Solutions Hybrides pour le Problème de Tournées de Véhicules

Sohaib Afifi et Aziz Moukrim

LGI2A

Roadef 2016

Plan

- Le problème
  - Définition du problème
  - Notations
  - Formulation mathématique
  - Nouvelle formulation
- Approches de résolution
  - Schéma global
  - Approches heuristiques
  - Bornes inférieures et coupes
- Expérimentation
  - Paramètres
  - Résultats
- Conclusions et perspectives

Définition du problème

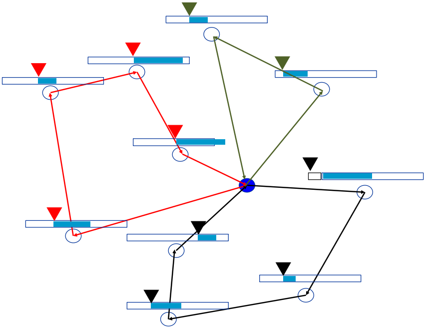
CVRPTW : un problème de tournées de véhicules +

- **des contraintes de fenêtres de temps**

  Pour chaque client on définit une fenêtre de temps dans laquelle le service doit être effectué.

- **des contraintes de capacité**

  Véhicules avec des capacités limitées pour servir les clients (demandes).



L'objectif

- Minimiser le nombre de véhicules utilisées
- Minimiser le coût des trajets
- Équilibrer les charges

Définition du problème - Notations

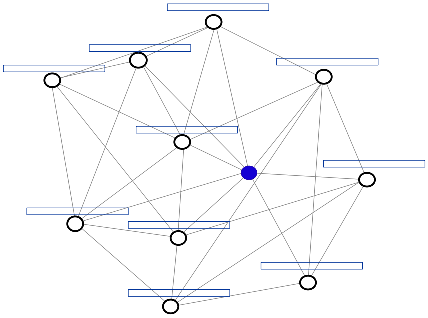
$\mathbf{G}(\mathbf{V}^+, \mathbf{A})$: graphe orienté
$\mathbf{V}$: ensemble de n clients
$\mathbf{A}$: ensemble des arcs

$\mathbf{K}$: ensemble de m véhicules

$\mathbf{T_{i,j}}$: temps de trajet entre i et j

$\mathbf{D_i}$: durée de service de i
$[\mathbf{e_i}, \mathbf{l_i}]$: fenêtre de temps
$\mathbf{Tmax}$: limite d'une tournée

$\mathbf{q_i}$: demande du client i
$\mathbf{Q}$: capacité des véhicules



Formulation mathématique

$$\mathbf{x_{ijk}} = \begin{cases} 1 & \text{Si véhicule } k \in K \text{ utilise l'arc } (i,j) \in A \\ 0 & \text{sinon} \end{cases}$$

$\mathbf{t_{ik}}$ = Début du service au client i par le véhicule k

$$\sum_{k \in K} \sum_{j:(i,j) \in A} x_{ijk} = 1 \qquad \forall i \in V \tag{1}$$

$$\sum_{j:(i,j) \in A} x_{ijk} - \sum_{j:(j,i) \in A} x_{jik} = 0 \qquad \forall i \in V \quad \forall k \in K \tag{2}$$

$$\sum_{i} q_i \times \sum_{j} x_{ijk} \leq Q \qquad \forall k \in K \tag{3}$$

$$t_{ik} + (T_{ij} + D_i) x_{ijk} \leq t_{jk} + l_i (1 - x_{ijk}) \qquad \forall (i,j) \in A \quad \forall k \in K \tag{4}$$

$$e_i \sum_{j:(i,j) \in A} x_{ijk} \leq t_{ik} \leq l_i \sum_{j:(i,j) \in A} x_{ijk} \qquad \forall i \in V \quad \forall k \in K \tag{5}$$

Un modèle avec $\mathbf{m} \times \mathbf{n^2}$ variables et contraintes

Nouvelle formulation - Variables

Passage à un modèle avec n^2 variables

$$\begin{matrix} x_{ijk} \\ t_{ik} \end{matrix} \longrightarrow \begin{matrix} \mathbf{z_{ij}} \\ \mathbf{r_i} \end{matrix}$$

