

## Module MAOA: Modèles et Applications en Ordonnancement et optimisation combinatoire

### Projet

#### *“Tournées de véhicules et Réseaux de télécommunications”*

Le projet qui vous est proposé dans ce module MAOA présente un problème d’Optimisation Combinatoire en Recherche Opérationnelle. Son objectif est de traiter un sujet issu d’un problème industriel jusqu’à la réalisation d’un logiciel de résolution en utilisant différentes méthodes approchées et exactes. Il s’agit ainsi de développer un logiciel permettant de résoudre des instances du problème en tenant compte de sa vitesse d’exécution et surtout sur la qualité de la solution obtenue.

Nous nous intéressons ici au problème d’Optimisation Combinatoire connu dans la littérature scientifique sous le nom de CVRP “Constrained Vehicle Routing Problem”, qui est très proche de la conception de réseaux de télécommunication dit “en boucle” de la technologie SONET.

Une première partie va s’intéresser au problème classique de tournées de véhicules. Une deuxième partie s’intéressera à des problèmes de conception de réseaux SONET sous des contraintes particulières: **La partie 2 est annulée compte-tenu du semestre troublé par les transports difficiles**

## 1 Le problème de tournée de véhicules (VRP)

Soit un graphe orienté complet  $G = (\mathcal{N}, A)$  possédant  $n+1$  sommets notés  $\mathcal{N} = \{0, 1, \dots, n\}$ : on appelle le sommet 0 le dépôt et les sommets  $\mathcal{N}_C = \mathcal{N} \setminus \{0\}$  les revendeurs (ou clients). Les arcs sont donc l’ensemble  $A \subseteq \{(i, j) \mid i, j \in \mathcal{N}, i \neq j\}$ . A chaque client  $i \in \mathcal{N}_C$ , on associe une valeur réelle positive  $d_i$  appelée *demande* du sommet  $i$ . A chaque arc  $(i, j) \in A$ , on associe un *coût*  $c_{ij}$  strictement positif correspondant au coût de transport du sommet  $i$  au sommet  $j$ . Le *problème de tournées de véhicules* (Vehicle Routing problem ou VRP) consiste à déterminer des tournées de véhicules partant du dépôt et passant livrer la quantité demandée par chacun des revendeurs.

On considère une flotte  $K = \{1, \dots, m\}$  de  $m$  véhicules identiques qui ont chacun une capacité maximale  $Q$ : un véhicule peut transporter au plus  $Q$  quantité de produits à partir du dépôt. On appelle *tournée* un circuit non vide de  $G$  passant par le sommet 0 et dont la somme des demandes des sommets du cycle est inférieure à  $Q$ . Comme le graphe  $G$  est complet, on peut écrire une tournée  $T$  comme une séquence  $(i_0, i_1, \dots, i_p)$  de sommets où  $i_0 = 0$  et  $\sum_{l=1}^p d_{i_l} \leq Q$ .

Le problème VRP consiste à déterminer un ensemble d’au plus  $m$  tournées  $(T_1, \dots, T_k)$ ,  $k \leq m$ , telles que, chaque revendeur soit servi par exactement une tournée et que la somme totale des coûts des arcs utilisés est minimum.

On suppose qu’un revendeur peut être servi par une unique tournée, *i.e.*  $d_i \leq Q$  pour tout  $i \in \mathcal{N}_C$ . On peut noter que dans une solution, les tournées n’ont aucun client en commun en dehors du sommet 0.

**Important:** il peut être très utile de considérer le problème à exactement  $m$  véhicules qui est souvent plus simple que le problème à au plus  $m$  véhicules.

## 1.1 Métaheuristique pour le VRP

- Heuristique initiale:

Le problème de partition des clients en  $m$  tournées (réalisables) est déjà un problème difficile. Il s'agit du problème du Bin-Packing qui revient à déterminer s'il existe une répartition des clients en  $m$  boîtes satisfaisant chacune la contrainte de sac-à-dos de la capacité d'un véhicule.

