Recherche Opérationnelle

Introduction, généralités, modélisations

J.-F. Scheid
Institut Elie Cartan de Lorraine/Télécom Nancy
Université de Lorraine

2025-26 (version du 28/09/25)

I. Introduction générale

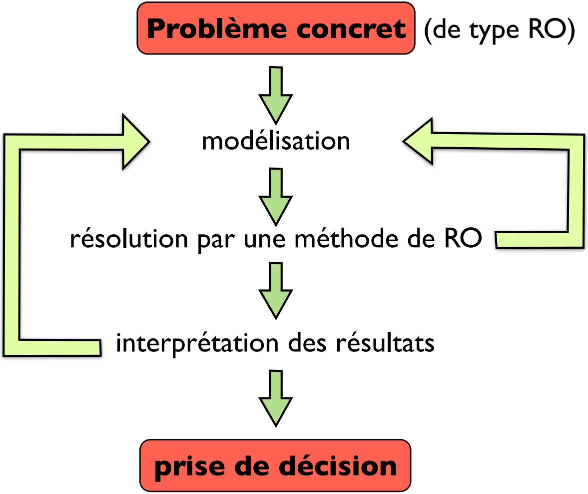
Le terme *recherche opérationnelle* (Operations Research en anglais) date de la seconde guerre mondiale où les anglo-saxons ont cherché à optimiser l'implantation de radars de défense anti-aérienne et à déterminer la taille optimale des convois d'approvisionnement.

Une définition possible de la RO:

Ensemble de méthodes d'analyse scientifique, à la frontière entre maths et informatique, des phénomènes d'organisation qui traite de l'**optimisation** de la conception et du fonctionnement des systèmes (industriels, économiques, énergétiques, numériques ...).

La RO est un outil d'**aide à la décision**.

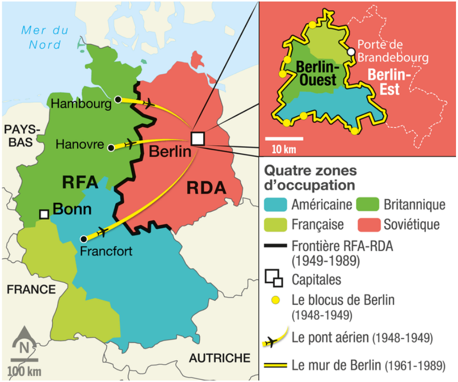
Schéma général en RO:



II. Exemples et domaines d'applications de la RO

1. **Un premier exemple historique : programmation linéaire (PL) pour l'organisation optimale de convois** (G. Dantzig, 1947).

Blocus de Berlin-Ouest en juin 1948 (15 mois).



Convois d'avions US et anglais pour approvisionner les 2 millions de personnes de Berlin-Ouest.

1 avion toutes les 45s au plus fort du trafic avec 278000 vols en 15 mois.

Il y a environ 1500 tonnes de nourriture et 3500 tonnes pour le charbon et l'essence à approvisionner *chaque jour*.

On dispose pour cela de 3 types d'avions avec les caractéristiques suivantes.

avion	tonnage (tonnes)	coût (unité monétaire)	disponibilité (nb d'avions/jour)
type 1	100	1000	10
type 2	200	2000	22
type 3	150	1200	10

Table: caractéristiques et contraintes des avions

Variables : il y a en fait 6 types différents d'avions car on ne mélange pas nourriture et charbon/essence.

☛ On introduit les variables

- X_{if} le nombre d'avions de type i avec une cargaison de nourriture
- X_{ic} le nombre d'avions de type i avec une cargaison de charbon/essence.

Objectif : déterminer le plan de vols (i.e. les variables X_{if} et X_{ic}) afin de minimiser le coût total défini par

$$1000X_{1f} + 1000X_{1c} + 2000X_{2f} + 2000X_{2c} + 1200X_{3f} + 1200X_{3c}$$

Contraintes : elles portent sur la variables et se traduisent par des inégalités.

