
PLANIFICATION DE LIVRAISONS

1 Contexte

Les transporteurs tels que Air Liquide pour fournir de l'oxygène ou de l'azote liquide, La Poste pour livrer des colis ou encore Total pour alimenter les stations services, essaient de planifier leurs livraisons pour les rendre optimales en réduisant les coûts de transports. Ces transporteurs ont une contrainte majeure dont ils doivent tenir compte : leurs camions ont une capacité de chargement limitée. Ceci entraîne l'utilisation de plusieurs véhicules pour livrer chaque client dans un délai fixé. Ce problème de planification de livraisons avec contrainte de capacité de chargement est plus connu dans la littérature sous le nom de CVRP : *Capacited Vehicle Routing Problem*, i.e. de tournée de véhicules avec capacité. Dans ce projet, nous traiterons uniquement le cas particulier où chaque véhicule a la même capacité de chargement et part du même dépôt pour assurer ses livraisons puis y retourne une fois sa tournée effectuée.

2 Formulation du problème de planification de livraisons

Considérons C_1, C_2, \dots, C_n , n clients devant chacun être livré de q_i unités. Tous les véhicules partent d'un même dépôt noté D . La capacité (en unité) de chaque véhicule est notée Q .

Le but est de trouver la planification optimale qui permettra au transporteur de réduire ses coûts. Ce problème d'optimisation combinatoire est multi-objectif par nature car les coûts associés sont nombreux : salaires des chauffeurs, entretiens des véhicules, achat du carburant, respect du délai... Dans ce projet, nous nous intéresserons uniquement aux coûts liés à la distance parcourue pour une tournée de véhicules : le but est donc de planifier les livraisons de manière à minimiser la distance parcourue totale par l'ensemble des véhicules.

Soit le véhicule V_1 qui livre les clients C_1, C_2 puis C_3 . Le coût associé à ce véhicule est calculé par :

$$\text{cout}(V_1) = \text{dist}(D, C_1) + \text{dist}(C_1, C_2) + \text{dist}(C_2, C_3) + \text{dist}(C_3, D)$$

où $\text{dist}(i, j)$ est la distance de i à j avec $i, j \in \{D, C_1, C_2, \dots, C_n\}$ et $i \neq j$.

L'objectif est donc de minimiser le coût total :

$$\text{cout_total} = \sum_v^{\mathcal{V}} \text{cout}(V_v)$$

où \mathcal{V} est le nombre de véhicules nécessaires pour effectuer l'ensemble des livraisons.

*Proposez une modélisation à l'aide des graphes de ce problème.
Quelles structures de données sont nécessaires ?*

3 Résolution du problème de planification de livraisons

La résolution de ce problème est assez facile tant que le nombre de clients à livrer est relativement petit. En effet, le problème devient de plus en plus difficile à résoudre dès lors que ce nombre augmente. C'est l'explosion combinatoire : le nombre de solutions envisageables croît exponentiellement par rapport au nombre de clients et le coût total lié à chacune de ces solutions ne peut donc pas être calculé en un temps raisonnable voire possible !

C'est pourquoi des heuristiques ont été proposées pour approcher la solution optimale. Ces méthodes sont conçues pour fournir rapidement une *bonne* solution au problème mais l'optimalité n'est pas garantie. Généralement, les solutions trouvées par les heuristiques sont des points de départ utilisés par d'autres méthodes plus complexes (*les métaheuristiques* par exemple) qui les amélioreront. Dans ce projet, nous proposons d'implanter l'une de ces heuristiques proposée par Beasley en 1983 dans la revue scientifique Omega sous le nom de méthode *route-first/cluster-second*.

3.1 La méthode route-first/cluster-second

Cette méthode peut être divisée en trois étapes principales :

- (i) Créer un *Tour géant*, noté T ,
- (ii) Construire le graphe auxiliaire H à partir de T (procédure SPLIT),
- (iii) Appliquer un algorithme de plus court chemin sur H .

Cette méthode fournit une solution convenable du point de vue de l'objectif à minimiser. Dans la suite, des indications sont données sur ces différentes étapes.