Introduction des variables $\mathbf{y_{ik}}$:

$$\mathbf{y_{ik}} = \begin{cases} 1 & \text{si la visite } i \text{ est effectuée par le véhicule } k \\ 0 & \text{sinon} \end{cases}$$

Nouvelle formulation - Contraintes

$$\sum_{j:(i,j)\in A} z_{ij} - \sum_{j:(j,i)\in A} z_{ji} = 0 \qquad \forall i \in V \tag{6}$$

$$e_i \leq r_i \leq l_i \qquad \forall i \in V \tag{7}$$

$$r_i + (D_i + T_{ij})x_{ij} \leq r_j + l_i(1 - x_{ij}) \qquad \forall (i,j) \in A,\ \forall k \in K \tag{8}$$

$$\sum_{i\in V} q_i \times y_{ik} \leq Q \qquad \forall k \in K \tag{9}$$

$$\sum_{k\in K} y_{ik} = 1 \qquad \forall i \in V \tag{10}$$

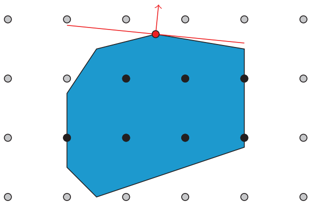
$$z_{ij} = 1 \Rightarrow \forall k \in K \quad y_{ik} = y_{jk} \qquad \forall (i,j) \in A \tag{11}$$

$$z_{0i} = z_{0j} = 1 \Rightarrow \forall k \in K \quad z_{ik} \neq z_{jk} \qquad \forall i,j \in V \tag{12}$$

Approches de résolution

Méthodes exactes

- Branch and bound
- Cutting planes
- Branch and cut



Méthodes heuristiques

- Des bonnes solutions
- Un temps de calcul raisonnable
- Sans garantie d'optimalité
- $\rightsquigarrow$ Intégration dans un solveur exact

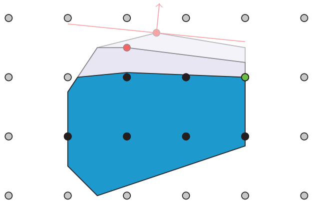
Approches de résolution

Méthodes exactes

- Branch and bound
- Cutting planes
- Branch and cut

Méthodes heuristiques

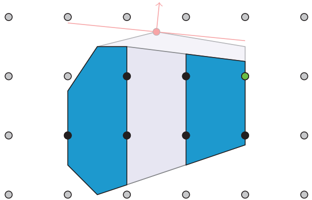
- Des bonnes solutions
- Un temps de calcul raisonnable
- Sans garantie d'optimalité
- $\rightsquigarrow$ Intégration dans un solveur exact



Approches de résolution

Méthodes exactes

- Branch and bound
- Cutting planes
- Branch and cut



Méthodes heuristiques

- Des bonnes solutions
- Un temps de calcul raisonnable
- Sans garantie d'optimalité
- $\rightsquigarrow$ Intégration dans un solveur exact

