

Projet REOP: Inventory Routing XXL

Axel Parmentier, Guillaume Dalle

Année scolaire 2020-2021

Contents

1	Questions de l'énoncé	2
1.1	Question 1	2
1.2	Question 2	3
1.3	Question 3	3
1.3.1	Graphe des camions	4
1.3.2	Variables	4
1.3.3	Contraintes	4
1.3.4	Coûts	5
1.3.5	PLNE complet	6
2	Analyse des données	6
3	Heuristique d'Axel	6
3.1	Relaxation du PLNE exact	6
3.1.1	Flot des camions	6
3.1.2	Flot des emballages	8
3.1.3	Relaxation tous emballages	9
3.2	Construction séquentielle des routes	9
4	Heuristique de Guillaume	9
4.1	Etape 1 : flot d'emballages	10
4.2	Etape 2 : quantités envoyées	10
4.3	Etape 3 : remplissage initial des camions	10
4.4	Etape 4 : fusion des routes	10
4.5	Etape 5 : suppression des routes inutiles	11
4.6	Etape 6 : échanges inter-routes	11
4.7	Etape 7 : répéter	11
5	Résultats numériques	11
6	Autres pistes	12
6.1	Exploitation d'un PLNE exact	12
6.2	Méthodes inspirées de la littérature sur les VRP	12
6.2.1	Reconnaître un VRP particulier	12
6.2.2	Sources académiques à creuser	13

Nous vous proposons ici une correction du projet, avec plusieurs éléments de réponse et quelques algorithmes intéressants.

1.1 Question 1

$$c_{uf}^r = c^{\text{cam}} + c^{\text{stop}} + \gamma d_{uf}$$

Notre problème d'optimisation se reformule :

$$\begin{array}{l|l} \min_{\mathbf{q}, \mathbf{s}} & \begin{array}{l} \sum_{u,f,j} c_{uf}^x q_{ufj} \\ + \sum_{u,j} c_u^s (s_{uj} - r_{uj})^+ \\ + \sum_{f,j} c_f^s (s_{fj} - r_{fj})^+ \\ + \sum_{f,j} c_f^{\text{exc}} (b_{fj}^- - s_{f(j-1)})^+ \end{array} \\ \text{s.c.} & \begin{array}{ll} s_{uj} = s_{u(j-1)} + b_{uj}^+ - \sum_f q_{ufj} & \forall (u, j) \\ s_{fj} = (s_{f(j-1)} - b_{fj}^-)^+ + \sum_u q_{ufj} & \forall (f, j) \\ \mathbf{s} \geq 0, \mathbf{q} \geq 0 & \\ \text{contraintes d'intégrité} & \end{array} \end{array}$$

Pour linéariser les parties positives $(\dots)^+$, notons qu'on a utilisé deux arêtes entre certaines paires de sommets : une arête limitée en capacité pour la partie gratuite, et une autre arête illimitée mais payante.

Type d'arc	Origine	Destination	Min	Max	Coût
Stock initial usine	stock_init	$(u, 0, \text{matin})$	s_{eu}^0	s_{eu}^0	
Stock initial fournisseur	stock_init	$(f, 0, \text{matin})$	s_{ef}^0	s_{ef}^0	
Stock final usine	$(u, J + 1, \text{matin})$	stock_fin			
Stock final fournisseur	$(f, J + 1, \text{matin})$	stock_fin			
Libération usine	libération	(u, j, matin)	b_{eu}^+	b_{eu}^+	
Demande fournisseur	(f, j, matin)	demande	b_{ef}^-	b_{ef}^-	
Cartons fournisseur	cartons	(f, j, matin)			c_{ef}^{exc}
Trajet $u \rightarrow f$	(u, j, matin)	(f, j, soir)			$c^{\text{cam}} + c^{\text{stop}} + \gamma d_a$
Stock usine journée	(u, j, matin)	(u, j, soir)			
Stock fournisseur journée	(f, j, matin)	(f, j, soir)			
Stock usine nuit gratuit	(u, j, soir)	$(u, j + 1, \text{matin})$		r_{eu}	
Stock fournisseur nuit gratuit	(f, j, soir)	$(f, j + 1, \text{matin})$		r_{ef}	
Stock usine nuit payant	(u, j, soir)	$(u, j + 1, \text{matin})$			c_{eu}^s
Stock fournisseur nuit payant	(f, j, soir)	$(f, j + 1, \text{matin})$			c_{ef}^s
Bouclage circulation	divers	divers			

