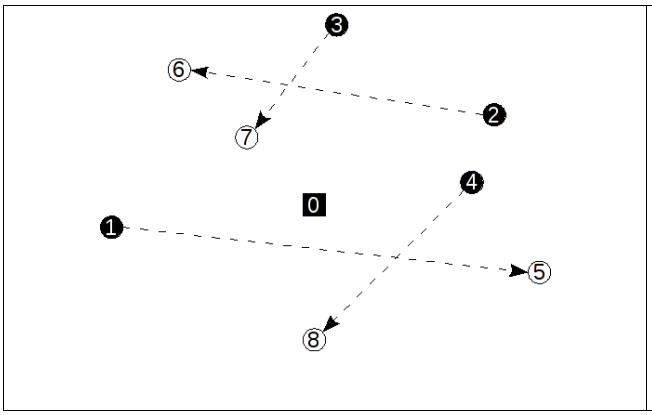
TP4 outils d'Aide à la Décision
Heuristiques pour le PDPTW

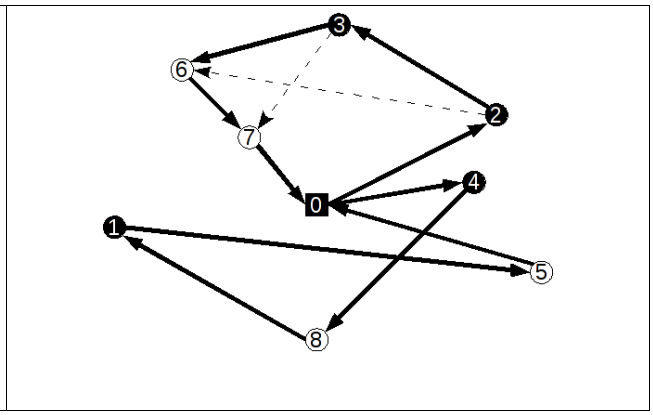
Duhamel - Lacomme

ISIMA 2ème année F2F3, 2 séances

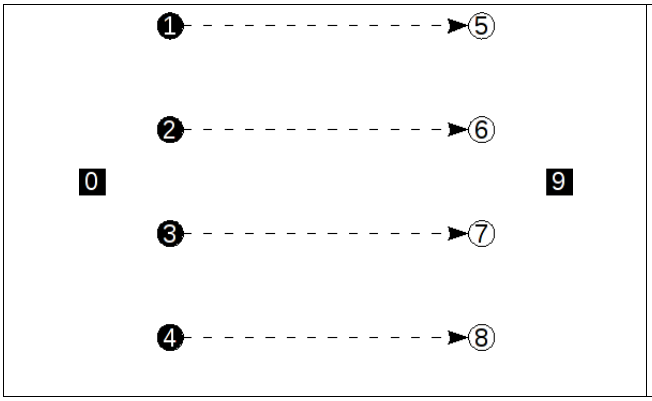
Présentation du problème : le problème de livraison et collecte (Pickup and Delivery Problem – PDP) consiste à définir un ensemble de routes de manière à satisfaire un ensemble de requêtes de transport. Une flotte de véhicules est disponible pour effectuer ces transports. Chaque véhicule a une capacité connue, un point de départ et un point d'arrivée (généralement le dépôt central). Une requête de transport est caractérisée par une quantité à transporter, un point de collecte (pickup) et un point de livraison (delivery). La requête ne doit être traitée que par un seul véhicule et la collecte doit être nécessairement réalisée avant la livraison. L'objectif est d'abord de minimiser le nombre de véhicules utilisés et, de manière secondaire, la distance totale parcourue par les véhicules. Une extension classique ajoute des contraintes temporelles sur les requêtes : à chaque point de livraison / collecte on associe une fenêtre de temps [A,B]. La première date correspond à l'heure d'ouverture et la seconde à l'heure de fermeture. Ainsi, le véhicule peut arriver avant l'heure A, mais il doit alors attendre l'ouverture. Par contre, il ne peut arriver après l'heure B. De plus, chaque collecte / livraison induit un certain temps de chargement / déchargement. Le PDP devient alors le problème de livraison et collecte avec fenêtre de temps (Pickup and Delivery Problem with Time Windows – PDPTW). Ce problème est un des problèmes centraux de l'industrie du transport, il est NP-difficile.



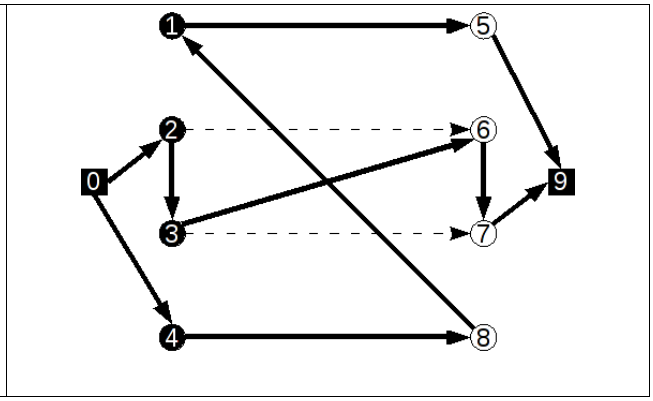
représentation géométrique d'une instance



représentation géométrique d'une solution



représentation logique de l'instance



représentation logique d'une solution

L'exemple ci-dessus illustre deux représentations d'une même instance, en laissant de côté les

fenêtres de temps et la capacité ; n=4 requêtes de transport (i,n+i) doivent être satisfaites. Dans la représentation logique, le dépôt est dupliqué en un sommet de sortie (0) et de retour (9).

