

Optimiser la distribution de repas

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

16 décembre 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 une destination désirée et m véhicules qui réalisent les tournées. Un client spécifie soit l'heure de départ désirée (requête *outbound*), soit l'heure d'arrivée désirée (requête *inbound*) avec parfois la déviation maximale de cet instant. À l'aide de ces informations, des fenêtres de temps sont construits dans lesquelles le véhicule doit passer pour respecter ces contraintes. Chaque véhicule possède une capacité maximale et une durée maximale pour sa tournée. Chaque client a un temps de transport maximum (on ne peut pas faire trop attendre un client déjà embarqué).

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

Notre sujet en tant que DARP

Dans notre projet il y a n repas qui seront transportés de leur cuisinier à leur destinataire et m véhicules qui réalisent les tournées. On peut voir le problème en tant que points de départ et points d'arrivée. Dans notre cas, que les heures d'arrivées sont spécifiées pour les repas, les heures de départ du transport sont calculées à l'aide de celles-ci. Ce ne sont donc que des requêtes *inbound*. Les véhicules (livreurs) possèdent une capacité maximale et la tournée d'un livreur a une durée limite tout comme le transport d'un repas.

Formulation du problème

Basé sur le doctorat de Jang-Jei Jaw [1] : "Solving large-scale dial-a-ride vehicle routing and scheduling problems".

L'algorithme heuristique garantie que

À chaque repas est associé une heure d'arrivée désirée,

Tiré de l'article de référence "A tabu search heuristic for the static multi-vehicle dial-a-ride problem." :

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 à 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 . On définit par requête *outbound* une requête de chez soi vers une destination, dont $e_i = 0$ et $l_i = T$, et on définit par requête *inbound* le retour dont $e_{i+n} = T$ et $l_{i+n} = 0$. La différence générale entre les deux requêtes est que l'utilisateur précise l'heure d'arrivée pour la requête *outbound* et l'heure de départ pour la requête *inbound*. À 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 tournée 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.

Les différentes approches

Tabu search

Les algorithmes tabu search commencent avec une solution initiale qu'ils essayent d'améliorer, itération par itération. On le nomme 'tabu' parce qu'on doit interdire (ou déclarer 'tabu') les solutions déjà visitées pour ne pas entrer dans un cycle. Les opérations qui sont effectuées pour améliorer une solution sont :

- enlever un couple (départ, destination) d'une route et l'insérer dans une autre route

- décaler un noeud soit de départ, soit de destination dans sa route
- permuter deux couples (départ, destination) de deux routes différentes

Pour la construction d'une solution initiale, les départs des requêtes sont répartis aléatoirement sur les différents véhicules et les destinations correspondantes sont ajoutées à la fin de la route de leur départ. Le voisinage d'une solution est l'ensemble de solutions pouvant être atteint par une simple opération : enlèvement d'un couple (départ, destination) d'une route et insertion dans une autre route. Les nouvelles solutions sont évaluées en fonction du coût et des contraintes (la capacité des véhicules, les fenêtres de temps, la durée limite d'un véhicule et la charge maximale d'un véhicule). Plus une solution viole de contraintes, plus elle est évaluée comme non favorable. Après un certain nombre d'itérations, il faut réarranger les nœuds pour chaque route tel que le coût de la route est réduit le plus possible. (<- A tabu search heuristics for ...) Dans la recherche actuelle, un algorithme qui s'appelle 'Granular Tabu Search' donne des bons résultats. Un voisinage granulaire est utilisé. Il s'agit d'un voisinage réduit ; c'est à dire il ne contient pas des solutions qui sont suspectées d'amener à une aggravation des coûts et des violations des contraintes. Une solution appartenant au voisinage est décidé par un 'threshold' (en français 'barrière'). Le critère de cette barrière est le temps de voyage. Seulement les solutions dont le temps de voyage est en-dessous de la barrière restent dans le voisinage. Il peut arriver qu'on ne trouve aucune solution qui réponde aux exigences de la barrière, on peut donc augmenter la barrière. (<- Granular Tabu Search)

Branch-and-cut

Branch-and-Cut (BAC) est une approche générique d'optimisation combinatoire. Il a été montré qu'un algorithme de type BAC peut résoudre un DARP ainsi que d'autres problèmes de routage. La première étape consiste à réduire la taille du problème. La seconde utilise un procédé appelé « relaxation continue » pour identifier certaines inégalités encadrant le problème qui seraient violées. Le but étant de transformer ce problème NP-hard en un problème qui est soluble en un temps polynomial. Le rétrécissement du problème effectué lors de la première étape se fait en premier lieu par une contraction de la fenêtre de temps dépendamment d'une requête out bound ou inbound. Deuxièmement, il est possible de supprimer un certain nombre d'arcs du graphe qui à coup sûr ne font pas parti d'une solution correcte. Enfin, il est également possible que certains clients ne soient pas compatibles avec d'autres. Il est important de réduire le problème à sa taille minimale pour éviter des calculs inutiles, atténuant ainsi le phénomène d'explosion combinatoire. La deuxième étape est un procédé heuristique consistant à identifier les solutions qui violent les inégalités précédemment mentionnées suivant une méthode de BAC.

