

TD modélisation RO

Ce TD d'introduction est là pour montrer quelques applications de la recherche opérationnelle, et par ce biais faire comprendre ce qu'est la recherche opérationnelle et à quoi elle peut servir.

Pour chacun des exercices suivants, le but est de redéfinir le problème posé mathématiquement, quitte à faire des hypothèses ou à simplifier des éléments réels. C'est ce qu'on appelle la modélisation. La modélisation a pour but de faire qu'à terme, un ordinateur puisse résoudre le problème posé à votre place.

Des outils de modélisation existent, citons: les graphes et la programmation mathématique. On pourra utiliser dans les exercices l'un ou les deux de ces outils. Il est à noter qu'il n'y a pas un modèle unique. Le modèle dépend des hypothèses et des simplifications adoptées. Il est important de bien distinguer les données et la sortie (le résultat) du modèle. Il n'est pas demandé de résoudre le problème.

Exercice 1. Estimer le risque (Réseaux télécom)

Dans un réseau de télécommunication filaire, combien de câbles au minimum doivent être détruits pour que 2 utilisateurs quelconques ne puissent plus communiquer ?

1-Modéliser le problème à l'aide d'un graphe. Quel problème doit-on résoudre dans ce graphe.

Exercice 4. Peut-on modéliser la bière et les chips (chaînes de sport)

Combien de chaînes de télévision au minimum dois-je regarder ou enregistrer pour voir tous les événements sportifs qui m'intéressent ? On suppose que les événements se déroulent à des instants différents ne se recouvrant pas. Chaque chaîne diffuse un certain nombre d'événements mais pas tous.

1-Modéliser à l'aide d'un ensemble et de parties de cet ensemble.

2-Modéliser par un programme linéaire en variables 0-1 avec une contrainte par événement sportif. Préciser bien les données d'une part et les variables d'autre part du problème.

Exercice 6. En retard ou non ? (ordonnancement de tâches)

Un projet est constitué de plusieurs tâches avec des durées données, des fenêtres de temps données, des contraintes de précédences entre tâches et des besoins en ressource donnés. On suppose, pour simplifier, qu'il n'y a un seul type de ressource. Peut-on réaliser le projet et si oui, quel est la durée minimum du projet ?

On suppose que les durées, le début et la fin de chaque fenêtre de temps sont des entiers.

1-Dans un premier temps, on suppose qu'il n'y a pas de contraintes de ressources. Modéliser avec un graphe. Modéliser par un programme linéaire.

2- Ecrire la contrainte de ressource

3-On tient compte maintenant des ressources. On discrétise le temps et à chaque instant t , on doit vérifier qu'il y a assez de ressources. Ecrire un modèle sous forme d'un programme linéaire en 0-1.

Exercice 7. Actualité (Placement de points de surveillance)

Où placer des points de surveillance afin que l'on puisse surveiller des manifestants placés en certaines places d'une ville ?

1-Modéliser à l'aide d'un graphe. Quelles sont les données ? Que cherche-t-on dans ce graphe ?

2-Formuler le problème par un programme linéaire en 0-1.

Exercice 9. Séquençage génétique

Les constituants d'un ADN sont les nucléotides A, C, G, T. Un ADN est constitué de deux brins identiques. Par exemple, un ADN est formé de 2 brins de 8 nucléotides et chaque brin s'écrit : ACGTACGT.

On ne sait pas donner le séquençement de nucléotides sur un ADN en entier. Par contre, on sait le faire sur des petits morceaux d'ADN. On casse donc un ADN en morceaux. Et sur chaque morceau, on détermine le séquençement des nucléotides. Chaque brin n'a pas été découpé aux mêmes endroits (la probabilité que cela arrive est quasi-nulle). On va justement se servir de cela pour reconstituer le séquençement de l'ADN en son entier. On regarde alors chaque paire de morceaux d'ADN et on vérifie s'ils se chevauchent (ou pas) c'est-à-dire si la fin de l'un des 2 morceaux est le début de l'autre. Si on arrive à produire une suite de chevauchements utilisant tous les morceaux, on a trouvé, probablement, le séquençement des nucléotides de l'ADN. Les fragments sont dits « libres » lorsqu'on a éliminé les fragments inclus dans un autre fragment. Par exemple : ACGT est inclus dans ACGTT.

Connaissant des fragments d'ADN, reconstituer l'ADN en entier.

1-Modéliser à l'aide d'un graphe. Peut-on mettre des poids sur les arcs ? Quel problème doit-on résoudre dans ce graphe ? Traiter l'exemple suivant : TACGA, ACCC, CTAAAG, GACA.