3.1.1 Créer un *Tour géant*

Un *Tour géant* représente une tournée de véhicule où est utilisé un unique véhicule n'ayant aucune limitation de chargement. Ici, le but est de trouver un ordre initial de livraison des clients, appelé *Tour géant*. Une méthode gloutonne, appelée *méthode du plus proche voisin*, est classiquement utilisée pour définir cet ordre initial, *i.e.* le choix du prochain client à livrer est effectué de proche en proche en choisissant parmi ceux non-encore planifiés le plus proche (en distance) du client courant. A noter que dans cette méthode, le dépôt n'est pas pris en compte dans la construction de cet ordre initial.

Décrire le fonctionnement de cette méthode gloutonne ?

Cet ordre dépend-il du premier client livré ?

La solution fournie par cette méthode, notée T , est utilisée comme donnée d'entrée à la procédure de découpage suivante qui construit le sous-graphe auxiliaire H [(ii)] sur lequel est ensuite appliqué l'algorithme de plus court chemin (*pcc*) [(iii)].

3.1.2 Construire le graphe auxiliaire – Procédure de découpage (*SPLIT*)

Prins, en 2004, a proposé une méthode efficace pour déterminer le graphe auxiliaire H associé au problème à partir d'un tour géant donné qui permet de résoudre le problème de planification avec contrainte de capacité en appliquant, *simplement*, un algorithme de *pcc* sur ce graphe. Le plus court chemin dans ce graphe, recouvrant tous les sommets, donne la planification des livraisons, *i.e.* pour chaque client on connaîtra son ordre dans la livraison et par quel véhicule il sera livré.

Procédure SPLIT

Données :

T tour géant
 Q capacité des véhicules
 n nombre de clients
 $dist$ distances entre $x, y \in \{D, C_1, C_2, \dots, C_n\}$
 q vecteur du nombre d'unités à livrer à chaque client

Résultat :

H sous-graphe auxiliaire de sommets h_i ($i \in \{0, \dots, n\}$) associé à T

Locales :

$cost$ coût courant /* distance parcourue par le véhicule courant */
 $load$ chargement courant /* chargement du véhicule courant */
 i, j indices utilisés pour parcourir les clients de T

Pour i de 1 à n

$j \leftarrow i$
 $load \leftarrow 0$ /* un nouveau véhicule est affrété */

Répéter

$load \leftarrow load + q(T_j)$

Si ($i = j$) **alors**

/* T_j est le premier client qui sera livré par le véhicule courant */
 $cost \leftarrow dist(D, T_i) + dist(T_i, D)$

Sinon

/* T_j est inséré à la fin de la livraison effectuée par le véhicule courant */
 $cost \leftarrow cost - dist(T_{j-1}, D) + dist(T_{j-1}, T_j) + dist(T_j, D)$

FinSi

Si ($load \leq Q$) **alors**

/* la livraison de T_j peut être assurée par le véhicule courant */
Créer un arc (h_{i-1}, h_j) de coût $cost$ dans H

FinSi

$j \leftarrow j + 1$

jusque ($j > n$) or ($load \geq Q$) /* La livraison de tous les clients a été planifiée
-ou- la livraison du client T_j entraîne une violation de la contrainte de capacité */

FinPour

Soit $T = (T_1, T_2, \dots, T_n)$ un *tour géant* où l'ensemble $\{T_1, T_2, \dots, T_n\}$ est en bijection avec l'ensemble des clients à livrer $\{C_1, C_2, \dots, C_n\}$. La procédure SPLIT donnée ci-dessus montre comment construire le graphe auxiliaire H associé au *tour géant* T .

Le nom de la méthode de Beasley est donc maintenant plus compréhensible : les clients sont tout d'abord (*first*) ordonnés sans tenir compte des contraintes de capacité pour générer un Tour géant (*route*) puis (*second*) ce tour est découpé en *clusters* qui forment les différentes livraisons caractérisées par chaque véhicule.

Définir et décrire le graphe auxiliaire H construit par la procédure SPLIT.
Quel algorithme de pcc allez-vous utiliser ? Pourquoi et comment ?

3.2 Mise en œuvre sur un exemple

Parfois un bon schéma vaut mieux qu'un beau discours... ou permet de mieux se figurer les choses !

Ci-dessous vous trouverez un exemple donné par Prins à ses étudiants pour expliquer cette procédure de découpage. 5 clients $\{a, b, c, d, e\}$ doivent être livrés. La matrice $dist$ donne les distances entre $x, y \in \{D, C_1, C_2, \dots, C_n\}$. Le vecteur q donne le nombre d'unités à livrer à chaque client.