- *Contraintes d'approvisionnement*
  - il faut fournir au moins 1500 tonnes de nourriture :

$$100X_{1f} + 200X_{2f} + 150X_{3f} \geq 1500$$

  - il faut fournir au moins 3500 tonnes de charbon et d'essence :

$$100X_{1c} + 200X_{2c} + 150X_{3c} \geq 3500$$

- *Contraintes sur les avions* (cf. tableau précédent)
  - on dispose d'un maximum de 10 avions de type 1 :

$$X_{1f} + X_{1c} \leq 10$$

- de même pour les avions de type 2 et 3 :

$$X_{2f} + X_{2c} \leq 22$$
$$X_{3f} + X_{3c} \leq 10$$

- *Pour éviter d'éventuelles solutions non physiques, on impose :*

$$X_{1f} \geq 0,\ X_{1c} \geq 0,\ X_{2f} \geq 0,\ X_{2c} \geq 0,\ X_{3f} \geq 0,\ X_{3c} \geq 0$$

et en fait on doit obtenir *des entiers*.

Ce problème a été résolu à la main en utilisant la méthode du simplexe.

Solution(s)

Sans imposer les contraintes *entières*, on obtient

$$[0.0, 0.0, 7.5, 10.0, 0.0, 10.0]$$

Soit :	0	avion du type 1 pour le transport de nourriture
	0	avion du type 1 pour le transport de charbon/essence
	7.5	avions de type 2 pour le transport de nourriture
	10	avions de type 2 pour le transport de charbon/essence
	0	avion du type 3 pour le transport de nourriture
	10	avions de type 3 pour le transport de charbon/essence

Avec un coût de 47000

Solution (suite)

☛ A partir de cette solution on peut "voir" que le 0.5 avion de type 2 est équivalent à utiliser un avion de type 1, donc on a la solution entière

$$[1, 0, 7, 10, 0, 10]$$

pour le même coût.

☛ On a aussi une autre solution entière :

$$[9, 0, 0, 13, 4, 6]$$

Ceci toujours pour le même coût de 47000.

2. **Problème de production**.

Détermination d'un plan optimal de production.

- Une entreprise fabrique plusieurs produits ou bien dispose de plusieurs procédés de fabrication d'un seul produit.
- *Contraintes* : cela nécessite des ressources particulières (énergies, matières premières, machines, personnel ...) en quantités limitées.
- Chaque produit ou production des procédés a une certaine utilité économique pour l'entreprise (bénéfice, capacité de production...).

Quelles quantités de produits/production des procédés l'entreprise doit-elle prévoir pour maximiser l'utilité économique?

☛ modélisation par programmation linéaire (simplexe).

x_i : quantité du produit/procédé i (variable de décision)

c_i : utilité économique unitaire du produit/procédé i (donnée)

$$\max_{\mathbf{x}} F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} = c_1 x_1 + \cdots c_n x_n \;\leftarrow \text{l'objectif}$$
$$\begin{cases} A\mathbf{x} \leq \mathbf{b} \;\leftarrow \text{les contraintes} \\ \mathbf{x} \geq 0 \end{cases}$$

Un exemple de problème de production.

Une usine fabrique 2 produits P_1 et P_2 nécessitant des ressources d'équipement, d'énergie et de matières premières disponibles en quantité limitée. Les quantités des ressources nécessaire pour produire une unité de P_1 de P_2 (disons 1kg) sont indiquées dans le tableau ci-dessous.

ressources	P_1 (unitaire)	P_2 (unitaire)	disponibilité
équipement	3	9	81
énergie	4	5	55
matière première	2	1	20

Une unité d'énergie P_1 et P_2 rapportent à la vente 6 euros et 4 euros par unité.

Problème: *Déterminer le plan de production (quantités de produits P_1 et P_2 à fabriquer) pour maximiser le bénéfice total venant de la vente des 2 produits.*

Variables : x_1, x_2 sont les quantités de P_1 et P_2 à fabriquer ($x_1, x_2 \in \mathbb{R}$)

- *Objectif*

  Bénéfice total $F(x_1, x_2) = 6x_1 + 4x_2$.

  On cherche

$$\max_{(x_1,x_2)} \left[F(x_1, x_2) = 6x_1 + 4x_2 \right].$$

- *Contraintes*
  - Disponibilité de chacune des ressources :

$$\begin{array}{ll} 3x_1 + 9x_2 \leq 81 & \text{(équipement)} \\ 4x_1 + 5x_2 \leq 55 & \text{(énergie)} \\ 2x_1 + x_2 \leq 20 & \text{(matière première)} \end{array}$$

  - Positivité des variables: $x_1, x_2 \geq 0$.

3. **Problèmes d'affectation**.

