Master Androide

Module MAOA

***Traveling Thief Problem* (TTP)**

*(*Problème du voyageur voleur*)*

Maroua BAHRI

maroua.bahri@lip6.fr

Pierre Fouilhoux

pierre.fouilhoux@lipn.univ-paris13.fr

2024-2025

SORBONNE UNIVERSITÉ

Projet: Traveling Thief Problem

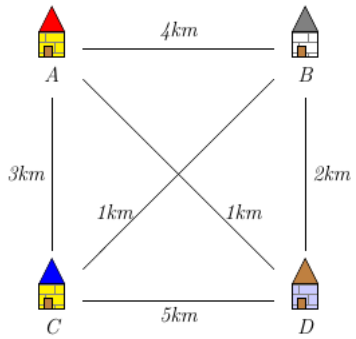
- Travail demandé pour ce projet:
  1. **Concevoir** des heuristiques pour les problèmes de TTP et KCTSP
  2. **Implémenter** ces heuristiques, les **évaluer**, et les **comparer** avec l'état de l'art
  3. **Documenter** sous forme de rapport décrivant les solutions ainsi que les expérimentations réalisées avec une analyse critique des résultats
- Vous pouvez former des binômes
- Ce projet sera noté en fonction de ces trois aspects et comptera pour 20% de la note finale (+ 40% examen1 et 40% examen2)
- Une mini-soutenance/présentation de 5 à 10min aura lieu le **22/01/2025 à 8h30**
- Soumission par email à:
  - maroua.bahri@lip6.fr
  - pierre.fouilhoux@lipn.univ-paris13.fr

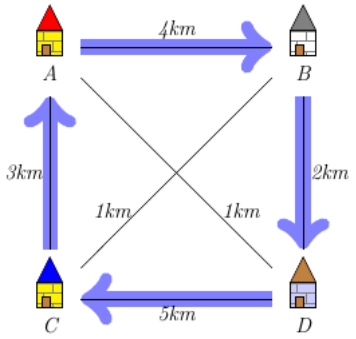
Traveling Thief Problem (TTP)

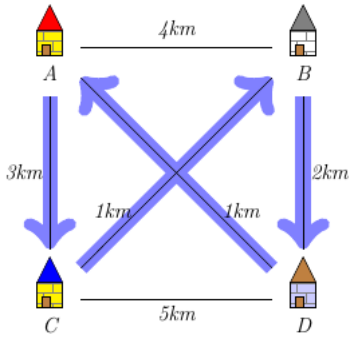
Combinaison de deux sous-problèmes bien connus:

- Problème du voyageur de commerce (TSP)

(Traveling Salesman Problem)







- Problème du sac à dos (KP)

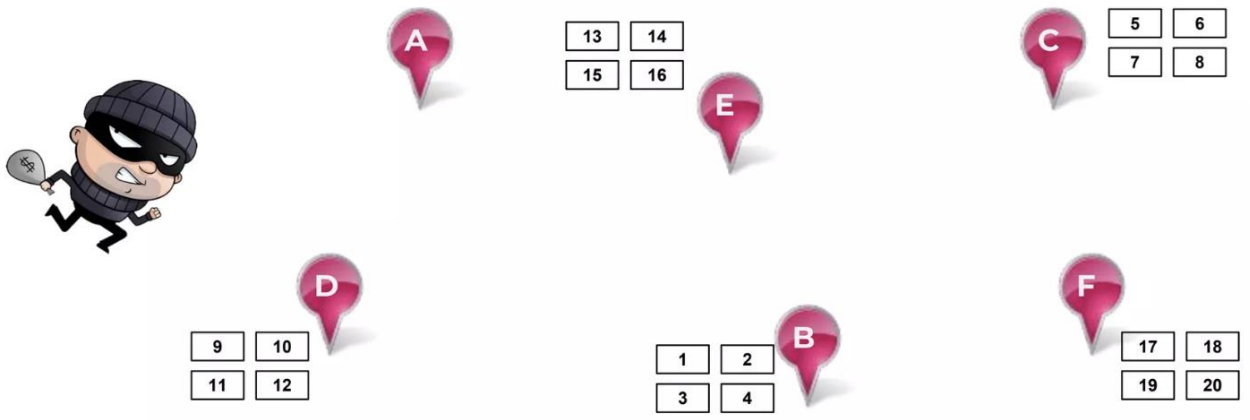
(Knapsack Problem)

Source d'images: Wikipédia

Traveling Thief Problem (TTP)

Étant donné un ensemble d'objets distribués sur un ensemble de villes, un voleur doit visiter chacune des villes et décider quels objets voler.

