Feuille de travaux pratiques n^o 3

Un problème de tournée de drones dans un contexte de déconfinement

1 Contexte

Nous sommes en mars 2021, cela fait un an que le pays est paralysé par un confinement semi-continu entrecoupé de nombreuses restrictions (accès à l'Université, couvre-feu...). Parallèlement à cela, les premiers signes du printemps sont visibles, les jours se rallongent, la nature se réveille, l'espoir revient avec un souhait renouvelé de liberté de la part de la population. Le gouvernement décide [1] alors de changer sa stratégie concernant le Covid-19, pour adopter une stratégie qui a fait ses preuves en Asie.

Afin de pouvoir relâcher la plupart des restrictions, un principe simple doit être rigoureusement appliqué : l'isolement total des personnes atteintes du Covid-19 jusqu'à la fin de leur période de contagion. Les personnes en situation de faiblesse sont complètement prises en charge par les hôpitaux. Ce sont les autres personnes qui nous intéresseront. Ces personnes seront simplement isolées dans leur domicile. Il est important de noter qu'on parle bien ici d'isolement total. Aucune sortie n'est permise, même pour faire des courses ou acheter des médicaments. Ces personnes sont donc livrées à domicile directement par des drones afin d'éviter la contamination de livreurs.

Dans chaque zone géographique, un unique dépot est réquisitionné pour stocker des ressources alimentaires et des médicaments. Nous supposerons que la capacité de stockage de cet entrepôt est suffisante pour répondre aux besoins des personnes en isolement de la même zone géographique. Les drones partiront pleinement chargés de ce dépot pour livrer les personnes en isolement. Le choix du drone a un inconvénient essentiel : sa capacité de chargement est limitée. Par conséquent, les livraisons des personnes en isolement devront être quotidiennes. Cela aura donc un coût qu'il faudra essayer de limiter autant que possible. Pour accomplir cela, nous devrons planifier des tournées de drones, i.e. les drones partiront du dépot et livreront plusieurs personnes, dans la limite de leur capacité de chargement, avant de rentrer au dépot. Nous supposerons que les drones sont tous identiques. Ils ont donc tous la même capacité de chargement. Par contre, du fait de la diversité des situations familiales, les besoins des personnes en isolement seront variables. Finalement, la réduction des coûts passera par une minimisation de la distance parcourue par l'ensemble des drones.

2 Définition du problème

Le problème considéré dans le contexte appartient à la classe des problèmes de tournées de véhicules [2]. De nombreux problèmes existent dans cette classe de problèmes, incluant souvent des contraintes spécifiques à certaines applications. Dans le cadre de ce projet, nous nous en tiendrons à la variante la plus basique, ce qui ne signifie en aucun cas que le problème sera simple.

Nous commençons par définir les données du problème. Nous notons l'ensemble des lieux $\{1, \ldots, n\}$ où l'indice 1 représente l'unique dépôt et les clients sont indicés de 2 à n. L'utilisation de ces indices nous permet de définir le distancier d'un lieu i à un lieu j par c_{ij} pour $i, j \in \{1, \ldots, n\}$. Nous notons d_i la demande de chaque client $i \in \{2, \ldots, n\}$ et Ca la capacité du véhicule de livraison. Nous supposerons qu'aucun client n'a une demande supérieure à la capacité du véhicule, et que les véhicules ne feront aucune livraison partielle à un client.

1. Ce scénario est fictif et n'annonce en aucun cas une décision à venir du gouvernement.
2. Le fait que les véhicules soient des drones n'est pas seulement dû au fait que le responsable du cours aime bien en placer dans ses sujets de projet. Une flotte de drones peut être utilisée de manière collaborative en vue de nombreux objectifs et ces objectifs requiert toujours de l'optimisation. Voir par exemple ce sujet de stage de M2 reçu le 3 mars : `https://www.rte-france.com/carrieres/nos-offres/stage-ingenieur-deploiement-drones`

Le but du problème est comme attendu de minimiser la distance parcourue par l'ensemble des véhicules tout en satisfaisant la demande de la totalité des clients.