Tableau 1: Arêtes du graphe de flot des emballages pour la Question 1

La seule difficulté liée à la modélisation par un b -flot est que certaines valeurs de b sont inconnues. On ne sait pas à l'avance (avant l'optimisation) combien de cartons il faudra faire entrer dans le graphe, ni combien de stocks chaque site conservera à la fin de l'horizon. Une façon élégante de surmonter cette difficulté est de créer des noeuds fictif **source** et **puits**, puis de boucler le graphe pour transformer le problème de flot en problème de circulation. Ainsi, les valeurs de b inconnues se transforment en valeurs de flot sur les nouvelles arêtes, et on se ramène exactement au cas du cours.

Le graphe obtenu est décrit exhaustivement dans le Tableau 1.

Dans la suite, puisque les camions pourront contenir plusieurs emballages, le flot des camions et le flot des emballages ne coïncideront plus. On gardera la notation \mathbf{q} pour le flot des emballages, tandis que le flot des camions sera noté \mathbf{x} .

1.2 Question 2

Si on fixe la longueur du camion et celle des emballages, autrement dit si la dimension du problème ne dépend que du nombre de sites, la formulation (11) contient un nombre polynomial de variables. En effet, les routes sont de longueur inférieure ou égale à 5. Cependant, cela signifie qu'il existe de l'ordre de $|U| \times |F|^4$ chemins possibles, ce qui est énorme vu la taille de l'instance **europe.csv**. Et en réalité, l'aspect emballages n'est pas juste une constante : il multiplie le nombre de routes possibles et rend cette formulation totalement inabordable.

1.3 Question 3

Supposons pour cette question que Renault utilise au plus K camions chaque jour. Ce ne sont pas forcément les mêmes de jour en jour, puisqu'ils ne reviennent pas dans leur usine de départ à la fin de la route. On peut alors reformuler le problème de façon compacte en optimisant le chemin et le chargement de chaque camion, ce qui évite d'énumérer toutes les routes.

1.3.1 Graphe des camions

Pour cela, on définit d'abord un graphe $G^c = (V^c, A^c)$ associé aux camions. Notons $T = 4$. Les sommets de ce graphe sont

$$V^c = \{\text{depart}\} \cup U \cup (F \times [T]) \cup \{\text{arrivee}\}$$

et ses arêtes sont

$$\begin{aligned} A^c = & \{\text{depart} \rightarrow u \mid u \in U\} \\ & \cup \{u \rightarrow (f, 1) \mid (u, f) \in U \times F\} \\ & \cup \{(f_1, t) \rightarrow (f_2, t+1) \mid (f_1, f_2, t) \in F \times F \times [T-1]\} \\ & \cup \{(f, t) \rightarrow \text{arrivee} \mid (f, t) \in F \times T\} \end{aligned}$$

On remarque qu'un $o-d$ chemin dans le graphe correspond à un trajet possible pour les camions.

1.3.2 Variables

Définissons maintenant les variables suivantes:

- x_{ajk} vaut 1 si le camion k du jour j emprunte l'arête $a \in A^c$ et 0 sinon
- y_{ujk} (resp. y_{fjk}) vaut 1 si le camion k du jour j part de l'usine u (resp. dessert le fournisseur f) et 0 sinon
- q_{eujk} (resp. q_{efjk}) est la quantité d'emballage e prise à l'usine u (resp. livrée au fournisseur f) par le camion k du jour j
- s_{euj} (resp. s_{efj}) est le stock d'emballage e de l'usine u (resp. du fournisseur f) au soir du jour j

Traduisons maintenant les contraintes.

