Combinatoire et routage

Cours 5. Tournées de véhicules

F. Clautiaux
Université de Bordeaux

francois.clautiaux@math.u-bordeaux.fr

(tiré de Christian Prins, Acquis et tendances de la recherche en tournées de véhicules)

Sommaire

Défiinition très générale

On a un problème de tournées de véhicules (vehicle routing problem, VRP) si :

- on doit visiter des sites
- pour y effectuer des tâches
- en utilisant des mobiles appelés véhicules
- avec un point de départ et de retour, le dépôt
- pour un optimiser un certain critère

Le problème est défini pour un graphe qui modélise un réseau réel (routier, électrique, ...) ou des déplacements possibles.

Des problèmes très répandus

Contexte	Graphe	Véhicules	Sites	Tâches
Logistique	routes	camions	points de vente	livraisons
Propreté urbaine	rue	camions	rues	collecte des déche
Gestion désastre	région	hélicoptères	villages	evacuation
Blanchisserie	couloirs	chariots	chambres	nettoyer
Relevé compteurs	électricité, eau,...	techniciens	compteurs	relevés
Assemblage	carlingue	bras de robot	position fixations	poser les fixation
Electronique	micro-proc	faisceau laser	transistor	graver les transisto

Tâches sur sommets vs. tâches sur arcs

Les problèmes les plus étudiés considèrent des tâches sur les sommets du graphe, comme des livraisons : **problèmes de tournées sur sommets** (node routing problems)

Les **problèmes de tournées sur arcs** (arc-routing, ARP) ont des tâches sur des arcs ou des arêtes, comme collecter des déchets ménagers dans une rue.

Equivalence entre tournées sur sommets et tournées sur arcs

Tout problème de tournées sur les arcs peut se reformuler comme un problème de tournées sur les sommets.

Méthode classique

pour chaque arête, on crée 2 sommets s_{ij} et s_{ji}, on ajoute le dépôt 0
$w(0, s_{ij}) = dist(0, i)$

$$w(s_{ij}, s_{kl}) = \begin{cases} 0, & \text{si } (i,j) = (k,l) \\ c(i,j), & \text{si } (i,j) = (l,k) \\ dist(i,k) & \text{sinon} \end{cases}$$

On obtient $2|E| + 1$ sommets.

Il existe une méthode récente (2015) où le nombre de sommets obtenu est $|E| + 1$.

La demande d'un arc (i, j) est répartie sur s_{ij} et s_{ji}.

Problèmes de tournées sur sommets

Capacitated Vehicle Routing Problem (CVRP).
Données

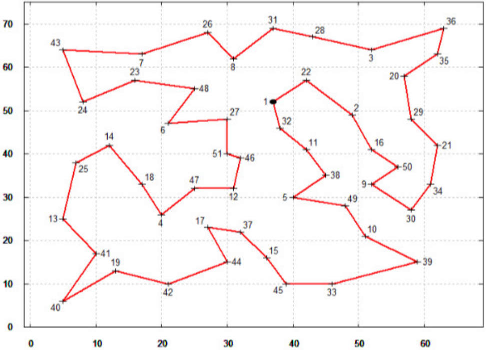
- graphe non orienté complet $G = (V, E, c)$ avec $n + 1$ sommets
- sommet 0 : dépôt avec véhicules identiques de capacité Q
- sommet 1 à n : clients avec demandes $q(i)$ pour un produit
- arêtes $(i,)$ avec des coûts $c(i, j)$ (distance, temps, ...)

Une tournée est un circuit de véhicules, partant du dépôt et visitant un ensemble de clients de demande totale inférieure à Q.

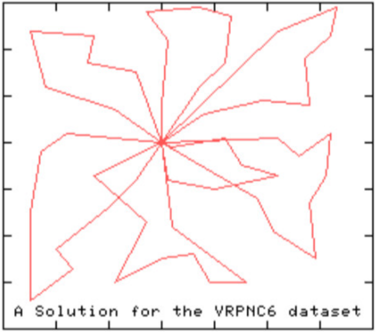
Objectif : déterminer un ensemble de tournées pour servir tous les clients, à coût total minimal

Le CVRP est NP-difficile car il généralise à la fois le TSP et le bin-packing.