Programmation linéaire en nombres entiers (PLNE)

a) un premier exemple de problème d'affectation en variables binaires : *affectation d'élèves à des cours*.

n élèves, m cours proposés avec capacités d'accueil limitées.
Chaque élève doit suivre 3 cours et donne une liste de préférence d'au moins 6 cours (par exemple)

- les variables :

$$x_{ij} = \begin{cases} 1 & \text{si l'élève } i \text{ suit le cours } j \\ 0 & \text{sinon} \end{cases}$$

- maximisation de la satisfaction collective (globale) :

$$\max_{x_{ij}} \sum_{i,j} c_{ij} x_{ij}$$

où $c_{ij} = \begin{cases} 1 & \text{si l'élève } i \text{ a choisi le cours } j \text{ dans sa liste de préférence} \\ 0 & \text{sinon} \end{cases}$

- ... et de nombreuses contraintes !

Contraintes linéaires sur les variables de type égalités/inégalités :

Toutes les contraintes s'écrivent avec des combinaisons linéaires des variables x_{ij}.

Par exemple : (chaque élève suit exactement 3 cours)

$$\sum_{j} x_{ij} = 3, \quad \text{pour tout } i$$

b) un autre exemple de problème d'affectation : le sudoku

2	·	·	·	·	**4**	·	**3**	·
·	**9**	·	·	·	·	·	**7**	·
·	·	·	·	**6**	·	·	**4**	·
3	**8**	·	·	·	**7**	·	·	·
6	·	·	·	·	·	**5**	·	·
·	·	**2**	·	·	·	**8**	·	·
·	·	·	**8**	·	·	**9**	·	·
·	·	·	·	**9**	·	·	·	**2**
·	·	**4**	·	·	**2**	·	**8**	·

2	1	5	7	8	**4**	6	**3**	9
4	**9**	6	2	5	3	1	**7**	8
8	3	7	1	**6**	9	2	**4**	5
3	**8**	9	5	2	**7**	4	1	6
6	7	1	9	4	8	**5**	2	3
5	4	**2**	3	1	6	**8**	9	7
1	2	3	**8**	7	5	**9**	6	4
7	6	8	4	**9**	1	3	5	**2**
9	5	**4**	6	3	**2**	7	**8**	1

Variables d'affectation : pour une case $(i, j) \in [\![1, 9]\!]^2$,

$$x_{ijk} = \begin{cases} 1 & \text{si le chiffre } k \in [\![1,9]\!] \text{ est affecté à la case } (i,j) \\ 0 & \text{sinon} \end{cases}$$

$\rightarrow$ maximisation de la fonction $f \equiv 1$ avec contraintes sur les x_{ijk}.

c) un dernier exemple de problème d'affectation *: mariages stables*

Trouver n mariages stables entre hommes et femmes.

Il y a *instabilité* si un homme et une femme préfèrent tous deux se mettre en couple plutôt que de rester avec son conjoint respectif.

Deux couples (X_1, Y_1) et (X_2, Y_2) sont *instables* si (par exemple) X_1 préfère Y_2 à Y_1 **et** Y_2 préfère X_1 à X_2.

Chaque homme et chaque femme exprime ses préférences parmi l'ensemble des individus de l'autre sexe (liste).

Un exemple (1=premier choix)

	Y_1	Y_2	Y_3
X_1	$(2,2)$	$(3,3)$	$(1,1)$
X_2	$(1,1)$	$(3,1)$	$(2,2)$
X_3	$(3,3)$	$(2,2)$	$(1,3)$

- $3 \times 2 \times 1 = 6$ mariages possibles.
- (X_1, Y_3), (X_2, Y_1), (X_3, Y_2) est stable.
- (X_1, Y_2), (X_2, Y_3), (X_3, Y_1) est instable.

Propriété. Il existe (au moins) une solution stable, donnée par :

- algorithme de Gale-Shapley (1962)
- programmation linéaire

Algorithme de Gale-Shapley.

À chaque itération, chaque homme *célibataire* se propose à la femme qu'il préfère parmi celles à qui il ne s'est jamais proposé. La femme considère alors toutes les propositions qui lui sont faites (y compris l'homme avec qui elle est déjà éventuellement) et elle se met en couple avec l'homme qu'elle préfère en rejetant tous les autres. On itère jusqu'à ce qu'il n'y ait plus d'homme *célibataire*.