Remarquons que si cette première phase de répartition est obtenue, une deuxième phase revient à résoudre  $m$  problèmes du voyageur de commerce (TSP) pour lequel des techniques gloutonnes sont bien connues (le plus proche voisin par exemple).

Pour résoudre l'étape de partition des clients, on peut lister trois techniques:

- tenter de résoudre exactement ce problème (donc sans prendre en compte les distances) par PLNE: cela n'est possible que pour des petite tailles.
- tenter de résoudre heuristiquement ce problème pour exactement  $m$  tournées. Il faut alors une méthode méta-heuristique dédiée.
- soit relaxer la contrainte du " $m$  tournées" en résolvant un problème avec un nombre potentiellement plus grand de tournées; puis en confiant l'objectif de réduire le nombre de tournées à une étape ultérieure.

Mais il existe de nombreuses heuristiques efficaces pour ce problème célèbres, n'hésitez pas à parcourir la littérature des heuristiques pour le CVRP... ou à en inventer vous-même.

- Méthode itérative:

A l'issue d'une étape initiale, on obtient une solution réalisable ou alors une solution utilisant plus de  $m$  tournées (chacune réalisable).

Une métaheuristique itérative peut alors être utilisée sur la base de plusieurs voisinages:

- les voisinages classiques du TSP (2-opt) pour améliorer chaque tournée indépendamment.
- la possibilité pour un client de changer de tournées
- supprimer une tournée vide
- éventuellement ajouter une tournée vide si le nombre est inférieur à  $m$

Si le nombre de tournées de la solution initiale est supérieur à  $m$ , il est possible d'utiliser dans un premier temps une fonction objective artificielle pour forcer à réduire le nombre de tournées.

D'autres méta-heuristiques sont bien entendu possible (algorithme génétique, etc).

## 1.2 Formulations PLNE pour le VRP

Il existe plusieurs formulations PLNE classiques pour ce problème, en voici deux qui nous intéressent pour ce projet

- une formulation compacte basée sur les inégalités de Miller-Tucker-Zemlin (MTZ)

- une formulation à nombre exponentiel de contraintes, basée sur les coupes

D'autres formulations existent, comme celles indexées sur les véhicules ou des formulations à nombre exponentiel de variables qui se résolvent par génération de colonnes: nous ne les explorerons pas ici.

**Attention:** si les instances ont des coûts symétriques, il peut être intéressant de prendre une formulation utilisant des variables non-orientées. Or la section suivante présente des versions orientées!

### 1.3 Formulation compacte

- *Formulation MTZ*

On considère des variables binaires  $x_{ij}$  si un véhicule utilise l'arc  $(i, j) \in A$  et des variables réelles  $w_i$  associées à chaque sommet  $i \in \mathcal{N}_C$ .

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \sum_{j=1}^n x_{0j} \leq m \end{aligned} \tag{1}$$

$$\sum_{i=1}^n x_{i0} \leq m \tag{2}$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \forall i \in \mathcal{N}_C, \tag{3}$$

$$\sum_{i=0}^n x_{ij} = 1 \quad \forall j \in \mathcal{N}_C, \tag{4}$$

$$w_i - w_j \geq d_i - (Q + d_i)(1 - x_{ij}) \quad \forall (i, j) \in A, i \neq 0, j \neq 0 \tag{5}$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \in A \tag{6}$$

$$0 \leq w_i \leq Q \quad \forall i \in \mathcal{N}_C$$

$$w_i \in \mathbb{R} \quad \forall i \in \mathcal{N}_C$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A.$$

Les contraintes de degré (1)-(4) imposent qu'il y ait exactement un arc entrant et un arc sortant pour chaque revendeurs, alors que le dépôt peut avoir jusqu'à  $m$  arcs entrants et  $m$  arcs sortants. Ces inégalités suffisent à décrire des tournées partant du dépôt, malheureusement il peut y avoir des sous-tournées (comme dans le problème du voyageur de commerce) qui ne passe pas par le dépôt. Pour briser les sous-tournées, on ajoute ici les contraintes MTZ (5) qui indiquent qu'à chaque arc  $(i, j) \in A$  avec  $i, j \neq 0$ , pris dans la solution, le sommet  $j$  a une valeur  $w_j$  inférieure à  $w_i - d_i$  (si  $x_{ij} = 1$ , alors  $w_j \leq w_i - d_i$ ). Ainsi, tout circuit qui ne passe pas le sommet 0 est impossible. En revanche, si l'arc n'est pas pris, l'inégalité reste bien valide (si  $x_{ij} = 0$ , alors  $w_j - w_i \leq Q$ ). Les contraintes MTZ ont une seconde utilité pour la formulation: elles permettent de limiter les tournées à charger au plus  $Q$  unités de produit