1.3.3 Contraintes

Trajet admissible Le trajet de chaque camion doit correspondre à un $o-d$ chemin dans G^c . Puisque \mathbf{x} est à valeurs binaires, il suffit d'imposer la conservation du flot en tout $v \in V^c$ (sauf aux noeuds **depart** et **arrivee**) :

$$\sum_{a \in \delta^-(v)} x_{ajk} = \sum_{a \in \delta^+(v)} x_{ajk} \quad (1)$$

La variable \mathbf{y} est définie en fonction de \mathbf{x} par le comptage des camions incidents à chaque site :

$$\begin{aligned} y_{ujk} &= \sum_{a \in \delta^-(u)} x_{ajk} \\ y_{fjk} &= \sum_t \sum_{a \in \delta^-(f,t)} x_{ajk} \end{aligned} \quad (2)$$

Chargement admissible Soit $q_e^{\max} = \lfloor L/\ell_e \rfloor$ la quantité maximum d'emballage e qu'un camion peut emporter. Le chargement se fait à l'unique usine visitée, et le déchargement se fait chez les fournisseurs visités :

$$\begin{aligned} q_{eujk} &\leq q_e^{\max} y_{ujk} \\ q_{efjk} &\leq q_e^{\max} y_{fjk} \end{aligned} \quad (3)$$

Le chargement à l'usine doit être égal au déchargement ultérieur :

$$\sum_{u \in U} q_{eujk} = \sum_{f \in F} q_{efjk} \quad (4)$$

Et sa longueur totale doit être inférieure à celle du camion :

$$\sum_{e \in E} \ell_e \left(\sum_{u \in U} q_{eujk} \right) \leq L \quad (5)$$

Evolution des stocks Le stock des usines évolue selon la dynamique imposée :

$$s_{euj} = s_{eu(j-1)} + b_{euj}^+ - \sum_{k \in K} q_{eujk} \quad (6)$$

Il doit rester positif :

$$s_{euj} \geq 0 \quad (7)$$

Le stock des fournisseurs évolue selon la dynamique imposée :

$$s_{efj} = \left(s_{ef(j-1)} - b_{efj}^- \right)^+ + \sum_{k \in K} q_{efjk} \quad (8)$$

Cela implique automatiquement sa positivité.

1.3.4 Coûts

Le coût total lié au camion k du jour j est donné par :

$$c_{jk} = c^{\text{cam}} + c^{\text{stop}} \sum_{f \in F} y_{fjk} + \gamma \sum_{a \in A^c} d_a x_{ajk}$$

Le coût total lié à l'usine u le jour j est donné par :

$$c_{uj} = \sum_{e \in E} c_{euj}^s (s_{euj} - r_{euj})^+$$

Le coût total lié au fournisseur f le jour j est donné par :

$$c_{fj} = \sum_{e \in E} c_{efj}^s (s_{efj} - r_{efj})^+ + \sum_{e \in E, j \in J} c_{efj}^{\text{exc}} (b_{efj}^- - s_{efj})^+$$

Par conséquent, la fonction objectif à minimiser s'exprime de la façon suivante :

$$f(\mathbf{x}, \mathbf{y}, \mathbf{q}, \mathbf{s}) = \sum_{j \in J} \left(\sum_{k \in K} c_{jk} + \sum_{u \in U} c_{uj} + \sum_{f \in F} c_{fj} \right)$$

1.3.5 PLNE complet

On obtient donc le problème d'optimisation suivant :

$$\begin{array}{ll}
\min_{\mathbf{x}, \mathbf{y}, \mathbf{q}, \mathbf{s}} & \left| \begin{array}{l} \sum_{j,k} (c^{\text{cam}} + c^{\text{stop}} \sum_{f \in F} y_{fjk} + \gamma \sum_{a \in A^c} d_a x_{ajk}) \\ + \sum_{e,u,j} c_u^{\mathbf{s}} (s_{euj} - r_{euj})^+ \\ + \sum_{e,f,j} c_f^{\mathbf{s}} (s_{efj} - r_{efj})^+ \\ + \sum_{e,f,j} c_f^{\text{exc}} (b_{efj}^- - s_{ef(j-1)})^+ \end{array} \right. \\
\text{s.c.} & \left| \begin{array}{l} (1), (2), (3), (4), (5), (6), (7), (8) \\ \mathbf{x} \geq 0, \mathbf{y} \geq 0, \mathbf{q} \geq 0, \mathbf{s} \geq 0 \\ \text{contraintes d'intégrité / binarité} \end{array} \right.
\end{array}$$

Les parties positives sont aisément linéarisables, et on obtient un PLNE comprenant un nombre de variables en

$$\mathcal{O}[(U + F)^2 JK + E(U + F)KJ]$$

2 Analyse des données

La première chose à faire avant de résoudre un problème est une exploration des données fournies. L'instance `europe.csv` est caractérisée par les dimensions suivantes :

- 25 usines
- 576 fournisseurs
- 28 types d'emballages

Quelques statistiques représentatives ont été tracées sur la Figure 2.

