

Optimiser la distribution de repas

Frantzen Christian, Küpper Marius, Baes Akira, Palmieri-Adant Emile

13 novembre 2015

Introduction

L'optimisation de la distribution des repas est un problème qui s'apparente fort au Dial-A-Ride-Problem (DARP). Il a fortement été étudié dans le but d'optimiser le transport des personnes âgées ou inaptes au déplacement. Dans le cadre de ce projet, ce sont des repas que l'on doit transporter des cuisiniers jusqu'aux clients. Ce problème implique un troisième personnage, le livreur. Le défi ici consiste à prendre en compte les contraintes de chacun et trouver une route de moindre coût qui satisfait les demandes de tous les clients.

Nous allons donc étudier le DARP plus en détail dans la prochaine section.

Description d'un DARP

Dans un DARP il y a n clients qui requièrent un transport depuis un point de départ jusqu'à la destination désirée. Chaque client doit spécifier une fenêtre de temps de leur heures désirées de départ ainsi que d'arrivée. Chaque véhicule possède une capacité maximale. Chaque client a un temps de transport maximum (on ne peut pas faire trop attendre un client déjà embarqué).

Il existe 2 types différents de DARPs, *statique* et *dynamique*. Un DARP statique part de l'idée que toutes les requêtes sont connues avant qu'un véhicule ne commence sa trajectoire. Un DARP dynamique, ici les requêtes ont lieu graduellement au cours du temps. La trajet est donc déterminé en temps réel. Nous allons d'abord considérer un modèle statique dans le cadre de ce projet.

Notre sujet en tant qu'un DARP

Dans notre projet il y a n couples de cuisiniers et de clients qui doivent voir leur repas transportés. On peut voir le problème en tant que points de départ et points d'arrivée. Deux fenêtres de temps sont spécifiées, une pour le départ (donnée par le cuisinier) et une pour l'arrivée (donnée par le client). Les véhicules (livreur) possèdent une capacité maximale et la tournée d'un livreur a une durée limite tout comme le transport d'un repas.

Formulation du problème

Le DARP est modélisé par un graphe $G = (V, A)$. $V = \{ v_0, v_1, v_2, \dots, v_{2n} \}$ représente l'ensemble des sommets du graphe. $A = \{ (v_i, v_j) : v_i, v_j \in V, i \neq j \}$ représente l'ensemble des arcs du graphe. v_0 correspond au dépôt et les $2n$ sommets restants correspondent aux origines et destinations des clients. La paire de sommet (v_i, v_{i+n}) définit une demande de transport. A chaque sommet est associé une charge q_i (avec $q_0 = 0$), une durée de service d_i (avec $d_0 = 0$) et une fenêtre de temps $[e_i, l_i]$ où e_i et l_i sont non négatifs. Les charges associées aux sommets v_1, \dots, v_n sont toujours positives et a contrario, les charges associées aux sommets v_{n+1}, \dots, v_{2n} sont toujours négatives. La variable T est le temps correspondant à la fin de toutes les tournées des véhicules, T_k étant la durée maximale d'une tournée d'un véhicule k . Dès lors, on définit par requête outbound une requête dont $e_i = 0$ et $l_i = T$. A l'opposé, on définit par requête inbound une requête dont $e_{i+n} = T$ et $l_{i+n} = 0$. A chaque arc (v_i, v_j) est associée une valeur c_{ij} positive étant le coût de déplacement et une valeur t_{ij} également positive étant le temps de déplacement. La variable L représente le temps maximum de trajet d'un repas.

Notre algorithme essaye donc de créer des routes dans le graphe G en respectant les contraintes suivantes :

- Chaque route commence et termine au dépôt ;
- Pour chaque requête i , les sommets v_i et v_{i+n} appartiennent à la même route et v_i est visité avant v_{i+n} ;
- La charge d'un véhicule k ne peut en aucun cas excéder une limite Q_k ;
- La durée totale d'une route k ne peut en aucun cas excéder la limite T_k ;
- Le temps de trajet d'un repas ne peut en aucun cas dépasser L ;
- Le coût dû au déplacement de chaque véhicule est minimisé.