par véhicule. En effet, on peut voir que  $w_i$  est une borne supérieure sur la quantité chargée dans le véhicule après le passage d'un véhicule au sommet  $i$ : sur une tournée  $t = (i_0, i_1, \dots, i_p)$  si on somme les contraintes MTZ des arcs, on obtient  $w_{i_1} - w_{i_p} \geq \sum_{l=1}^p d_{i_l}$ , donc si  $w_{i_p}$  est la charge du véhicule au retour de sa tournée,  $w_{i_1}$  est la charge de la tournée, or  $w_i \leq Q$  pour chaque revendeur  $i$ .

L'inégalité (6) n'est pas nécessaire à la formulation. Par contre, elle est bien valide car les deux variables ne peuvent être toutes les deux à un en même temps. On peut également déduire sa validité à partir de l'inégalité (5) par somme de Chvátal: en prenant  $d = \min(d_i, d_j)$ , et en sommant les inégalités MTZ pour  $i$  et pour  $j$ , on peut déduire que  $x_{ij} + x_{ji} \leq \frac{2Q}{Q+d}$  or par hypothèse sur les  $d_i$ , on sait que  $2Q \leq Q + d$ : par arrondi de Gomory, on obtient alors l'inégalité (6). Vous pouvez remarquer expérimentalement leur grande utilité.

- *Autre formulations compactes*

Vous pouvez trouver dans la littérature des formulations basées sur des flots, à l'image de celle vue pour le TSP en cours.

## 1.4 Formulation à nombre exponentiel de contraintes

Il y a trois pistes possibles pour construire un PLNE pour le CVRP:

- utiliser une formulation compacte (comme les MTZ)
- couper tous les points entiers en séparant exactement une famille d'inégalités (en nombre exponentiel) qui suffise à briser tous les sous-tours (utiliser le callback lazy)
- choisir l'option a) ou l'option b) puis la renforcer en coupant des points fractionnaires par des séparations heuristiques (en utilisant le callback usercut)

Dans le cas b), il est à noter qu'il faut séparer exactement toutes les inégalités permettant de briser les sous-tours. Or on va voir par la suite que de telles inégalités ont une séparation NP-difficile dans le cas général! Heureusement, cette séparation devient facile pour couper un point entier! Il y a aussi la possibilité d'utiliser uniquement des variables non-orientés!

Remarque: il existe aussi des formulations à nombre exponentiel de variables: mais ce n'est pas au programme de projet.

- *Inégalités de coupes-CVRP*

Les inégalités suivantes, dites de coupes-CVRP, permettent de briser tous les sous-tours.

$$\sum_{i \in W} \sum_{j \in \{0, 1, \dots, n\} \setminus W} x_{ij} \geq \left\lceil \frac{\sum_{i \in W} d_i}{Q} \right\rceil \quad \forall W \subset \{1, \dots, n\}, W \neq \emptyset, \quad (7)$$

Ces contraintes (7) sont dites *contraintes de coupes-CVRP* ou même de *coupes*. L'idée principale est la suivante: la demande totale d'un ensemble  $W$  de clients est  $\sum_{i \in W} d_i$ , si elle

est non nulle, le nombre de tournées nécessaires pour servir les clients de  $W$  est supérieur à  $\frac{\sum_{i \in W} d_i}{Q}$ . On peut montrer que cette inégalité est suffisante pour formuler le problème VRP.