On voit d'ores et déjà qu'il est illusoire d'espérer satisfaire toute la demande des fournisseurs. On voit aussi que la capacité de stockage chez l'ensemble des fournisseurs est bien plus importante que celle des usines (sans doute à cause de leur nombre), peut-être obtiendra-t-on donc un coût de stockage assez faible pour eux.

3 Heuristique d'Axel

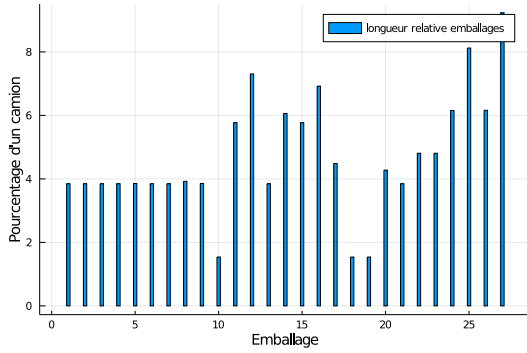
3.1 Relaxation du PLNE exact

Le problème de la formulation donnée pour la Question 3 est que sa taille augmente avec le nombre de camions nécessaires, dont on ne connaît pas de borne assez petite dans le cas général. Il est donc intéressant de proposer une formulation alternative qui ne dépende pas de K . Comme on le verra, cette nouvelle approche dégrade une partie des contraintes, c'est pourquoi elle fournit en fait une relaxation du problème initial.

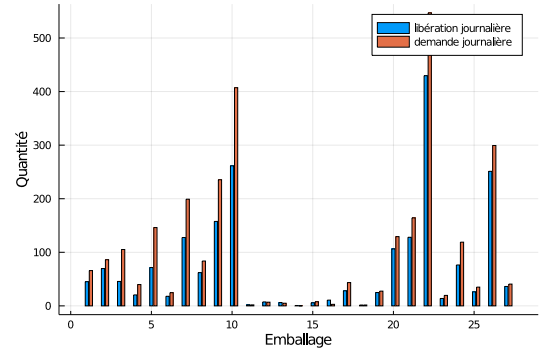
3.1.1 Flot des camions

On adopte une approche différente par rapport à la Question 3, en enrichissant le graphe des camions $G^c = (V^c, A^c)$ défini à la Section 1.3.1. Ses sommets sont toujours les mêmes, mais ses arêtes ont désormais un coût, donné par le Tableau 2.

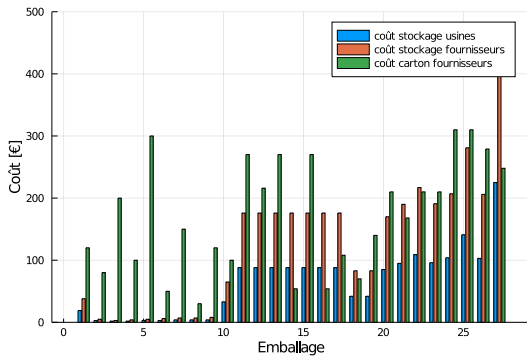
On associe à ce graphe un flot \mathbf{x}_j pour chaque jour j : la variable entière $x_{aj} \in \mathbb{N}$ compte ainsi le nombre de camions empruntant l'arête $a \in A^c$ ce jour-là.



