Algorithmique pour l'optimisation

↳ Recettes pour maximiser des trucs.

I) Introduction

algorithme : "suite finie et non ambigüe d'instructions et d'opérations permettant de résoudre une classe de pb".

= recette de cuisine

optimisation : "cherche à modéliser, ~~optimiser~~ analyser, résoudre [...] les problèmes qui consistent à trouver des minimaux ou des maximaux"

= maximiser des trucs

Recettes •
- → Recettes génériques
- → Adapter / Variantes
- → Recettes ultra-spécifiques

} cadre du cours

• On va rester au niveau du livre de cuisine
↳ Théorique.

Optimisation → Modélisation : transformer pb réel en un problème mathématique
↳ Réfléchir et discuter la MOD.

→ Résoudre.
↳ Qu'est ce que ça signifie ?
↳ Pourquoi maximiser ?

} cadre du cours.

- Compromis qualité de la solution / temps.
  - ↳ Heuristique ?
  - ↳ Approximation ?

  } Aéronautique / Medecine vs Production
  GPS vs Modèles météo
  - ↳ Un problème NP-difficile est-il difficile ? Ex : sac à dos.
  - ↳ Importance des paramètres Ex : sac à dos.

- Difficulté de l'algo / Explicabilité.
  - ↳ Risques d'erreurs
  - ↳ Maintenance

  Ré → On va se focaliser sur des méthodes génériques & simples à mettre en place

- Certificats
- Robustesse → à un petit changement de données
  → à une erreur dans les données
- Recalculabilité

Exemples de problèmes d'optimisation

- Plus court chemins (GPS)
- Arbres couvrants min (routage).
- Optimisation industrielle
  Etant données des contraintes d'équipement et de ressource, comment maximiser le profit de l'entreprise ?
- Classification et clustering.
  Etant données un ensemble X de données que l'on représente comme des points d'un espace en 2-dimensions.
  On évalue la proximité comme la distance (norme 2). Deux données sont similaires si leur distance est ≤ 1.
  Trouver le plus grand cluster.

- Echange de reins / Parcours Sup.

On a n donneurs et m receveurs. $B_1, \dots, B_m$.
$A_1, \dots, A_n$

On a un ensemble de paires donneurs / receveurs compatibles (A_i, B_j).

But : Trouver un ensemble maximum de paires compatibles
$\Leftrightarrow$ Maximiser le nombre de personnes transplantées.

- Problèmes de transport

On a n entrepots $E_1, \dots, E_n$ qui ont chacun S_i I-phones
m clients $C_1, \dots, C_m$ tq chaque C_j a un besoin d_j de la ressource
Pour tout (i,j), on a un coût $c_{i,j}$ de transport (pour une unité).
Comment minimiser le coût global ?

- Ordonnancement

Etant données n tâches $T_1, \dots, T_n$ tq chaque tâche T_i doit être effectuée entre $[a_i, b_i]$. Aucune tâche ne peut être fractionnée
Combien peut-on effectuer au maximum ? Combien de jours sont nécessaires pour tout faire ?

II. Modélisation

A) Modélisation via des graphes

Graphe $G = (V, E)$
sommets, arêtes

Outil naturel pour représenter des problèmes discrets.

* clustering
  ↳ Clique
* Echange de reins
  ↳ Couplage
* Ordonnancement
  ↳ IS & coloration + Graphes de conflits.

B) Modélisation via PL

PL : optimisation d'une fonction linéaire sous des contraintes elles-même linéaires

$$\begin{cases} \max \ c^t x \\ \text{Soumis à} \ Ax \leq b \\ (x \geq 0) \end{cases}$$

objectif

variables de décision

vecteur de contraintes

matrice de contraintes

la plupart du temps

{ x / x satisfait les contraintes } s'appelle le domaine $\Leftrightarrow$ Région acceptable

En général
n variables
m contraintes

Ou

$$\max \sum_{i=1}^{n} c_i x_i$$

$$\text{Soumis à} \begin{cases} \sum a_{1i} x_i \leq b_1 \\ \sum a_{2i} x_i \leq b_2 \\ \vdots \\ \sum a_{ni} x_i \leq b_n \end{cases}$$

$\forall i \quad x_i \geq 0$

⚠ $a_{i,j}$: scalaires
b_i, c_i : scalaires

⚠ Seules les x_i sont des variables !!!

⚠ Le max peut être un min.

Exemple : Fabrication d'allumettes

Dovetail produit deux types d'allumettes, longues & courtes.
Pour produire des allumettes il faut :

- Du bois
- Des boîtes
- Une machine (temps).

- Pour ~~Une~~ 100 000 boîtes d'allumettes courtes : (resp. longues)
  1 m^3 cube de bois (resp. 3 m^3 cube).
- 900 000 boîtes max par an. peuvent être fabriquées
- 700 000 grandes boîtes et 600 000 petites.
- Stock de 18 m^3 de bois.

Profit : 300 € pour 100.000 grandes boîtes
200 € ——— petites boîtes

Construire un PL :

1) Quelles sont les variables ?