Quel est le meilleur chemin et le meilleur choix d'objets pour faire le meilleur profit ?

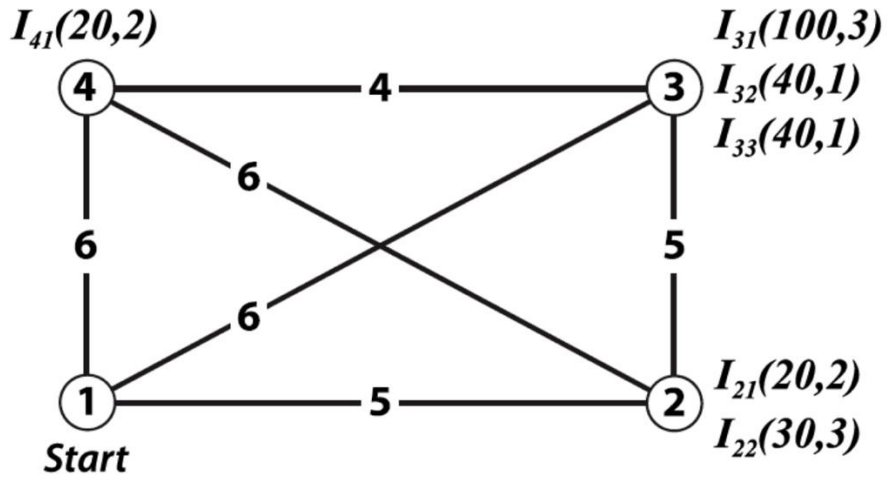


Applications: Logistique et transport, planification d'itinéraires pour la collecte de ressources, …

Source de l'image: El Yafrani et al., A fitness landscape analysis of the Travelling Thief Problem. GECCO, 2018

Traveling Thief Problem (TTP)

- Le profit dépend de:
  - La valeur totale des objets volés
  - La longueur de la tournée
- La longueur de la tournée dépend :
  - Du chemin pris (TSP)
  - De la vitesse du voleur (déterminée par le poids total des objets volés dans le sac)



La tournée doit commencer et finir par la ville 1 (aucun objet n'existe dans la ville de départ/arrivée)

Source de l'image: Polyakovskiy et al., A Comprehensive Benchmark Set and Heuristics for the Travelling Thief Problem. GECCO, 2014

Traveling Thief Problem (TTP)

Données d'entrée/contraintes:

- Un ensemble de $\boldsymbol{n}$ villes $\boldsymbol{N} = \{1..n\}$
  - Avec les distances $\boldsymbol{d_{i,j}}$ entre chaque paire (i, j) de villes
- Un ensemble de $\boldsymbol{m_i}$ objets pour chaque ville: $\boldsymbol{M_i} = \{1, \dots, m_i\}$, où chaque objet $\boldsymbol{k}$ dans la ville $\boldsymbol{i}$ est caractérisé par:
  - Un poids $\boldsymbol{w_{ik}}$ et sa valeur $\boldsymbol{p_{ik}}$
- Un sac à dos avec une capacité maximale $\boldsymbol{W}$
  - Sous la contrainte de sac à dos: $\sum_{i=1}^{n} \sum_{k=1}^{m_i} w_{ik} \leq W$
- Un coût de location $\boldsymbol{R}$ du sac à dos (pour chaque unité de temps)
- Une vitesse de déplacement variable $\boldsymbol{v(w)}$ qui dépend du poids w transporté
  - La vitesse diminue quand le poids augmente
  - La vitesse maximale est $\boldsymbol{v_{max}}$ et la vitesse minimale est $\boldsymbol{v_{min}}$

Traveling Thief Problem (TTP)

- Le but est de trouver un tour passant par toutes les villes et la décision de prendre ou non un objet de chaque ville
  - Le parcours du voleur: tournée $\boldsymbol{\Pi} = (x_1, \dots, x_n)$
  - Les objets à voler à chaque ville: plan de collecte *(packing plan)* $\boldsymbol{P} = (y_{21}, \dots, y_{2m_2}, \dots, y_{nm_n})$