Schéma global

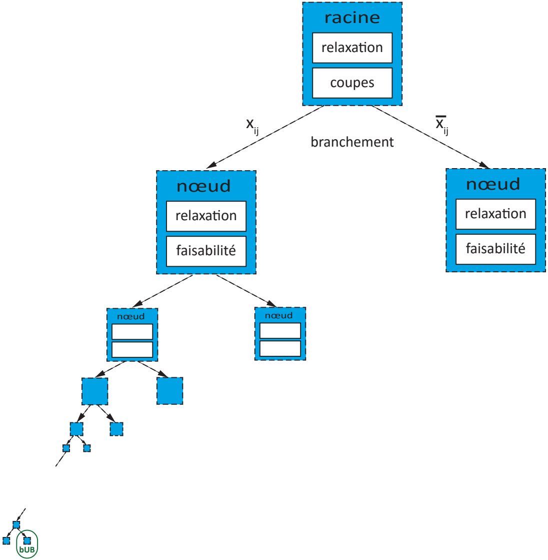


Schéma global

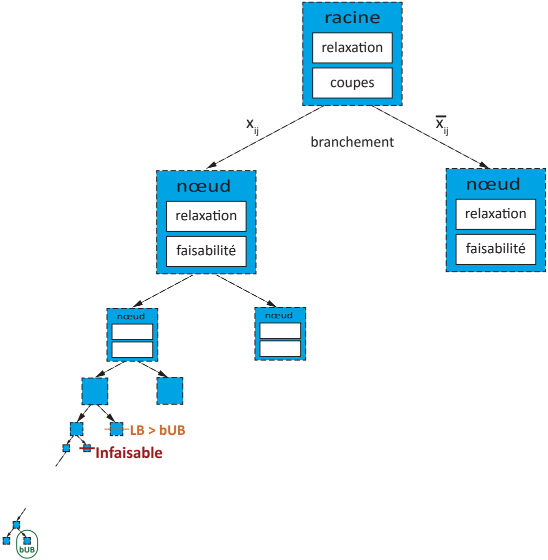


Schéma global

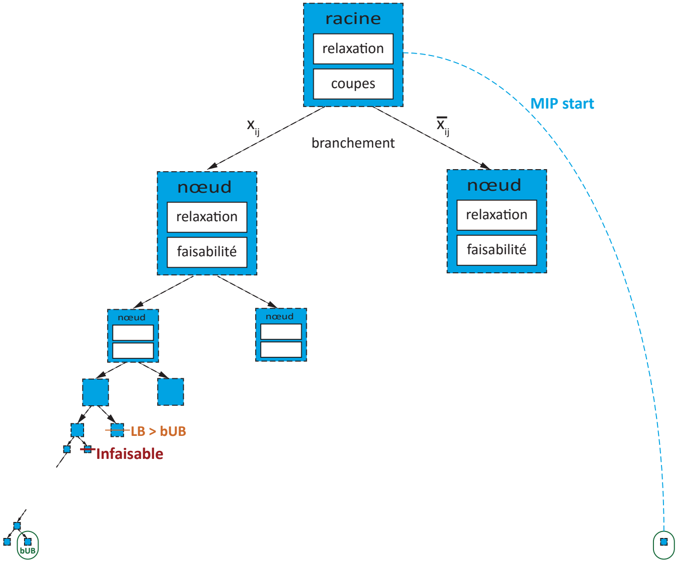


Schéma global

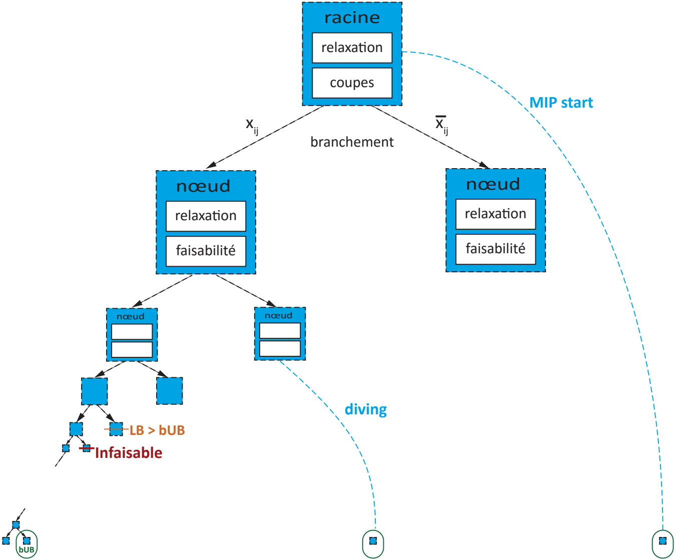
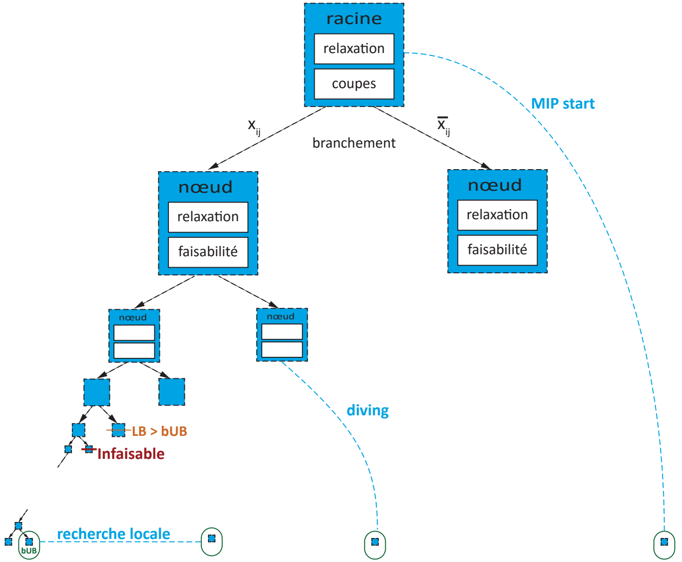


