Introduction à AMPL

(Logiciel de modélisation et de résolution de problèmes d'optimisation)

Julien Ah-Pine (julien.ah-pine@eric.univ-lyon2.fr)

Université Lyon 2

M1 Informatique 2015-2016

Introduction

- AMPL signifie "A Mathematical Programming Language"
- C'est un **langage de modélisation algébrique** pour décrire et résoudre des problèmes d'optimisation. Il en existe d'autres comme GAMS, LINGO, ...
- Il existe une version libre mais limitée (à 300 variables et 300 contraintes) pour les étudiants téléchargeable à l'adresse suivante : `http://www.ampl.com/DOWNLOADS/index.html`

Fonctionnement

- Etant donné un problème d'optimisation, AMPL permet de résoudre celui-ci de la manière suivante :
  - Ecrire le problème en langage AMPL en isolant la formulation du modèle (fichier .mod) des paramètres (ou données) du modèle (fichier .dat)
  - Sélectionner un solver qui est un logiciel (indépendant d'AMPL) permettant de résoudre un ou plusieurs types de problèmes d'optimisation
  - AMPL permet d'invoquer plusieurs solver. Autrement dit AMPL permet de passer le problème décrit dans son langage au solver pour le résoudre
  - Une fois le problème résolu par le solver, AMPL permet de récupérer le résultat et d'interpréter celui-ci dans son langage

Fonctionnement (suite)

- Pour utiliser AMPL on a en général recours à 3 fichiers :
  - Un fichier .mod dans lequel on définit formellement le modèle
  - Un fichier .dat dans lequel on définit les valeurs numériques ou les données du cas d'étude
  - Un fichier .run dans lequel on définit le problème à résoudre (en faisant notamment appel à un fichier .mod et un fichier .dat) et qui est exécuté par AMPL

- Pour ajouter des commentaires il faut utiliser la commande #
- Toutes les commandes AMPL se terminent par ;
- Pour réinitialiser ampl (lorsque vous rencontrez des erreurs) il faut utiliser la commande `reset;`

Fichier de modélisation .mod

- Le fichier .mod est composé (de manière non exhaustive), des commandes suivantes :
  - `set` *nom-de-l-ensemble* **attribut** ;
  - `param` *nom-du-paramètre* **attribut** ;
  - `var` *nom-de-la-variable-de-décision* **attribut** ;
  - `minimize` *nom-de-la-fonction-objectif-f* : *définition-de-f* ;
  - `subject to` *nom-de-la-contrainte-h* : *définition-de-h* ;

Fichier des données .dat

- Le fichier .dat est composé (de manière non exhaustive), des commandes suivantes :
  - `data ;` (optionnel)
  - `set` *nom-de-l-ensemble* `:=` *définition* `;`
  - `param` *nom-du-paramètre* `:=` *valeur-du-paramètre* `;`
    (s'il s'agit d'une constante)
  - `param` *nom-du-paramètre* `:=`
    ind-premier-élément val-premier-élément
    $\vdots$
    ind-dernier-élément val-dernier-élément `;`
    (s'il s'agit d'un vecteur)

Fichier des données .dat (suite)

- `param` *nom-du-paramètre* :
  ind-première-col ... *ind-dernière-col* :=
  ind-première-lig *val-prem-lig-prem-col* ... *val-prem-lig-dern-col*
  ⋮
  ind-dernière-lig *val-dern-lig-prem-col* ... *val-dern-lig-dern-col* ;
  (s'il s'agit d'une matrice)

Fichier d'exécution .run

- Le fichier .run est composé (de manière non exhaustive), des commandes suivantes :
  - `model` *problem.mod* ;
  - `data` *problem.dat* ;
  - `option solver` **nom-du-solver** ;
  - `solve ;`
  - `display` *nom-de-la-variable-de-décision* ;

Quelques solveurs associés à AMPL

- `CPLEX` : problèmes linéaires et quadratiques
- `MINOS` : problèmes linéaires et non-linéaires

Exemple

Une entreprise fabrique deux biens (des pièces mécaniques par exemple). Ces fabrications nécessitent l'utilisation de deux ateliers dont les capacités de production exprimées en heures d'usinage sont de 12. Supposons que :

- Chaque unité du 1er produit nécessite 2h d'usinage dans l'atelier 1 et 1h dans l'atelier 2
- Chaque unité du 2ème produit nécessite 1h d'usinage dans l'atelier 1 et 2h dans l'atelier 2

Sachant que la marge sur le 1er produit est $p_1 = 4$ et que celle sur le 2ème produit est de $p_2 = 3$, déterminer un programme mathématique qui modélise le problème de l'optimisation de la marge de l'entreprise sous les contraintes de production décrites précédemment