Habituellement, la première étape avant la résolution d'un problème est d'en effectuer une modélisation. Plusieurs modélisations directes ont déjà été proposées dans la littérature pour le problème. Cependant, ces modélisations ont l'inconvénient de demander un nombre exponentiel de contraintes par rapport au nombre de lieux n. Elles sont donc difficilement exploitables. Afin de résoudre de manière exacte le problème, nous allons donc chercher à effectuer un traitement sur les données pour écrire une autre modélisation. Ce sera l'objet de la méthode exacte présentée dans la section 3.

Comme nous l'avons observé dans le TP2, la résolution exacte d'un problème d'optimisation combinatoire difficile peut demander un temps d'exécution très important. Dans un contexte opérationnel, il y a en pratique un temps limite à ne pas dépasser avant de fournir une solution aux décideurs. Si nous ne disposons pas d'une méthode exacte permettant de fournir une solution optimale en un temps raisonnable, nous nous tournerons alors vers des méthodes approchées qui fournirons une solution admissible si possible de bonne qualité en un temps court. Nous considérons dans la section 4 une variante de l'heuristique de Clark et Wright.

3 Résolution exacte

Si 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. On notera $\mathcal{S}$ l'ensemble des sous-ensembles de clients qui peuvent être regroupés dans une même tournée. Plus formellement, $\mathcal{S}$ est défini par

$$\mathcal{S} := \left\{ S \subset \{2, \dots, n\} \mid \sum_{j \in S} d_j \leq Ca \right\}.$$

Ensuite, pour chaque regroupement de clients $S \in \mathcal{S}$, il reste à déterminer la plus courte tournée passant une fois par chaque client, en partant du dépôt avant d'y revenir, tout en minimisant la distance. Il s'agit ici à nouveau d'un problème de voyageur de commerce, mais cette fois sur un ensemble restreint de lieux : $\{1\} \cup S$. Pour illustrer l'ensemble $\mathcal{S}$, on considère un exemple didactique.

Exemple : On suppose que $n = 6$, $Ca = 10$, $d_2 = d_4 = d_6 = 2$ et $d_3 = d_5 = 4$. 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	$\{2\}$	$\{2,3\}$	$\{2,3,4\}$	$\{2,3,4,6\}$	$\{2,3,5\}$	$\{2,3,6\}$	$\{2,4\}$	$\{2,4,5\}$	$\{2,4,5,6\}$
Longueur tour	668	714	730	803	1208	787	685	1179	1209
Indice	10	11	12	13	14	15	16	17	18
Regroupement	$\{2,4,6\}$	$\{2,5\}$	$\{2,5,6\}$	$\{2,6\}$	$\{3\}$	$\{3,4\}$	$\{3,4,5\}$	$\{3,4,6\}$	$\{3,5\}$
Longueur tour	758	1162	1192	741	524	541	1065	747	1056
Indice	19	20	21	22	23	24	25	26	27
Regroupement	$\{3,5,6\}$	$\{3,6\}$	$\{4\}$	$\{4,5\}$	$\{4,5,6\}$	$\{4,6\}$	$\{5\}$	$\{5,6\}$	$\{6\}$
Longueur tour	1195	744	496	1020	1153	702	554	1055	604

Connaissant l'ensemble $\mathcal{S}$ ainsi que la longueur de la plus plus courte tournée pour chaque regroupement de l'ensemble $\mathcal{S}$, 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. Pour cela, nous allons finalement poser un modèle. Tout d'abord, nous associons une variable de décision à chaque tournée de l'ensemble $\mathcal{S}$.

$$x_j = \begin{cases} 1 & \text{si on choisit la tournée indicée } j, \\ 0 & \text{sinon,} \end{cases}$$

où $j \in \{1, \ldots, |\mathcal{S}|\}$. Dans le cas de l'exemple, il y a donc 27 variables. En notant l_i la longueur de la plus courte tournée visitant chaque client du regroupement j, la fonction objectif (à minimiser) s'écrit simplement

$$z = \sum_{j=1}^{|\mathcal{S}|} l_j x_j.$$

Il reste à écrire les contraintes garantissant que chaque client est visité une seule fois. Pour cela, nous écrirons une contrainte pour chaque client $i \in \{2, \ldots, n\}$. Cette contrainte demande d'identifier le sous-ensemble $S_i \subset \{1, \ldots, |\mathcal{S}|\}$ d'indices de tournées contenant le client i. Nous avons donc les contraintes suivantes.