Schéma global

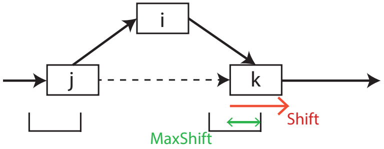


Approches heuristiques - Heuristique constructive

Un algorithme d'insertion à la meilleure position

- Pré-calculer le décalage maximal autorisé pour chaque visite $\rightarrow$ un coût d'évaluation en $\mathcal{O}(1)$

$$\textsc{MaxShift}_{r(p)} \leftarrow \min \begin{cases} l_{r(p)} - \textsc{Start}_{r(p)} \\ \textsc{Wait}_{r(p+1)} + \textsc{MaxShift}_{r(p+1)} \end{cases}$$



- Un algorithme d'insertion **`Insert(S)`** en $\mathcal{O}(n^2)$
- Un algorithme d'initialisation **`Initilize()`**$\equiv$**`Insert(`**$\emptyset$**`)`** en $\mathcal{O}(n^3)$

Approches heuristiques - Recherche locale

Dans l'espace de recherche défini par $G(V, A)$:

- **Exchange** : Échanger les positions de deux clients
- **OrOpt** : Déplacer une séquence dans la même tournée
- **2-Opt*** : Croiser deux arcs de deux tournées différentes
- **SingleMove** : Changer la position d'un client
- **Destruction/Réparation** : Replacement d'un nombre aléatoire de clients

Bornes inférieures et coupes

Vérification de la faisabilité par rapport aux bornes inférieures sur **le nombre de véhicules** nécessaires pour visiter tous les clients

- Clique
- Bin-packing
- Raisonnement énergétique

Bornes inférieures classiques - Cliques

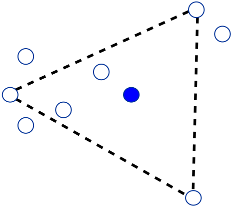
Construction du graphe des incompatibilités entre clients défini par : $G^V_{inc} = (V, E_V)$ où $E_V = \{(i, j) \in A^{noeud} : i \| j\}$.

n

n

0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	0	0	0
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0

$G^V_{inc}(V, E_V)$

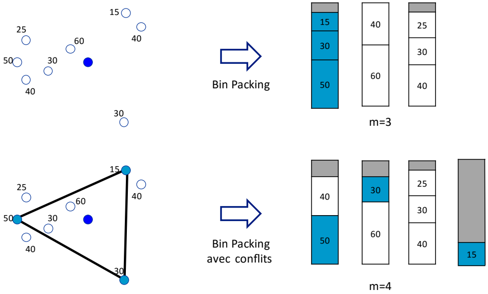


LB_{Clique}

La taille de **la clique maximum** du graphe G^V_{inc} notée LB_{Clique} est une borne inférieure sur le nombre de routes pour VRPTW.

Bornes inférieures classiques

Chaque véhicule est considéré comme une boîte de taille Q et chaque demande d'un client comme un article de taille q_i.