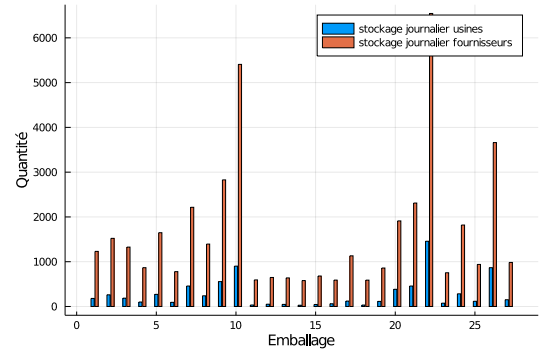
(a) Longueur relative des emballages



(b) Offre et demande



(c) Coûts de stockage et d'expédition par carton



(d) Capacités de stockage

Figure 2: Analyse préliminaire des données de `europe.csv`

Type d'arc	Origine	Dest.	Coût
Création camion	depart	u	c^{cam}
Trajet sur $a = u \rightarrow (f, 1)$	u	$(f, 1)$	γd_a
Trajet sur $a = (f_1, t) \rightarrow (f_2, t + 1)$	(f_1, t)	$(f_2, t + 1)$	γd_a
Destruction camion	(f, t)	arrivee	t_c^{stop}
Camions inutiles	depart	arrivee	
Bouclage circulation	arrivee	depart	

Tableau 2: Arêtes du graphe de flot des camions pour l'heuristique d'Axel

Type d'arc	Origine	Destination	Min	Max	Coût
Trajet sur $a = u \rightarrow (f, 1)$	(u, j, matin)	$(f, j, 1, \text{midi})$		Lx_{aj}	$\frac{c^{\text{cam}} + c^{\text{stop}} + \gamma d_a}{\ell_e}$
Trajet sur $a = (f_1, t) \rightarrow (f_2, t + 1)$	(f, j_1, t, midi)	$(f, j_2, t + 1, \text{midi})$		Lx_{aj}	

Tableau 3: Modification du graphe de flot des emballages dans les deux heuristiques constructives

bleu : uniquement dans l'heuristique d'Axel

rouge : uniquement dans l'heuristique de Guillaume

3.1.2 Flot des emballages

On définit maintenant un autre (multi-)graphe $G^e = (V^e, A^e)$, cette fois-ci associé aux emballages. Il est similaire à celui proposé pour la Question 1, mais nécessite quelques ajustements puisque cette fois les emballages peuvent aussi aller d'un fournisseur vers un autre. Une représentation graphique est donnée sur la Figure 3.

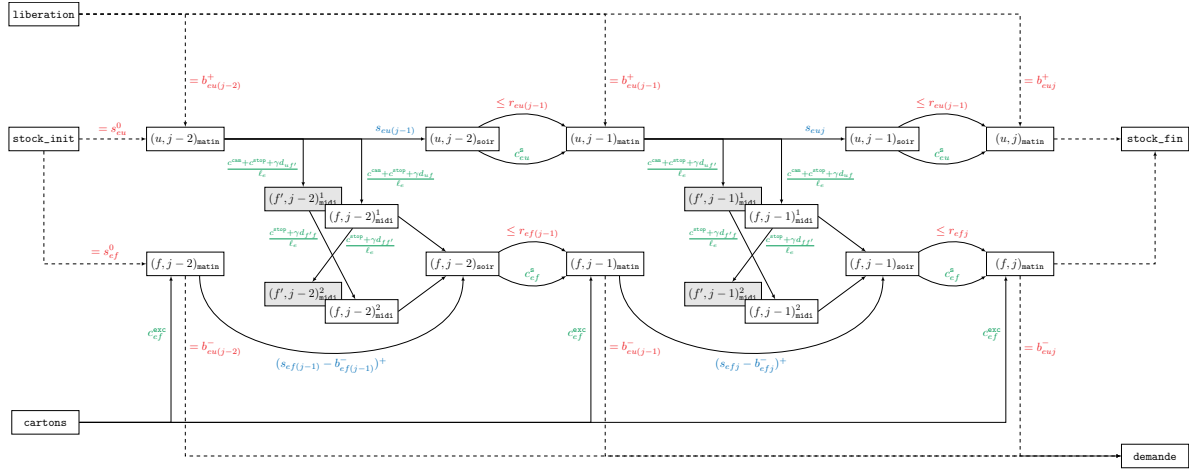


Figure 3: Graphe de flot complet des emballages (zoomez !)

Comme on le constate sur la figure, ses sommets sont donnés par

$$\begin{aligned}
 V^e = \{ & \text{source, puits} \} \\
 & \cup ([J] \times U \times \{\text{matin}, \text{soir}\}) \\
 & \cup ([J] \times F \times \{\text{matin}, \text{soir}\}) \\
 & \cup ([J] \times F \times [t] \times \{\text{midi}\})
 \end{aligned}$$

et ses arêtes sont décrites dans le Tableau 1 avec quelques modifications recensées dans le Tableau 3.