TSP vs. tournées



TSP – Instance EIL51 de la TSPLIB
$n = 51,\ z^* = 426$



CVRP – Instance CMT n° 6
$n = 50,\ z^* = 555.43$, 7 tours

Problèmes de tournées sur arcs

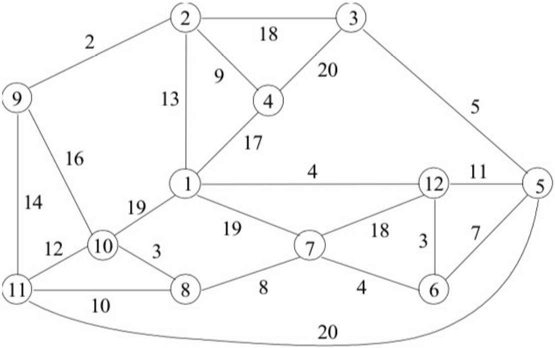
Le capacitated arc routing problem est l'équivalent du CVRP

Il faut traiter un sous-ensemble d'arêtes E_R avec des demandes $q(i,j)$ et des coûts de traitement $s(i,j)$.

Les meilleures méthodes exactes gèrent directement le problème sur arc et ne cherche pas à le transformer en problème sur sommets.

Cas particulier NP-difficile même à un véhicule : problème du postier rural (RPP) : traverser au moins une fois chaque arête de E_R à meilleur coût.

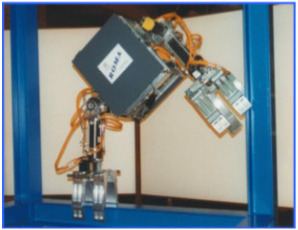
Exemple d'instance CARP



Village avec 22 rues et leurs longueurs.
Dépôt au sommet 1 avec des véhicules de 5 tonnes.
Chaque rue nécessite une tonne de sel.

Exemple atypique

Problème traité par l'Université de Valence.
Inspection des structures 3D par un robot télécommandé.



Minimiser la consommation d'énergie (monter est plus coûteux !)

Problème de postier rural venteux !

Complications possibles

Réseau :

- graphes orientés ou mixtes
- interdiction de tourner
- arêtes "venteuses"
- multi-graphe
- temps de parcours dépendant de l'heure
- dépôts multiples

Véhicules

- flotte hétérogène
- autonomie des véhicules
- compartiments, remorques
- temps de travail

Clients :

- contraintes d'accès
- fenêtres de temps
- visites multiples
- ordre partiel sur les clients
- visites synchronisées
- demandes stochastiques
- collecte et livraison

Divers :

- horizon multi-période
- grande taille
- location-routing
- inventory-routing

Intérêt des problèmes de tournées

Grande importance économique

- partie aval des chaînes logistiques (livraison)
- partie amont (collecte)
- bien d'autres applications

L'optimisation permet de réduire les coûts (km, durées), mais aussi l'impact environnemental (carburant, bruit, ...).

Une littérature scientifique très foisonnante (croissance "exponentielle") : 480 références pour 1960-1999, 863 entre 2000 et 2006, 3545 entre 2007 et 2013 (selon SCOPUS)

Sommaire

Modèles à flots de véhicules : rappel TSP

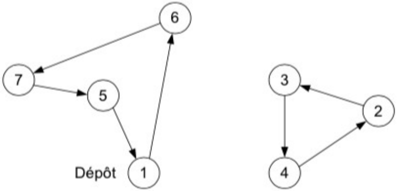
$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\sum_{j\in V} x_{ij} = 1, \quad \forall i \in V$$

$$\sum_{j\in V} x_{ji} = 1, \quad \forall i \in V$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in A$$