  $y_{ik} = 1$ si et seulement si l'objet k à la ville i est sélectionné
- Afin de maximiser le gain:

$$Z(\Pi, P) = B(P) - R * T(\Pi, P), \text{ où}$$

  - $B(P)$ est le bénéfice, cad, le total des valeurs des objets choisis:

$$B(P) = \sum_{i=1}^{n} \sum_{k=1}^{m_i} p_{ik} y_{ik}$$

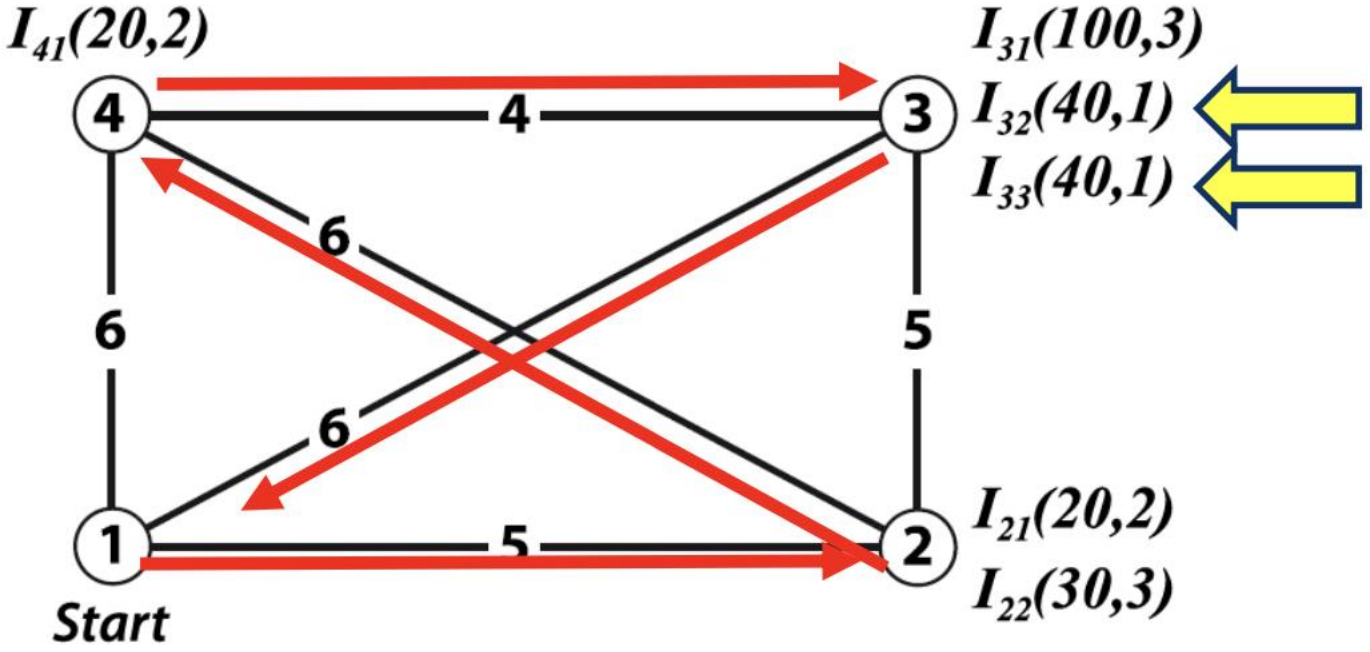
  - $T(\Pi, P)$ est le temps que le voleur prend pour la tournée

$$T(\Pi, P) = \frac{d_{x_n x_1}}{v_{max} - v W_{x_n}} + \sum_{i=1}^{n-1} \frac{d_{x_i x_{i+1}}}{v_{max} - v W_{x_i}}$$

  où $\boldsymbol{v} = \frac{v_{max} - v_{min}}{W}$, et $\boldsymbol{W_{x_i}}$ représente le poids chargé dans le sac quand le voleur part de la ville i

Illustration de la fonction objective

- Entrée: $W = 3, v_{max} = 1, v_{min} = 0.1, R = 1$



- Profit de la tournée $\Pi = (1, 2, 4, 3)$ et du choix d'objets $P = (0, 0, 0, 0, 1, 1)$ est:

$$Z(\Pi, P) = 40 + 40 - 1.\left(15 + \frac{6}{1 - 0{,}3\,.\,2}\right) = 80 - 30 = 50$$

