



**HAL**  
open science

## Problème de tournée avec pénalités de départ et de retard

Hugo Chevrotton, Yannick Kergosien, Jean-Charles Billaut

► **To cite this version:**

Hugo Chevrotton, Yannick Kergosien, Jean-Charles Billaut. Problème de tournée avec pénalités de départ et de retard. ROADEF 2019, Feb 2019, Le Havre, France. hal-02087851

**HAL Id: hal-02087851**

**<https://hal.science/hal-02087851>**

Submitted on 2 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Problème de tournée avec pénalités de départ et de retard

Hugo Chevroton, Yannick Kergosien, Jean-Charles Billaut

Université de Tours

LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002, 64 avenue Jean Portalis, 37200 Tours

hugo.chevroton@univ-tours.fr, yannick.kergosien@univ-tours.fr,

jean-charles.billaut@univ-tours.fr

**Mots-clés :** *Livraison, matheuristique, modélisation*

## 1 Introduction

Cet article aborde un problème de tournée où un ensemble de clients doit être livré à un ensemble de dates souhaitées. Si cette date est dépassée, le livreur effectuant la tournée verse au client un dédommagement proportionnel à son retard, appelé pénalité de retard. La livraison doit être effectuée en une seule tournée. Le coût de la tournée est égal à la distance totale parcourue.

Comme autre aspect du problème, cette tournée s'effectue dans un contexte plus large qui implique que la date de départ du véhicule n'est pas connue. Celle-ci est une variable de décision à prendre dans une fenêtre de temps  $[a, b]$ . Un coût externe s'applique et est fonction de la date de départ choisie.

L'objectif de ce problème est de minimiser le coût total composé des pénalités de retard, du coût de la tournée et du coût externe.

La version de ce problème considérant une date de départ fixe et ne prenant pas en compte le coût de tournée peut être assimilée à un problème d'ordonnancement noté  $1|s_{ij}|\sum w_i T_i$  selon la notation de Graham [1] (où  $T_i$  est le retard de la tâche  $i$ ,  $w_i$  son poids, et  $s_{i,j}$  indique un temps de setup dépendant de la séquence). Ce problème est prouvé NP-difficile [2].

La prise en compte des pénalités de retard est peu abordée dans la littérature des problèmes de livraison. On peut néanmoins citer quelques articles comme [3] et [4] qui considèrent des fenêtres de temps sur lesquelles les clients doivent être servis et où les dépassements génèrent des pénalités.

Depuis quelque années, les problèmes intégrant simultanément des problématiques d'ordonnancement et de livraison sont de plus en plus étudiés. Par exemple, dans [5], un fabricant produit des commandes pour ses clients et les fait livrer par une flotte de véhicules. Dans [6], les auteurs abordent un cas concret d'impression et de distribution de journaux aux États-Unis. Cependant, dans l'état de nos connaissances, il n'existe pas dans la littérature d'article où les auteurs cherchent explicitement à résoudre un problème de tournée sur une fenêtre de date de départ variable.

Le concept général de la méthode est présenté dans la Section 2 de l'article. La Section 3 détaille la deuxième partie de la méthode et présente des heuristiques. La Section 4 détaille les résultats expérimentaux.

## 2 Approche globale de résolution

La méthode proposée se décompose en deux étapes. La première étape a pour objectif d'évaluer l'évolution des coûts de la tournée et des pénalités de retard en fonction de la date de départ du véhicule. La seconde étape consiste en la résolution d'un problème pour trouver la date de départ et la tournée qui optimise le critère.

