

# Sujet 1: Introduction des logiciels pour l'optimisation

MSE3113: Outils et logiciels pour l'optimisation

Andrew J. Miller

Dernière mise au jour: October 26, 2011

Tous les fichiers peuvent être téléchargé à  
<http://www.math.u-bordeaux1.fr/~amiller/cours/>.

## Dans ce sujet...

- 1 Logiciels pour l'optimisation
- 2 Tableurs et langues algébriques de modélisation (LAMs)
- 3 Planification de production à Ajax

- 1 Logiciels pour l'optimisation
- 2 Tableurs et langues algébriques de modélisation (LAMs)
- 3 Planification de production à Ajax

# Motivations

- On se trouve souvent dans les situations qui sont **trop complexes** pour qu'un humain puisse considérer tous les aspects importants de la situation, et tous les relations entre eux, en même temps.
- On a besoin de résoudre des modèles avec une quantité de données réalistes, c'est à dire trop grand pour trouver une solution réalisable de bonne qualité sans aide informatique.
- On a aujourd'hui la capacité de ramener, garder, et accéder aux **quantités de données énormes**.
  - logiciels de base de données
  - variété et capacité des réseaux informatiques
- De plus en plus on a accès a une **capacité de calcul énorme**.

On a des capacités et ressources informatiques immenses. On a besoin de les utiliser. Mais...

## Comment utiliser ces ressources?

...ça ne sert pas à grande chose si on ne peut pas utiliser ces ressources pour **prendre des meilleures décisions**.

Il faut savoir utiliser des outils qui peuvent fournir **l'aide à la décision**.

Beaucoup de ces outils sont des **logiciels d'optimisation**.

## Trois ingrédients

Chaque outil de calcul qu'on peut utiliser pour optimiser des problèmes de taille pratique comprend trois éléments:

- un solveur : le moteur qui fait l'optimisation
- une langue de modélisation : l'interface qui nous permet de communiquer le modèle au solveur
- un environnement de développement : des editeurs et autres logiciels qui nous permet de créer le modèle que le solveur va lire.

Ces ingrédients peuvent aussi se trouver (et s'acheter) séparément.

# Solveurs

C'est la module qui fait le calcul mathématique.

Vous apprenez les méthodes qu'emploient les solveurs dans les autres cours (programmation linéaire et non-linéaire, optimisation de flots des réseaux, programmation en variables entières, etc.).



# Langues de modélisation

La langue est composée de la vocabulaire, la syntaxe, etc., qui définissent un modèle pour le solveur, et qui donnent le solveur les commandes de résoudre le modèle et de sortir des données.

Important : Il s'agit toujours d'une **langue formelle**. Le solveur comprends exactement ce que vous le communiquez et non pas ce que vous *voulez* communiquer!

# Environnements de développement

Un environnement de développement aide l'utilisateur aux activités suivantes

- écrire un modèle correctement dans la langue associée
- utiliser la bonne syntaxe
- trouver et corriger des erreurs
- lancer le solveur
- sortir et analyser les résultats, etc.

- 1 Logiciels pour l'optimisation
- 2 Tableurs et langues algébriques de modélisation (LAMs)
- 3 Planification de production à Ajax

# Tableurs

## Exemples de logiciels

- Tableurs : [Excel](#), [IBM Lotus Symphony](#), [OpenOffice.org Calc](#), [Gnumeric](#), ...
- Solveurs de Tableurs : [Solveur](#), [What's Best \(Lindo\)](#), [Premium Solver Platform \(Frontline Systems\)](#), ...

# Tableurs

- Le solveur est un macro complémentaire qui fait le calcul sur les données et le modèle précisés dans les cellules d'un fichier tableur.
- La langue de modélisation est ici le format du fichier .xls.
- L'environnement de développement est le logiciel qui ouvre et montre les fichiers .xls (Microsoft Office, Lotus Notes, etc.)  
En pratique, ces deux éléments sont inséparables pour les logiciels de tableur.

## Tableurs: exemple

voir “Ajax.xls”

## Langue algébrique de modélisation (LAM)

Un des grands problèmes avec un tableur<sup>1</sup>, c'est que la logique spatiale imposée par le logiciel ne correspond pas du tout à la logique du modèle.

La structure du modèle se voit souvent beaucoup plus clairement dans une formulation algébrique.

Les **langues algébriques de modélisation** sont conceptualisées pour permettre de voir cette structure aussi dans les fichiers qui sont créés pour être passés au solveur.

---

<sup>1</sup>Nous en verrons beaucoup d'autres.

# Langues algébriques de modélisation (LAMS)

Dans ce cours, on va utiliser plutôt la suite des logiciels de FICO Xpress :  
[http://www.fico.com/en/Products/DMTools/Pages/  
FICO-Xpress-Optimization-Suite.aspx](http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx)

La langue algébrique de modélisation appartenant à cette suite s'appelle **Mosel**.

