

Feuille de travaux pratiques n° 4

Une méthode simple pour la résolution exacte d'un problème de tournée de véhicule avec capacité

1 Contexte : Blade Flyer

Nantes 2024 ; cela fait 5 ans que l'explosion de la bulle des communications a eu lieu, entraînant l'abandon des services estimés trop lents et faisant notamment disparaître des entreprises historiques comme "La poste". Les infrastructures (routières, réseaux de communication) peinent à suivre la charge de la demande, et le service personnalisé s'est imposé. Plus que jamais, l'optimisation est indispensable afin de rationaliser les décisions.

Une révolution a définitivement changé les sociétés de livraison, l'agence aérienne nationale a rendu en mai 2019 un avis favorable sur l'exploitation de drones¹ pour assurer des livraisons à courte distance, les "Blade Flyer". Petit engin autonome volant, le système apparaît fiable, flexible et indépendant des encombrements du trafic routier. Cette technologie ouvre la possibilité de proposer une livraison personnalisée, dans une fenêtre de temps serrée pour le rendez-vous, soulageant le trafic routier et présentant un faible impact en terme d'émission CO2.

Partant d'un dépôt géolocalisé où se trouvent les produits, un drone est capable de transporter une certaine quantité de produits, visiter plusieurs clients en déposant la livraison attendue, et rentrer au dépôt. L'activité dépôt-client(s)-dépôt définit une tournée.

La "FlyBlade Corporation" qui construit les drones souhaite développer un service sur Nantes fondé sur cette technologie innovante. Etant donné

- la géolocalisation du dépôt,
- la disponibilité actuelle d'un seul type de drone, de capacité unique,
- pour chaque client, sa géolocalisation et sa demande.

Votre travail consiste à mettre en place une solution informatique performante qui, connaissant les données du problème (lieu dépôt, lieux clients, demande clients, capacité du drone), calcule les tournées de façon à livrer tous les clients, tout en minimisant la distance totale parcourue par le drone.

Cependant en pratique, afin que ces drones puissent effectuer un service fonctionnel et efficace, des méthodes d'optimisation aussi bien exactes qu'approchées sont utilisées de manière conjointe. Un itinéraire optimal est d'abord défini puis ajusté en cas de modification des données au cours de la tournée. En particulier, il est possible de livrer non pas en un lieu prédéfini mais là où on se trouve le client, le drone identifiant celui-ci et sa position à l'aide des informations fournies automatiquement par le GSM du client². Le drone doit dans ce cas être capable de modifier l'itinéraire prédéfini, à l'aide de méthodes approchées fondée sur l'Intelligence Artificielle (algorithme génétiques, colonies de fourmis, réseaux de neurones, logique floue...) capables de déterminer une solution proche de l'optimalité pour des problèmes de grande taille, en utilisant peu de ressources mémoire et CPU³.

2 Définition du problème

Nous commençons par définir les données du problème. Nous notons l'ensemble des lieux $\{0, \dots, n\}$ où n est le nombre de clients, l'ensemble des clients est indexé de 1 à n et l'indice 0 représente le dépôt. L'utilisation de ces indices nous permet de définir le distancier d'un lieu i à un lieu j par c_{ij} pour $i, j \in \{0, \dots, n\}$. Nous notons d_i la

1. Le concept existe et a été récemment présenté par amazon, à raison d'un seul client à courte distance par tournée.

2. Perspective techniquement réalisable aujourd'hui.

3. À suivre... Ce contexte est inspiré de problèmes réels, de technologies existantes, d'expression de besoins crédibles et de l'ambiance cyberpunk du film "Blade Runner" (1982, Ridley Scott).

demande de chaque client $i \in \{1, \dots, n\}$ et Ca la capacité du véhicule de livraison. Nous supposons qu'aucun client n'a une demande supérieure à la capacité du véhicule.

Ces données étant posées, il ne reste plus qu'à définir les variables de décision, et à écrire la modélisation... ce qui est loin d'être immédiat ici ! Les décisions à prendre sont en effet nombreuses et une décomposition du problème s'impose.

Nous nous plaçons d'abord dans le cas (utopique) où la capacité du véhicule de livraison est suffisante pour satisfaire la demande de la totalité des clients, il n'y a alors qu'une seule tournée à effectuer partant du dépôt, visitant une seule fois chaque client, avant un retour au dépôt, tout en minimisant la distance parcourue. Il s'agit alors d'un problème de voyageur de commerce.

Bien entendu, il est généralement nécessaire d'effectuer plusieurs tournées pour satisfaire la demande des clients. Une première question à se poser, concerne les sous-ensembles de clients qu'il (n')est (pas) possible de regrouper dans une même livraison. Si on note S un sous-ensemble de clients et que $\sum_{i \in S} d_i > Ca$, on ne pourra pas inclure l'ensemble des clients de S dans une même tournée. Ensuite, pour chaque sous-ensemble de clients qu'il est possible de regrouper, il reste à déterminer la tournée passant une fois par chaque client, en partant du dépôt avant d'y revenir, tout en minimisant la distance. Une fois ces informations connues, nous pourrions poser une modélisation. Pour illustrer, on considère un exemple didactique.