Cette technique algorithmique appliqué au DARP à l'avantage d'être très efficace pour résoudre des petites instances du problème. Pour des plus larges sets de données, elle devient beaucoup moins intéressante.

Descriptions d'articles de référence

A tabu search heuristic for the static multi-vehicle dial-a-ride problem (Jean-François Cordeau et Gilbert Laporte) 2002

Dans cet article est décrit un algorithme de recherche tabou ('Tabu Search'). Leur approche consiste à établir un ensemble de solutions initiales sans se préoccuper des contraintes de temps et de charge. Est ensuite déterminé une fonction de coût régie par plusieurs paramètres ajustables ce qui permettra de construire les solutions. Le fait d'être souple quant à la violation des contraintes initiales est caractéristique de cet algorithme décrit dans cet article. Il se base sur l'évaluation de voisinages pour construire un trajet optimal minimisant le temps total de trajet ainsi que la durée d'un trajet pour un client. L'article décrit un algorithme 'Tabu Search' sur lequel nous allons nous baser pour résoudre notre DARP.

A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows (Marco Diana et Maged M. Dessouky) 2002

Dans leur article, les auteurs 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 LaTorre et José-María Peña) 2014

Dans leur article, les auteurs décrivent une façon pour résoudre des DARP de grandes tailles. Pour maîtriser la taille du problème, les différentes requêtes qui sont proches d'un point de vue géographique et horaire sont regroupées, 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 indépendamment des autres groupes. La construction de la solution finale est construite à partir des différentes sous-solutions du problème. L'article est pertinent pour notre projet parce qu'il montre une façon de 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 un 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. Par ailleurs 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 et Roberto Wolfer Calvo) 2013

Dans cet article, un algorithme 'Granular Tabu Search' est présenté. C'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. 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 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.

Large Neighborhood Search For Dial-a-Ride Problems (Siddhartha Jain et Pascal Van Hentenryck) 2011

Cet article présente une variante "First Feasible Probabilistic Acceptance" de la méthode appelée Large Neighborhood Search qui fait une recherche de voisinage sans fixer de contrainte d'amélioration des solutions partielles jusqu'à trouver des solutions faisables qui sont choisies sur bases probabilistes. La méthode est dite plus efficace dans l'article pour des recherches à contraintes à temps limité que les méthodes tabu de l'époque (2011), le rendant approprié pour du DARP semi-dynamique, et donne des résultats non-optimaux comparables aux algorithmes état-de-l'art de l'époque. Cet article peut nous donner une autre piste dans la recherche d'un algorithme efficace de résolution de DARP.

Bibliographie

- [1] Jang-Jei Jaw, Solving large-scale dial-a-ride vehicle routing and scheduling problems, *FTL report (Massachusetts Institute of Technology. Flight Transportation Laboratory)*, R84-3 (1984), 90-131
- [2] A tabu search heuristic for the static multi-vehicle dial-a-ride problem (Jean-François Cordeau, Gilbert Laporte) 2002
- [3] A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows (Marco Diana, Maged M. Dessouky) 2002
- [4] A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios (Santiago Muelas, Antonio LaTorre, José-María Peña) 2014
- [5] The dial-a-ride problem : models and algorithms (Jean-François Cordeau et Gilbert Laporte) 2007
- [6] A Granular Tabu Search algorithm for the Dial-a-Ride Problem (Dominik Kirchler, Roberto Wolfer Calvo) 2013
- [7] Recent Models and Algorithms for One-to-One Pickup and Delivery Problems revised (Jean-François Cordeau, Gilbert Laporte et Stefan Ropke) 2007
- [8] A Branch-and-Cut Algorithm for the Dial-a-Ride Problem (Jean-François Cordeau) 2003
- [9] Large Neighborhood Search For Dial-a-Ride Problems (Siddhartha Jain et Pascal Van Hentenryck) 2011