Sous-tours



Méthode MTZ (Miller-Tucker-Zemlin)

$$t_i \geq 0, \quad \forall i = 1, \ldots, n$$
$$t_i + c_{ij} \leq t_j + M.(1 - x_{ij}), \quad \forall i = 1, \ldots, n, j = 2, \ldots, n, i \neq j$$

Méthode classique

$$\sum_{(i,j) \in s} x_{ij} \leq |s| - 1, \quad \forall s \in S$$

Modèle pour le CVRP asymétrique

Adaptation du modèle TSP : ajout d'un indice pour les véhicules.

Note : il existe un modèle sans indice de véhicules.

$$
\begin{aligned}
\min z = \sum_{i=0}^{n} c_{ij} x_{ij} & \\
\sum_{j=0, j\neq i}^{n} x_{ij} = 1, & \quad \forall i = 1, \ldots, n \\
\sum_{j=0, j\neq i}^{n} x_{ji} = 1, & \quad \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{0i} \leq m & \\
q_i \leq t_i \leq Q, & \quad \forall i \in 1, \ldots, n \\
t_i + q_j \leq t_j + Q(1 - x_{ij}), & \quad \forall i = 0, \ldots, n, j = 0, \ldots, n, i \neq j \\
x_{ij} \in \{0, 1\}, & \quad \forall i = 0, \ldots, n, j = 0, \ldots, n, i \neq j
\end{aligned}
$$

Modèle pour le CVRP classique

Variables : x_e : nombre de traversées de l'arête e.

$$\min z = \sum_{e \in E} c_e x_e$$

$$\sum_{e \in \delta(i)} x_e = 2, \quad \forall i \neq 0$$

$$\sum_{e \in \delta(0)} x_e = 2m$$

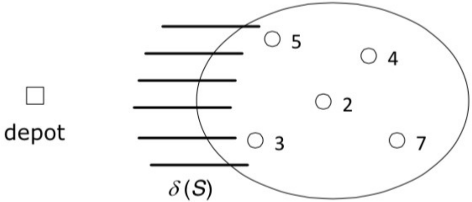
$$\sum_{e \in \delta(S)} x_e \geq 2 \left\lceil \sum_{i \in S} q_i / Q \right\rceil, \quad \forall S \in \mathcal{S}$$

$$x_e \in \{0, 1\}, \quad \forall e \notin \delta(0)$$

$$x_e \in \{0, 1, 2\}, \quad \forall e \in \delta(0)$$

Comprendre les contraintes

Les contraintes $\sum_{e\in\delta(S)} x_e \geq 2\lceil\sum_{i\in S} q_i/Q\rceil$ empêchent les sous-tours et servent de contraintes de capacité.



S sous-ensemble de 5 clients :

- demande totale 21
- capacité des véhicules $Q = 10$
- d'où au moins 3 véhicules pour S

Il faut au moins 6 arêtes dans $\delta(S)$.

Résolution

Les modèles à flots de véhicules peuvent être résolus par :

- résolution directe par un solveur. C'est possible avec les modèles MTZ, mais limité à $n \simeq 15$
- branch-and-bound. Bornes inférieures basées sur des arbres couvrants ou relaxation langrangienne des contraintes de sous-tour : $n \simeq 40$
- branch-and-cut : on relâche les contraintes difficiles et on ajoute les contraintes si nécessaire : $n \simeq 80$

Méthode de branch-and-cut

Méthode de recherche arborescente.

On renforce la borne inférieure en ajoutant des inégalités valides.

