TP n^o 4
Méthode de résolution exacte d'un problème de tournée de véhicule avec capacités et profits

1 BladeFlyer II : conquest of water

Au lendemain de l'élection de son président [1], Trumpland est tombé dans le chaos suite au violent rechauffement climatique qui a frappé le pays ; une guerre interne y règne depuis maintenant plusieurs années où la clef de survie est l'eau potable. La situation est post-apocalyptique ; exaspérées par la situation de crise, les populations se sont révoltées, vivent en communautés autonomes, en essayant de maintenir un semblant d'ordre à l'intérieur de chaque communauté, tandis qu'entre communautés la conquête de l'eau fait rage.

La collecte de l'eau est devenue une activité mortelle ; les pertes humaines étaient incalculables avant que la mission ne soit confiée à des drones robustes et rapides. Chaque communauté de survivants couvre un territoire de points de pompage et dispose d'un drone. Lors des épisodes de pluie, les niveaux en eau aux points de pompage sont remis au maximum et une course contre la montre s'engage dès la fin de la pluie : chaque communauté envoie son unique drone pour récupérer le maximum d'eau avant que les points de pompage soient mis à sec par d'autres communautés.

On dispose d'un territoire sur lequel sont positionnés des points de pompage. Les coordonnées dans le plan des points de pompage sont connues a priori ; elles respectent l'inégalité triangulaire. Chaque point de pompage contient une réserve d'eau, ici aussi connue a priori.

Le drone est localisé à sa base où se situe une citerne de grande capacité (position de la base et capacité de la citerne connues). Il a une autonomie de vol exprimée en unité de distance (km par exemple). Il dispose d'un réservoir embarqué d'une capacité maximale donnée, qui lui permet de transporter de l'eau. On suppose que ses performances en vol ne sont pas altérées par la quantité d'eau transportée. Et on suppose que la contenance à vide du réservoir est au moins égale au volume d'eau disponible à chaque point de pompage. Quand un drone repasse par sa base, son réservoir est vidé dans la citerne et son autonomie de vol est rétablie au maximum de sa capacité (opérations de changement/recharge de batteries que l'on suppose de temps négligeable).

La première question (situation sans la concurrence d'une autre communauté) s'exprime comme suit. Quelles sont les tournées que doit réaliser le drone afin de rapporter toute l'eau disponible à la citerne sachant que, un point de pompage est visité une seule fois (donc soit toute l'eau est emportée compte tenu de la capacité résiduelle du drone, soit la totalié est laissée sur place si le drone ne peut pas tout emporter en un seule visite ; autrement dit, la quantité d'eau disponible en un point n'est pas divisible), de manière à minimiser la distance totale parcourue par le drone au cours de sa mission.

On suppose à présent qu'une seconde communauté va procéder dans les mêmes conditions, avec son drone (appelons les drones par A, le vôtre, et B, l'adverse), ayant les mêmes caractéristiques. Les deux drones ne connaissent pas leurs plans de vol réciproques. Et donc si un drone visite un point de pompage, le second s'y présentera éventuellement si le point est dans sa tournée mais inutilement car toute l'eau sera déjà emportée. On suppose que les drones ne se gênent pas dans leurs mouvements (pas de collision).

La seconde question (situation de mise en concurrence des deux communautés) reprend les éléments de la première question avec en plus ces nouveaux éléments et la volonté que le drone A affiche le meilleurs rendement (score) que le drone B en fin de journée.

1. Ce contexte imaginaire est inspiré de problèmes d'optimisation réels, de technologies existantes, d'expression de besoins crédibles, de l'ambiance cyberpunk du film "Blade Runner" (1982, Ridley Scott) et de Mad Max II, film d'anticipation dystopique (1981, George Miller), plus une pointe de "Stratego" (1947), jeu de société de stratégie et de bluff.

2 Définition du problème

Nous commençons par définir les données du problème. Nous notons l'ensemble des lieux $\{0, \ldots, n\}$ où n est le nombre de points de pompage, l'ensemble des points de pompage est indicé de 1 à n et l'indice 0 représente la base. L'utilisation de ces indices nous permet de définir le distancier d'un lieu i à un lieu j par c_{ij} pour $i, j \in \{0, \ldots, n\}$. Nous notons d_i la quantité d'eau disponible en chaque point de pompage $i \in \{1, \ldots, n\}$ et Ca la capacité du drone. Nous supposerons qu'aucun point de pompage n'a une quantité supérieure à la capacité du drone. La capacité d'autonomie de vol du drone n'est pas prise en compte en première approche (mais l'intégrer ne pose aucune difficulté supplémentaire).

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 drone est suffisante pour emporter toute l'eau à tous les points de pompages, il n'y a alors qu'une seule tournée à effectuer partant de la base, visitant une seule fois chaque point de pompage, avant un retour à la base, 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 emporter toute l'eau des points de pompage. Une première question à se poser, concerne les sous-ensembles de points de pompage qu'il (n')est (pas) possible de regrouper dans une même livraison. Si on note S un sous-ensemble de points de pompage et que $\sum_{i \in S} d_i > Ca$, on ne pourra pas inclure l'ensemble des points de pompage de S dans une même tournée. Ensuite, pour chaque sous-ensemble de points de pompage qu'il est possible de regrouper, il reste à déterminer la tournée passant une fois par chaque point de pompage, en partant de la base avant d'y revenir, tout en minimisant la distance. Une fois ces informations connues, nous pourrons 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 points de pompage 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 points de pompage 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 $i \in \{1, \ldots, 27\}$. En notant l_i la longueur de la plus courte tournée visitant chaque point de pompage du regroupement $i \in \{1, \ldots, 27\}$, 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 point de pompage est visité une seule fois. Pour le point de pompage 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,
- le première tournée ($x_{25} = 1$) ne visite que le point de pompage 4 pour une longueur de 554,
- la seconde tournée ($x_4 = 1$) visite dans l'ordre les points de pompage 5, 1, 3 et 2 pour une longueur de 803.

Nous résumons la méthode appliquée à cet exemple :

1. Énumérer les regroupements possibles de points de pompage,
2. Pour chacun de ces regroupements, déterminer la plus courte tournée partant de la base, visitant une fois chaque point de pompage, et revenant à la base,
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 points de pompage.

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 drone,
- La troisième ligne indique la quantité d'eau à relever aux points de pompage,
- 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 11 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 points de pompage,
- 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 points de pompage, 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...

– 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 !

– Bonus : traiter la seconde question (situation de concurrence). Pour évaluer la qualité de votre solution et donc de votre stratégie, on vous communiquera (au moins) un jeu de données qui correspond au résultat de la stratégie de l'adversaire (en temps simulé).
L'idéal étant de faire jouer en temps réel les deux solutions pour découvrir en live qui est le gagnant. Ce n'est pas très compliqué mais éloigné de l'objet de ce cours (avis aux amateurs cependant, un bonus++ sera bien évidemment accordé).