Remarque importante : contrairement au graphe des camions G^c , qui ne concerne qu'un seul jour, le graphe des emballages G^e s'étend sur tout l'horizon $[J]$. On associe donc à ce graphe d'emballages un unique flot \mathbf{z} couvrant l'ensemble des jours de l'horizon. Ce dernier représente le mouvement de tous les types d'emballages à travers le réseau, ainsi que leur stockage sur les différents sites.

Le mouvement des emballages étant conditionnée à la présence de camions, leur flot \mathbf{z} doit dépendre de la valeur de \mathbf{x} . Lorsqu'on considère un seul emballage à la fois, les contraintes de

capacité indiquées dans le Tableau 3 sont suffisantes. Lorsqu'on considère plusieurs emballages simultanément, ces contraintes de capacité prennent la forme d'un *bin-packing* car elles combinent les différents types : pour toute arête de trajet $a^e \in A^e$, on trouve l'arête correspondante $a^c \in A^c$ et on impose

$$\sum_e z_{a^e} \leq Lx_{a^c j} \quad (9)$$

Dans les deux cas, ces restrictions correspondent à la même idée : on ne peut envoyer des emballages à travers une arête que s'il y a suffisamment de camions pour les transporter.

3.1.3 Relaxation tous emballages

En rassemblant ces deux éléments, on obtient l'intégralité de notre problème :

$$\min \left(\mathbf{c}^e \cdot \mathbf{z} + \sum_j \mathbf{c}^c \cdot \mathbf{x}_j \right) \quad \text{s.c.} \quad \left\{ \begin{array}{l} \mathbf{x}_j \text{ est un flot entier sur } G^c \quad \forall j \\ \mathbf{z} \text{ est un flot entier sur } G^e \\ \text{contraintes liant } \mathbf{z} \text{ à } \mathbf{x} \end{array} \right.$$

Cependant, il ne s'agit pas d'une reformulation exacte du problème initial, car les contraintes sur les camions ont été légèrement relâchées. Par exemple, dans cette version, rien n'interdit aux camions d'échanger des emballages lors de leurs arrêts chez les différents fournisseurs. De plus, la contrainte de capacité pour les trajets des emballages est donnée par xL , où x est le nombre de camions passant par l'arête, ce qui est moins restrictif que d'insérer les emballages dans x camions de capacité L .

La raison sous-jacente est que pour avoir un couplage précis entre emballages et camions, il faut individualiser les camions. Or, comme on l'a vu à la Question 3, cela augmente fortement la dimension du problème. La relaxation présentée ici propose un compromis, sacrifiant un peu de précision contre une dimension plus faible.

3.2 Construction séquentielle des routes

Cette relaxation peut être utilisée pour optimiser simultanément sur l'ensemble des emballages, mais cela n'est pas forcément pertinent. En effet :

- la taille du problème devient très grande
- la reconstruction des camions n'est pas évidente à cause des imprécisions détaillées ci-dessus.

C'est pourquoi il est intéressant de l'utiliser de façon séquentielle, en considérant les emballages par ordre de taille décroissante et en remplissant les camions au fur et à mesure.

Cette heuristique n'a pas été implémentée en entier.

4 Heuristique de Guillaume

Nous présentons maintenant l'heuristique qui a été implémentée dans notre code. Elle est décrite de façon assez vague, un pseudocode devrait être bien plus rigoureux.

4.1 Etape 1 : flot d'emballages

On commence par optimiser les emballages envoyés par chaque usine à chaque fournisseur, mais en négligeant les contraintes de contenance des camions. Pour cela, un graphe similaire à celui de la Section 3.1.2 est utilisé, mais les limites de capacité sur les arêtes de type "trajet" sont remplacées par une approximation des coûts de transport. Comme le montre le Tableau 3, le coût de chaque arête de type "trajet" est défini comme le coût du "camion fractionnaire" occupé par un emballage e suivant cette arête :

$$c_{uf} = \frac{\ell_e}{L} (c^{\text{cam}} + c^{\text{stop}} + \gamma d_{uf}) \qquad c_{f_1 f_2} = \frac{\ell_e}{L} (c^{\text{stop}} + \gamma d_{f_1 f_2})$$

À partir d'une solution réalisable du problème initial, on peut toujours construire un flot admissible de valeur inférieure ou égale : cela justifie qu'on obtient ainsi une borne inférieure.