(i) Créer un tour géant

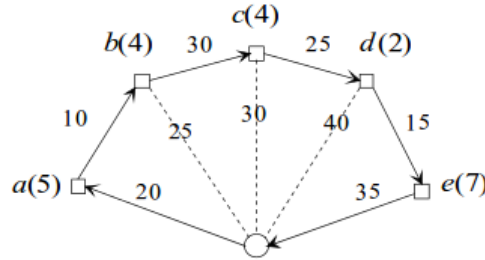
$dist$:

$$\begin{matrix} & D & a & b & c & d & e \\ D & \begin{pmatrix} - & 20 & 25 & 30 & 40 & 35 \end{pmatrix} \\ a & \begin{pmatrix} 20 & - & 10 & 15 & 40 & 50 \end{pmatrix} \\ b & \begin{pmatrix} 25 & 10 & - & 30 & 40 & 35 \end{pmatrix} \\ c & \begin{pmatrix} 30 & 15 & 30 & - & 25 & 30 \end{pmatrix} \\ d & \begin{pmatrix} 40 & 40 & 40 & 25 & - & 15 \end{pmatrix} \\ e & \begin{pmatrix} 35 & 50 & 35 & 30 & 15 & - \end{pmatrix} \end{matrix}$$

q :

$$\begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} \begin{pmatrix} 5 \\ 4 \\ 4 \\ 2 \\ 7 \end{pmatrix}$$

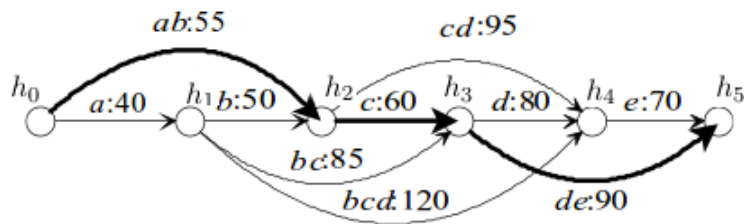
Ici, a est choisi comme le premier client à livrer. Le *tour géant* pour visiter ces 5 clients est donc donné par $T = (a, b, c, d, e)$.



Sur ce tour géant sont précisées la distance à considérer entre chaque client successif et la distance entre le dépôt et chacun des clients. Pour chaque client, le nombre d'unités à livrer est indiqué entre parenthèses.

(ii) Construire le graphe auxiliaire

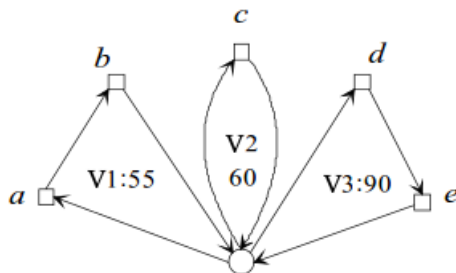
Ensuite, la procédure SPLIT construit le graphe auxiliaire H présenté ci-dessous sachant que la capacité des véhicules ne peut excéder une quantité de 10 unités.



Ce graphe permet de modéliser toutes les tournées envisageables à partir du tour géant T construit ci-dessus en commençant les livraisons par le client a . Par exemple, l'arc $(a:40)$ représente une première tournée où seul le client a doit être livré par un véhicule partant du dépôt et y revenant directement après la livraison effectuée, l'arc $(ab:55)$ représente une première tournée où les clients a puis b sont livrés par un même véhicule. Une solution naïve à ce problème est d'utiliser 5 véhicules pour livrer chacun des 5 clients, soit la solution $\{(a); (b); (c); (d); (e)\}$ pour un coût total de 300 ($40 + 50 + 60 + 80 + 70$). Cette solution est facilement visible sur le graphe ci-dessus comme le chemin pris par les arcs horizontaux.

(iii) Appliquer un algorithme de plus court chemin

Un algorithme de plus court chemin appliqué sur le graphe permet d'obtenir la solution optimale à partir de ce tour géant (voir arcs en gras sur le graphe H). Soit la solution $s = \{(a, b); (c); (d, e)\}$ où seulement 3 véhicules sont nécessaires pour assurer le bon fonctionnement des livraisons. Le premier véhicule (V_1) effectuera une distance de 55, le deuxième (V_2) de 60 et le troisième (V_3) de 90. Ici, $\text{cout_total}(s) = 205$.