Fichier profit.mod

```
#Paramètres du problème
param nb_produit integer;#nombre de produits
param nb_atelier integer;#nombre d'ateliers
set P:=1..nb_produit;#ensemble des produits (suite d'entiers)
set A:=1..nb_atelier;#ensemble des produits (suite d'entiers)
param prix_produit{P};#vecteur des prix des produits
param temps_produit_atelier{A,P};#matrice des coefficients des contr
param capacite_atelier{A};#vecteur des capacités max des ateliers

#Variables de décision
var quantite_produit{P}>=0;#vecteur des variables de décision

#Fonction objectif et contraintes du problème
maximize profit : sum{i in P} prix_produit[i]*quantite_produit[i];
#Contraintes du problème : chaque atelier a une contrainte
subject to heures_max_atelier{i in A}:
sum{j in P}temps_produit_atelier[i,j]*quantite_produit[j]
<= capacite_atelier[i];
```

Fichier profit.dat

```
data;

param nb_produit := 2;#donc P = {1,2}
param nb_atelier := 2;#donc A = {1,2}
param prix_produit :=
1 4
2 3;
param temps_produit_atelier :
1 2 :=
1 2 1
2 1 2;
param capacite_atelier :=
1 12
2 12;
```

Fichier profit.run

```
model profit.mod;
data profit.dat;
option solver lpsolve;
solve;
display quantite_produit;
display profit;
```

Exécution et output

- Pour exécuter un programme AMPL :
  - Exécution sous Linux : `ampl profit.run` ou `cat profit.run | ampl`
  - Exécution sous Windows : exécuter au préalable `sw` puis à l'invite : `sw: ampl profit.run`
  - Sous Linux ou Windows on peut intéragir avec AMPL en exécutant la commande `ampl` pour le premier ou en tapant `sw: ampl` (après avoir exécuter `sw`) pour le second. On exécute alors de manière intéractive et les unes après les autres les commandes contenues dans `profit.run`
- Voici la sortie que nous obtenons :

```
LP_SOLVE 4.0.1.0: optimal, objective 28
2 simplex iterations
quantite_produit [*] :=
1  4
2  4
;

profit = 28
```

Exercice 1

- Allez sur la page `http://www.ampl.com/DOWNLOADS/` et dans la section "AMPL Command Line download for Windows", télécharger le fichier `ampl.mswin64.zip` sur le bureau puis décompresser l'archive
- Connectez-vous au BV et télécharger les fichiers `profit.run`, `profit.mod`, `profit.dat` en les sauvegardant dans le répertoire `ampl.mswin64`
- Ouvrez chaque fichier `profit.*` avec un éditeur de texte comme `bloc-notes`
- Dans le répertoire `ampl.mswin64`, lancer le programme `sw`
- A l'invite `sw:` entrez la commande `ampl profit.run`. Qu'observez-vous ?
- A l'invite `sw:` entrez la commande `ampl`. Qu'observez-vous ?
- A l'invite `ampl:` entrez successivement les commandes `model profit.mod;`, `data profit.dat;`, `option solver cplex;`, `solve;`, `display quantite_produit;`. Qu'observez-vous ?

Exercice 2

- Utilisez AMPL pour déterminer le minimiseur de la fonction $f(x) = x^4 - 14x^3 + 60x^2 - 70x$ avec les contraintes de bornes suivantes : $0 \leq x \leq 2$
  - Ouvrez un fichier `ex2.mod` dans lequel vous écrirez le modèle avec deux paramètres lb et ub comme étant les bornes de la variable de décision x
  - Ouvrez un fichier `ex2.dat` dans lequel vous écrirez les valeurs des paramètres lb et ub
  - Ouvrez un fichier `ex2.run` dans lequel vous écrirez les commandes pour résoudre le problème avec AMPL et afficher la valeur optimale de x trouvée. Vous utiliserez pour cela le solveur `minos`

Exercice 3

- Utilisez AMPL pour déterminer le minimiseur de la fonction $f(x) = \frac{1}{2}x^2 - \sin(x)$ avec les contraintes de bornes suivantes : $0 \leq x \leq 1$
  - Ouvrez un fichier `ex3.mod` dans lequel vous écrirez le modèle avec deux paramètres *lb* et *ub* comme étant les bornes de la variable de décision x
  - Ouvrez un fichier `ex3.dat` dans lequel vous écrirez les valeurs des paramètres *lb* et *ub*
  - Ouvrez un fichier `ex3.run` dans lequel vous écrirez les commandes pour résoudre le problème avec AMPL et afficher la valeur optimale de x trouvée. Vous utiliserez pour cela le solveur `minos`