(D'après Wikipédia)

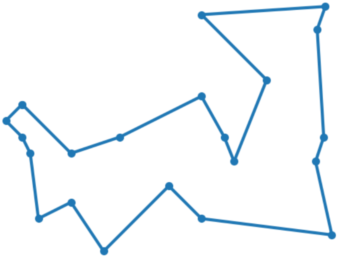
```
Entrée : Deux ensembles finis M (d'hommes) et W (de femmes) de cardinal n ;
         Une famille L de relations de préférences ;
Sortie : Un ensemble S de couples engagés (homme ; femme) ;
fonction mariageStable {
    Initialiser tous les m ∈ M et w ∈ W à célibataire
    tant que ∃ homme célibataire m qui peut se proposer à une femme w {
       w = femme préférée de m parmi celles à qui il ne s'est pas déjà proposé
       si w est célibataire
         (m, w) forment un couple
       sinon un couple (m', w) existe
         si w préfère m à m'
           (m, w) forment un couple
            m' devient célibataire
         sinon
           (m', w) restent en couple
    }
    Retourner l'ensemble S des couples engagés
}
```

Quelques applications du problème des mariages stables:

- Services Internet distribués (page web, vidéos,...) avec utilisateurs/serveurs.
  - Les utilisateurs préfèrent accéder à des serveurs de proximité pour des réponses rapides (liste de préférences des serveurs pour chaque utilisateur).
  - Les serveurs préfèrent servir des utilisateurs dont le coût est faible (liste de préférences des utilisateurs pour chaque serveur).
- Parcours-Sup (h=formation, f=étudiant)

4. **Le problème de voyageur de commerce (TSP)**.

Un voyageur de commerce doit visiter n villes. Les distances entre les villes sont données.



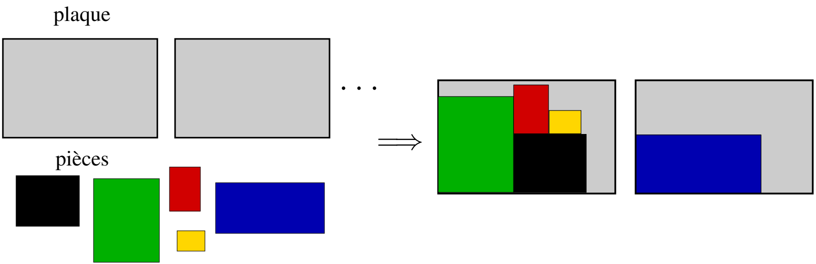
Trouver le plus court trajet passant par les n villes et revenant au point de départ.

Plusieurs modélisations/méthodes possibles :

- programmation linéaire (en nombres entiers/binaires).
- programmation dynamique : on décompose le problème en sous-problèmes et on résout les sous-problèmes des plus petits aux plus grands en stockant les résultats intermédiaires.
- algorithmes génétiques.

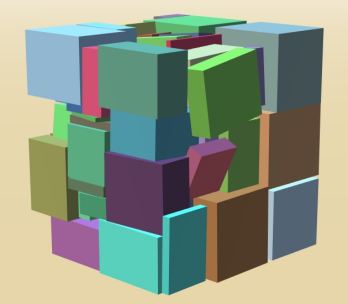
5. **Placement optimal de pièces 1D/2D/3D (Bin Packing)**.

En 2D : on dispose de plaques rectangulaires toutes identiques dans lesquelles on veut placer des pièces rectangulaires sans chevauchement. Les pièces à placer ont des dimensions différentes.



Trouver le placement optimal des pièces pour minimiser le nombre de plaques utilisées.

et aussi en **3D** ...

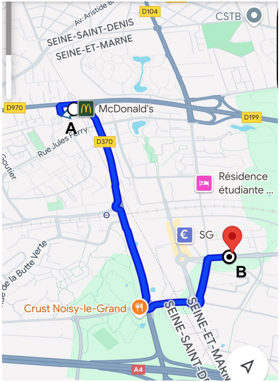


Daniel Hartmeier/ benzedrine.ch

→ méthodes heuristiques (du grec *eurisko*, "je trouve" : art de faire des découvertes, d'inventer) : fournir rapidement une solution réalisable mais pas nécessairement optimale.

- *algorithmes gloutons* (greedy algorithm) : faire un choix *localement* optimal
- *méthodes évolutionnistes* : algos génétiques, colonies de fourmis, ...

6. **Plus court chemin dans un graphe**.

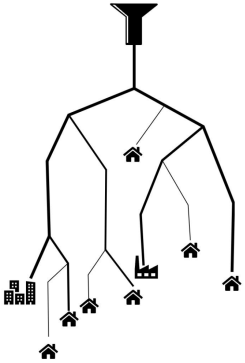


Un graphe est composé de sommets et d'arêtes auxquelles sont associées des distances (entre deux sommets).