On remarque aussi que ce problème correspond au relâché linéaire de la borne inférieure obtenue par l'heuristique d'Axel. Sa valeur est donc une borne inférieure moins élevée que celle obtenue par Axel, mais ce problème présente néanmoins plusieurs avantages :

- On se convainc facilement que les arêtes de type $(f_1, t) \rightarrow (f_2, t + 1)$ ne seront jamais utilisées dans une solution optimale. Par conséquent, il est inutile de les inclure dans le graphe, ce qui évite un facteur F^2 dans le nombre de variables.
- Il s'agit d'un PL et non d'un PLNE, donc sa solution est extrêmement rapide.
- Sa résolution se décompose par emballages, contrairement à l'heuristique d'Axel dans le cas où on veut en tirer une borne inférieure.

4.2 Etape 2 : quantités envoyées

Rappelons que le flot obtenu à l'étape précédente n'est positif que sur les arêtes menant d'une usine vers un fournisseur. Les valeurs du flot sur ces arêtes nous fournissent donc des candidates naturelles pour fixer les quantités d'emballages e envoyées de u à f le jour j : on les note z_{eufj} .

4.3 Etape 3 : remplissage initial des camions

On veut maintenant remplir et router des camions pour acheminer ces quantités exactes, tout en payant un coût de transport minimal. Pour cela, on commence par construire des routes de longueur 1. Pour chaque jour j , pour chaque paire (u, f) , on considère tous les emballages que u doit livrer à f ce jour-là. Ces emballages sont agencés dans des camions avec l'heuristique *First Fit Decreasing*.

4.4 Etape 4 : fusion des routes

Les routes de longueur 1 n'étant pas rentables, on va maintenant chercher à mutualiser les camions. Cela peut se faire séparément pour chaque usine et chaque jour, en fusionnant itérativement la paire de camions qui nous permet l'économie la plus importante.

Dans un deuxième temps, on cherche aussi à fusionner des paires de camions partant de la même usine mais à des jours différents : cette étape est plus coûteuse car elle nécessite un recalcul partiel des stocks pour les sites concernés.

A chaque fois qu'une route est créée, l'ordre de ses étapes est toujours optimisé : ce n'est pas très coûteux car il n'y a que $4! = 24$ permutations possibles au maximum.

Instance	U	F	Cout final	Cout relaxation	Camions	Gap	Temps calcul
SLOVENIE	1	9	1451795	1445202	58	0.0	0.7
MAROC	2	12	345101	335463	39	0.03	0.8
PORTUGAL	1	23	382328	372465	31	0.03	1.2
ROUMANIE	1	76	1352498	1326457	222	0.02	4.5
ESPAGNE	6	113	844173	570675	822	0.48	12.7
FRANCE	14	184	1311451	872976	1264	0.5	31.8
EUROPE-SO	9	148	796508	403270	1038	0.98	21.8
EUROPE-O	14	248	1576511	1010427	1396	0.56	48.4
EUROPE	25	576	4245676	2861820	2977	0.48	175.1

Tableau 4: Performances de l’heuristique de Guillaume
Les temps de calcul sont donnés en secondes et les coûts en euros

4.5 Etape 5 : suppression des routes inutiles

Puisque l’étape définissant les quantités envoyées raisonnait sur des camions fractionnaires, il se peut tout à fait qu’un camion ait été créé pour desservir un fournisseur lointain avec peu d’emballages, et qu’il n’ait jamais été fusionné par la suite. Pour remédier à ce problème, on rajoute une étape de suppression des routes les plus coûteuses.

4.6 Etape 6 : échanges inter-routes

Enfin, on fait quelques mouvements impliquant des paires de routes : échanges et don d’étapes sont les deux voisinages implémentés.

4.7 Etape 7 : répéter

Puisque la solution a évolué depuis la première phase de la recherche locale, on peut répéter ces diverses étapes plusieurs fois, éventuellement en changeant leur ordre, jusqu’à stabilisation.