## 2.1 Première étape : Construction d'une fonction de coût

La date de départ d'un véhicule peut être choisie dans un ensemble discret de dates comprises entre une date de départ  $a$  et une date de départ  $b$ . Cette discrétisation correspond par exemple à une réalité terrain où le départ d'un véhicule ne se planifie qu'à la minute ou que toutes les 10 minutes. L'efficacité de la livraison est évaluée d'après deux critères.  $CP$  représente les coûts dus aux pénalités de retard versés aux clients.  $CT$  représente les coûts de transport. Pour une date de départ  $t$ , la tournée optimale selon les critères  $CP$  et  $CT$  peut être différente de celle optimale à la date de départ  $t + 1$ . La courbe  $CP(t) + CT(t)$  est une fonction linéaire par morceaux non décroissante. Le but de la première étape est de construire cette fonction. La Fig.(1) présente un profil de cette courbe. La méthode de construction est détaillée dans la Section 3.

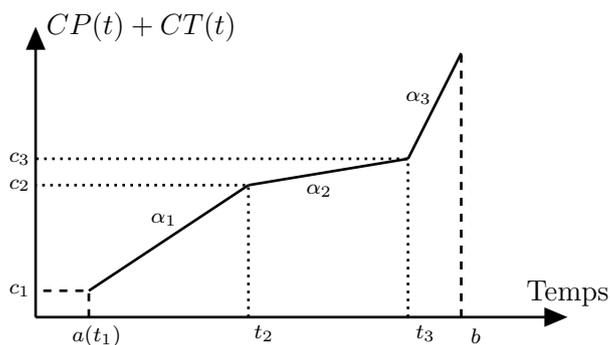


FIG. 1 – Coûts en fonction de la date de départ du véhicule

Soit  $S$  le nombre de segments de cette courbe. Chaque segment est associé à une tournée. Une même tournée peut être également associée à plusieurs segments. On note  $t_k$  la date de changement de pente de  $\alpha_{k-1}$  à  $\alpha_k$ ,  $\forall k \in S$ .  $c_k$  représente les coûts de retard et de livraison pour un départ du véhicule à la date  $t_k$  et en utilisant la tournée associée au segment.

De nouveaux segments sont créés au fur et à mesure du décalage de la date de départ. Deux événements provoquent la création d'un nouveau segment  $k$  :

- la tournée du segment  $k - 1$  génère un nouveau ou plusieurs clients en retard (cette tournée sera associée au nouveau segment mais  $\alpha_k$  augmentera par rapport à  $\alpha_{k-1}$ ),
- une nouvelle tournée moins coûteuse est trouvée pour la date de départ  $t_k$ .

## 2.2 Deuxième étape : Résolution du modèle global

La deuxième étape vise à intégrer la fonction de coût établie à la première étape dans la fonction objectif du problème et à résoudre le problème par le programme linéaire en nombres entiers présenté par modèle ci-dessous.

$D$  la variable représentant la date de départ du véhicule. Deux types de variables sont introduites :  $x_k$  et  $d_k$ ,  $\forall k \in \{1, \dots, S\}$ .  $x_k$  vaut 1 si la date de départ  $D$  appartient au  $k^{eme}$  segment de la fonction de coût, 0 sinon.  $d_k$  représente la position de  $D$  sur ce segment.

Dans cette formulation,  $f(D)$  représente le coût externe dépendant de la date de départ  $D$ .

La contrainte (1) définit  $D$  en fonction des  $x_k$  et  $d_k$ . La contrainte (2) assure que  $D$  n'appartient qu'à un seul segment. Les contraintes (3) et (4) limitent  $d_k$  à la taille du segment  $k$  si  $D$  appartient à  $k$ , ou imposent  $d_k$  à 0 sinon.

