

Projet 2 : Pick-and-Deliver Traveling Salesman Problem (PD-TSP)

On s'intéresse ici à une variante du problème du voyageur de commerce avec chargement, dans lequel le véhicule n'effectue pas uniquement une collecte d'objets, mais doit également gérer des opérations de *livraison* et de *ramassage* en différents points du tour. Ce problème est appelé *Pick-and-Deliver Traveling Salesman Problem* (PD-TSP). Il modélise naturellement des situations réelles de logistique : services de courrier, tournées de collecte et distribution, maintenance itinérante, robotique mobile, etc.

Le PD-TSP modélise des systèmes logistiques où un véhicule :

- transporte initialement un ensemble d'objets à livrer,
- collecte de nouveaux objets dans les différentes villes,
- doit visiter chaque ville exactement une fois,
- subit un **coût de déplacement dépendant du poids transporté**.

Dans ce projet, nous étudions deux variantes :

1. une variante à coût linéaire en fonction du poids,
2. une variante à coût quadratique (pénalités croissantes dues, par exemple, à la surcharge, risque de casse, difficulté d'organisation interne du chargement).

1 Formalisation du PD-TSP

1.1 Données communes

Input

- Un ensemble de villes $N = \{1, \dots, n\}$ et les distances $d_{i,j}$ entre chaque paire (i, j) .
- Un véhicule ayant une capacité maximale W_{\max} .
- Un poids initial transporté par le véhicule, noté W^0 .

Chaque ville contient deux types d'objets :

Objets à livrer (déjà présents dans le véhicule). Pour chaque ville i , un ensemble L_i d'objets à livrer, chacun ayant un poids w_{ik}^L . La livraison réduit le poids transporté.

Objets à ramasser. Pour chaque ville i , un ensemble R_i d'objets à collecter, chacun ayant :

- un profit p_{ik}^R ,
- un poids w_{ik}^R .

Output

- un tour hamiltonien visitant chaque ville exactement une fois,
- un plan de chargement $P = \{z_{ik}, y_{ik}\}$:
 - $z_{ik} = 1$ si l'objet $k \in L_i$ est livré en ville i ,
 - $y_{ik} = 1$ si l'objet $k \in R_i$ est ramassé en ville i ,

1.2 Évolution de l'inventaire

À la sortie de la ville i , le poids transporté est :

$$W_i = W_0 - \sum_{k \in L_i} w_{ik}^L z_{ik} + \sum_{k \in R_i} w_{ik}^R y_{ik},$$

où :

- $z_{ik} = 1$ si l'objet à livrer k est effectivement déposé en i ,
- $y_{ik} = 1$ si l'objet à ramasser k est collecté à i .

Chaque étape doit respecter la contrainte :

$$W_i \leq W_{\max}.$$

1.3 Coût du déplacement

Le coût de déplacement entre deux villes i et j dépend du poids W_i transporté en sortie de la ville i .

Deux variantes sont considérées :

Variante linéaire.

$$C(W_i) = a W_i$$

où a est un paramètre positif représentant un coût marginal par unité de poids.

Variante quadratique.

$$C(W_i) = a W_i + b W_i^2$$

où $b > 0$ modélise une *surcharge croissante* lorsque l'inventaire devient important.

1.4 Fonction objective

On cherche une tournée Π et un plan de chargement P maximisant :

$$Z(\Pi, P) = \sum_{i,k} p_{ik}^R y_{ik} - \sum_{(i,j)} C_{ij}(W_i) x_{ij}.$$

L'objectif exprime un compromis entre :

- maximiser les profits des objets collectés,
- minimiser les coûts de déplacement dépendants du poids.

2 Manipulation des instances

Pour mener à bien les expérimentations des heuristiques appliquées au PD-TSP, plusieurs sources de données peuvent être utilisées. L'objectif est de disposer à la fois de petites instances pour faciliter le débogage et la compréhension du comportement des heuristiques, ainsi que d'instances plus importantes, permettant une évaluation comparative plus robuste.

2.1 Instances existantes dans la littérature

Une source particulièrement adaptée est le *Pickup-and-Delivery Site* développé par H. Hernández-Pérez¹. Ce site historique propose des instances.