$$\sum_{j \in S_i} x_j = 1, i \in \{2, \ldots, n\}.$$

Dans le cas de l'exemple, nous avons $S_2 = \{1, \ldots, 13\}$, $S_3 = \{2, 3, 4, 5, 6, 14, 15, 16, 17, 18, 19, 20\}$...

Nous obtenons finalement un problème d'optimisation combinatoire appelé problème de *partitionnement d'ensemble*. Il ne reste plus qu'à résoudre ce problème pour achever la résolution.

Pour l'exemple, nous obtenons les résultats suivants :
— la valeur optimale pour la fonction objectif est 1357,
— le première tournée ($x_{25} = 1$) ne visite que le client 5 pour une longueur de 554,
— la seconde tournée ($x_4 = 1$) visite dans l'ordre les clients 6, 2, 4 et 3 pour une longueur de 803.

Nous résumons la méthode :

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.

4 Résolution approchée

La méthode de Clark et Wright est initialisée par une solution composée des tournées suivantes : $1 \to i \to 1$ pour $i \in \{2, \ldots, n\}$. Cette solution est nécessairement admissible puisque nous faisons l'hypothèse qu'aucun client n'a une demande supérieure à la capacité du véhicule. La longueur totale associée à cette solution est $\sum_{i=2}^{n} c_{1i} + c_{i1}$.

Le principe de la méthode est ensuite d'améliorer itérativement la solution en fusionnant des tournées. On suppose par exemple que l'on souhaite que i et j se suivent dans une même tournée. En partant de deux des tournées initiales $1 \to i \to 1$ et $1 \to j \to 1$, on obtient la tournée $1 \to i \to j \to 1$. La longueur de cette tournée fusionnée est $c_{1i} + c_{ij} + c_{j1}$, et la somme de la longueur des tournées initiales est $c_{1i} + c_{i1} + c_{1j} + c_{j1}$. Le gain réalisé avec cette fusion est donc $g_{ij} = c_{i1} + c_{1j} - c_{ij}$. Si g_{ij} est positif et que la demande $d_i + d_j$ des clients i et j ne dépasse pas la capacité Ca du véhicule, il est alors intéressant de réaliser cette fusion.

Bien entendu, les fusions pourront ensuite concerner des tournées passant par plusieurs clients. Nous pourrons fusionner de la même façon des tournées du type $1 \to \ldots \to i \to 1$ et $1 \to j \to \ldots \to 1$, pour obtenir une tournée $1 \to \ldots \to i \to j \to \ldots \to 1$. Le gain G_{ij} restera alors $c_{i1} + c_{1j} - c_{ij}$.

Par conséquent, suite à la construction de la solution initiale, une matrice des gains G définie par les gains g_{ij} sera calculée pour $i, j \in \{2, \ldots, n\}$ (et $i \neq j$). Dans le cas de l'exemple, nous aurons alors la matrice suivante.

	2	3	4	5	6
2	×	478	479	60	531
3		×	479	22	384
4			×	30	398
5				×	103

Nous tenons ici compte du fait que le distancier de l'exemple est symétrique. Il en sera d'ailleurs de même pour toutes les instances numériques considérées dans ce projet. Dans ce cas, la matrice des gains G est aussi symétrique et seule la partie supérieure de la matrice nous est alors utile. La symétrie du distancier aura aussi un impact sur les fusions possibles.

Les valeurs de la matrice des gains sont ensuite triées pour être parcourues de manière décroissante. Une itération de l'algorithme consiste alors à considérer une valeur de la matrice des gains définie par un couple de clients (i, j) et de réaliser la fusion des tournées contenant i et j, à la condition bien entendu que cette fusion soit possible.

Sur l'exemple, nous obtenons les itérations suivantes.

