Rapport du projet du TOP

(Chemanack Thierry & Glandier Quentin)

Introduction

Le **Team Orienteering Problem** **(TOP)** et le **Traveling Salesman Problem** **(TSP)** sont deux challenges fondamentaux en optimisation combinatoire, largement étudiés dans le domaine de la recherche opérationnelle. Ces problèmes appartiennent à la classe des problèmes de routage, et ils trouvent leur application dans divers domaines tels que la logistique, la distribution, le transport, et la planification d'itinéraires (Google Maps est une application par excellence de ces problèmes).
Le **TSP** consiste à trouver le parcours le plus court qui permet à un voyageur de visiter chaque ville d'une liste donnée exactement une fois et de revenir à sa ville d'origine.Par ailleurs, le **TOP** représente une variante du PVC où plusieurs voyageurs, véhicules sont impliqués dans la recherche des parcours optimaux tout en maximisant la somme des récompenses associées aux visites de différentes villes en respectant des contraintes (capacité, temps,..).
De nombreuses approches et heuristiques ont été développées pour les résoudre efficacement. Nous avons eu à utiliser et à adapter **l'algorithme de Clark & Wright** afin de pouvoir résoudre ce problème de TOP. Par ailleurs, dans le but de voir effectivement que le TOP est un problème NP-difficile, Nous avons fait un algorithme qui permet de déterminer une solution exacte. Pour ce fait, nous avons utilisé la librairie **Gurobi de Python**.

I.Algorithme basée sur une formulation linéaire

Cet algorithme correspond au fichier **Algo_formulation_lineaire**, il est grandement basé sur une modélisation ou formulation mathématique et linéaire du TOP(pour ce faire, nous nous sommes servis du fichier branch-and-cut comme fichier de référence et donc un screenshot est donné plus-bas). Après exécution de l'algorithme avec une variation des variables décisionnelles, nous nous sommes effectivement rendu compte que le TOP est un problème NP-difficile. En effet, le nombre de véhicules ou alors le nombre de clients au-dessus d'un certain seuil affectent énormément le temps d'exécution de l'algorithme. Avec **Gurobi**, ici, qui peut être considéré comme solveur, au-dessus de **17 clients et 3 véhicules**, le temps de résolution de l'algorithme se compte **en minutes**. Au-delà de **25**

clients, il devient impossible pour une simple machine de pouvoir exécuter cet algorithme; tant il est ***excessivement gourmand en mémoire, en temps et en espace***. On a eu à exécuter cet algorithme avec plusieurs instances. Avec une instance de *11 clients et 2 véhicules*, l'exécution a mis une **dizaine de secondes**, avec une instance de *17 clients et 3 véhicules*, le temps d'exécution a été un peu plus d'une cinquantaine de secondes. Les résultats affichés par Gurobi sont très explicites et présentent clairement les temps d'exécution et comment ceux-ci sont répartis sur les grandes parties de l' algorithme.
Réaliser cet algorithme basé sur une formulation linéaire nous revenait à faire en quelque de la **programmation par contrainte** **PPC**, avec ***des variables, leurs domaines, une fonction objective et des contraintes sur les variables***.
Après de nombreuses recherches, nous nous sommes appuyés sur la formulation linéaire suivante:

1 Formulation linéaire

Nous utilisons V^-, V^d, V^a et V pour désigner respectivement les ensembles $\{1, ..., n\}$, $V^- \cup \{d\}$, $V^- \cup \{a\}$ et $V^- \cup \{d, a\}$. Le TOP peut être formulé en programmation linéaire en nombres entiers (PLNE) avec les variables de décision y_{ir} et x_{ijr} : $y_{ir} = 1$ si le client i est servi par le véhicule r et 0 sinon ; $x_{ijr} = 1$ si l'arc (i, j) est utilisé par le véhicule r pour visiter le client i puis le client j et 0 sinon.

$$\max \sum_{i \in V^-} \sum_{r \in F} y_{ir} P_i \tag{1}$$

$$\sum_{r \in F} y_{ir} \leq 1 \quad \forall i \in V^- \tag{2}$$

[*] Ce travail est partiellement soutenu par le Conseil Régional de Picardie et le Fonds Européen de Développement Régional (FEDER), dans le cadre du projet PRIMA.

