplémentaire :

```
r_2: +2 x1 -x2 >= 0;
```

4. Compléter le problème en ajoutant une variable supplémentaire  $x_3$  dont le coefficient dans la fonction objectif est 3, dans la contrainte  $r_1$  est +1 et dans la contrainte  $r_2$  est -2.

## 3 Affectation de projets

Le programme `Model.c` présente une implémentation du problème d'affectation des projets à des groupes suivant leurs souhaits. Les deux fichiers `PB.data` et `Proj.data`.

Le programme modélise le problème d'affectation de projet où on cherche à minimiser le mécontentement global.

1. Faire tourner le programme avec les exemples donnés. Afficher le problème résolu.
2. Programmer la version de ce problème où on cherche à minimiser le mécontentement maximal.