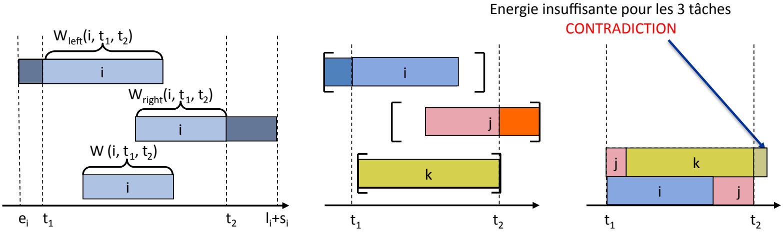
LB_{BP}

Toute borne inférieure LB_{BP} sur le nombre de boîtes nécessaires pour ranger ces articles est une borne inférieure valide pour le VRPTW.

Raisonnement énergétique - Principe sur PMSP

Principe sur PMSP (problème d'ordonnancement sur machines parallèles) :

- Détecter l'infaisabilité d'une instance pour un nombre de machines donné.
- Ajuster les fenêtres de temps.

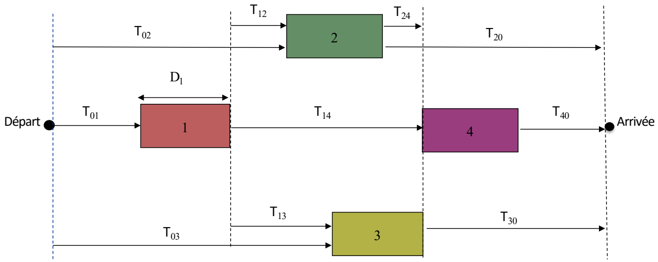


Test de faisabilité

si $\exists[t_1, t_2],\ W(t_1, t_2) > m \times (t_2 - t_1)$ alors l'instance est **infaisable**.

Raisonnement énergétique - Adaptation au mVRPTW

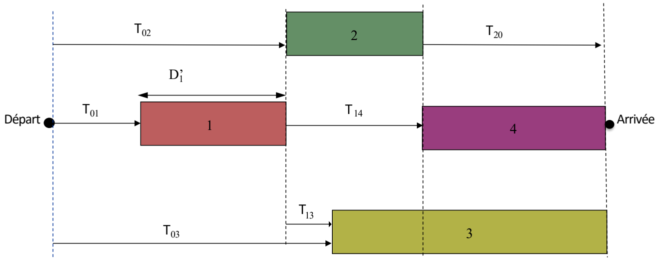
- Relaxation de l'instance du m-VRPTW
- Intégration des temps de trajets
- Ajout de tâches fictives pour le départ et l'arrivée



Un exemple d'instance mVRPTW

Raisonnement énergétique - Adaptation au mVRPTW

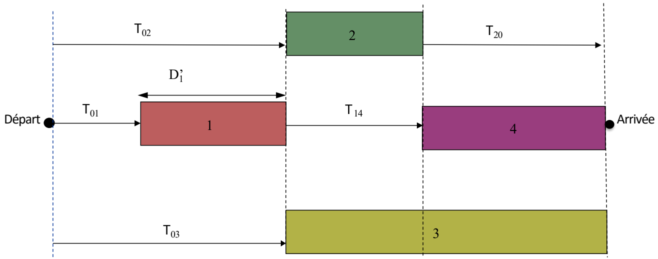
- Relaxation de l'instance du m-VRPTW
- Intégration des temps de trajets
- Ajout de tâches fictives pour le départ et l'arrivée



Un exemple d'instance mVRPTW

Raisonnement énergétique - Adaptation au mVRPTW

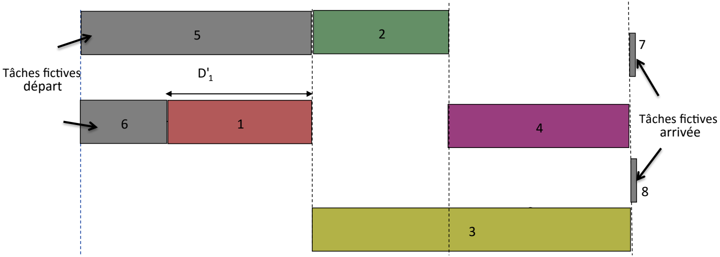
- Relaxation de l'instance du m-VRPTW
- Intégration des temps de trajets
- Ajout de tâches fictives pour le départ et l'arrivée