Les inégalités de coupes ont un algorithme de séparation NP-difficile pour le cas général. Mais pour le cas entier, le problème est très simple. Il suffit de regarder s'il existe un cycle ne passant pas par 0: s'il n'en existe pas, aucune inégalité de coupes n'est violée par ce point entier; s'il existe un tel cycle  $W$ , on ajoute l'inégalité de coupes pour  $W = V(C)$ . Ainsi on peut utiliser ces inégalités dans une séparation ne coupant que les points entiers (en callback lazy); ceci donne une résolution exacte du problème.

- *Renforcement*

Les inégalités de coupes peuvent être également utilisées pour couper des points fractionnaires (en callback usercut): c'est même très efficace!

L'idée est la suivante: on sait efficacement les séparer par des méthodes approchées (taboo search) basées sur les inégalités où la borne est relaxée en  $\frac{\sum_{i \in W} d_i}{Q}$  pour lesquelles le problème de séparation est alors polynomial. Voir l'article [1].

Remarque: on aimerait trouver la valeur de droite la plus grande telle que l'inégalité reste valide: mais il n'est pas simple de déterminer une telle borne.

## 1.5 Travail à fournir sur la partie 1

A partir des instances de la librairie classique du CVRP (voir lien sur le site), vous devez:

- proposer une technique heuristique efficace pour résoudre des instances de toute taille
- proposer une formulation PLNE efficace pour résoudre des instances jusqu'à une taille de  $n$  clients.

La valeur  $n$  citée doit être suffisamment important pour pouvoir juger si votre heuristique est capable de fournir des solutions proches de l'optimum: pour cela vous proposerez des jeux d'essai de comparaison.

Une démarche possible pour la PLNE est la suivante:

- coder une formulation compacte MTZ pour avoir une base de référence pour les autres PLNE en variable orientée. Il est possible de rechercher une autre formulation compacte (type flot)
- coder la formulation avec les coupes-CVRP à la fois avec la séparation des points entiers et la séparation des points fractionnaires.

D'autres approches sont possibles: n'hésitez pas à en faire part aux encadrants.

Remarque: la librairie classique du CVRP vous fournit des valeurs optimales pour plusieurs instances. Le but de ce projet est d'apprendre à retrouver ces valeurs: vous pouvez néanmoins les utiliser au besoin dans votre travail.

## 2 Le problème de réseaux en boucle

La partie 2 est annulée compte-tenu du semestre troublé par les transports difficiles

## 3 Contenu du projet

### 3.1 Cadre du sujet

La difficulté de résolution d'un problème d'optimisation réaliste comme celui du VRP avec prise en compte des compétences intègre ainsi deux problèmes très classiques d'Optimisation Combinatoire en Recherche Opérationnelle: le problème de tournée de véhicules et celui d'affectation. Ce problème est apparu récemment dans la littérature scientifique et représente un challenge en RO de part sa difficulté à combiner les difficultés.

Dans ce sujet, il vous est demandé:

- **partie 1:** de traiter le problème CVRP classique dans le cadre de la description faite dans la section 1
- **partie 2:** [A VENIR]

Il est souhaité que l'ensemble des binômes aient ainsi une diversité importante dans les sujets et dans les approches considérées.

Le projet est ainsi divisé en deux parties. Pour chacune des parties, il est demandé de mettre en œuvre les trois programmes: résolution heuristique, résolution exacte et évaluation expérimentale.

### 3.2 Résolution heuristique

Même si le but de ce projet (ainsi que de l'UE MAOA) est une résolution exacte du problème, il est utile de commencer votre étude par la mise en place d'une heuristique ou d'une méta-heuristique. Cela vous permettra d'appréhender facilement le problème et ses instances, ainsi que d'avoir un élément de comparaison pour le reste du projet.

D'autre part, la nature NP-difficile des problèmes traités issus d'applications réelles rendent nécessaires la mise au point d'une telle approche heuristique.

Les objectifs sont:

- permettre de produire une première solution utile aux méthodes exactes,
- produire une solution quel que soit la taille de l'instance à traiter,
- la comparaison entre cette méthode et les bornes obtenues par les méthodes exactes vous permettra dans certains cas d'avoir une garantie expérimentale de vos solutions.