Exemple : On suppose que $n = 5$, $Ca = 10$, $d_1 = d_3 = d_5 = 2$ et $d_2 = d_4 = 4$. Le distancier est défini par la matrice suivante.

$$c = \begin{pmatrix} 0 & 334 & 262 & 248 & 277 & 302 \\ 334 & 0 & 118 & 103 & 551 & 105 \\ 262 & 118 & 0 & 31 & 517 & 180 \\ 248 & 103 & 31 & 0 & 495 & 152 \\ 277 & 551 & 517 & 495 & 0 & 476 \\ 302 & 105 & 180 & 152 & 476 & 0 \end{pmatrix}$$

Le tableau ci-dessous récapitule les 27 regroupements possibles, et donne la longueur de la plus courte tournée visitant les clients de ces regroupements.

Indice	1	2	3	4	5	6	7	8	9
Regroupement	{1}	{1, 2}	{1, 2, 3}	{1, 2, 3, 5}	{1, 2, 4}	{1, 2, 5}	{1, 3}	{1, 3, 4}	{1, 3, 4, 5}
Longueur tour	668	714	730	803	1208	787	685	1179	1209
Indice	10	11	12	13	14	15	16	17	18
Regroupement	{1, 3, 5}	{1, 4}	{1, 4, 5}	{1, 5}	{2}	{2, 3}	{2, 3, 4}	{2, 3, 5}	{2, 4}
Longueur tour	758	1162	1192	741	524	541	1065	747	1056
Indice	19	20	21	22	23	24	25	26	27
Regroupement	{2, 4, 5}	{2, 5}	{3}	{3, 4}	{3, 4, 5}	{3, 5}	{4}	{4, 5}	{5}
Longueur tour	1195	744	496	1020	1153	702	554	1055	604

Connaissant ces informations, il reste à choisir plusieurs de ces tournées afin de visiter tous les clients une seule fois, en minimisant la distance cumulée des tournées choisies. À partir des informations données dans le tableau ci-dessus, nous pouvons enfin poser un modèle. Nous associons une variable de décision à chaque tournée.

$$x_i = \begin{cases} 1 & \text{si on choisit la tournée } i, \\ 0 & \text{sinon,} \end{cases}$$

où $i \in \{1, \dots, 27\}$. En notant l_i la longueur de la plus courte tournée visitant chaque client du regroupement $i \in \{1, \dots, 27\}$, la fonction objectif (à minimiser) s'écrit simplement

$$z = \sum_{i=1}^{27} l_i x_i.$$

Il reste à écrire les contraintes garantissant que chaque client est visité une seule fois. Pour le client 1, les regroupements possibles sont indicés de 1 à 13. On a donc la contrainte

$$\sum_{i=1}^{13} x_i = 1.$$

Les autres contraintes s'écrivent de manière similaire. On obtient un problème de *partitionnement d'ensemble*.

Il ne reste plus qu'à résoudre ce programme linéaire en variables binaires pour obtenir le résultat :

- la valeur optimale pour la fonction objectif est de 1357,
- la première tournée ($x_{25} = 1$) ne visite que le client 4 pour une longueur de 554,
- la seconde tournée ($x_4 = 1$) visite dans l'ordre les clients 5, 1, 3 et 2 pour une longueur de 803.

Nous résumons la méthode appliquée à cet exemple :

1. Énumérer les regroupements possibles de clients,
2. Pour chacun de ces regroupements, déterminer la plus courte tournée partant du dépôt, visitant une fois chaque client, et revenant au dépôt,
3. En déduire une instance de problème de partitionnement d'ensemble, puis la résoudre.

Pour le point 2., on pourra (au moins dans un premier temps) faire une énumération de l'ensemble des tournées possibles. *Indication* : On peut représenter une tournée par l'ordre de visite des villes.

3 Travail à effectuer

Une implémentation de cette méthode en faisant appel au solveur GLPK est demandée. Clairement, on ne souhaite pas ici "juste" résoudre un Programme Linéaire en variables binaires. Par conséquent, il sera nécessaire de faire appel au solveur GLPK en tant que bibliothèque de fonctions dans un code C/C++. Cette implémentation permettra ensuite d'observer le fonctionnement de la méthode, de pointer ses faiblesses, et éventuellement de proposer des améliorations.

Plusieurs fichiers sont disponibles sur madoc dans l'archive `ProjetRO.zip` :

- Le fichier `Projet_NOMS.c` contient un squelette à compléter,
- Deux dossiers d'instances numériques A et B.

Les instances numériques sont des fichiers textes dont le format est donné par :

- La première ligne indique le nombre de lieux,
- La deuxième ligne indique la capacité du véhicule,
- La troisième ligne indique la demande des clients,
- Les lignes suivantes indiquent le distancier.

Le fichier des données de l'exemple est indiqué ci-dessous.

```
6
10
2 4 2 4 2
0 786 549 657 331 559
786 0 668 979 593 224
549 668 0 316 607 472
657 979 316 0 890 769
331 593 607 890 0 386
559 224 472 769 386 0
```

La date limite de remise des projets est fixée au mercredi 2 avril à 18h. Le code source devra être remis sur madoc et le rapport sera déposé dans le casier de votre enseignant de TP. Le rapport devra contenir :

- Une description de l'algorithme d'énumération des regroupements possibles des clients,
- Une description de l'algorithme d'énumération des tournées,
- Une description (et une justification) des structures de données choisies pour l'implémentation,
- Une analyse des résultats issus de la résolution des instances numériques par votre implémentation. En particulier, on pourra considérer le nombre de regroupements possibles de clients, la taille de ces regroupements, le temps CPU de résolution totale, le temps CPU nécessaire pour la construction de l'instance de partitionnement d'ensemble...
- Si le temps le permet : des propositions d'améliorations et des retours sur ces améliorations. Les enseignants de TP seront heureux d'en discuter avec vous !