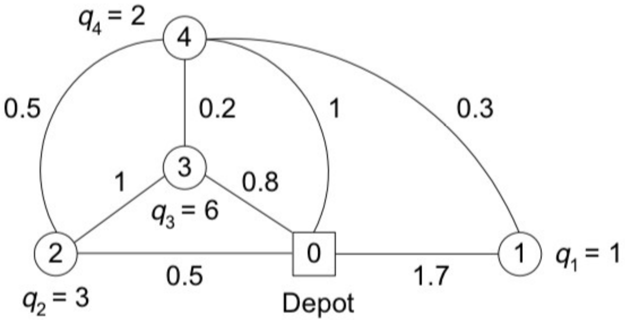
Schéma de séparation : fixer à 1 ou 0 une variable fractionnaire.

Nombreuses inégalités valides (comb, hypotour, multistar, ...)

- Il faut détecter celles qui sont violées dans la solution courante (problème de séparation)
- ce problème peut être NP-difficile (par exemples les contraintes de capacité)
- on utilise alors une heuristique
- on l'applique sur le graphe support, qui contient une arête e si $x_e \neq 0$ dans la solution courante.

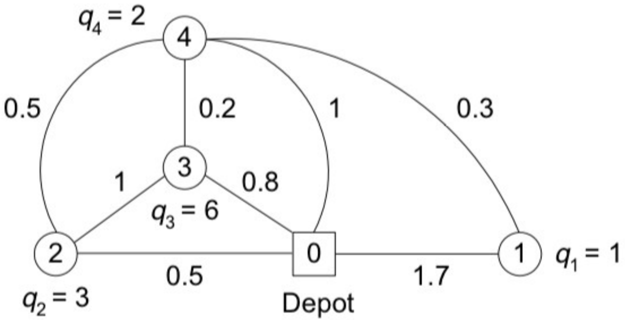
Violation de contrainte de capacité

Cas à 4 clients et 2 véhicules de capacité 10. Graphe support :



Violation de contrainte de capacité

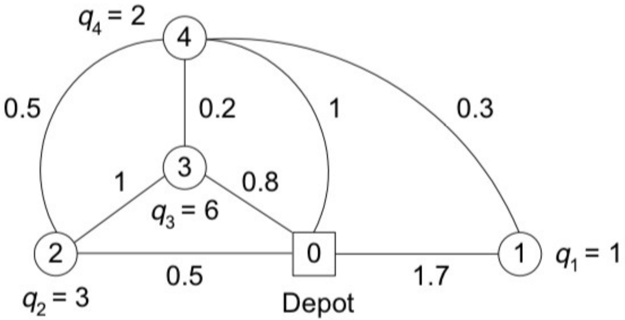
Cas à 4 clients et 2 véhicules de capacité 10. Graphe support :



Pour $S = \{2, 3, 4\}$, il faut 2 véhicules : second membre = 4. Or la somme vaut $2,6$.

Violation de contrainte de capacité

Cas à 4 clients et 2 véhicules de capacité 10. Graphe support :



Pour $S = \{2, 3, 4\}$, il faut 2 véhicules : second membre = 4. Or la somme vaut $2, 6$.
On peut ajouter la contrainte $x_{41} + x_{40} + x_{30} + x_{20} \geq 4$ au PL.

Modèles set-covering

On considère l'ensemble R des tournées possibles. Pour chaque route r, on crée une variable y_r indiquant si on choisit cette route ou non.

Modèles set-covering

On considère l'ensemble R des tournées possibles. Pour chaque route r, on crée une variable y_r indiquant si on choisit cette route ou non.

Problème de couverture.

$$\min z = \sum_{r \in R} c_r y_y$$

$$\sum_{r \in R} a_{ir} y_r \geq 1, \forall i \in V \setminus \{0\}$$

$$\sum_{r \in R} y_r = m$$

$$y_r \in \{0,1\}, \forall r \in R$$

Modèle très flexible, pouvant gérer de nombreuses contraintes.

Résolution par génération de colonnes

Le modèle est très fort, mais il a $O(2^n)$ variables !

On va le résoudre en générant uniquement les variables nécessaires.

Algorithme de génération de colonnes

- Choisir un petit ensemble initial de variables S
- Résoudre le problème avec les variables de S
- Chercher quelle serait la variable non générée la plus intéressante (meilleur coût réduit)
- Si on en trouve une, alors on l'ajoute.
- Sinon on s'arrête.