TTP
Time and weight dependent TSP + contrainte de KP

Vous pouvez considérer deux formulations du problème:

1. Une première version du Time and weight dependent problem qui est appelé Cumulative TSP avec la contrainte de KP (KCTSP). Il y a une donnée supplémentaire:
   - K coût de location du sac à dos (par km du transport d'un kg transporté)
   - Le poids transporté à la ville x_i est alors $W_{x_i} = \sum_{k=1}^{i} \sum_{j=1}^{m_i} w_{kj} . y_{kj}$
   - fonction **linéaire** qui ne considère pas l'aspect de vitesse (v_{min}, v_{max}) du TTP [3][4]

$$\sum_{i=1}^{n} \sum_{k=1}^{m_i} p_{ik} y_{ik} - K \left(d_{x_n x_1} W_{x_n} + \sum_{i=1}^{n-1} d_{x_i x_{1+1}} W_{x_i} \right)$$

2. Une deuxième version qui est le TTP avec une fonction objective Z qui est **non-linéaire**

Travail demandé

- L'idée est d'étudier les deux variantes en utilisant plusieurs méthodes de la littérature. Approfondissez au moins une des méthodes en s'appuyant sur la littérature
- Dans le cadre de ce projet, il est demandé:
  - Une visualisation des données et des solutions pour les deux versions
  - Une heuristique gloutonne
  - Une heuristique itérative
  - Une formulation compacte PLNE pour la version KCTSP résolu par GUROBI
  - Une comparaison des résultats obtenus pour les trois méthodes et les deux problèmes (en termes de qualité, temps de calcul, tailles résolues, ...)

Implémentation et comparaison

1. Implémenter et tester les heuristiques de la réf [1]

   [1] Polyakovskiy et al., A Comprehensive Benchmark Set and Heuristics for the Travelling Thief Problem. GECCO, 2014

Les détails concernant l'implémentation et les données sont disponibles dans le [2] CEC 2014 competition: https://cs.adelaide.edu.au/~optlog/CEC2014Comp/

2. Concevoir, implémenter et tester vos propres solutions:
   1. Heuristiques gloutonnes
   2. Heuristiques itératives
   3. Programmation linéaire en nombre

Entrée: Il faut choisir une **limite** sur le nombre d'itérations ou le temps d'exécution pour chaque heuristique

[1] Polyakovskiy et al., A Comprehensive Benchmark Set and Heuristics for the Travelling Thief Problem. GECCO, 2014

Implémentation et comparaison

- Les jeux de données et le code:
  - Pour tester et évaluer vos solutions, vous pouvez utiliser l'ensemble d'instances dans [2] (9 jeux de données)
- Pour l'implémentation de vos solutions, utiliser le langage de programmation qui vous arrange
- Rendre un code source complet et documenté des solutions (et les baselines utilisées) ainsi que les jeux de données utilisées pour les expérimentations
  - Le code et les données doivent être organisés dans une structure de répertoires claire
  - Options de partage recommandées: Un lien privé de dépôt GitHub, Drive, Dropbox, OneDrive, GitLab ou autre.
  - Fournir un lien de partage qui doit rester valide pendant la période d'évaluation, avec les droits d'accès en lecture accordés

[2] Site web de CEC 2014 competition http://cs.adelaide.edu.au/~optlog/CEC2014Comp/

Références

[1] Polyakovskiy et al., A Comprehensive Benchmark Set and Heuristics for the Travelling Thief Problem. GECCO, 2014

[2] Site web de CEC 2014 competition http://cs.adelaide.edu.au/~optlog/CEC2014Comp/

[3] Abeledo et al., The Time Dependent Traveling Salesman Problem: Polyhedra and Branch-Cut-and-Price Algorithm. Experimental Algorithms. SEA 2010. LNCS, 6049 (2010).

[4] Cordeau et al., Analysis and Branch-and-Cut Algorithm for the Time-Dependent Travelling Salesman Problem Transportation Science, 48-1 pp. 46-58 (2014).

Voir plus:

[5] Bonyadi et al., The travelling thief problem: the first step in the transition from theoretical problems to realistic problems. IEEE CEC, 2013

[6] Faulkner et al., Approximate Approaches to the Traveling Thief Problem, GECCO, 2015

- Questions?
- Quels sont les binômes ?