Ce modèle comporte  $S$  variables binaires,  $S + 1$  variables continues et  $S + 3$  contraintes.

$$\begin{aligned} \min \quad & \sum_{k=1}^S (x_k c_k + d_k \alpha_k) + f(D) \\ \text{s.t.} \quad & D = \sum_{k=1}^S (x_k t_k + d_k) \end{aligned} \quad (1)$$

$$\sum_{k=1}^S x_k = 1 \quad (2)$$

$$d_k \leq x_k (t_{k+1} - t_k - 1) \quad \forall k \in \{1, \dots, S-1\} \quad (3)$$

$$d_S \leq x_S (b - t_S) \quad (4)$$

$$a \leq D \leq b \quad (5)$$

$$d_k \geq 0, x_k \in [0, 1] \quad \forall k \in \{1, \dots, S\}$$

### 3 Calcul de la fonction $CP(t) + CT(t)$

#### 3.1 Algorithme général

A chaque tournée  $p_i$ , on associe une fonction de coût de livraison et de retard donnée par  $CP_{p_i}(t) + CT_{p_i}(t)$  et dépendant de la date de départ  $t$  de la tournée  $p_i$ . Étant donné un ensemble  $W$  de tournées, la fonction  $CP(t) + CT(t)$  de cette ensemble est définie par :

$$CP(t) + CT(t) = \min_{i \in W} (CP_{p_i}(t) + CT_{p_i}(t))$$

On note  $t_{p_i}$  la plus petite date de départ telle que la tournée  $p_i$  est la meilleure selon la fonction de coût suivante :  $t_{p_i} = \min_{t \in [a, b]} (CP_{p_i}(t) + CT_{p_i}(t))$

L'algorithme proposé est basé sur un algorithme de voisinage qui explore un ensemble des tournées afin de trouver le meilleur sous-ensemble  $W$  selon la fonction  $CP(t) + CT(t)$ . Chaque nouvelle tournée explorée est ajoutée à l'ensemble  $W$  si elle améliore la fonction.

Les figures Fig.(2), Fig.(3) et Fig.(4) illustrent le cas d'un ensemble  $W$  constitué de deux tournées  $p_A$  et  $p_B$  et du résultat de la fonction  $CP(t) + CT(t)$ . Pour une date de départ de tournée comprise dans l'intervalle  $[1, 6]$  ou dans l'intervalle  $[12, 13]$ , la tournée  $p_B$  est la meilleure alors que la tournée  $p_A$  est meilleure sur l'intervalle  $[6, 11]$ .

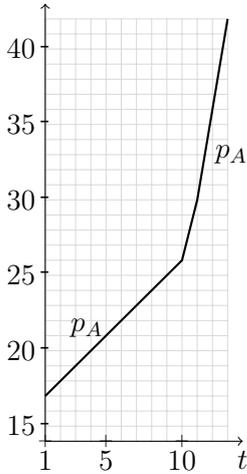


FIG. 2 –  $CP_{p_A}(t) + CT_{p_A}(t)$

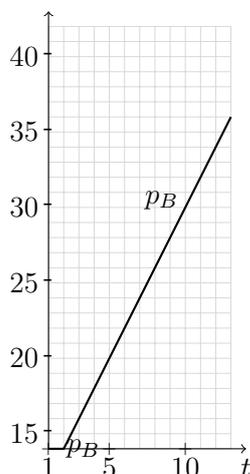


FIG. 3 –  $CP_{p_B}(t) + CT_{p_B}(t)$

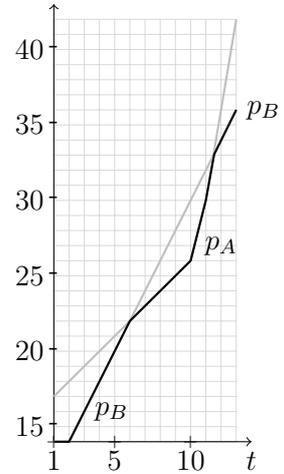


FIG. 4 –  $CP(t) + CT(t)$

L'algorithme de calcul de la fonction  $CP(t) + CT(t)$  est décrit dans l'Algo.1.

Tout d'abord, une tournée initiale  $p_{init}$  est déterminée par la fonction *Calcul de la tournée initiale()* décrite dans la prochaine section. La tournée  $p_{init}$  est ajoutée à l'ensemble  $W$  et à un ensemble noté *Candidats*.