Si les variables sont fractionnaires, on doit utiliser une méthode de branchement (branch&price)

Sous-problème

Comment générer la colonne la plus attractive dans le modèle ?

On cherche celle qui va avoir le meilleur coût réduit.

$c_k + \sum_{i \in V \setminus \{0\}} a_{ik} \lambda_i - \lambda_0 < 0$

Problème de plus court chemin élémentaire avec contraintes de ressources entre deux copies du dépôt avec des coûts $c_{ij} - \lambda_i$ sur les arcs.

NP-difficile au sens fort, mais on peut utiliser la programmation dynamique pour le résoudre en pratique

Pour des grandes instances, on peut aussi générer des tournées non élémentaires.

Extensions : tournées presque élémentaires

Quand on autorise les tournées non élémentaires, la relaxation est plus faible.

Les techniques les plus récentes utilisent des versions intermédiaires :

- tournées sans k-cycles : on peut repasser par i uniquement après avoir visité au moins k autres sommets (mémoire locale)
- ng-routes : on peut repasser par i uniquement après être passé par un sommet "non voisin". La notion de voisinage est paramètrée par ng : on sélectionne les ng sommets les plus proches.

En pratique, il est rarement nécessaire d'utiliser des grandes valeurs de k ou ng pour obtenir de bonnes solutions.

Résultats les plus récents

Les méthodes les plus récentes sont basées sur la génération de colonnes.

Fukasawa et al. (2006). Branch, cut, price, k-cycles

Baldacci et al. (2011). ng-routes, pas de branchement, génération de toutes les routes attractives, et résolution directe par un solveur pour finir.

Pecin et al. (2015). Branch, cut, price, avec tous les raffinements de la littérature. Permet de fermer toutes les instances de la littérature jusque 200 sommets !

Bilan méthodes exactes

Pour des jeux de données de bonne taille, les méthodes exactes arrivent à résoudre de nombreuses variantes de problèmes de tournées.

Les modèles directs ne sont pas efficaces.

Les méthodes plus complexes demandent un travail conséquent d'implémentation aujourd'hui.

Futur proche : bibliothèques libres pour implémenter de manière efficace les méthodes de branch&price

Difficulté : le sous-problème de génération des variables reste ad-hoc dans le cas général et nécessite un soin très particulier.

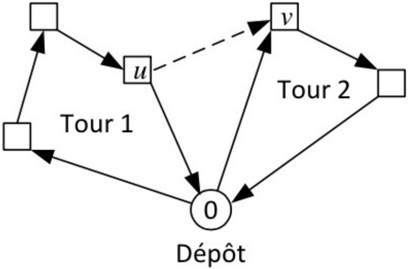
Sommaire

Heuristiques constructives

Il existe quatre grandes familles de méthodes.

- Fusion de tournées
- Construction séquentielle ou parallèle de tournées par insertions
- Cluster first - route second.
- Route first - cluster second.

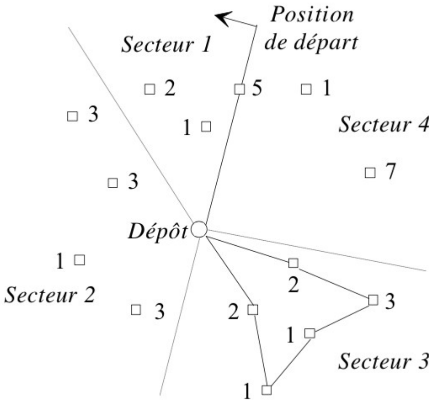
Heuristique de Clarke et Wright



On crée une tournée par client, puis on effectue à chaque itération la fusion de deux tournées qui minimise la variation de coût. $\Delta_{uv} = c(u,v) - c(u,0) - c(0,v)$

Il existe des implémentations en $O(n^2 \log n)$.