5 Résultats numériques

Nous présentons les résultats de l’heuristique de Guillaume sur diverses sous-instances, afin d’illustrer notamment l’augmentation du temps de calcul et la performance décroissante des algorithmes selon la taille de l’instance¹. Les calculs ont été réalisés sur un ordinateur portable présentant les caractéristiques suivantes :

Modèle MacBook Pro (13", 2017)

Processeur 3,5 GHz Intel Core i7 double cœur

Mémoire 16 Go 2133 MHz LPDDR3

¹Les instance agrégées sont définies comme suit :

- EUROPE-SO : Espagne, Portugal, Maroc
- EUROPE-O : France, Royaume-Uni, Allemagne, Belgique, Pays-Bas, Italie

Critère	Valeur
Cout stock usines	29816
Cout stock fournisseurs	473
Cout cartons fournisseurs	2134438
Cout camions	444150
Cout arrêts	391800
Cout kilometres	1260937
Taux remplissage moyen	0.64
Nombre etapes moyen	3.31

Tableau 5: Analyse de la solution trouvée sur EUROPE

6 Autres pistes

6.1 Exploitation d'un PLNE exact

Pour exploiter la formulation PLNE du problème bien qu'elle ne passe pas à l'échelle, on pourrait appliquer une *variable neighborhood search* en optimisant itérativement sur de petits sous-ensembles des variables, choisis en fonction de la géométrie du problème. Dans le vocabulaire de l'optimisation continue, cette méthode porte le nom de *block coordinate descent*.

Une alternative consisterait à utiliser la formulation exacte donnée dans les Equations (11a)-(11b) de l'énoncé à l'aide de méthodes de décomposition. Ce PLNE (ou plutôt sa relaxation continue) pourrait se prêter à une génération de colonnes, qui sélectionnerait au fur et à mesure les nouvelles routes les plus prometteuses. Cependant, ces techniques sont avancées et leur implémentation est complexe, c'est pourquoi nous avons privilégié des heuristiques plus simples dans ce corrigé. De plus, une brève réflexion sur le sujet ne nous a pas permis (à Axel et moi) de formuler un sous-problème facile à résoudre.

6.2 Méthodes inspirées de la littérature sur les VRP

Le problème que nous traitons est une variante du *Vehicle Routing Problem* (VRP) [Toth et al., 2014]. Par conséquent, pour améliorer les résultats présentés ici, il serait intéressant de se plonger dans la littérature scientifique associée pour y dénicher des méthodes plus avancées.

6.2.1 Reconnaître un VRP particulier

Notre variante du VRP présente les caractéristiques suivantes :

- Graphe sous-jacent dirigé et dense
- Coûts de transport satisfaisant l'inégalité triangulaire
- Routes de longueur maximale 5
- *Multiple Commodity* (MC) : Plusieurs marchandises sont transportées dans un même camion
- *Split Delivery* (SD) : Un client peut être desservi par plusieurs camions, mais attention contrairement au SDVRP les items ne sont pas infiniment divisibles (*Discrete Splits*). Apparemment il existe toujours une solution optimale entière quand même ?

- *Inventory Routing* (IR) : On ne s'intéresse pas uniquement à optimiser des routes, mais également à définir les quantités transportées pour gérer au mieux les stocks et les besoins de différentes entités.

6.2.2 Sources académiques à creuser

- Dror and Trudeau [1989]
- Dror et al. [1994]
- Archetti and Speranza [2012]
- Section 9.5 de Toth et al. [2014]
- Yu et al. [2008]

Bibliographie

- C. Archetti and M. G. Speranza. Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, 19(1-2):3–22, 2012. ISSN 1475-3995. doi: 10.1111/j.1475-3995.2011.00811.x.
- M. Dror and P. Trudeau. Savings by Split Delivery Routing. *Transportation Science*, 23(2):141–145, May 1989. ISSN 0041-1655. doi: 10.1287/trsc.23.2.141.
- M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50(3):239–254, May 1994. ISSN 0166-218X. doi: 10.1016/0166-218X(92)00172-I.
- P. Toth, D. Vigo, P. Toth, and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Society for Industrial and Applied Mathematics, USA, 2014. ISBN 978-1-61197-358-7.
- Y. Yu, H. Chen, and F. Chu. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189(3):1022–1040, Sept. 2008. ISSN 0377-2217. doi: 10.1016/j.ejor.2007.02.061.