La boucle principale de l'algorithme, décrite de la ligne 4 à 12, extrait une tournée dans la liste *Candidats* et l'améliore à l'aide d'une recherche locale. La fonction *Sélectionner tournée()* sélectionne la tournée  $p_i$  ayant la plus petite date de départ  $t_{p_i}$ . La recherche locale, décrite en Section 3.3, est appliquée à la tournée  $p_i$ . Chaque tournée explorée dans la recherche locale est ajoutée à  $W$  si elle améliore la fonction  $CP(t)+CT(t)$ . On note  $p_{best}$  la tournée finale ne pouvant plus être améliorée par la recherche locale. L'ensemble *Candidats* est alors mis à jour de manière à contenir l'ensemble des tournées de  $W$  qui n'ont pas encore été améliorées par la recherche locale. Ainsi, l'ensemble  $V$  est déterminé de manière à obtenir l'ensemble des tournées ayant été ajoutées à  $W$  lors de la recherche locale à l'exception de la tournée  $p_{best}$ . Cette ensemble  $V$  est ajouté à l'ensemble *Candidats*. Puis l'ensemble des tournées n'appartenant plus à  $W$  est retiré de l'ensemble *Candidats*. Ce processus est répété jusqu'à ce que l'ensemble *Candidats* soit vide.

---

### Algorithm 1

---

```

1:  $p_{init} \leftarrow \text{Calcul de la tournée initiale}()$ 
2:  $\text{Candidats} \leftarrow \{p_{init}\}$ 
3:  $W \leftarrow p_{init}$ 
4: tant que  $\text{Candidats} \neq \emptyset$  faire
5:    $p_i \leftarrow \text{Sélectionner tournée}(\text{Candidats})$ 
6:    $\text{Candidats} \leftarrow \text{Candidats} \setminus p_i$ 
7:    $W' \leftarrow W$ 
8:    $\{p_{best}, W\} \leftarrow \text{Recherche Locale}(p_i, W)$ 
9:    $V \leftarrow W \setminus (W' \cup p_{best})$ 
10:   $\text{Candidats} \leftarrow \text{Candidats} \cup V$ 
11:   $\text{Candidats} \leftarrow \text{Candidats} \cap W$ 
12: fin tant que

```

---

## 3.2 Calcul de la tournée initiale

Deux stratégies sont utilisées pour calculer la tournée initiale :

- La première est basée sur un algorithme de plus proche voisin (PPV).
- La deuxième consiste à visiter les clients par ordre de date de livraison croissante (EDD).

## 3.3 Recherche locale

La recherche locale part d'une tournée initiale  $p_i$ , explore son voisinage à l'aide d'un opérateur de voisinage, sélectionne le meilleur voisin et réitère le processus à partir de ce voisin. La recherche locale prend fin dès qu'elle ne permet plus d'améliorer la tournée courante. Toutes les tournées explorées sont évaluées en considérant une date de départ égale à  $t_{p_i}$ . L'opérateur de voisinage consiste à réinsérer un client à n'importe quelle position dans la tournée. Le voisinage est déterminé en testant tous les clients pour toutes les positions possibles. Enfin, chaque tournée explorée est ajoutée à  $W$  si elle améliore la fonction  $CP(t)+CT(t)$  en un point. A noter qu'une tournée peut être retirée de  $W$  si elle n'améliore plus cette fonction.

## 3.4 Convergence de la méthode

Étant donné qu'il existe un nombre fini de tournées optimales minimisant la fonction  $CP(t)+CT(t)$ , soit l'algorithme converge vers cet ensemble de tournées optimales soit vers un ensemble de tournées sous optimales qu'il n'est plus possible d'améliorer avec la recherche locale. Le critère d'arrêt (un ensemble vide de candidats) est nécessairement atteint puisqu'à chaque

itération, l'algorithme s'assure de n'ajouter dans l'ensemble *Candidats* que des tournées qui ont strictement amélioré l'ensemble *W*.