Cluster-first, route second



Sweep heuristic, Gillette et Miller, 1974

- Tri des clients par angle polaire croissant par rapport au dépôt
- Création d'un secteur angulaire / véhicule
- Application d'une heuristique de TSP pour construire les tournées

Route-first, cluster second

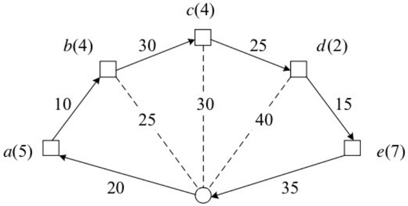
Idée principale : construire d'abord un "tour géant", puis le découper en sous-tours.

Cette heuristique nécessite :

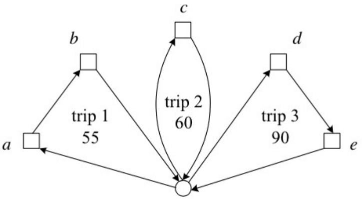
- une heuristique pour le TSP
- une méthode de découpage

Dans de nombreux cas, on peut utiliser la programmation dynamique pour découper le tour !

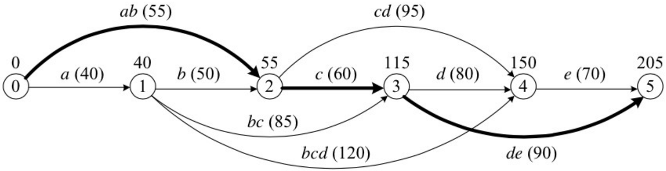
Procédure de découpage



1. Giant tour $T = (a,b,c,d,e)$



3. Optimal splitting, cost 205



2. Auxiliary graph H of possible trips for $Q = 10$ (shortest path in boldface)

Remarques sur le découpage

Un tour optimal pour le TSP ne donne par forcément un optimum pour le CVRP : des heuristiques fonctionnent aussi bien.

Le découpage est optimal pour l'ordre défini par le tour géant et coûte $O(n^2)$. On peut gérer de nombreuses contraintes en gardant une complexité polynomiale.

La méthode est désormais très utilisée (plus de 74 articles avant 2014).

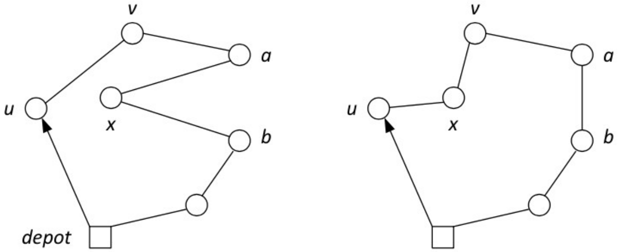
Recherche locale

Toutes les bonnes méthodes méta-heuristiques pour les VRP sont basées sur des recherches locales.

Il existe de nombreux voisinages pour les tournées :

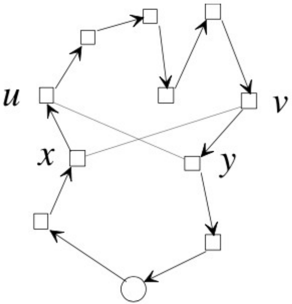
- relocate
- 2-opt sur une tournée
- k-opt sur une tournée
- 2-opt* sur deux tournées

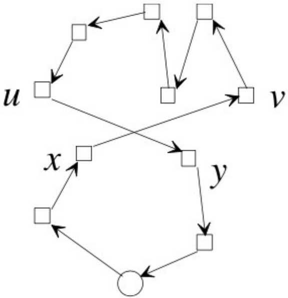
Relocate dans une tournée



Déplacement d'un client x après un sommet u :
$|N(s)| = O(n^2)$
Pas de probème de faisabilité pour le CVRP
$\Delta = c(u,x) + c(x,v) + c(a,b) - c(u,v) - c(a,x) - c(x,b)$, calculable en $O(1)$.