Partant d'un sommet A, on veut atteindre un sommet B en parcourant la distance minimale.

Approche par programmation dynamique: algorithme de Dijkstra, algorithme A^* (Google Maps)

7. **Flot maximal dans un graphe**.
Des châteaux d'eau alimentent des villes à travers un réseau de distribution d'eau comportant aussi des stations de pompages. Les canalisations ont des débits limités (on parle de capacités).



On peut associer un graphe dont les sommets représentent les châteaux d'eau, les stations de pompage et les villes; les arêtes représentent les canalisations du réseau.

Ce problème peut se modéliser de plusieurs façons.

- Programmation linéaire
- Problème de flot maximal dans un graphe valué par des capacités sur les arêtes (cf. cours P. Moyal)

☛ On cherche le débit maximal à faire passer dans le réseau, tout en respectant la demande minimale des villes.

III. Quelques éléments de modélisation pour la PL (entière)

1. *Implication logique.* Soit $x, y \in \{0, 1\}$.

$$\begin{array}{ccc} (x = 0 \Rightarrow y = 0) & \Leftrightarrow & x \geq y \\ (x = 1 \Rightarrow y = 1) & \Leftrightarrow & x \leq y \end{array}$$

2. *OU logique.* Soit $x, y \in \{0, 1\}$.

$$\begin{array}{ccc} (x = 0 \text{ ou } y = 0) & \Leftrightarrow & x + y \leq 1 \\ (x = 1 \text{ ou } y = 1) & \Leftrightarrow & x + y \geq 1 \end{array}$$

3. *Contrainte de seuil.* Soit $x \in [0, M]$ avec $K \leq M$.

(si $x < K$ alors $x = 0$) ou bien de façon équivalente (si $x > 0$ alors $x \geq K$)

$$\Leftrightarrow \quad Ky \leq x \leq My, \quad y \in \{0, 1\}$$

4. *Restriction à un ensemble discret de valeurs.*

$$\left\{\begin{array}{l}\bullet\ x \text{ doit prendre ses valeurs réelles (distinctes) dans } \{p_1, \ldots, p_k\} \\ \bullet\ y_i \in \{0,1\} \text{ est la } \textit{fonction indicatrice} \text{ de } \{x = p_i\} \text{ i.e. } y_i = 1 \text{ si et seulement si } x = p_i.\end{array}\right.$$

$$\Leftrightarrow \quad \left\{\begin{array}{l} x = \sum_{i=1}^{k} p_i y_i, \quad y_i \in \{0,1\} \\ \sum_{i=1}^{k} y_i = 1 \end{array}\right.$$

Remarques générales.

- Tous les problèmes précédents sont des *problèmes de combinatoire*. Par ex., pour le problème du voyageur de commerce, il y a $n!$ possibilités.

  Avec $n = 20$, l'énumération des trajets possibles à une vitesse d'un million par seconde, prendrait 77094 années ...

  $\rightsquigarrow$ *nécessité d'avoir des méthodes efficaces, exactes ou approchées (méthode = modélisation + résolution)*

- **Formalisation.**
  Tous les problèmes mentionnés ci-dessus et présentés dans ce cours peuvent se formaliser de la façon suivante.

$$\max\{f(x),\ x \in X\}$$

  - $f : X \to \mathbb{R}$ est la fonction **objectif** qui peut être linéaire, quadratique, nonlinéaire.
  - X est l'ensemble des solutions possibles dites **réalisables** (les contraintes). L'ensemble X est fini mais en général de très grande taille.

Remarque importante

Pour un problème de *minimisation*, on se ramène à une *maximisation* (ou inversement) avec la formule

$$\min f = -\max(-f)$$

Objectif du cours.

- savoir modéliser un problème de RO par programmation linéaire.
- connaitre les principes de la méthode de résolution du simplexe.
- connaître des principes des résolutions de la PL en nombres entiers.
- savoir utiliser un solveur de PL.
- acquérir des notions de programmation dynamique.

Contenu du cours.

Chap. 1. Programmation linéaire
- modélisation; méthode du simplexe (rappels et compléments)
- dualité, simplexe dual.
- solveurs de PL

Chap. 2. Programmation linéaire en nombres entiers
- méthode Branch & Bound
- méthode des coupes

Chap. 3. Programmation dynamique/plus court chemin dans un graphe