$$\sum_{j \in V^a} x_{djr} = \sum_{j \in V^d} x_{jar} = 1 \qquad \forall r \in F \tag{3}$$

$$\sum_{i \in V^a \setminus \{k\}} x_{kir} = \sum_{j \in V^d \setminus \{k\}} x_{jkr} = y_{kr} \qquad \forall k \in V^-, \forall r \in F \tag{4}$$

$$\sum_{i \in V^d} \sum_{j \in V^a \setminus \{i\}} C_{ij} x_{ijr} \leq L \qquad \forall r \in F \tag{5}$$

$$\sum_{(i,j) \in U \times U} x_{ijr} \leq |U| - 1 \qquad \forall U \subseteq V^-, |U| \geq 2, \forall r \in F \tag{6}$$

$$x_{ijr} \in \{0, 1\} \qquad \forall i \in V, \forall j \in V, \forall r \in F \tag{7}$$

$$y_{ir} \in \{0, 1\} \qquad \forall i \in V^-, \forall r \in F$$

La fonction objectif (1) consiste à maximiser la somme des profits collectés. Les contraintes (2) assurent que chaque client est servi au plus par un véhicule, alors que les contraintes (3) et (4) garantissent la connectivité de chaque tournée. La limite imposée sur le temps de trajet est assurée par les contraintes (5), les sous-tours sont interdits par les contraintes (6) et les contraintes (7) traduisent celles d'intégrité.

Concernant la complexité de cette algorithme, elle est **exponentielle**

II.Algorithme avec Clark & Wright

2.Explication du choix de l’Heuristique de Clark & Wright

L’heuristique de Clark & Wright est l’une des heuristiques les plus utilisées dans les problèmes de TOP. Ceci dit, il y a plusieurs raisons et arguments qui font d’elle l’une des incontournables. Voici quelques raisons pour lesquelles cette heuristique devraient être choisie:

-Simplicité : L'algorithme de Clark and Wright est relativement simple à comprendre et à mettre en œuvre par rapport à certaines autres méthodes plus complexes.En effet, les notions de marguerites, d'économies et de fusion sont facilement compréhensibles. Cela rend cette heuristique accessible, en particulier pour des problèmes de taille moyenne où des solutions exactes peuvent être coûteuses en termes de ressources computationnelles.

-Efficacité : Étant une heuristique, Clark and Wright ne garantit pas une solution optimale, mais elle génère souvent des solutions de qualité en un temps raisonnable. Pour de nombreux problèmes de routage de véhicules réalistes, obtenir une solution rapidement peut être crucial.

-Adaptabilité : L'approche de Clark and Wright peut être adaptée pour traiter différentes variantes du PVC, y compris le TOP. Elle est flexible et peut être ajustée pour prendre en compte des contraintes spécifiques ou des caractéristiques particulières d'un problème donné. Dans notre algorithme, nous avons adapté les profits à la notion d’économies.

-Évolutivité : L'algorithme fonctionne bien pour des instances de taille moyenne à grande, ce qui le rend approprié pour des applications réelles où le nombre de villes ou de clients peut être substantiel.

2.Principe et fonctionnement de l’algorithme

Pour résoudre ce problème de TOP , nous nous sommes servis d’une adaptation de l’heuristique de Clark & Wright, nous avons adapté **les profits** à ce qui est censé être les notions d’économies (On aurait pu le faire avec le **profit(client)/ (distance(depart, client)+distance(client,arrivée))**).
Nous avons commencé premièrement par créer toutes les marguerites possibles, c'est-à-dire le triplet (départ, client, arrivée).
Par la suite, nous avons trié les marguerites par profits décroissants, et dans la même lancée pour les différents clients.

Après avoir trié les clients en fonction des profits importants, on s'est attelé à ce que le parcours ou la visite des clients soit suivant cet ordre. Ceci dans le but de pouvoir recueillir le maximum de profits possibles.
La contrainte majeure dans notre TOP est le temps maximal de parcours des véhicules. Ayant la liste des clients triés. La contrainte ou condition pour faire une fusion (ou ajout d'un client à la solution) d'un client i (visite d'un client par un véhicule), est que ***la somme du temps parcouru du départ au client i plus le temps de parcours du client i au client i+1 plus le temps de parcours de la distance max entre les clients et l'arrivée doit être inférieure ou égale au temps Tmax donné dans le problème***.
Le parcours des clients par les véhicules est ***séquentiel***, et donc les pour les clients n'ayant pas pu être visités par le premier véhicule, ils pourront potentiellement être parcouru par les véhicules suivants(tant que la condition est vérifiée).

En ce qui concerne la complexité, elle est ***polynomiale.***

Conclusion

Le TOP représente un défi stimulant dans le domaine de la recherche opérationnelle, offrant des implications pratiques significatives pour la planification des itinéraires dans divers contextes tels que la logistique, le transport et la gestion des ressources. En ayant travaillé sur ce problème, les acquis et découvertes à la fin de ce projet sont nombreux et ceci nous motive encore plus à nous y immerger et contribuer à sa résolution. Par exemple, on a aisément compris comment des applications telles que Google Maps et Plans fonctionnent. Le TOP demeure un domaine actif de recherche, et de nouvelles approches émergent constamment pour relever les défis liés à la complexité algorithmique et à la résolution optimale de ce problème captivant.