4 Description d'un problème par fichier texte

Nous vous proposons d'utiliser le format suivant de fichiers pour décrire le problème de planification avec contrainte de capacité où tous les véhicules partent du même dépôt. Soit :

- 1 ligne avec le nombre de clients à livrer(n),
- 1 ligne avec la capacité maximale en unité des véhicules (Q),
- 1 ligne avec les demandes de livraison q_i ($i \in \{1, \dots, n\}$) en unité de chaque client $\{C_1, C_2, \dots, C_n\}$,
- $n + 1$ lignes avec les $n + 1$ distances entre x, y où $x, y \in \{D, C_1, C_2, \dots, C_n\}$,

L'exemple donné précédemment est ainsi représenté dans un fichier par :

5					
10					
5	4	4	2	7	
0	20	25	30	40	35
20	0	10	15	40	50
25	10	0	30	40	35
30	15	30	0	25	30
40	40	40	25	0	15
35	50	35	30	15	0

5 Cours Moodle

La plateforme Moodle sera utilisée pour mettre à disposition des instances du problème et pour rendre les différents rapports demandés.

5.1 Inscription

Le cours se trouve dans la catégorie :

“ Polytech Lille et EPU > Formation ingénieur Polytech Lille > GIS > GIS3 ”

sous le nom Projet SD-GC.

La clé d'inscription est `sdgc2019`.

Note : c'est le même cours et la même clé d'inscription pour les GIS3 et GIS2A3.

5.2 Instances à résoudre

Dans le fichier `exemple.dat`, vous retrouverez l'exemple utilisé dans le sujet.

Les instances suivantes sont proposées pour que vous puissiez tester votre programme de résolution. Elles ont été construites pour avoir des caractéristiques différentes. Pour chacune d'elles, nous vous indiquons le coût total obtenu par la méthode route-first/cluster-second en commençant par livrer le client 1, puis la meilleure solution pouvant être obtenue en précisant le client de départ (ceci à titre indicatif car la recherche de cette meilleure solution n'est pas demandée, c'est un plus). A noter, qu'ici les coûts sont obtenus en explorant les sommets dans l'ordre numérique.

Dans le fichier `cvrp_100_1_det.dat`, vous trouverez une instance où :

- le coût total* est 14289,27 lorsque le tour géant commence par le client 1,
- le coût total* est 14162,33 lorsque le tour géant commence par le client 3.

Cette instance a été construite pour ne présenter aucune distance identique entre les clients. Qu'importe l'implémentation choisie des différents algorithmes, vous devez trouver ces résultats.

Dans le fichier `cvrp_100_1_r.dat`, vous trouverez une instance où :

- le coût total* est 1014,55 lorsque le tour géant commence par le client 1,
- le coût total* est 992,7 lorsque le tour géant commence par le client 59.

Dans le fichier `cvrp_100_1_c.dat`, vous trouverez une instance où :

- le coût total* est 882,29 lorsque le tour géant commence par le client 1,
- le coût total* est 856,51 lorsque le tour géant commence par le client 13.

6 Travail à rendre à votre tuteur

Le **mercredi 15 mai avant 13h50**, déposer sur moodle un rapport d'analyse et de conception (selon votre tuteur, veuillez lui rendre une version imprimée), contenant :

- Précisions éventuelles du cahier des charges,
- Analyse (modélisation + méthode de résolution) et conception du problème proposé,
- Choix et justification des différentes structures de données utilisées pour représenter le problème au cours des différentes étapes de résolutions,
- Algorithmes de résolution appliqués à votre structure de données : méthode pour créer le tour géant, procédure SPLIT et pcc,
- Pseudo-code du programme principal (correspondant au `main` en C) du projet.

Des questions sont posées au cours des sections pour vous aider dans votre réflexion. Votre analyse doit y apporter des réponses.

A la fin du projet, au plus tard le **vendredi 7 juin 8h**, déposer sur moodle un rapport final (selon votre tuteur, veuillez lui rendre une version imprimée), contenant :

- Partie analyse (modélisation + méthode de résolution) et conception du premier rapport avec des compléments éventuels,
- Mode d'emploi (commandes de compilation et de lancement, description des fichiers d'entrées et sorties),
- Description des exemples traités et résultats obtenus,
- Conclusion (point sur ce qui a été fait / non fait), améliorations possibles,
- Bilan personnel sur le projet.

et les sources et fichiers exemples (si vous en avez d'autres).