Petite introduction à AMPL et CPLEX

B. CORNÉLUSSE

Université de Liège
Département d'électricité, électronique et informatique

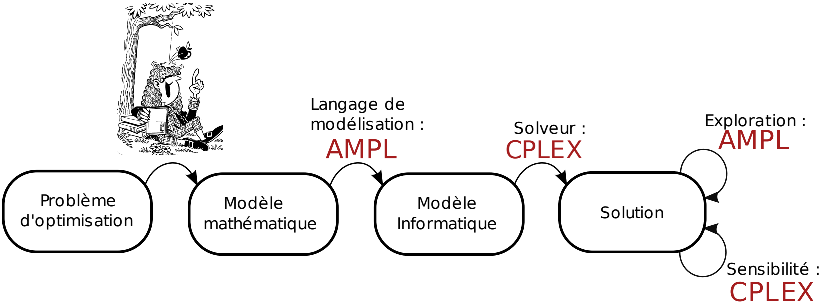
Liège, Mars 2008

Sommaire

Sommaire

Introduction

- AMPL :
  1. décrire le modèle sous forme informatique,
  2. explorer la solution.
- CPLEX :
  1. résoudre le problème,
  2. réaliser l'analyse de sensibilité.



Sommaire

Description

[Répétition 1, question 4.1]
Une société lance un appel d'offre pour construire un nouveau bâtiment. L'entreprise de construction (...) doit réaliser les travaux le plus vite possible. Les principales tâches à accomplir :

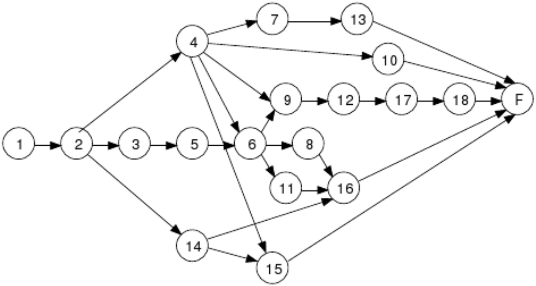
Tâche	Description	Durée	prédécesseur(s)	réduction max	sur-coût/ semaine
1	Installer le site de construction	2	-	0	-
2	Terrassement	16	1	3	30
...					
17	Réaliser les aménagements extérieurs	9	12	3	16
18	Réception du bâtiment	1	17	0	-

- Les durées sont en semaines.
- Certaines tâches ne peuvent débuter qu'après que d'autres tâches soient terminées.

Quel est le nombre minimum de semaines pour terminer le travail ?

Modèle mathématique

- On introduit une tâche fictive de durée 0, à la fin.
- Ensemble $TASKS = \{1, ..., N\}$, N est la tâche fictive.
- Durée de la tâche i : DUR_i.
- On symbolise les relations entre les tâches par un graphe de précédence, $G = (TASKS, ARCS)$, $ARCS$ est l'ensemble des arcs (i, j) (la tâche i précède la tâche j) :



Modèle mathématique (...)

- Variables : $start_i$, début de la tâche i.
- Formulation du problème :

$$
\begin{aligned}
\min \;& start_N \\
s.t \;& \forall i \in TASKS, \; start_i \geq 0 \\
& \forall (i,j) \in ARCS, \; start_i + DUR_i \leq start_j
\end{aligned}
$$

Modélisation avec AMPL (scheduling1.mod)

```
## Exemple 1

# Parametres
param N;
set ARC dimen 2;
param DUR{1..N};

# Variables
var start{1..N} >= 0;

# Objectif
minimize endTime: start[N];

# Contraintes
subject to prec{(i,j) in ARC}: start[i] + DUR[i] <= start[j];
```

Dans un fichier séparé, les données : (scheduling1.data)

```
data; # Pour l'exemple 1.
param N := 19;
param : DUR :=
      1     2
      2     16
.
.
.
      17    9
      18    1
      19    0;
set ARC := (1,2) (2,3) (2,4) (3,5) (4,6) (5,6) (4,7) (6,8)
    (4,9) (6,9) (4,10) (6,11) (9,12) (7,13) (2,14) (4,15)
    (14,15) (8,16) (11,16) (14,16) (12,17) (17,18) (10,19)
    (13,19) (15,19) (16,19) (18,19);
```

Calculer la solution du problème (`scheduling1.run`)

Exécuter `ampl scheduling1.run`.

```
#Lire le modele
model scheduling1.mod;

#Lire les donnees
data scheduling1.data;

# Choisir le solveur
option solver cplex;

# Options du solveur
option cplex_options 'sensitivity';

# Appeler le solveur
solve;

# Analyser les resultats
# La solution
display start > scheduling1.sol;

# Les contraintes
display {j in 1.._ncons}
  (_conname[j],_con[j].down,_con[j].current,_con[j].up,_con[j]) >> scheduling1.sol;

# Les variables
display {j in 1.._nvars}
  (_varname[j],_var[j].up,_var[j].current,_var[j].down) >> scheduling1.sol ;
```

Pour commencer, utiliser l'interpréteur. Exécuter `ampl`, puis rentrer les commandes de `scheduling1.run` (et d'autres !) unes à unes.

Résultats, voir `scheduling1.sol`

```
CPLEX 10.1.0: optimal solution; objective 64
0 dual simplex iterations (0 in phase I)
start [*] :=
 1   0
 2   2
 3  18
 4  18
 5  27
 6  37

.
.
.

16  46
17  54
18  63
19  64
;
```

Sommaire

Description

[Répétition 1, question 4.2]

- La durée obtenue à la question 4.1 est trop longue.
- Pour la réduire la société est prête à payer à l'entreprise de construction un bonus de 30000 euros par semaine gagnée.
- L'avant-dernière colonne de la table ci-dessus représente le nombre maximum de semaines qui peuvent être gagnées par tâche.
- La dernière colonne représente le coût additionnel associé par semaine (exprimé en tranches de 1000 euros).

Quelle est la durée minimale pour réaliser les travaux si l'entreprise de construction souhaite maximiser son profit ?

Modèle

Soit $OBJ1$ la durée minimale solution du problème de la question 1. On introduit de nouvelles variables $save_i, \forall i \in TASKS \backslash \{N\}$, représentant le temps épargné sur la tâche i.

$$\begin{aligned}
\max \quad & (OBJ1 - start_N)\, 30000 - \sum_{i \in TASKS \backslash \{N\}} REDCOST_i \, save_i \\
s.t \quad & \forall i \in TASKS, \; start_i \geq 0 \\
& \forall i \in TASKS \backslash \{N\}, \; 0 \leq save_i \leq MAXRED_i \\
& \forall (i,j) \in ARCS, \; start_i + DUR_i - save_i \leq start_j
\end{aligned}$$

Voir `scheduling2.mod`, `scheduling2.data` et `scheduling2.run`

Sommaire

Informations pratiques

- Version “étudiante” d’AMPL et CPLEX sur le site `http://www.ampl.com/DOWNLOADS/index.html`. Vous devez télécharger deux choses : AMPL et CPLEX. Pour y parvenir, lire les instructions se trouvant sur cette page.
- Fichiers utilisés dans cettes présentation : `http://www.montefiore.ulg.ac.be/~louveaux/optim.html`
- Je vous conseille de lire cette introduction à AMPL : `http://www.core.ucl.ac.be/wolsey/ampl.pdf`