Qu'est-ce qui "bouge" ? Sur quoi on a un pouvoir d'agir → Nombre de boîtes fabriquées.

x_1 : nombre de grandes boîtes produites (/100.000)

x_2 : ——— petites boîte produites (/100 000)

2) Quel est l'objectif ?

Qu'est ce qu'on veut minimiser / maximiser ?

→ max $300\,x_1 + 200\,x_2 \iff \max 3x_1 + 2x_2$

3) Quelles sont les contraintes ?

- Bois : $3x_1 + x_2 \leq 18$
- Machine : $x_1 + x_2 \leq 9$
- Boîtes : $x_1 \leq 7$ (grandes boîtes)
  $x_2 \leq 6$ (petites boîtes)
- Positivité $x_1, x_2 \geq 0$
  $(x_1, x_2 \in \mathbb{N} \times \frac{1}{100\,000})$

Finalement :

$$\max\ 3x_1 + 2x_2$$
$$\text{soumis à}\quad \begin{aligned} 3x_1 + x_2 &\leq 18 \\ x_1 + x_2 &\leq 9 \\ x_1 &\leq 7 \\ x_2 &\leq 6 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$\iff$

$$\max\ 3x_1 + 2x_2$$
$$\text{soumis à}\quad \begin{pmatrix} 3 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 18 \\ 9 \\ 7 \\ 6 \end{pmatrix}$$

Remarques

- Intégralité de la solution

  Résoudre un PL en nombre réel est plus simple & plus rapide.

  Ici on approxime bien la solution

- Vrai PL :

  Milliers voir millions de contraintes et variables

Exemple 2 : Transport

$E_1, \dots, E_n$ entrepôts, m clients $C_1, \dots, C_m$

C_j a un besoin d_j. E_i a un stock S_i.

cout $c_{i,j}$.

Variables : "Combien E_i envoie à C_j"

↳ $x_{i,j}$.

contraintes :

- Stock : $\forall i \quad \sum_{j=1}^{m} x_{i,j} \leq S_i$.
- Demande $\forall j \quad \sum_{i=1}^{n} x_{i,j} \geq d_j$
- Positivité : $\forall i,j \quad x_{i,j} \geq 0$

Objectif :

$$\min \sum_{i,j} c_{i,j} \cdot x_{i,j}$$

Exemple 3 : Ordonnancement

n tâches $T_1, \dots, T_n$

1 intervalle par tâche $[a_i, b_i]$

But : effectuer le nombre maximum de tâches

Variables : "Faire ou ne pas faire ?"

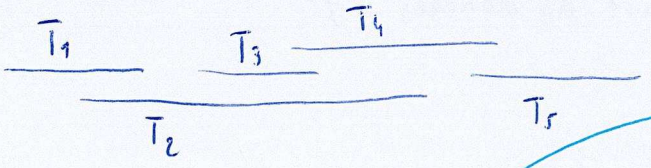
$\forall i$, $x_i = \begin{cases} 0 & \text{si on n'effectue pas la tâche} \\ 1 & \text{si on l'effectue} \end{cases}$ $\quad x_i \in \{0,1\}$

⚠ C'est l'interprétation de la variable. càd la manière dont on voudrait qu'elle se comporte

↳ il faudra vérifier après avoir conçu le PL que le comportement est conforme à ce qu'on souhaite

Contraintes

- Chaque fois que deux intervalles s'intersectent, il ne peuvent pas être effectués tous les deux.



Si T_i et T_j s'intersectent $\Rightarrow x_i + x_j \leq 1$

↳ Une contrainte par arête du graphe de conflit

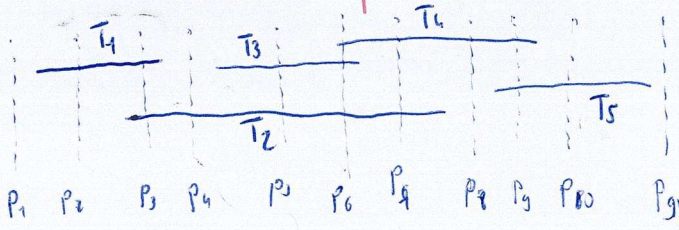
Objectif

$$\max \sum_i x_i$$

Remarque :

Ici on a un PLNE (Programme Linéaire en Nombres Entiers) et on n'a pas le choix ⟶ Grosse différence

⚠ Autre modélisation possible ⚠



1 contrainte par "période de temps".

⟶ $2n+1$ tels intervalles

Pour la période P_6 :

$$x_2 + x_3 + x_4 \leq 1$$

Plus formellement :

$$\forall j, \sum_{i \text{ contient } P_j} x_i \leq 1.$$

Qu'est-ce que ça change ?

→ Nombre de contraintes

Linéaire vs Quadratique

→ "Structure" du PL

Un peu être résolu efficacement, l'autre non.

Bilan : La modélisation est une étape cruciale

Bien réfléchir car elle impacte la suite !

(comme la structure de données ...)