## 4 Résultats expérimentaux

Les résultats expérimentaux ont été réalisés sur ordinateur avec les caractéristiques suivantes : processeur Intel Core i7-7820HQ, CPU 2.90GHz et 16,0 Go de RAM. Les algorithmes ont été développés en C++.

Dans un premier temps, nous présentons les instances générées afin de tester l'approche proposée. Puis différentes variantes de l'algorithme sont comparées à une méthode exacte en utilisant un critère de comparaison basé sur le calcul d'une aire entre deux courbes.

### 4.1 Instances

Un ensemble de 320 instances sont générées de la manière suivante. L'ensemble des clients et le dépôt sont placés aléatoirement dans un carré de taille de 10 unités de temps et la distance de Manhattan est utilisée pour calculer les temps de trajets. Cette distance a du sens en agglomération, où les rues se croisent à angle droit. Les coûts de livraison sont égaux aux temps de trajets. La pénalité à verser à un client par unité de retard est tirée aléatoirement selon une loi normale d'espérance 5 et d'écart type 2. Les dates de livraison souhaitées par les clients sont tirées de manière uniforme dans une fenêtre de temps. La première date de cette fenêtre est fixée à une durée totale de tournée arbitrairement grande, notée  $M$  et égale au nombre de clients  $n$  multiplié par la plus grande distance entre deux clients de l'instance. Les dates de livraison sont tirées dans une fenêtre de taille 400,  $[M, M + 400]$ . Concernant les bornes de la fenêtre de départ du véhicule, la première date  $a$  vaut 0 et la dernière date  $b$  correspond à la dernière date de livraison souhaitée par un client. Pour chaque nombre de clients compris entre 5 et 20, 20 instances sont générées.

### 4.2 Méthodes comparées

Les deux principales tournées initiales utilisées sont les tournées  $p_v$  (plus proche voisin) et  $p_d$  (ordre de dates de livraison) décrites en Section 3.2.

Plusieurs méthodes de calcul de la fonction  $CP(t) + CT(t)$  sont comparées :

- *PPV* : La fonction est donnée par l'heuristique du plus proche voisin.
- *Version 1* : La fonction est donnée par l'Algo.1 en utilisant une stratégie d'initialisation PPV.
- *Version 2* : La fonction est donnée par l'union des meilleures tournées obtenues par deux exécutions de l'Algo.1, une première fois initialisée par la stratégie PPV et une deuxième fois initialisée par la stratégie EDD.
- *Version 3* : La fonction est donnée par l'union des meilleures tournées obtenues par huit exécutions de l'Algo.1 initialisé de huit manières différentes : par la stratégie PPV, EDD et 6 autres variantes basées sur PPV et EDD. Ces variantes se distinguent par le premier client visité et le sens de parcours des séquences données par PPV et EDD.
- *OPT* : la fonction optimale calculée par une procédure par séparation et évaluation développée lors de travaux antérieurs.

### 4.3 Indicateur de comparaison

Afin de comparer les fonctions  $CP(t) + CT(t)$  trouvées par les différentes méthodes, nous nous appuyons sur un indicateur de comparaison. Cet indicateur noté  $EA(X)$  est basé sur l'aire séparant la courbe de la fonction  $CP^X(t) + CT^X(t)$  donnée par une méthode  $X$  et celle donnée par la méthode exacte  $OPT$  et notée  $CP^*(t) + CT^*(t)$ .