Les trois composants (en gros) de cette suite de logiciel sont

- environnement: Xpress-IVE (**I**ntegrated **V**isual **E**nvironment)
- langue: Xpress-Mosel
- solveur: Xpress-Optimizer (orthographe à l'américain maintenant)



# Langues algébriques de modélisation

Autres LAMs disponible sur le marché :

- GAMS (**G**eneral **A**lgebraic **M**odeling **S**ystem)
  - environnement: environnement GAMS
  - langue: format .gms
  - solveurs: CPLEX, Conopt, MINOS, etc.
- AMPL (<http://www.ampl.com/>)
  - environnement: AMPL Studio, “experimental graphical interfaces” variées
  - langue: formats .mod (modèle) et .dat (données)
  - solveurs: CPLEX, Conopt, GuRoBi, MINOS, etc.
- ILOG (<http://www.ilog.com/products/cplex/>)
  - environnement: OPL Development Studio
  - langue: OPL (Optimization Programming Language)
  - solveur: ILOG CPLEX

## Langues algébriques de modélisation

A remarquer:

- Les solveurs qu'utilisent GAMS et AMPL viennent tous des autres entreprises. On peut même appeler Xpress-Optimiser de l'environnement GAMS ou pour résoudre les fichiers .gms; c'est pareil pour AMPL.  
Ce n'est pas le cas avec Xpress-MOSEL, ni avec ILOG OPL.
- Souvent, on n'est pas obligé à utiliser l'environnement associé à une LAM n'est pas fixée, lui non plus.  
Même dans ces cas, c'est presque toujours mieux à utiliser l'environnement associé.
- Attention! Toutes les versions de ces logiciels ne marche pas sous tous les systèmes d'exploitation.  
Par exemple, et notamment, Xpress-Optimizer et Xpress-Mosel marchent sous Unix, *mais non pas Xpress-IVE.*

## Langues algébriques de modélisation: exemple

voir "Ajax.mos"

## Pourquoi utiliser un tableur?

- Facilité et ubiquité d'utilisation:
  - Il est relativement facile à l'apprendre.
  - On n'est pas obligé a acheter encore une licence de logiciel.
- Facilité de manipulation des données:
  - Des prototypes pour tester des idées se font vite.
  - Les données sont souvent sauvegardées dans les formats tableurs.
  - Même s'il s'agit de manipuler des données sauvegardées par les logiciels de base de données (Access, Oracle, etc.), c'est **(normalement)** relativement facile à les transporter entre ces logiciels et des tableurs.

## Pourquoi utiliser une LAM?

- **Maintenance**: Il est beaucoup plus facile à identifier, trouver, et corriger des erreurs avec une LAM. C'est surtout le cas si le modèle sera modifié.
- **Documentation**: Un modèle se comprend plus facilement par ceux qui ne l'ont pas fait, s'il est écrit dans une LAM au lieu d'un tableur.
- **Séparation de modèle et de données**: Une LAM nous exige à bien distinguer entre le **modèle**, qui décrit la structure du problème, et les **données**, qui représentent les caractéristiques du problème qui peuvent changer d'une résolution à l'autre.
- **Augmentation de taille**
- **Augmentation de dimension**

# Résumé des avantages/inconvénients des tableurs

## Avantages

- Facilité et ubiquité d'utilisation
- Facilité de manipulation des données

## Inconvénients

- Facilité d'introduction des erreurs
- Difficultés de modification
- Difficultés de maintenance
- Grandes difficultés de distinction entre les *données* (qui peuvent tous changer) et le *modèle* (qui incorpère la *structure* du problème)

# Résumé des avantages//inconvénients des LAMs

## Avantages

- Séparation de modèle mathématique et des données
- L'expression du modèle dans la langue se rassemble à l'expression algébrique (meilleure abstraction)
- Facilité relative d'identifier des erreurs, et de distinguer entre erreurs de modélisation, erreurs de syntaxe, et erreurs de données
- Beaucoup plus facile à ajouter des données (agrandir la taille) ou d'ajouter des dimensions (agrandir la complexité) sans faire des erreurs

## Inconvénients

- Des efforts sont nécessaires pour apprendre la syntaxe et la vocabulaire de la langue.
- Les licences sont souvent payantes, et surtout pour des problèmes qu'on s'intéresse à résoudre.

- 1 Logiciels pour l'optimisation
- 2 Tableurs et langues algébriques de modélisation (LAMs)
- 3 Planification de production à Ajax



## Une situation simplifiée

- 3 produits
- deux lignes de fabrication
  - Chacun a sa capacité limitée.
  - Produits 1 et 2 se font sur ligne "A", produit 3 sur ligne "3".
- capacité de main d'oeuvre
- Il faut maximiser le profit.

# RAPPEL: Principes des modélisation

Dans chaque programme mathématique, il faut bien distinguer et définir

- **Indices** : Les *dimensions* du modèle; les ensembles pour lesquelles toutes les autres éléments sont définis.
- **Données** : Les éléments du modèle qu'on **ne peut pas** changer
- **Variables**
- **Fonction objective**
- **Contraintes**

Pour utiliser (correctement) des logiciels, il est **très important** de **ne pas confondre** ces éléments.

# Modèle algébrique

$$\begin{aligned} \max \quad & \sum_{i=1}^3 (R_i - C_i)x_i \\ \text{s.à} \quad & x_1 + x_2 \leq ACAP \\ & x_3 \leq CCAP \\ & \sum_{i=1}^3 H_i x_i \leq L \\ & x_i \geq 0, i = 1, \dots, 3 \end{aligned}$$

## Une modification

Comment modéliser le problème s'il y a aussi des coûts fixes?

*[20000, 15000, 15000]*

A voir: les fichiers `Ajax.xls`, `Ajax.mod`, `Ajax_fixedcosts.mod`

# A souvenir

- Trois ingrédients d'un logiciel pour l'optimisation
  - environnement
  - langue
  - solveur
- Avantages et inconvénients des tableurs et ceux des LAMs
- Rappel : principes de modélisation
- Syntaxe fondamentale des modèles utilisés