2-opt sur une tournée



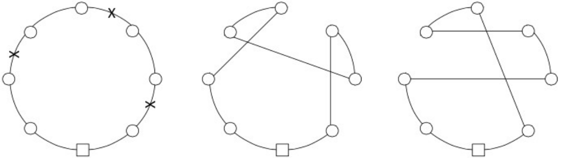


On remplace deux arcs par deux autres.
$|N(s)| = O(n^2)$

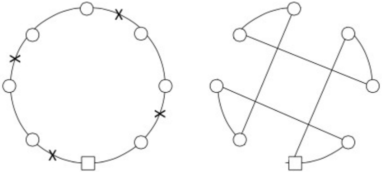
Pas de probème de faisabilité pour le CVRP.

$\Delta = c(x, v) + c(u, y) - c(x, u) - c(v, y)$, calculable en $O(1)$.

3-opt et 4-opt sur une tournée



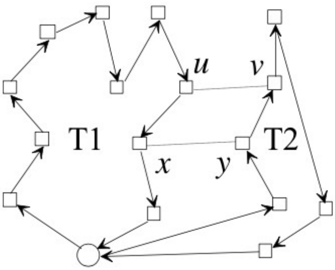
$3 - opt$: plusieurs reconnexions possibles.

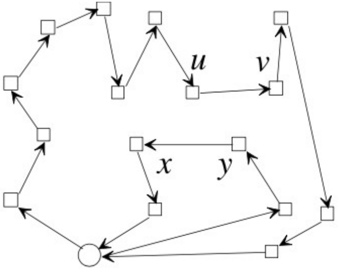


$4 - opt$, version double-bridge.

La complexité augmente vite, en $O(n^k)$ pour $k-$opt.

2-opt* sur deux tournées





Ou "crossover move" : $|N(s)| = O(n^2)$
$\Delta = c(u,v) + c(y,x) - c(u,x) - c(y,v)$

Test de capacité en $O(1)$ au lien de $O(n)$ si on pré-calcule pour chaque tournée T sa charge $load(T)$ et pour chaque sommet i, la demande cumulée $load(T,i)$ jusque i.

Recherche locale avec pré-calculs

Tour T, fenêtres $[b_i, e_i]$, heures d'arrivéer t_i. Insertion avant T_4.
Pré-calcul du retard maximum S_k du client T_k sans violer une fenêtre ensuite (w_i désigne l'attente à T_i).

$S_r = e_r - t_r + w_r$

Pour $i = r, r-1, \ldots, k$, $S_i = \min(S_{i+1}, e_i - t_i + w_i)$

L'ensemble de ces marges est pré-calculable en $O(n)$. Ici une insertion de retard θ est possible si $\theta \leq S_4 = 2$.
Calculable en $O(1)$.

Que précalculer ?

On peut généraliser la technique de précalcul en décomposant les mouvements en opérations de concaténations portant sur des séquences de sommets.

Pour chaque critère utile Z, on précalcule $Z(\sigma)$ pour chaque séquence de sommets σ dans la solution courante.

Il faut quelques opérateurs.

- Initialisation : calcule $Z(\sigma)$ si σ a un seul sommet
- Ajout en tête : déduit Z si un sommet x est ajouté devant σ
- Ajout en queue : déduit Z si x est ajouté après σ
- Concaténation : déduit Z si deux séquences σ et σ' sont concaténées ($\sigma \oplus \sigma'$).

Exemple de précalcul : CVRP

Quantités à pré-calculer : charge $Q(\sigma)$ et longueur $L(\sigma)$.

Initialisation : si $\sigma = (x)$, $Q(\sigma) = q(x)$ et $L(\sigma) = 0$

Ajoute en queue : $Q(\sigma \oplus x) = Q(\sigma) + q(x)$
$L(\sigma \oplus x) = L(\sigma) + c(last(\sigma), x)$

Concaténation : $Q(\sigma \oplus \tau) = Q(\sigma) + Q(\tau)$
$L(\sigma \oplus x) = L(\sigma) + c(last(\sigma), first(\tau)) + L(\tau)$

L'initilisation et l'ajout en queue permettent de faire en $O(n^2)$ les précalculs pour toutes les séquences !