Exercice 4

- Utilisez AMPL pour déterminer le minimiseur de la fonction multidimensionnelle $f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4$
  - Ouvrez un fichier `ex4.mod` dans lequel vous écrirez le modèle avec trois variables de décision x_1, x_2, x_3. Vous écrirez la fonction objectif $(x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4$ avec ces 3 variables. Remarquez que dans ce problème et tel qu'il est défini, il n'y a pas de paramètres donc pas de fichier .dat puisque toutes les valeurs sont entrées en dur dans le .mod.
  - Lancez `ampl`, choisissez comme solveur `minos`, et entrez le modèle défini précédemment
  - Résoudre le problème en utilisant la commande `solve` et afficher le minimiseur à l'aide de la commande `display`

Exercice 4 bis

- Utilisez AMPL pour déterminer le minimiseur de la fonction multidimensionnelle $f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4$ mais dans ce cas vous écrirez un programme générique.
  - Ouvrez un fichier `ex4_bis.mod` dans lequel vous écrirez le modèle avec un ensemble `I` des indices du vecteur $\mathbf{x}$ des variables de décision. Vous définirez une fonction objectif sous la forme générique suivante : $\sum_{i \in \mathtt{I}} b_i(x_i + c_i)^{p_i}$ où $\mathbf{b}, \mathbf{c}, \mathbf{p}$ sont des vecteurs de paramètres.
  - Ouvrez un fichier `ex4_bis.dat` dans lequel vous instancierez le problème particulier de cette exercice
  - Lancez `ampl`, choisissez comme solveur `minos`, et entrez le modèle défini précédemment
  - Résoudre le problème en utilisant la commande `solve` et afficher le minimiseur à l'aide de la commande `display`

Exercice 5

- Modélisez et résolvez l'exercice 3.3 du TD avec AMPL
  - Ouvrez un fichier `ex5.mod` dans lequel vous écrirez le modèle sous forme initiale (non standard) en déclarant les variables x1 et x2 telles que celles-ci soient ≥ 0 et en écrivant les contraintes directement sans devoir déclarer des paramètres dans un fichier .dat
  - Lancez `ampl`. Entrez successivement les commandes suivantes `model ex5.mod;`, `solve;`. Qu'observez-vous ?
  - Considérer alors le modèle avec les contraintes telles ques les membres de droite sont 2 et 2 au lieu de 1 et 3. Ouvrez un fichier `ex5_bis.mod` dans lequel vous écrirez ce nouveau modèle. Entrez ensuite les mêmes commandes AMPL que précédemment après avoir réinitialisez `ampl` avec la commande `reset;`. Qu'observez vous ? Donnez la valeur des variables obtenues.
  - Ouvrez un fichier `ex5_ter.mod` dans lequel vous écrirez le modèle étudié précédemment sous forme standard en prenant $M = 100$. Répétez les mêmes commandes que précédemment. Qu'observez vous ?

Rappel du PL initial de l'exercice 3.3 du TD

$$\max f(x_1, x_2) = x_1 + 2x_2 \; slc \begin{cases} x_1 + x_2 \leq 1 \\ \frac{3}{2}x_1 + x_2 \geq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

Exercice 6

- Modélisez et résolvez l'exercice 3.5 du TD (entreprise de production postes télé) avec AMPL
  - Ouvrez un fichier `ex6.mod` dans lequel vous écrirez le modèle avec comme paramètre l'ensemble `M` de moniteurs; les vecteurs `montage_heure_moniteur`, `test_heure_moniteur`, `prix_moniteur`; les constantes `capacite_montage`, `capacite_test`, `capacite_transfo`, `capacite_tccouleur` et enfin le vecteur des variables de décision `quantite_moniteur`
  - Ouvrez un fichier `ex6.dat` dans lequel vous écrirez les valeurs des différents paramètres
  - Ouvrez un fichier `ex6.run` dans lequel vous écrirez les commandes pour résoudre le problème avec AMPL et afficher la valeur optimale trouvée. Vous utiliserez pour cela le solveur `cplex`

Rappel de l'énoncé de l'exercice 3.5 du TD

Un constructeur de postes de télévision possède 4 modèles à son catalogue : le portatif NB (M1), le standard NB (M2), le standard couleur (M3) et le couleur de luxe (M4). L'entreprise comporte un atelier de montage et un de tests. Les durées nécessaires pour le montage et test des différents modèles sont données dans la table 1.

	M1	M2	M3	M4
Montage (en heures)	8	10	12	15
Tests (en heures)	2	2	4	5

Table : Durées nécessaires pour le montage et le test des différents modèles

La force de travail de l'atelier de montage est de 6000 heures/mois, celle de l'atelier de tests est de 1500 heures/mois et les profits des postes M1, M2, M3 et M4 sont respectivement de 400, 600, 800 et 1000 euros. L'entreprise dispose chaque mois de 450 transformateurs et de 300 tubes cathodiques couleur. On a besoin d'un transformateur dans chaque poste (NB ou couleur). La quantité disponible de tubes cathodiques NB n'est pas limitée.