# TD 0 : Modélisation

Recherche opérationnelle S3.

2024

*Ce TD d'introduction tente de vous montrer quelques applications pratiques à la recherche opérationnelle, et, par ce biais, de vous faire comprendre ce qu'est la recherche opérationnelle et à quoi elle peut servir. Pour chacun des exercices suivants, le but est de redéfinir mathématiquement le problème posé, quitte à faire des hypothèses ou à simplifier des éléments réels, **de sorte qu'un ordinateur puisse répondre au problème à votre place**. On attend une réponse sous la forme suivante : quelle sont les données en entrée et quelle est la sortie attendue. **Il n'est pas demandé de résoudre le problème**. Il n'existe pas qu'une seule réponse mais certaines se justifient mieux que d'autres. Utilisez tous les outils à votre disposition : graphes, systèmes d'équations et d'inéquations, ensembles, listes, formules booléennes,...*

▶ **Correction**

Toutes les corrections, sont indicatives, vous pouvez en proposer d'autres. Il peut être intéressant de laisser les élèves réfléchir et d'argumenter sur les différentes propositions.

## 1 Modélisations faciles

*Les problèmes suivants sont simples à modéliser, dans le sens où ils laissent peu de place à l'interprétation.*

**Exercice 1 — *Estimer le risque***

Dans un réseau de télécommunication filaire, combien de câbles, au minimum, doivent être détruits avant que deux utilisateurs, quels qu'ils soient, ne puissent plus communiquer ?

▶ **Correction**

Solution la plus évidente :

**Entrée :** un graphe $G = (V, E)$ non orienté connexe

**Sortie :** un sous ensemble $E'$ de $E$ de taille minimum tel que retirer $E'$ de $G$ déconnecte le graphe.

**Exercice 2 — *Jouons un peu***

Résoudre un sudoku.

▶ **Correction**

On peut proposer plusieurs solutions : coloriage de graphe, formule booléenne, programme linéaire. La plus triviale étant bien probablement la définition même du sudoku :

On appelle $\varepsilon$ le symbole *vide*.

**Entrée :** un entier $n$, une matrice $A$ carrée de $\mathcal{M}_{n^2}(\{\varepsilon, 1, 2, \ldots, n^2\})$

**Sortie :** une matrice $B$ carrée de $\mathcal{M}_{n^2}(\{1, 2, \ldots, n^2\})$ telle que

- pour tout $1 \leq i, j \leq n^2$, si $A_{i,j} \neq \varepsilon$ alors $A_{i,j} = B_{i,j}$
- pour tout $1 \leq i, j, k \leq n^2$ tels que $j \neq k$, $B_{i,j} \neq B_{i,k}$
- pour tout $1 \leq i, j, k \leq n^2$ tels que $i \neq k$, $B_{i,j} \neq B_{k,j}$
- pour tout $1 \leq i, j, k, l \leq n^2$ tels que $i \neq k$ ou $j \neq l$, soient $q_i, q_j, q_k$ et $q_l$ les quotients de $i-1, j-1, k-1$ et $l-1$ par $n$, si $q_i = q_k$ et $q_j = q_l$ alors $B_{i,j} \neq B_{k,l}$.

Si une telle matrice n'existe pas, on ne renvoie rien.

**Exercice 3 — *Soyons sport***

Pourriez-vous me donner les 5 meilleurs pays aux JO ?

▶ **Correction**

Il faut définir *meilleur*. On va utiliser les définitions suivantes :

— un pays $p$ est associé au nombre de médailles d'or $p_o$, d'argent $p_a$ et de bronze $p_b$ qu'il a remporté

— soient deux pays $p$ et $q$, on dit que $p \preceq q$ si et seulement si

  — $p_o < q_o$

  — ou $p_o = q_o$ et $p_a < q_a$

  — ou $p_o = q_o$ et $p_a = q_a$ et $p_b \leq q_b$

**Entrée :** un ensemble de pays $\mathcal{P}$.

**Sortie :** un sous ensemble $\mathcal{Q}$ de 5 pays de $\mathcal{P}$ tels que pour tout pays $p \in \mathcal{P} \backslash \mathcal{Q}$ et pour tout pays $q \in \mathcal{Q}$, $p \preceq q$.

**Exercice 4 — *Peut-on aussi modéliser la bière et les chips ?***

Combien de chaînes dois-je regarder/enregistrer pour voir tous les évènements sportifs qui m'intéressent ?

