Rapport TP1 - Recherche opérationnelle

Kawtar LYAMOUDI - Nihal BELMAKHFI

November 25, 2022

1 Première partie: Assemblage

Dans cette partie, on a choisi le format (lp), parceque l'énoncé est sur une entreprise bien particulière. Alors les données ne doivent pas etre modifiées. Pour maximiser le bénéfice de l'entreprise, il fallait tout d'abord représenter le nombre des voitures (de luxe et standard), on a choisi les variables NL et NS. Pour les deux cas PL et PLNE, on a trouvé les meme solutions. L'entreprise bénéfécie plus de la fabrication des voitures de luxe. La matrice de contraintes de la solution est bien creuse.

```
Objective:  margetotale = 10285714.29 (MAXimum)
```

Figure 1: fonction objective du problème d'assemblage PL

```
Objective:  margetotale = 10284000 (MAXimum)
```

Figure 2: fonction objective du problème d'assemblage PLNE

2 Deuxième partie: Gestion de personnel

Dans cette partie, on n'a pas le nombre de personnes et de travaux, c'est pour cela qu'on a opté pour les format (mod) et (dat). Dans les données, on a mis: les personnes, les travaux et un tableaux de couts. La variable (travail) montre le travail affecté à chaque personne. Cette valeur est de type binary. Elle est égale à 1 si le travail en indice a été attribué à la personne en indice. Les contraintes sont: chaque personne n'effectue qu'un seul travail et inversement un travail n'est affecté qu'à une seule personne.

```
Objective:  CoutFormation = 16 (MINimum)
```

Figure 3: fonction objective du problème de gestion de personnel

Pour obtenir cette fonction objective, on n'a choisi comme données:

```
data;

set PERSONNES :=
firstPerson
secondPerson;

set TRAVAUX :=
firstJob
secondJob;

param coutformation: firstJob secondJob :=
firstPerson 5 12
secondPerson 20 11;

end;

```

Figure 4: données utilisées pour le problème de gestion de personnel

3 Troisième partie: E-commerce

3.1 Introduction

Parmi les problématiques d'optimisation émergeant en e-commerce, se trouvent l'affectation de commandes de clients aux magasins, compte tenu des coûts financiers et/ou environnementaux associés à la livraison des colis, à la préparation des commandes et à la gestion des différents stocks. Nous nous intéresserons particulièrement au problème d'affectation de commandes et tournées de véhicules pour différents magasins d'une même enseigne ou franchise à coût/impact minimal.

Dans toute cette partie, on va utiliser une format (mod) et (dat) pour séparer le modèle des valeurs et pouvoir réutiliser la même solution par la suite pour des valeurs numériques différentes.

	F1	F2
D1	2	0
D2	1	3

(a) Fluides demandés par commande

	F1	F2
M1	2.5	1
M2	1	2
M3	2	1

(b) Stocks de fluides par magasin

	F1	F2
M1	1	1
M2	2	3
M3	3	2

(c) Coûts unitaires par magasin d'origine

Figure 5: Données Cas 1

3.2 Cas particulier 1

Dans cette partie, on a choisi un format (mod) et (dat). Car c'est un cas général qu'on veut modéliser. On modélise ce modèle avec des fluides, des commandes et des magasins fournisseurs. ReponseDemande est une variable qui englobe la demande, le fluide livré et le magasin et vérifie qu'une commande bien particulière est satisfaite. Les données sont paramètres: Demandes, Stocks et CoutUnitaire. Les contraintes sont: le fluide demandé sur l'une des demandes est bien livré tout en prenant en considération la livraison par chaque magasin. Ainsi que la quantité demandée, elle doit être inférieure à la quantité au stock (on a le fluide en stock magasin que l'on souhaite. On cherche à minimiser le coût total sur l'ensemble des magasins. La solution respecte bien le stocker les demandes. La matrice de la solution est creuse.

```
6 Objective:  CoutLivraison = 9.5 (MINimum)
7
```

Figure 6: fonction objective du problème e-commerce cas1

3.3 Cas particulier 2

Dans cette partie, on reprend le même modèle du cas 1 avec quelques modifications, à savoir discrétisassions de la demande. Pour cela, on doit avoir forcement des valeurs entières dans les demandes. On conclut donc, que le cas 1 est une généralisation du cas 2.

```
6 Objective:  CoutLivraison = 10 (MINimum)
7
```

Figure 7: fonction objective du problème e-commerce cas2

3.4 Cas particulier 3

Dans les cas précédents, on prends en compte uniquement le coût payé par les clients pour acheter les produits, nous souhaitons désormais prendre en compte également les coût d'expédition des colis des magasins aux clients. On considère alors deux frais différents de livraison, le premier est un coût fixe, et le deuxième est un coût variable dépendant de la quantité transportée. On a alors définit une nouvelle variable binaire (Expedie), qui vaut 1 si le magasin m expédie un colis vers le client en question, et vaut 0 sinon. On ajoute également les deux constantes CoutExpeditionFixe et CoutExpeditionVariable qui représentent les frais d'expéditions fixes et variables d'un colis entre chaque paire (Demandes,Magasins).