L'indicateur de comparaison  $EA(X)$  ne s'applique qu'à un intervalle de temps jugé intéressant  $[a_X^*, b_X^*]$ , avec  $a_X^* = \min(a_X, a_{OPT})$  et  $b_X^* = \min(b_X, b_{OPT})$  tel que :

- $a_X$  désigne la plus grande date de départ à laquelle tous les clients sont livrés à l'heure par une tournée minimisant le coût de livraison.
  - $b_X$  désigne la plus petite date à laquelle tous les clients sont livrés en retard.
- L'indicateur de comparaison  $EA(X)$  est donné par la formule suivante :

$$EA(X) = \frac{\sum_{t \in [a_X^*, b_X^*]} [(CP^X(t) + CT^X(t)) - (CP^*(t) + CT^*(t))]}{\sum_{t \in [a_X^*, b_X^*]} [CP^*(t) + CT^*(t)] - CT^*(a_X^*)(b_X^* - a_X^*)}$$

Cet indicateur représente le ratio entre la différence de surface donnée par les deux fonctions  $CP^X(t) + CT^X(t)$  et  $CP^*(t) + CT^*(t)$ , et la surface de  $CP^*(t) + CT^*(t)$  moins la surface induite par le coût de livraison  $CT^*(a^*)$  incompressible du à la longueur minimale d'une tournée.

Un exemple graphique, donné par la Fig.(5), représente les différentes surfaces impliquées dans le calcul de  $EA(X)$ .

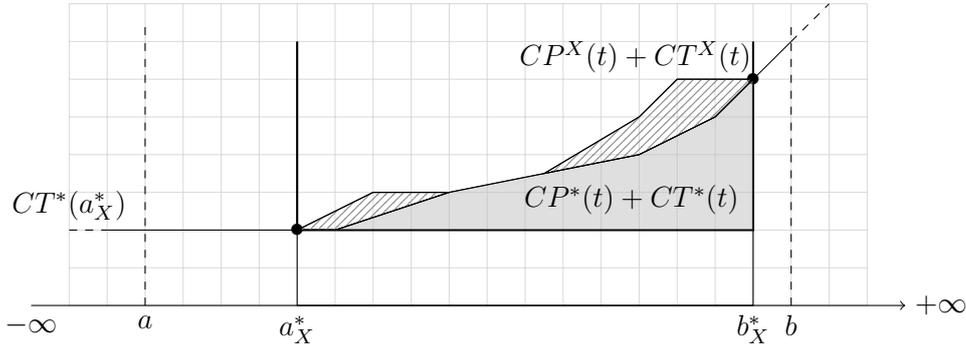


FIG. 5 – Comparaison d'une méthode  $X$  avec  $OPT$

#### 4.4 Analyse des résultats

L'approche proposée vise à être intégrée dans la résolution d'un problème plus global avec plusieurs tournées à déterminer. Cette approche a donc pour objectif d'évaluer une seule solution d'une tournée et pourra être amenée à être appelée de très nombreuses fois dans une méthode de résolution pour le problème global. Le temps de résolution étant très important, nous nous sommes intéressés principalement aux performances de la résolution de l'étape 1 de l'approche, puisque l'étape 2 nécessite la résolution d'un modèle mathématique de petite taille dépendant fortement de la fonction  $f(D)$  du problème global.

Les Fig.(6) et (7) présentent l'évolution du temps de résolution de la méthode  $OPT$  et des *Versions 1, 2 et 3* en fonction du nombre de clients. A noter que pour un nombre de clients donné, une moyenne du temps de résolution sur 20 instances est calculée. Le temps de résolution de  $OPT$  est de l'ordre de la seconde jusqu'à 10 clients puis augmente exponentiellement. La Fig.(7) montre que les méthodes *Version 1* et *Version 2* résolvent l'ensemble des instances en moins d'une demi seconde. La méthode *Version 3* a les mêmes performances pour les instances en dessous de 15 clients mais son temps de calcul augmente significativement à partir de 15 clients. Pour chaque méthode, le temps de résolution est proportionnel au nombre d'appels de l'Algo.1. Le temps de résolution de la méthode  $PPV$  n'est pas présenté sur la Fig.(7) car il est inférieur à une milliseconde pour chaque instance.

