Rapport RO : TP1

HUANG Julien
AFKER Samy

Département Sciences du Numérique - Deuxième année
2023-2024

Table des matières

Table des figures

1 Introduction

Le but de ce TP est, dans un premier temps, de modéliser des problèmes d'optimisation en recherche opérationnelle et ensuite de les résoudre en se basant sur le solver **GLPK**.

2 Assemblage

Variables de Décision :

$$n_C : \text{le nombre de vélos cargos produits par semaine.}$$
$$n_S : \text{le nombre de vélos standards produits par semaine.}$$

Objectif :

$$\text{Maximiser la marge } M = 700n_C + 300n_S$$

Contraintes :

$$\begin{aligned}
&\text{Heures de travail : } 0.06n_C + 0.05n_S \leq 60\\
&\text{Espace de stockage : } 2.5n_C + n_S \leq 1500\\
&\text{Limite de vélos cargos : } n_C \leq 700\\
&x_C \geq 0, \quad n_S \geq 0
\end{aligned}$$

Il est préférable de modéliser ce problème avec un PLNE. En effet on ne peut pas vendre des demi-vélos. Au vu de la simplicité du problème et de la non nécessité de rentrer des données, nous avons choisi d'utiliser le format .lp . Enfin, nous obtenons comme solution optimale : **232** cargos et **920** simples donc **438400** euros en terme de bénéfice maximal.

3 Affectation avec prise en compte des préférences

Pour l'ensemble des problèmes qui suivent, nous allons utiliser le format gmpl. Ce dernier prenant des fichier de données en entrée, il est plus simple de réaliser des tests.

Notations :

$$N : \text{nombre de personnes (membres de l'équipe)}$$
$$T : \text{nombre de tâches}$$

Variables de Décision :

$$c_{ij} : \text{Score de préférence de la personne } i \text{ pour la tâche } j$$
$$x_{ij} : \text{Variable binaire indiquant si la personne } i \text{ est affectée à la tâche } j$$

Objectif :

$$\text{Maximiser la satisfaction } S = \sum_{i=1}^{N} \sum_{j=1}^{T} c_{ij} \cdot x_{ij}$$

Contraintes :

Chaque personne doit être affectée à exactement une tâche :

$$\sum_{j=1}^{T} x_{ij} = 1 \quad \forall i$$

Chaque tâche doit être effectuée par exactement une personne :

$$\sum_{i=1}^{N} x_{ij} = 1 \quad \forall j$$

Variables binaires :

$$x_{ij} \in \{0, 1\} \quad \forall i, j$$

4 Applications en optimisation pour l'e-commerce

4.1 Cas particulier 1.1

Notations :

i : Indice des magasins.
j : Indice des fluides.
k : Indice des commandes.

Paramètres du problème :

d_{jk} : Demande de fluide j de la commande k.
c_{ij} : Coût unitaire de la quantité de fluide j provenant du magasin i.
s_{ij} : Stock disponible dans le magasin i pour le fluide j.

Variables de Décision :

x_{ijk} : Quantité de fluide j provenant du magasin i pour la commande k.

Objectif :

$$\text{Minimiser le prix : } P = \sum_{i} \sum_{j} \sum_{k} c_{ij} \cdot x_{ijk}$$

Contraintes :

Chaque commande doit être satisfaite :

$$\sum_{i} \sum_{j} x_{ijk} = d_{jk} \quad \forall k$$

Les stocks disponibles ne doivent pas être dépassés :

$$\sum_{k} x_{ijk} \leq s_{ij} \quad \forall i, j$$

Les quantités de fluides livrées doivent être positives :

$$x_{ijk} \geq 0 \quad \forall i, j, k$$

Les solutions du problème dans les cas PL et PLNE sont dans les fichiers solutions Ecommerce1_1.sol.txt et Ecommerce1_1PLNE.sol.txt

4.2 Cas particulier 1.2

Ce problème est similaire au cas précedent avec les contraintes des coûts de transports en plus. Il s'agit ici de contraintes de type bigM. Notre problème devient :

Notations :

i : Indice des magasins.
j : Indice des fluides.
k : Indice des commandes.