Puisqu'on a définit une nouvelle variable, et il faut également définir des contraintes sur celle-ci, notamment une contrainte qui oblige que la variable (Expedie) est égale à 1 lorsque le magasin expédie effectivement un colis au client. Donc on a définie la contrainte (Expedition). Finalement on a ajouté les frais d'expédition (fixe et variable) au cout total payé part le client c.à.d à la fonction qu'on veut minimiser (CoutLivraision). On a choisi de travailler encore en PLNE pour ce cas parce qu'il ressemble plus au cas 2 qu'au cas 1. Sur les figures suivantes on trouve les données numériques utilisées pour tester et la solution obtenue par glpsol.

	M1	M2	M3
D1	110	90	100
D2	110	90	100

(d) Coûts fixes d'expédition d'un colis entre chaque paire : point de demande, magasin

	M1	M2	M3
D1	10	1	5
D2	2	20	10

(e) Coûts variables d'expédition d'un colis entre chaque paire : point de demande, magasin

Figure 8: Données numériques Cas3

```
6 Objective:  CoutLivraison = 368 (MINimum)
7
```

Figure 9: fonction objective du problème e-commerce cas2

3.5 Cas particulier 4

Ce dernier cas est indépendant des autres, on a un magasin ALPHA qui embauche un livreur pour assurer toutes les livraisons (des colis dont on a parlé précédemment), et on veut optimiser le chemin suivis par ce dernier pour livrer aux différents clients.

On a alors définit une set de Clients (y compris le magasin) et une variable $Passages[c1, c2]$ binaire qui prends en paramètres deux clients et évalue si le livreur a passé de c1 vers c2. Et pour les constantes on définit une matrice $Distance[c1, c2]$ qui représente le tableau des valeurs donné dans le sujet. Par la suite on définit des contraintes sur Passages, qui doit être une matrice des 0 et des 1, avec un seul 1 sur chaque ligne et chaque colonne (à partir d'un clients c1 on passe à un seul autre client c2 et on doit plus repasser par c1).

La fonction qu'on cherche à optimiser est DistanceTotal qui est la somme des Distance[c1,c2] multipliée par la variable Passages[c1,c2] (parce qu'on prends pas en compte la distance entre deux clients sauf si le livreur a passé entre les deux).

Mais après avoir définit ces variables et contraintes de base, on avait encore un résultat faux, car le livreur fait des sous-cycles (Sub-Cycles) entre les différents Clients, notamment une boucle (Alpha,C1,C2) et une boucle (C3,C4,C5), ce qui aboutit à une distance minimal de 6. Donc pour remédier à ce problème on s'est basé sur la formulation de Miller–Tucker–Zemlin (MTZ), et on a ajouté une variable Ordre (dummy variable) qui garde une trace de l'ordre dans lequel les clients sont visités, à partir du magasin ALPHA.

$$Ordre_{C_i} \leq Ordre_{C_j}$$

implique que le client i est visitée avant le client j. On a imposer plusieurs contraintes sur Ordre, notamment, si n est le cardinal du set CLIENTS: $1 \leq Ordre_i \leq n-1$ et on a également la condition suivante pour tout i, j:

$$Ordre_{C_j} + (n-2) \geq Ordre_{C_i} + (n-1) * Passage[c_i, c_j]$$

où le terme constant (n-2) fournit un écart suffisant pour que le cas de $Passage[c_i, c_j] = 0$ n'impose pas de relation (combinaison linéaire) entre $Ordre_{C_i}$ et $Ordre_{C_j}$.

En d'autres terme, en imposant ces contraintes sur la variable Ordre, on impose que le livreur avance dans un sens unique et ne peut pas faire de demi-tours pour revenir à un clients déjà desservie. Et en définissant cette variable que pour i entre 2..n, on assure qu'il va effectivement revenir au point de départ c.à.d le magasin après avoir terminé les livraison. On obtient alors une distance minimale de 22 qui est effectivement la solution optimale à ce problème.

	ALPHA	C1	C2	C3	C4	C5
ALPHA	-	1	1	10	12	12
C1	1	-	1	8	10	11
C2	1	1	-	8	11	10
C3	10	8	8	-	1	1
C4	12	10	11	1	-	1
C5	12	11	10	1	1	-

(f) matrice des distances (magasin ALPHA et 5 clients à livrer)

Figure 10: Données numériques cas4

```
Objective:  DistanceTotal = 22 (MINimum)
```

Figure 11: fonction objective du problème e-commerce cas4

4 Quatrième partie: Conclusion

Le travail en groupe sur ce TP rendu était vraiment utile pour la compréhension des notions vue en Cours et en TD. Les premiers cas n'étaient pas très difficile, et il servaient bien comme base pour avancer vers des cas de plus en plus délicats. Le cas 4 était le plus dur à traiter et il fallait prendre du temps pour visualiser l'objectif, le modéliser mathématiquement et discuter avec plusieurs de nos camarades de classe pour y arriver. Finalement on a réussi à proposer une solution qui marche et cela nous a motivé pour essayer de faire l'exercice Bonus. Mais par contrainte de temps on a pas terminer ce dernier donc il n'est pas inclue dans l'archive rendue.