Les Fig.(8) et Fig.(9) présentent l'évolution de l'indicateur de comparaison  $EA(X)$  de la méthode  $PPV$  et des *Versions 1, 2 et 3* en fonction du nombre de clients. Pour un nombre de clients donné, une moyenne de cet indicateur sur 20 instances est calculée. Dans le cas de  $PPV$ , cet indicateur varie entre 6% et 14%, alors que pour les méthodes *Version 1, 2 et 3*, cet indicateur plafonne autour de 2%, 1% et 0.5% respectivement. On note que  $EA(\text{Version } 3) \leq EA(\text{Version } 2) \leq EA(\text{Version } 1)$  étant donné que l'Algo.1 est déterministe et que les résultats de la méthode *Version 1* sont inclus dans les résultats de *Version 2* qui sont inclus dans les résultats de *Version 3*. En comparant la méthode  $PPV$  et la *Version 1*, on remarque que

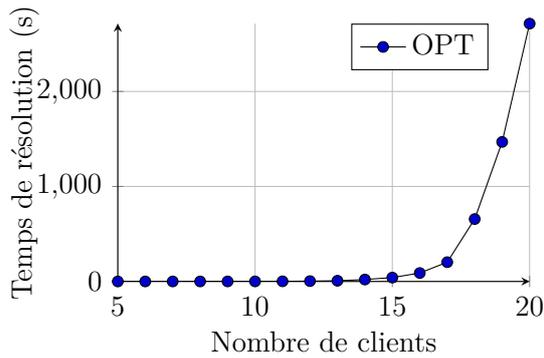


FIG. 6 – Temps de résolution de *OPT*

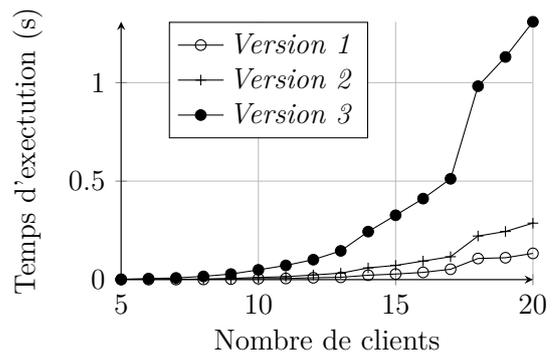


FIG. 7 – Temps d'exécution des méthodes

l'exécution de l'Algo.1 permet de gagner entre 6% et 12%. Cependant, l'évolution de l'indicateur de comparaison  $EA(X)$  pour *Version 2* et *Version 3* (en exécutant plusieurs fois l'Algo.1) n'est pas aussi importante, au plus 1% de *Version 1* à *Version 2* et au plus 0.5% de *Version 2* à *Version 3*.

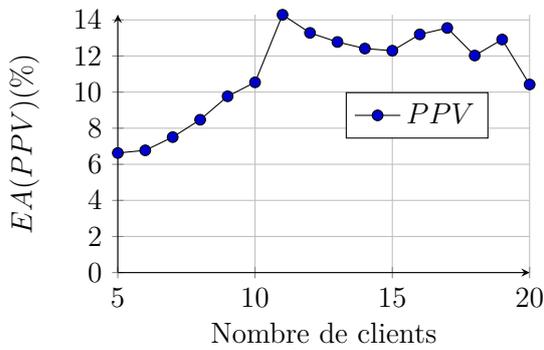


FIG. 8 –  $EA(PPV)$

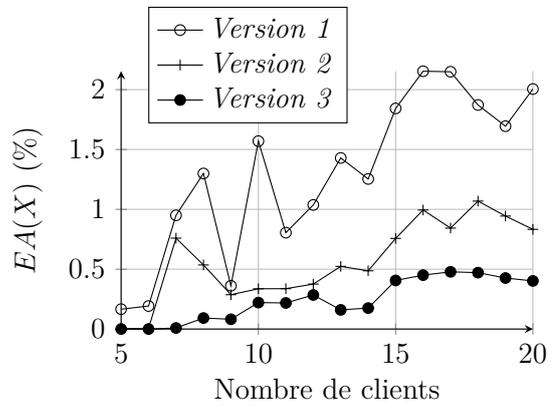


FIG. 9 –  $EA(X)$  pour les différentes méthodes

Au vu du compromis temps de résolution et qualité des solutions, la méthode *OPT* semble la plus efficace pour des instances avec un nombre de clients inférieur à 10. Au delà de 10 clients, la *Version 2* semble la plus pertinente.