▶ **Correction**

Il s'agit d'un problème de couverture par ensembles. Les éléments à couvrir sont les évènements et les ensembles sont les chaînes.

**Entrée :**

— Un ensemble d'évènements sportifs $\mathcal{X}$

— un ensemble $\mathcal{S} \subset \mathcal{P}(\mathcal{X})$ de parties de $\mathcal{X}$

**Sortie :** Un sous ensemble $C$ de $\mathcal{S}$ tel que $\bigcup_{s \in C} s = \mathcal{X}$.

# 2 Modélisations intermédiaires

*Les problèmes suivants sont un peu plus difficiles. Il faut, avant de modéliser, vous poser des questions, envisager des possibilités, bref, éviter de foncer tête baissée dans une modélisation qui ne prendrait pas en compte tous les éléments importants à considérer.*

**Exercice 5 — *Par où passer ?***

Comment décider où placer les différents panneaux routiers indiquant des villes ?

▶ **Correction**

La difficulté de cet exercice réside dans le fait que, contrairement à ce qu'on pourrait croire, on ne va pas forcément rechercher le plus court chemin. Renvoyer cette solution créerait un désordre monstrueux dans la circulation routière car tout le monde passerait au même endroit. Le but de cet exercice est de faire réfléchir à ça. Il peut être bon d'essayer de créer un petit exemple qui créerait un trafic problématique avec cette solution.

Voici une proposition : on associe à chaque couple de villes/villages un entier correspondant à une estimation du nombre de personnes qui circulent entre ces villes, à chaque route, on estime le nombre de personnes qui peuvent passer sur ces routes en même temps. Les chemins qu'on va créer entre les villes doivent faire en sorte que sur une même route, il n'y ait pas plus de personne que le nombre maximal.

On crée pour chaque ville $w$ un arbre couvrant $T_w$ du réseaux (l'ensemble des chemins des autres villes vers $w$), les panneaux routiers suivront la direction de cet arbre pour rejoindre cette ville. On peut ensuite estimer combien de personnes vont rouler sur chaque tronçon de route. Soit un arbre couvrant $T_w$, deux nœuds $(v, w)$ et une arête $e$, on définit la variable $x(T_w, v, w, e)$ binaire comme valant 1 si et seulement si $e$ appartient au chemin de $v$ à $w$ dans $T_w$ ; autrement dit tous les gens qui veulent se rendre de $v$ à $w$ passeront par $e$.

**Entrée :**

— un graphe non orienté $G = (V, E)$ connexe

— pour chaque couple de nœuds $v$ et $w$, on associe un entier $\omega(v, w) \in \mathbb{N}$

— pour chaque arête $e$, on associe un entier $c(e) \in \mathbb{N}$

**Sortie :** pour chaque nœud $w$, construire un arbre couvrant $T_w$ de $G$ de sorte que, pour toute arête $e$, $\sum_{v,w \in V^2} \omega(v, w) \cdot x(T_w, v, w, e) \leq c(e)$.

**Exercice 6 — *En retard ou non ?***

Déterminer si un projet, constitué d'un ensemble de tâches, peut être fini dans les temps avec les ressources données.

▶ **Correction**

Le plus simple est probablement de définir un programme linéaire ici. De nombreuses contraintes sont possibles, en voici une proposition.

**Entrée :**

— Un ensemble de tâches $\mathcal{T}$ dont une tâche $START$ de début et une tâche $END$ de fin

— un graphe de précédence orienté sans circuit $\mathcal{G} = (\mathcal{T}, A)$ (chaque arc $(t, t')$ indique que $t$ doit être terminé avant de commencer $t'$), la tâche $START$ est une source et la tâche $END$ est un puits

— pour chaque tâche $t$, un entier $\omega(t)$ (le nombre de ressource utilisées)

— pour chaque tâche $t$, un entier $d(t)$ (la durée de la tâche) ; $d(START) = 0$ et $d(END) = 0$

— pour chaque tâche $t$, une fenêtre de temps $[b(t), e(t)]$ pendant laquelle la tâche accomplie

— un nombre maximal de ressources $c$

**Sortie :** Un entier $x_t$ associé à chaque tâche tel que :

— $x_{START} = 0$

— $x_{END}$ est minimum

— pour toute tâche $t$, $b(t) \leq x_t$ et $x_t + d(t) \leq e(t)$

— pour toute arête $(t, t')$ de $G$, $x_t + d(t) \leq x_{t'}$

— pour tout entier $n$, $\sum_{t : n \in [x_t, x_t + d(t)]} \omega(t) \leq c$

**Exercice 7 — *Actualité***

Où placer les forces de l'ordre pour assurer la sécurité d'une foule ?