Itération	Paire	fusion possible ?	Si oui, Tournée obtenue
1	(2,6)	oui (demande = 4)	$1 \to 2 \to 6 \to 1$
2	(2,4)	oui (demande = 6)	$1 \to 4 \to 2 \to 6 \to 1$
3	(3,4)	oui (demande = 10)	$1 \to 3 \to 4 \to 2 \to 6 \to 1$
4	(2,3)	non (2 est à l'intérieur d'une tournée)	×
5	(4,6)	non (4 est à l'intérieur d'une tournée)	×
6	(3,6)	non (3 et 6 sont dans la même tournée)	×
7	(5,6)	non (demande = 14)	×
8	(2,5)	non (2 est à l'intérieur d'une tournée)	×
9	(4,5)	non (4 est à l'intérieur d'une tournée)	×
10	(3,5)	non (demande = 14)	×

Au final, nous obtenons sur cet exemple les mêmes tournées que pour la résolution exacte, la tournée $1 \to 5 \to 1$ n'ayant pas été fusionnée avec une autre.

Plusieurs remarques peuvent être faites suite à cet exemple. Tout d'abord, du fait du caractère symétrique du distancier, la paire (2,4) et la paire (4,2) sont la même. Cela justifie donc la fusion réalisée dans la seconde itération. De manière générale, une fusion entre deux tournées peut être réalisée si les deux villes de la paire se trouvent chacune juste avant ou juste après le dépôt dans la tournée. Pour réaliser cette fusion, Il sera éventuellement utile d'inverser l'ordre de visite d'une tournée. Ce sera par exemple le cas si on souhaite fusionner une tournée du type $1 \to i \to \ldots \to 1$ et une tournée du type $1 \to j \to \ldots \to 1$. En inversant l'ordre de visite de la première tournée (ce qui ne change rien à la distance parcourue dans le cas symétrique), on obtient $1 \to \ldots \to i \to 1$ et la fusion est ensuite immédiate.

Ensuite, le déroulement de l'exemple est (volontairement) maladroit. Les itérations 4, 5, 6, 8 et 9 auraient en effet pu être évitées. Dès la fin de l'itération 2, on peut supprimer à l'avance la paire (4,6) puisque les lieux 4 et 6 sont maintenant dans la même paire. De plus, le lieu 2 est maintenant définitivement à l'intérieur d'une tournée, ce qui signifie qu'il ne se trouve pas juste avant ou juste après le dépôt dans une tournée. On peut par conséquent supprimer toutes les paires incluant le lieu 2. De même, à l'issue de l'itération 3, la paire (3,6) peut être supprimée ainsi que toutes les paires contenant le lieu 4. En tenant de ces suppressions, vérifier que la fusion de deux tournées est possible demande juste de vérifier que la demande des clients ne dépassera pas la capacité du véhicule.

5 Travail à effectuer

Une implémentation des méthodes présentées dans les sections 3 et 4 est attendue. Ces méthodes devront être implémentée en utilisant Julia/JuMP. Cette implémentation permettra ensuite d'observer le fonctionnement des méthodes, de pointer leurs faiblesses, et éventuellement de proposer des améliorations.

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

— Le fichier `Projet_Base.jl` contient deux fonctions utiles pour démarrer le projet : une fonction de lecture des données du problème, une fonction de résolution du problème du voyageur de commerce.

— Deux dossiers d'instances numériques `A` et `B`.

Remarque : la méthode utilisée dans le package `TravelingSalesmanExact` de Julia est très similaire à ce qui avait éte proposé dans le projet proposé en 2018. Pour les plus curieux, ce sujet a également été déposé sur madoc.

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 les demandes 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 vendredi 2 avril à 23h59. Le code source ainsi que le rapport devront être déposés sur madoc. Le rapport devra contenir :
- Une description (et une justification) des structures de données choisies pour l'implémentation,
- Une description de l'algorithme d'énumération des regroupements possibles des clients,
- 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 partionnement d'ensemble dans le cas de la méthode exacte, on pourra aussi évaluer la qualité de la solution admissible obtenue par la méthode approchée...
- 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 !