## 5 Conclusion et perspectives

Cet article présente un problème de tournée dans lequel un ensemble de clients doit être servi pour un ensemble de dates souhaitées. La livraison est effectuée en une seule tournée. Des coûts de pénalité de retard et de livraison sont pris en compte. De plus, ce problème s'inscrit dans un contexte plus large où la date de départ du véhicule n'est pas connue et influence la fonction objectif par un coût externe.

Une approche générale de résolution en deux étapes a été présentée. Dans un premier temps, une fonction  $CP(t) + CT(t)$  est construite, représentant les coûts de livraison et de pénalité en fonction de la date de départ du véhicule dans un intervalle de temps  $[a, b]$ . Dans un second temps, une formulation mathématique modélise la fonction  $CP(t) + CT(t)$  et permet de prendre en compte les coûts externes dépendant de la date de départ.

Des heuristiques à base de recherche locale sont présentées pour calculer la fonction de coût  $CP(t) + CT(t)$ . Les temps d'exécution et la qualité des fonctions  $CP(t) + CT(t)$  générées par les différentes méthodes sont comparées en utilisant un indicateur de qualité ad-hoc. Les résultats montrent que l'approche proposée permet d'obtenir des temps de calcul très courts.

Ainsi l'approche proposée peut être intégrée dans une méthode de résolution d'un problème plus global avec plusieurs tournées à déterminer.

L'approche proposée est suffisamment générique pour prendre en compte de nouveaux coûts (coûts fixes pour le paiement des pénalités clients ou encore des coûts de transport évoluant au fil du temps) sachant que la modélisation mathématique de l'étape 2 n'impose pas que la fonction  $CP(t) + CT(t)$  soit continue.

Plusieurs perspectives de recherche peuvent être envisagées. Lors de la génération de la fonction  $CP(t) + CT(t)$ , d'autres opérateurs de recherche locale peuvent être redéfinis afin de s'adapter à d'autres types d'instances. D'autres méthodes de résolution comme l'algorithme tabou ou génétique peuvent également être testées.

## Références

- [1] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979) *Optimization and Approximation in Deterministic Sequencing and Scheduling : a Survey*, Annals of Discrete Mathematics, Volume 5, Pages 287-326.
- [2] Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. (1982) *Deterministic and Stochastic Scheduling. Recent developments in deterministic sequencing and scheduling : a survey, deterministic and stochastic scheduling*. Dempster, M.A.H., Lenstra, J.K. and Rinnooy Kan, A.H.G. (eds.), Reidel, Dordrecht, 35-73.
- [3] Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y. (1997) *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Transportation Science, vol.31, 170-186
- [4] Duygu Taş, Nico Dellaert, Tom van Woensel, Ton de Kok (2013) *Vehicle routing problem with stochastic travel times including soft time windows and service costs*, Computers & Operations Research, Volume 40, Issue 1, Pages 214-224.
- [5] Kai Li, Chao Zhou, Joseph Y-T. Leung, Ying Ma, (2016), *Integrated production and delivery with single machine and multiple vehicles*, Expert Systems with Applications, Volume 57, Pages 12-20.
- [6] Wen-Chyuan Chiang, Robert Russell, Xiaojing Xu, David Zepeda, (2009) *A simulation/metaheuristic approach to newspaper production and distribution supply chain problems*, International Journal of Production Economics, Volume 121, Issue 2, Pages 752-767