Descriptions d'articles de référence

A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows

Dans leur article, Marco Diana et Maged M. Dessouky décrivent une solution du DARP en tenant compte de contraintes horaires (time windows) et de la situation géographique des différents clients. Les time windows admissibles de chaque requête sont définis à l'aide des données des clients (heure de départ souhaitée ou heure d'arrivée souhaitée) et d'un temps de service à chaque arrêt, vu que les auteurs se concentrent particulièrement sur le transport de personnes à mobilité réduite. Cet article est intéressant pour notre projet car on y décrit une façon comment choisir un bon point de départ de

chaque route pour les différents véhicules et on y décrit une façon comment insérer de nouvelles requêtes dans des routes déjà établies en profitant du système des time windows.

A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios (Santiago Muelas , Antonio La-Torre , José-María Peña)

Dans leur article, les auteurs décrivent une façon pour résoudre des DARPs de grande taille. Pour maîtriser la taille du problème, les différents requêtes qui sont proches d'un point de vue géographique et horaire sont regroupés, puis ces groupes sont traités comme sous-problèmes. Chaque sous-problème est résolu et la route optimale de sa solution est renvoyé à l'algorithme appelant. Chaque groupe est traité indépendant des autres groupe. La construction de la solution finale est construite à partir des différents sous-solutions du problème. L'article est pertinent pour notre projet parce qu'il montre une façon comment traiter un DARP de grande taille.

The dial-a-ride problem : models and algorithms (Jean-François Cordeau et Gilbert Laporte) 2007

Cet article résume la littérature scientifique à propos des DARP. Il donne un aperçu des différents algorithmes qui ont été publiés pour attaquer ce problème, chacun avec une description des contraintes (time windows, véhicules, ...) et une nombre n d'utilisateurs que l'algorithme est capable de gérer dans un temps raisonnable. Cet aperçu peut nous servir pour apprécier quelle structure l'algorithme convient le plus pour résoudre notre problème. Aussi les différents modèles mathématiques sont présentés et la différence entre le modèle statique et le modèle dynamique est expliquée.

A Granular Tabu Search algorithm for the Dial-a-Ride Problem (Dominik Kirchler, Roberto Wolfer Calvo) 2013

Dans cet article, un algorithme 'Granular Tabu Search' est présenté, qui est une amélioration de l'algorithme 'Tabu Search' décrit par l'article de référence. 'Granular', granulaire en français, parce que les voisinages qui sont utilisés pour l'amélioration de la solution initiale sont sous forme réduite, c'est-à-dire que les trajets qui ont une faible probabilité d'appartenir à de bonnes solutions sont exclus. L'article peut nous fournir des informations pour améliorer notre algorithme.

Recent Models and Algorithms for One-to-One Pickup and Delivery Problems revised (Jean-Francois Cordeau, Gilbert Laporte et Stefan Ropke) 2007

Dans cet article sont décrits des algorithmes (2007) pour résoudre les PDP (Pickup & Delivevy Problem), dont un branch-and-cut pour DARP développé par Cordeau et Ropke en pages 16-19 ("3.1.2 A branch-and-cut algorithm for the PDPTW and the DARP"). Branch-and-cut est présenté comme plus efficace pour donner une solution optimale sur de petits problèmes. Dans ce modèle où tous les véhicules sont identiques, on pose une variable à minimiser avec une série de nouvelles contraintes et d'inégalités sur lesquelles on applique un algorithme branch-and-cut. C'est une version améliorée du Branch-and-cut Cordeau de 2003. L'article semble présenter l'état de l'art de l'époque en matière de Branch-and-cut pour DARP.

A Branch-and-Cut Algorithm for the Dial-a-Ride Problem (Jean-Francois Cordeau) 2003

Dans cet article est décrit la première version d'un Branch-and-cut pour DARP décrit par Cordeau peu après son article sur la méthode heuristique tabu. Sont présentés une somme à minimiser sous de nombreuses contraintes et inégalités qui décrivent les limitations du problème. L'article peut nous informer sur les débuts de la méthode Branch-and-cut.