Si on insère une chaîne de clients α entre les deux portions σ et τ d'une tournée, on calcule facilement Q et L pour $\sigma \oplus \alpha \oplus \tau$.

Exemple de précalcul : logistique humanitaire

Critère d'urgence : minimiser la somme des heures d'arrivée aux sites affectés

Temps de déplacement :		10		5		8		6	
Noeud :	0	$\rightarrow$	2	$\rightarrow$	1	$\rightarrow$	3	$\rightarrow$	0
Heure d'arrivée :			10		15		23		29

Coût : 10 + 15 + 23 = 48.
Une insertion décale les heures d'arrivée suivantes et une séquence inversée change de coût.
On introduit :
$D(\sigma)$: durée pour visiter les sommets de σ $C(\sigma)$: coût cumulatif pour un départ au temps 0 $W(\sigma)$: surcoût pour un décalage de 1 de l'heure de départ
Initialisation : $D(x) = 0$, $C(x) = 0$, $W(x) = 1$
Concaténation : $D(\sigma \oplus \tau) = D(\sigma) + d(\mathit{last}(\sigma), \mathit{first}(\tau)) + D(\tau)$
$C(\sigma \oplus \tau) = C(\sigma) + W(\tau) \times [D(\sigma) + d(\mathit{last}(\sigma), \mathit{first}(\tau))] + C(\tau)$
$W(\sigma \oplus \tau) = W(\sigma) + W(\tau)$

Sommaire

Méthodes les plus efficaces à ce jour

Méthodes exactes : branch and cut and price. Principalement quand le problème peut bien être décomposé en tournées indépendantes. Sinon, branch and cut peut être utile.

Heuristiques : hybrides entre recherche locale et algorithme génétiques, mais aussi recherche à grands voisinages.

Les instances historiques sont désormais très bien résolues par les méthodes de la littérature.

Tournées dans les services

De nouveaux problèmes de tournées apparaissent avec le développement des services.

Soins à domicile : synchronisation

Relevés de de compteurs à distance

Orienteering problems : version sac-à-dos de l'objectif

Transport de fonds, gardiennage : tournées changeant tous les jours

Considérations envrionnementales

Minimisation des GES

Tournées de véhicules électriques

Problèmes multi-échelon

Collecte sélective des déchets

Réduction des pesticides (tournées sur des parcelles)

Logistique humanitaire

Intérêt croissant.

- récents désastres
- création d'un center pour la logistique humanitaire à Georgia Tech

Problèmes cumulatifs ; minimisation du temps moyen de secours

Problèmes avec plusieurs objectifs

VRP généralisés : tournées choisissant un point de livraison dans chaque région

Problèmes riches

Problèmes combinant plusieurs classiques et/ou considérant plus de contraintes. On veut résoudre ave un seul code un problème plus réaliste et tous ses cas particuliers classiques.

Exemples :

- multi-dépôt et périodique
- flotte hétérogène et retours et fenêtres de temps
- collecte et ramassage
- ...

Problèmes riches

Combiner le VRP avec un autre problème d'optimisation :

- Planification de la production
- Gestion et localisation des stocks
- Chargement d'objets 2D et 3D

Combinaison de problèmes de tournées sur sommets et sur arcs.

Données incertaines

Temps de trajet, quantités, présence/absence d'un client, etc.

VRP temps réel et dynamique : donnéees révélées au cours du temps (client imprévus). On peut parfois réoptimiser.

Approches stochastiques (avec lois de proba)

- demandes aléatoires
- durée de trajet et de service

Approches d'optimisation robuste

- ensembles d'incertitude
- demandes incertaines
- temps de trajet incertains