### 3.3 Résolution exacte et bornes par PLNE

Il vous est demandé de mettre en place au moins une méthode exacte **performante** ou un calcul de **bornes pertinentes**, en utilisant la programmation linéaire en nombres entiers (PLNE).

Les objectifs possibles sont:

- résoudre exactement des instances de tailles réduites,
- fournir des bornes pour le problème pour des tailles moyennes,
- utiliser les techniques d'arrondi à partir des solutions fractionnaires,

Les tailles réduites vous permettront également de vérifier si vos solutions heuristiques sont de bonne qualité.

Nous vous proposons, comme lors des séances de TME, d'utiliser l'outil CPLEX avec concert technology.

### 3.4 Evaluation expérimentale

Comme évoqué ci-dessus, une évaluation expérimentale est nécessaire pour valider vos différentes méthodes approchées et exactes, en utilisant des bornes pour obtenir des garanties expérimentales. La meilleure façon de convaincre de la qualité de votre travail est de pouvoir donner des statistiques (courbes, chiffres éloquentes,...) pour illustrer la performance de vos outils. N'hésitez pas à avoir recours à des tableurs ou des logiciels de tracé de courbes (gnuplot par exemple).

## 4 Instances

Voir les sources des instances et les sections de chacune des parties.

Il vous est demandé d'utiliser un ensemble d'instances de tailles et de dimensions suffisantes pour effectuer des tests expérimentaux. La nature de ces instances (particularités, corrélation,...) est aussi à examiner de près.

Il est aussi très utile d'avoir une visualisation de ces instances (voir document concernant les instances).

## 5 Rendu du projet

Le projet nécessite un travail personnel qui s'étale sur tout le semestre: il vous sera demandé:

- un rapport synthétique
- une production logicielle
- une validation expérimentale
- une présentation finale de votre projet devant vos camarades.

Le projet est à réaliser en binômes.

Il vous est demandé un travail régulier afin de pouvoir rendre le projet complet à temps: n'hésitez pas à discuter par mail avec les encadrants tout au long du projet sur des questions constructives ou des problèmes particuliers d'ordre théorique ou technique. Le sujet est assez libre pour laisser place à l'intuition et à l'initiative.

### 5.1 Evaluation du projet

Plusieurs facteurs seront pris en compte dans l'évaluation:

- Le but de ce projet est clairement d'aboutir à un produit fini: c'est-à-dire à un logiciel complet permettant de résoudre un problème de Recherche Opérationnelle allant de l'instance à une solution compréhensible. Pour cela, vous pouvez par exemple utiliser en entrée un fichier texte contenant les données du problème pratique et en sortie une visualisation compréhensible de la solution (graphique,...).
- Un des objectifs est d'obtenir un logiciel performant. Vous devez fournir une évaluation de

ses performances par rapport aux instances fournies et à des instances générées aléatoirement, en utilisant des mesures expérimentales.

- L'apport d'idées nouvelles (originalité), de recherche dans la littérature scientifique, de mise au point d'algorithmes fins,... sera bien évidemment pris en compte.

Le projet proposé est vaste: à chacun de trouver la partie et le développement qui convient à ses capacités informatiques et mathématiques.

## 5.2 Calendrier du projet

**Mercredi 9 octobre 2019:** début du projet.

**Mercredi 16 octobre 2019: séance de travail avec les encadrants:** Constitution des binômes

**Mercredi 6 et 27 novembre, 18 décembre 2019 et 8 janvier 2020:** Séances de travail avec les encadrants.

**Fin de la partie 1:** **La partie 2 est annulée compte-tenu du semestre troublé par les transports difficiles**

**Soutenances:** Mercredi 22 Janvier 2020

**Date limite de rendu de projet:** Vendredi 24 Janvier 2019.

## References

- [1] P. Augerat and J.M. Belenguer and E. Benavent and A. Corberán and D. Naddef Separating capacity constraints in the CVRP using tabu search European Journal of Operational Research 106-2, 546-557 (1998)  
<https://www.sciencedirect.com/science/article/pii/S0377221797002907>