▶ **Correction**

Il s'agit, en gros, d'un problème de $k$-centre généralisé. Il faut bien différencier l'ensemble des zones qu'on peut surveiller et l'ensemble des zones où on peut agir (on peut surveiller sans pouvoir agir et inversement).

**Entrée :**

— Un ensemble de positions $V$

— un ensemble de positions $W \subset V$ où la foule se trouvera

— un graphe de surveillance $G = (V, E)$ non orienté (chaque arête $(u, v)$ signifie qu'on voit $v$ depuis $u$ et inversement)

— un graphe d'intervention $H = (V, F)$ orienté (chaque arc $(u, v)$ signifie qu'on peut intervenir en $v$ depuis $u$)

— un entier $k$ (le nombre d'équipes que l'on peut disposer)

**Sortie :** Un ensemble $U$ de $k$ nœuds de $V$ tels que pour tout nœud $w$ de $W$, $w$ a un voisin dans $G$ qui soit dans $U$ et un prédécesseur dans $H$ qui soit dans $U$.

**Exercice 8 — *Tortue ninja !***

Comment couper une pizza $n$ fois pour maximiser le nombre de parts résultant ?

▶ **Correction**

Plusieurs questions se posent ici

— doit-on faire des coupes droites ?

— doit-on passer par le centre de la pizza ?

— la pizza a-t-elle un centre ? (elle peut être carrée)

— doit-on faire des parts égales ?

— la pizza est-elle plate ?

Remarque : si on fait des coupes droites sans passer par le centre et sans faire de parts égales, la réponse est $\binom{n}{0} + \binom{n}{1} + \binom{n}{2}$ si $n \geq 2$ (On the number of regions in an m-dimensional space cut by n hyperplanes, Chungwu Ho and Seth Zimmerman ou `http://mathworld.wolfram.com/CircleDivisionbyLines.html`).

**Entrée :** deux entiers $n$ et $m$

**Sortie :** $n$ hyperplans de $\mathbb{R}^m$ tels que le nombre de régions formées par ces hyperplans est maximal

**Exercice 9 — *Séquençage génétique***

Connaissant un ensemble de fragments du même ADN, reconstituer l'ADN entier.

▶ **Correction**

Il faut ici un minimum de connaissance sur le séquençage. Une méthode utilisée consiste à découper l'ADN en plein de petits morceaux car c'est la seule manière de lire dans l'ordre les nucléotides (on ne sait pas le faire sur l'ADN entier). Du coup, on se retrouve avec plein de bout d'ADN qu'il faut réassembler.

L'avantage c'est qu'on a pas massacrer qu'un seul brin d'ADN, on a traité plein de cellules en même temps. Chaque brin n'a pas été découpé de la même manière (la probabilité que ça arrive est ridicule). On va donc se servir de ça pour reconstituer l'ADN. On regarde deux morceaux et on vérifie s'ils se chevauchent (si la fin d'un des deux morceaux est le début de l'autre).

Si on arrive à produire une suite de chevauchements utilisant tous les morceaux, le résultat est très probablement un brin d'ADN complet. On obtient un problème de chemin hamiltonien.

**Entrée :** un graphe orienté $G = (V, A)$, chaque nœud est associé à un morceau d'ADN et deux morceaux sont reliés par un arc si la fin du premier est le début du second

**Sortie :** un chemin hamiltonien de $G$.

## 3 Modélisations difficiles

*Dans les problèmes suivants, on vous donne peu d'information, ce qui ne rend pas la modélisation très simple. En général, la solution est d'aller voir la personne qui vous a posé le problème et de lui poser des questions. On espère que votre chargé de TD va savoir y répondre.*

▶ **Correction**

Je ne propose pas de correction car je pense qu'il n'y a pas vraiment de réponse à ces exercices. C'est beaucoup trop flou. Il peut être intéressant de discuter avec les élèves pour qu'ils essaient de répondre à cette question : quelles informations me manque-t-il ?

**Exercice 10 — *Un peu de chimie***

Étant donnée une molécule, déterminer sa forme 3D.

**Exercice 11 — *Transports en commun***

Comment adapter les transports en commun aux fréquentations touristiques d'une ville au cours de l'année ?

**Exercice 12 — *Jusque dans l'espace !***

Le lanceur qui décollera demain va-t-il exploser ?