Un exemple d'instance mVRPTW

Raisonnement énergétique - Adaptation au mVRPTW

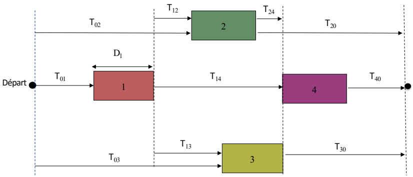
- Relaxation de l'instance du m-VRPTW
- Intégration des temps de trajets
- Ajout de tâches fictives pour le départ et l'arrivée



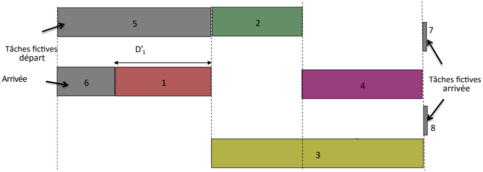
Un exemple d'instance mVRPTW

Raisonnement énergétique - Adaptation au mVRPTW

- Relaxation de l'instance du m-VRPTW
- Intégration des temps de trajets
- Ajout de tâches fictives pour le départ et l'arrivée

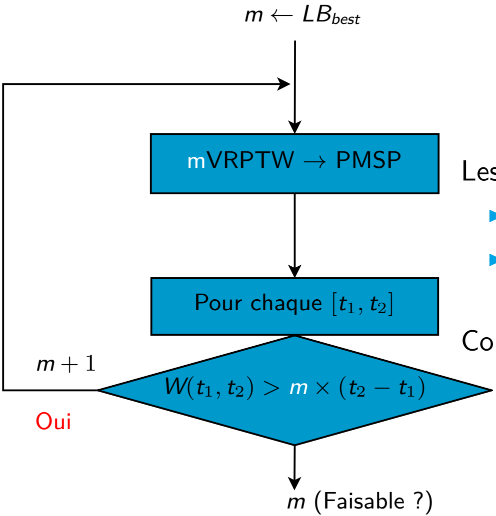


Un exemple d'instance mVRPTW



Les activités pour PMSP

Raisonnement énergétique - L'algorithme



Les intervalles :

- $t_1 \in \{e_i, l_i, e_i + s_i, i \in V\}$
- $t_2 \in \{l_i + s_i, e_i + s_i, l_i, i \in V\}$

Complexité : $\mathcal{O}(n^3)$

Expérimentation

Paramètres :

Mip Start

- Schéma de la méta heuristique
- Nombre d'itérations
- Temps limite

Recherche locale

- Nombre d'itérations, tentatives
- Temps limite

Diving

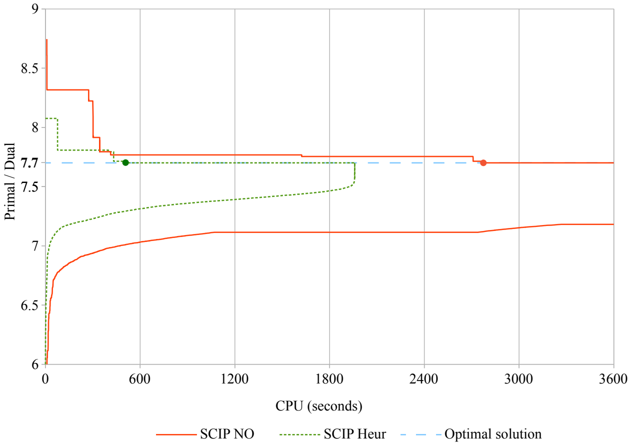
- Nombre d'itérations
- Profondeur

Coupes

- Profondeur
- Densité du graphe

Résultats

Exemple



Conclusions et perspectives

Conclusions

- Un nouveau modèle pour le CVRPTW
- Une méthode hybride (schéma exacte + heuristiques d'accélération + coupes dédiées)
- Des solutions **optimales** plus rapidement

Perspectives

- Études des paramètres
- Testes sur des grandes instances
- Développement d'une boite noire
- Traitement d'autres variantes

Merci