Récupération des instances

1. Un jeu d'essai standard pour le PDPTW a été créé par Li & Lim en 2001. On peut le retrouver sur la page (http://www.sintef.no/Projectweb/TOP/Problems/) avec les meilleures solutions trouvées. Chaque point (dépôt, livraison ou collecte) possède des coordonnées 2D et la distance entre deux points est la distance euclidienne.

Manipulations élémentaires

2. Définir les structures de données adaptées au stockage des données et à la manipulation efficaces des solutions. La représentation de la solution est plus complexe que celle du TP précédent car on doit stocker un ensemble de tournées et, dans chacune, la date de passage du véhicule sur chaque point (pour accélérer les calculs de faisabilité temporelle).

Heuristiques de construction

Les contraintes temporelles compliquent la génération d'une solution réalisable. Par contre, le nombre de véhicule est libre. Voici deux heuristiques conçues initialement pour le VRP :

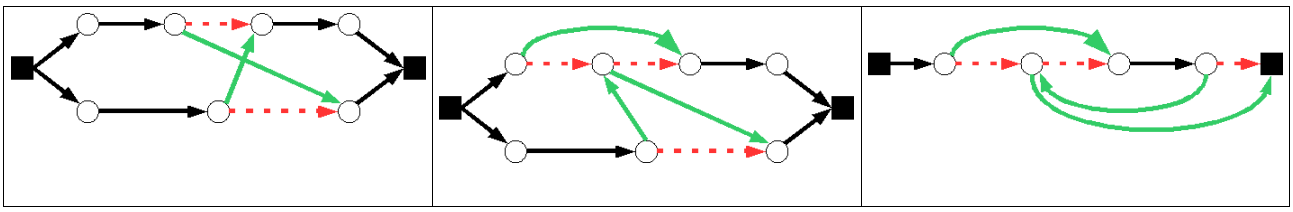
- L'heuristique d'insertion : pour le PDPTW, on crée une nouvelle tournée. Parmi les requêtes non encore traitées, on regarde celle qui peut s'insérer le mieux à la fin de la tournée. Si une telle requête existe, on l'insère. Sinon on crée une nouvelle tournée. On itère tant qu'il reste une requête non traitée.
- L'heuristique Savings de Clarke & Wright : pour le PDPTW, on crée initialement une tournée pour chaque requête de transport. Ensuite, pour tout arc (i,j) où i est la livraison d'une requête et j la collecte d'une autre, on calcule le gain $g_{ij} = c_{i0} + c_{0j} – c_{ij}$. On trie ces gains par ordre décroissant. Ensuite, dans cet ordre, g_{ij} étant le gain courant, on regarde s'il existe une route commençant par j et une route terminant par i. Si c'est le cas, on vérifie la compatibilité au niveau des fenêtres de temps. Si c'est bon, on fusionne ces routes en supprimant les arcs (0,j) et (i,0) et en ajoutant l'arc (i,j). Puis on passe au gain suivant.

3. Implémenter l'heuristique d'insertion telle que décrite. Étudier la possibilité d'insérer la requête n'importe où dans la tournée en cours de remplissage, puis la possibilité que la livraison et la collecte ne soient pas traitées consécutivement.
4. Implémenter la variante suivante de l'heuristique d'insertion : on suppose que les requêtes sont mises dans un ordre donné. On part d'une solution vide. Dans cet ordre, on prend la requête courante et on effectue l'action suivante : soit on peut l'insérer dans la tournée en cours (et on le fait) soit on crée une nouvelle tournée avec cette requête. Une amélioration consiste à voir si on peut insérer la requête courante dans une des tournées ayant été ouvertes jusqu'à présent.
5. Implémenter l'heuristique Savings.

Heuristiques d'amélioration

Le principe d'amélioration est identique à celui vu en cours. Les voisinages sont différents car ils doivent prendre en compte à la fois la capacité, le temps et l'ordre collecte → livraison. Plusieurs voisinages sont intéressants :

- le 2-Opt* consiste à échanger les dernières requêtes de deux tournées
- le Or-Opt consiste à déplacer une séquence de requêtes (1 ou 2) dans une autre tournée
- au sein d'une tournée, les opérations de déplacement d'un sommet ou plusieurs sommets (livraison ou collecte) permettent d'optimiser localement la tournée.



2-Opt* Or-Opt Shift

Les mouvements ci-dessus sont valides pour le TSP. Il faut les adapter pour le PDPTW.

Métaheuristiques

6. Choisir une métaheuristiques. GRASP et les algorithmes génétiques / mémétiques sont de bons candidats. Attention à l'efficacité de la recherche locale. À noter l'article de S. Ropke et D. Pisinger « a unified heuristic for vehicle routing problems with backhauls » dans European Journal of Operational Research 2006 qui présente une approche originale et très efficace pour le PDPTW.

Modalités de remise du travail

Comme pour le TP2, nous attendons un rendu sous forme d'une fichier tar.gz ou zip contenant les sources du programme et le rapport décrivant les choix réalisés et présentant les tableaux de résultats. Le programme principal doit accepter en entrée le nom de l'instance à traiter. Il doit ressortir une ligne contenant le temps de calcul et la valeur de la meilleure solution trouvée (nombre de véhicules et distance totale). La date limite de remise du travail est fixée au 9 janvier. Au delà de cette date, la pénalité sera proportionnelle au nombre de jours de retard.