Paramètres du problème :

d_{jk} : Demande de fluide j de la commande k.
c_{ij} : Coût unitaire de la quantité de fluide j provenant du magasin i.
s_{ij} : Stock disponible dans le magasin i pour le fluide j.
F_{ki} : Coût fixe correspondant au magasin i pour la commande k.
V_{ki} : Coût variable correspondant au magasin i pour la commande k.
M : La constante de la contrainte bigM. Il suffit de prendre une valeur suffisemment grande par rapport aux valeurs possibles pour la variable objectif (100 par exemple).

Variables de Décision :

x_{ijk} : Quantité de fluide j provenant du magasin i pour la commande k.
y_{ki} : Indique s'il existe une commande provenant du magasin i pour la commande k.

Objectif :

$$\text{Minimiser } Z = \sum_i \sum_j \sum_k (V_{ki} + c_{ij}) \cdot x_{ijk} + \sum_i \sum_k F_{ki} \cdot y_{ki}$$

Contraintes :

Chaque commande doit être satisfaite :
$$\sum_i \sum_j x_{ijk} = d_{jk} \quad \forall k$$
Les stocks disponibles ne doivent pas être dépassés :
$$\sum_k x_{ijk} \leq s_{ij} \quad \forall i, j$$
Les quantités de fluides livrées doivent être positives :
$$x_{ijk} \geq 0 \quad \forall i, j, k$$
Contraintes de liens entre x_{ijk} et y_{ki} avec Big M :
$$y_{ki} \leq \sum_j x_{ijk} \quad \forall i, k$$
$$\sum_j x_{ijk} \leq M \cdot y_{ki} \quad \forall i, k$$

Les solutions du problème dans les cas PL et PLNE sont dans les fichiers solutions Ecommerce1_2.sol.txt et Ecommerce1_2PLNE.sol.txt

4.3 Cas particulier 2

Il s'agit du problème du voyageur vu en cours. Pour ce problème, nous nous sommes basés sur la formulation de Miller–Tucker–Zemlin (page wikipédia : Travelling salesman problem).

Variables de décision :

x_{ij} (binaire) : Indique si le trajet entre le point i et le point j est effectué.
u_i (entière) : L'ordre de livraison du client i.

Paramètres du problème :

d_{ij} : La distance entre le point i et le point j.
$n_{clients}$: Le nombre de clients.

Objectif :

$$\text{Minimiser la distance totale parcourue : } D = \sum_i \sum_j d_{ij} \cdot x_{ij}$$

Contraintes :

À partir du point i on ne peut aller que vers un seul point :
$\sum_j x_{ij} = 1 \quad \forall i$
Le point i ne peut avoir qu'un seul prédecesseur :
$\sum_i x_{ij} = 1 \quad \forall j$
Il est impossible de faire un trajet d'un point i vers ce même point :
$\sum_i x_{ii} = 0 \quad \forall i$
L'ordre de livraison d'un point i est entre 1 et $n_{clients}$:
$u_i \geq 1 \quad \forall i$
$n_{clients} \geq u_i \quad \forall i$
Respect de l'ordre de passage :
$u_i + (n_{clients} - 1) \geq u_j + n_{clients} \cdot x_{ij} \quad \forall i, j$

Enfin, nous obtenons une distance minimale de 22 en suivant le chemin suivant :
$ALPHA \rightarrow C1 \rightarrow C4 \rightarrow C5 \rightarrow C3 \rightarrow C2 \rightarrow ALPHA$.

5 Conclusion

Ce TP nous a permis de mettre en pratique nos différentes connaissances théoriques acquises en cours de Recherche Opérationnelle. Nous avons pu résoudre des problèmes concrets en utilisant le solver **GLPK**.