Correction

Exercice 1 . Réseaux télécom

On modélise par un graphe $G=(V,E)$ non orienté. V =les utilisateurs , E =liaisons filaires (entre utilisateurs)

Définition d'un cocycle : soit A un sous-ensemble propre de V (i.e. $\emptyset \subset A \subset V$), le cocycle induit par A est $\omega(A)=\{e=[i,j] \in E : (i \in A \text{ et } j \notin A) \text{ ou } (j \in A \text{ et } i \notin A)\}$ i.e. les arêtes de G avec exactement une extrémité dans A .

Trouver $\emptyset \subset A \subset V$ tel que le cocycle $\omega(A)$ soit de cardinal minimum.

Il existe un algorithme polynomial basé sur la recherche de flot maximum.

Exercice 4. Chaînes de sport

$S=\{e_i \mid i=1,\dots,n\}$ ensemble des évènements sportifs

$P=\{p_j \mid j=1,\dots,m\}$ un ensemble de m parties de S . P modélise les m chaînes diffusant les évènements sportifs.

Trouver un ensemble de parties de P de cardinal minimum qui « couvre » l'ensemble des évènements sportifs : $j_i \in \{1,\dots,m\} \mid i=1,\dots,n$ tel que $S \subset \bigcup_{i=1}^n p_{j_i}$ et p est minimum.

Programme linéaire en 0-1.

Variables : $x_j \mid j=1 \text{ à } m$ t.q. $x_j=1$ si on prend la partie p_j et 0 sinon.

$$\min \sum_{j=1}^m x_j$$

$$\text{Sous les contraintes} \quad \begin{cases} \sum_{j=1}^m a_{ij} x_j \geq 1 & i = 1, \dots, n \\ x_j \in \{0,1\} & j = 1, \dots, m \end{cases}$$

Avec $a_{ij}=1$ si l'évènement e_i est retransmis par la chaîne j i.e. $e_i \in p_j$, $a_{ij}=0$ sinon.

Exercice 6. Ordonnancement de tâches

Les données sont :

n nombre de tâches (on numérote les tâches de 1 à n), H horizon temporel sur lequel se déroule le planning (on suppose que le planning démarre à l'instant 0), R quantité totale de ressource

Pour chaque tâche i de 1 à n : d_i durée tâche i , $[b_i, e_i]$ fenêtre de temps de la tâche i , ω_i quantité de ressource requise par tâche i

1° Modèle linéaire sans tenir compte de la ressource R .

Les variables sont : x_i = date de début de la tâche $i=1$ à n

$f \geq 0$ est date de fin de l'ensemble du projet

Les contraintes sont :

- Fenêtre de temps

Pour tout $i=1$ à n : $x_i \in [b_i, e_i - d_i]$

- Contrainte de précédence

Pour tout $i=1$ à n et tout i' suivante de i , $x_i + d_i \leq x_{i'}$

Il faut définir l'objectif. On veut minimiser la durée totale. Il suffit de rajouter une tâche fictive suivante de toutes les tâches et de minimiser sa date d'exécution.

Pour tout $i=1$ à n : $x_i + d_i \leq f$

Et on minimise f .

2° Pour tenir compte des ressources, il faut rajouter des contraintes telles que à chaque instant t la somme des ressources requises n'excède pas R .

Pour tout $t \in [0, H]$: $\sum_{i: t, q, t \in [x_i, x_i + d_i[} w_i \leq R$

La somme est prise sur les tâches i en cours d'exécution à l'instant t . L'intervalle dans la somme est ouvert à droite car à l'instant $x_i + d_i$ la tâche i est terminée et ne consomme donc plus la ressource. L'étendue de cette somme dépend des variables x_i du problème et cette formulation est par conséquent non linéaire.

3° Discrétisation du temps et modèle linéaire en 0-1

On discrétise le temps t de 0 à H .

Les variables sont :

$x_{it}=1$ si la tâche i démarre à l'instant t , 0 sinon

$f \geq 0$ est date de la fin du projet

Les contraintes sont les suivantes.

- Une tâche démarre à un seul instant

pour toutes les tâches $i=1, \dots, n$: $\sum_{t=0}^H x_{it} = 1$

- Une tâche doit démarrer et terminer dans sa fenêtre de temps

pour toutes les tâches $i=1,\dots,n$: $x_{it}=0$ pour tout $t \notin [b_i, e_i - d_i]$

ou plus simplement $b_i \leq \sum_{t=0}^H t x_{it} \leq e_i - d_i$

- Contraintes de précédence : si la tâche i démarre à t alors i' (tâche suivante de i) doit démarrer à $t+d_i$ ou $t+d_i+1$ etc...

