

Recherche Opérationnelle :

TP1 et 2

Sandra U. Ngueveu (ngueveu@laas.fr), Arthur Claviere, Aloïs Duguet,
Alexandre Dupaquis, Quentin Fabry

2020

Table des matières

1	Exemple de Modélisation en PL(NE) avec Julia/JuMP	2
2	Applications en optimisation pour l'e-commerce	2
2.1	Cas particulier 1	2
2.2	Cas particulier 2	3
2.3	Cas particulier 3	3
2.4	Cahier des charges du problème général	3
3	BONUS	4

Le but est de modéliser et résoudre différents problèmes fournis en section 2 de ce document, à l'aide du package JuMP de Julia qui permet de faire appel à différents solveurs de PLNE à travers la même interface.

Rendus attendus (à soumettre sur moodle à l'emplacement dédié) : archive .zip contenant :

- un rapport synthétique (voir instructions sur Moodle)
- un notebook .ipynb contenant tous les problèmes traités (et permettant aux évaluateurs de les rejouer si besoin)

MENTION SPECIALE

Données (assainies) et cahier des charges du 2.4 tirés du stage de Louis Rivière et du projet Région-FEDER ONESTOCKPERFORMANCE 2016-2018.



1 Exemple de Modélisation en PL(NE) avec Julia/JuMP

Récupérer sur Moodle un notebook montrant un exemple de modélisation et résolution avec Julia/JuMP+solveur PLNE du problème de fabrication de ciment vu en cours et TD.

L'intérêt de JuMP est de proposer une interface unique à partir de laquelle les solveurs utilisés sont interchangeables à quelques options près. Pour le TP vous pourrez utiliser CBC ou GLPK qui sont opensource. La liste complète des solveurs disponibles est accessible à l'adresse <https://jump.dev/JuMP.jl/stable/installation/> (avec des instructions pour les installer et les utiliser avec JuMP).

2 Applications en optimisation pour l'e-commerce

Parmi les problématiques d'optimisation émergeant en e-commerce, se trouvent l'affectation de commandes de clients aux magasins, compte-tenu des coûts 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. Nous nous plaçons donc dans la peau du gestionnaire de l'ensemble des magasins, qui doit dépenser le moins d'argent possible pour satisfaire les demandes des clients.

2.1 Cas particulier 1

Les tableaux (a), (b) et (c) représentent à titre d'exemple des demandes en fluide émanant de différentes commandes et les coûts pour qu'une unité de fluide soit disponible dans un magasin. Chaque magasin dispose d'un volume de stockage limité. Modéliser et résoudre à l'aide d'un programme linéaire. Tester avec différents jeux de données de votre choix.

	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 de disponibilité par magasin

2.2 Cas particulier 2

En réalité, il ne s'agit pas de fluide mais de produits préconditionnés et une commande d'un client peut être constituée de plusieurs produits en quantités différentes. Le problème d'affectation consiste à déterminer le nombre de produits de chaque type livrés par chaque magasin à chaque client. Modifier et résoudre la formulation précédente pour tenir compte de la discrétisation de la demande. Tester avec différents jeux de données.

2.3 Cas particulier 3

A présent on souhaite prendre en compte les coûts d'expédition des colis des magasins aux clients. Chaque magasin expédie, vers chaque client qu'il dessert, un unique colis contenant tous les produits fournis par ce magasin à ce client. Modifier la formulation précédente pour modéliser le problème résultant et résoudre avec les données du tableau (d). Tester ensuite avec différents jeux de données.

	M1	M2	M3
D1	1	0	0
D2	0	2	1

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

2.4 Cahier des charges du problème général

L'objectif est de modéliser un problème de minimisation des trajets de livraisons des commandes (i.e. distance à parcourir) pour les livreurs des magasins. On fera donc abstraction des autres coûts pour se focaliser sur un problème mono-objectif. Le PLNE proposé doit prendre en entrée les données suivantes : les commandes à satisfaire (quantités et types de produits), les niveaux de stocks disponibles au sein des différents magasins, les temps de trajet entre les différents sites. A partir de ces données, le modèle doit pouvoir déterminer comment répartir les commandes entre les magasins et quelle est la tournée de livraison que doit réaliser le livreur de chaque magasin.

2.4.1 Données

- D : l'ensemble des commandes (équivalent à un ensemble clients car 1 client = 1 commande)

- P : l'ensemble des produits
- M : l'ensemble des magasins
- $N = M \cup D$: l'ensemble des noeuds (représentant les différents sites)
- $V = \{(i, j) \mid i, j \in N^2\}$ l'ensemble des arcs entre les noeuds.
- R : l'ensemble des $r_{ij} \in \mathbb{R}$: valeur de l'arc allant de i vers j représentant la distance à parcourir/temps de trajet entre les sites i et j
- Q : l'ensemble des $q_{dp} \in \mathbb{N}$: quantité de produit $p \in P$ dans la commande $d \in D$.
- S : l'ensemble des $s_{mp} \in \mathbb{N}$: stock de produit $p \in P$ dans le magasin $m \in M$.

2.4.2 Objectif

L'objectif est de minimiser la distance totale parcourue pour la livraison.

2.4.3 Contraintes

1. chaque magasin dispose de son propre livreur/camion qui sera en charge de livrer en une seule tournée tous les produits qui proviennent de son magasin (pas de limite de capacité sur les camions).
2. pour chaque magasin qui expédie au moins 1 produit, son livreur/camion débute sa tournée au magasin, visite une seule fois chacun des clients qu'il doit servir et retourne au magasin en fin de tournée.
3. chaque commande doit être satisfaite en totalité
4. une commande peut être satisfaite par un unique magasin qui livre tous les produits qui la composent, ou alors par plusieurs magasins, qui fournissent chacun une partie des produits
5. aucun magasin ne peut faire livrer plus de produits qu'il n'en possède en stock

3 BONUS

Modéliser avec Julia les versions génériques des problèmes vu en TD, ainsi que ceux de la fiche supplémentaire (quelques problèmes classiques) et tester avec différents jeux de données générés aléatoirement.