Les fichiers fournis sont généralement au format `.tsp`, semblable au format et contiennent :

- les coordonnées des villes
- la capacité du véhicule
- des demandes pouvant être positives (collecte) ou négatives (livraison)

ce qui correspond très naturellement à nos objets à ramasser ou livrer dans le PD-TSP. Ainsi, ces instances constituent un excellent point de départ pour tester les heuristiques linéaires et non-linéaires développées dans ce projet.

Des instances supplémentaires peuvent également être trouvées sur des bibliothèques de problèmes de tournées telles que TSPLIB ou CVRPLIB, en adaptant, lorsque nécessaire, les demandes de livraison/collecte à la structure du PD-TSP. Ces adaptations seront mentionnées dans votre rapport si vous choisissez de les utiliser.

2.2 Jeux d'instances artificielles

Outre les instances issues de la littérature, vous êtes encouragés à générer vos propres jeux d'instances pour affiner vos expérimentations. Ce type d'instances artificielles est particulièrement utile lorsque l'on souhaite analyser plus finement l'influence de certains paramètres ou isoler des configurations spécifiques.

Quelques idées pour générer vos propres jeux de données :

- utiliser une configuration géographique issue d'instances TSP (par exemple TSPLIB) et y associer aléatoirement des objets à livrer et/ou à ramasser
- fixer des profits et poids selon des distributions contrôlées
- varier la quantité de poids initial W^0 pour évaluer l'effet de la charge de départ
- tester différents régimes de capacité W_{\max} : faible, moyenne ou élevée ;
- construire de petites instances synthétiques ($n = 10$ à 20 villes) pour comprendre le comportement élémentaire de vos heuristiques.

Ces instances générées peuvent constituer de petites batteries de tests pour comparer rapidement différentes variantes d'heuristiques, avant de passer aux instances plus exigeantes du *Pickup-and-Delivery Site*.

Il est recommandé de :

- commencer par de petites instances (10–30 villes) afin de valider les choix de modélisation et de vérifier que le calcul des poids W_i est correct
- utiliser ensuite les instances plus structurées et plus riches, pour les comparaisons systématiques
- documenter, dans votre rapport, les sources des instances utilisées et les éventuelles transformations réalisées

Ainsi, l'ensemble de ces sources vous permettra de construire un ensemble d'expériences varié, cohérent et scientifiquement robuste pour l'évaluation de vos méthodes sur les deux variantes du PD-TSP.

¹<https://hhperez.webs.ull.es/PDsite/index.html>

3 Travail demandé pour ce projet

3.1 Visualisation

Produire une visualisation des villes, des distances, du plan de chargement (livraisons et ramassages), de la tournée obtenue.

3.2 Heuristiques gloutonnes

Pour les deux versions (linéaire et quadratique), proposer au moins une heuristique :

- livrer le plus tôt possible pour réduire le poids,
- ramasser les objets les plus profitables,
- stratégie d'insertion, nearest neighbor pondéré, etc.
- considérer dans la fonction objective les distances entre les villes (TSP).

3.3 Méthodes itératives

Développer une ou plusieurs méthodes itératives améliorantes :

- considérer dans la fonction objective les distances entre les villes (TSP).

3.4 Modélisation et résolution PLNE

Élaborer une **formulation compacte** pour :

- la version linéaire du PD-TSP,

et la résoudre avec **GUROBI**.

3.5 Comparaison expérimentale

Comparer :

- heuristiques gloutonnes,
- méthodes itératives,
- méthode exacte (PLNE GUROBI),
- linéaire vs quadratique.

Critères :

- qualité de la solution,
- temps de calcul,
- taille d'instance résolue,
- sensibilité au paramètre b (quadratique).

4 Références

- Veenstra et al., *Pickup-and-Delivery TSP with Handling Costs*. EJOR, 2017.
- Carrabs et al., *Variable Neighborhood Search for the PD-TSP with LIFO Loading*. INFORMS, 2007.
- Malli et al., *A computational study for the inventory routing problem*. <https://arxiv.org/pdf/2007.14740>., 2020.