Pour tout $i=1$ à n et tout i' suivante de i , pour tout $t=0$ à H : $x_{it} \leq \sum_{t'=t+d_i}^H x_{i't'}$

Ou encore de façon plus compacte :

Pour tout $i=1$ à n et tout i' suivante de i : $\sum_{t=0}^H t * x_{it} + d_i \leq \sum_{t=0}^H t * x_{i't}$

- Définition de la date de la tâche de fin qui est après toutes les tâches

Pour tout $i=1$ à n : $\sum_{t=0}^H t * x_{it} + d_i \leq f$

- Contraintes de ressource : à chaque instant t la somme des ressources des tâches en cours n'excède pas la ressource disponible

Pour tout $t=0$ à H : $\sum_{i=1}^n \sum_{\Delta=0}^{\min(t, d_i-1)} w_i x_{i, t-\Delta} \leq R$

L'objectif est : minimiser f .

Voici un exemple avec $n=7$ tâches, numérotées de 1 à 7, et $R=5$ employés.

tâche	durée	tâches précédentes	nombre d'employés
A=1	6	-	3
B=2	3	-	2
C=3	6	-	1
D=4	2	B	1
E=5	4	B	3
F=6	3	D , A	3
G=7	1	F , E , C	2

Le modèle a été codé en *MathProg* et résolu avec le « solveur » libre *glpk*. Arbitrairement, on a pris $H=20$, et pour chaque tâche, la fenêtre de temps $[0,H]$. Voici ci-dessous le résultat obtenu, pour chaque tâche *début* indique la date à laquelle la tâche doit démarrer :

```

INTEGER OPTIMAL SOLUTION FOUND
Time used: 0.1 secs
Memory used: 0.5 Mb (571175 bytes)
Display statement at line 32
f.val = 14
tache 1 debut 0 duree 6
tache 2 debut 0 duree 3
tache 4 debut 4 duree 2
tache 3 debut 6 duree 6
tache 6 debut 6 duree 3
tache 5 debut 9 duree 4
tache 7 debut 13 duree 1
Model has been successfully processed
C:\Users\afaye\Documents\Enseignement\MRO\TD>

```

Exercice 7. Actualité (Placement de points de surveillance)

1° On modélise par un graphe non orienté. Les sommets V = les places de la ville, les arêtes E = les rues. L'ensemble des manifestations auront lieu dans $W \subset V$. Il faut trouver un ensemble de sommets S tel que de S les forces de l'ordre puissent intervenir i.e. S t.q. pour $w \in W$ il existe une arête $[s, w]$ avec $s \in S$ ou bien $w \in S$ (une place où se trouve la police est surveillée). Dans le cas où $W = V$ il s'agit de trouver un ensemble de sommets absorbants.

2° Programme linéaire en 0-1

Variables : $x_i = 1$ si le sommet i est sélectionné comme point de surveillance, 0 sinon

Contraintes :

Pour tout $w \in W$: $x_w + \sum_{i \in \Gamma(w)} x_i \geq 1$ où $\Gamma(w)$ désigne les voisins de w .

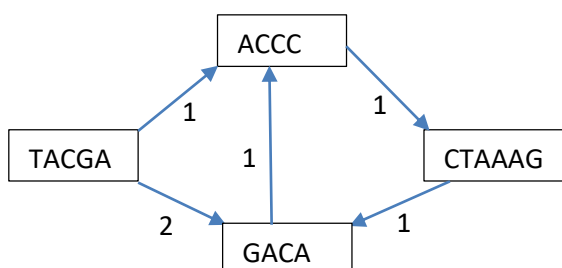
On définit l'objectif. On veut minimiser les points de surveillance i.e. le cardinal de S ce qui revient à :

$$\min \sum_{i \in V} x_i$$

Exercice 9. Séquençage génétique

On modélise à l'aide d'un graphe orienté : les morceaux (fragments) sont les sommets du graphe, il y a un arc entre 2 morceaux (fragments) si et seulement si la fin du premier coïncide avec le début du second. On met un poids sur les arcs correspondant au nombre de nucléotides communes.

Traitons l'exemple suivant : TACGA, ACCC, CTAAAG, GACA.



Il faut un enchainement de tous les morceaux (fragments) c'est-à-dire, dans le graphe, un chemin hamiltonien (i.e. qui passe par tous les sommets).

Ici, il y en a deux :

TACGA→ACCC→CTAAAG→GACA de poids total 3

TACGA→GACA→ACCC→CTAAAG de poids total 4

On préférera le deuxième dont le poids est plus fort.

Le brin d'ADN est TAC**GACACC**CTAAAG. On a mis en gras les nucléotides se « recouvrant ».

Les 2 brins d'ADN ont donc été cassés selon le schéma suivant :

TACGA | C | ACCC | TAAAG

TAC | GACA | CC | CTAAAG

Les fragments en vert sont ceux qui restent après avoir rendu le système de fragments libre i.e. après avoir éliminé les fragments inclus dans un autre.