

# Recherche à voisinage large pour les problèmes de voyageurs de commerce statiques ou dynamiques avec fenêtres temporelles

Cédric Pralet<sup>1</sup>

ONERA/DTIS, Université de Toulouse

cedric.pralet@onera.fr

**Mots-clés :** *voyageur de commerce, fenêtres temporelles, transitions dépendantes du temps, recherche à voisinage large*

## 1 Introduction

Dans ces travaux, nous considérons la résolution du problème de voyageur de commerce avec fenêtres temporelles (ou TSPTW pour *Traveling Salesman Problem with Time Windows*) et son extension aux problèmes dits time-dependent (ou TDTSPWT pour *Time-Dependent TSPWT*). Dans ces problèmes, on considère un ensemble de clients devant être visités par un véhicule sachant qu'il existe pour chaque client une fenêtre temporelle autorisée pour la visite et que le temps de transition entre deux clients peut dépendre de la date à laquelle la transition est réalisée. Plus spécifiquement, nous nous intéressons à la définition de méthodes de résolution incomplètes pour calculer très rapidement des solutions faisables ou des solutions minimisant la date de fin des visites des clients. Dans la littérature, diverses méthodes de résolution incomplètes ont déjà été définies pour les TSPTW (recherche heuristique, recherche tabou, recherche à voisinage variable, recherche à voisinage large...) [1], et les méthodes de résolution de TDTSPWT sont plutôt centrées sur les méthodes complètes [3]. Par rapport à la littérature, la méthode introduite exploite au maximum la structure du graphe de précedence associé au problème. De plus, nous introduisons des méthodes pour gérer des problèmes dynamiques dans lesquels des clients peuvent être ajoutés ou retirés, avec un effort pour résoudre chaque nouveau problème de manière incrémentale.

## 2 Algorithme de résolution de problèmes statiques

La méthode incomplète proposée est une recherche à voisinage large (ou LNS pour *Large Neighborhood Search*), qui utilise intensivement un graphe de précedences obligatoires entre visites de clients (cf. figure 1). Partant d'une séquence initiale de visite, la méthode enchaîne des phases de destruction et de réparation de la solution courante. A chaque itération, l'opération de destruction consiste à retirer des clients de manière à obtenir une séquence de visite partielle, et l'opération de réparation reconstruit une séquence de visite complète en tentant de réinsérer optimalement les clients retirés. La notion d'optimalité utilisée à chaque itération consiste à privilégier d'abord les séquences de visite qui minimisent la somme des retards par rapport aux dates de fin des fenêtres temporelles, ensuite les séquences qui se terminent le plus tôt possible, et enfin les séquences qui minimisent la somme des temps de transition entre visites successives. Une des originalités de l'algorithme est que la méthode de destruction s'autorise à retirer des clients tant qu'un paramètre du graphe de précedence entre clients appelé la *largeur d'insertion* est inférieur à une certaine borne  $W_{max}$  fixée à l'avance, sachant que dans ce cas la complexité de la méthode de réparation par programmation dynamique est seulement linéaire en le nombre de clients. Au-dessus de ces mécanismes de base, lorsque l'algorithme considère qu'un optimum local est atteint, des opérations de perturbation et de redémarrage sont également utilisées. L'algorithme obtenu est appelé ImaxLNS pour *Iterated Maximum LNS* [4].

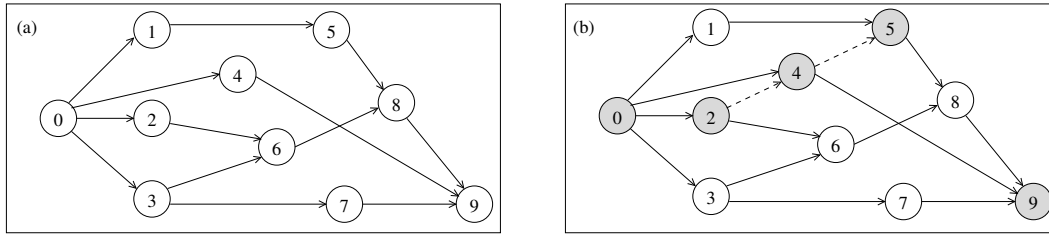


FIG. 1 – Graphe de précédence entre clients : (a) version initiale obtenue à partir des fenêtres temporelles associées aux clients ; (b) version avec une séquence partielle de visite [0, 2, 4, 5, 9]

### 3 Algorithme de résolution de problèmes dynamiques

Pour traiter les problèmes dynamiques dans lesquels des clients sont retirés, la méthode supprime simplement les visites de clients qui ne sont plus à prendre en compte. Si le plan initial était faisable, le nouveau plan obtenu reste faisable pour autant que la propriété FIFO soit satisfaite. Pour traiter les problèmes dynamiques dans lesquels de nouveaux clients sont ajoutés, une méthode en deux temps est utilisée. En première approche, une insertion gloutonne qui ne change pas l’ordre des clients déjà présents dans la séquence de visite courante est tentée (recherche d’une “meilleure” position d’insertion). En deuxième approche, si l’insertion gloutonne a échoué à produire un plan faisable, le graphe de précédence entre clients est actualisé et l’algorithme ImaxLNS est appelé. L’actualisation du graphe de précédence est un point clé dans le sens où elle nécessite de prendre en compte à la fois les retraits et les ajouts de clients, ainsi que le fait que des calculs de réduction transitive du graphe de précédence peuvent être invalidés du fait de la modification des clients impliqués dans le problème.

### 4 Résultats expérimentaux

Les algorithmes proposés ont été évalués avec de nombreuses instances de tests disponibles dans la littérature, pour des problèmes statiques et dynamiques. ImaxLNS a permis de retrouver en moins d’une seconde en moyenne toutes les meilleures solutions connues sur 467 instances classiques de TSTPW. Côté TDTSPW [2], l’algorithme a permis de fournir 63 nouvelles solutions faisables et 13 nouvelles meilleures solutions par rapport à une approche complète récente [3], sur des benchmarks définis par Arigliano et al. [2]. Du côté des problèmes dynamiques, l’approche a été appliquée avec succès pour traiter des problèmes OPTW (*Orienteering Problem with Time Windows*) dans lesquels des clients apportant un certain profit peuvent être sélectionnés ou non, en laissant à chaque étape de modification des clients sélectionnés seulement quelques centaines de millisecondes pour recalculer une solution.

Ces travaux sont réalisés avec le support du Programme d’Investissements d’Avenir (projet BPI PSPC LiChIE mené par AIRBUS Defence and Space).

### Références

- [1] K. Amghar, J.-F. Cordeau, and B. Gendron. A general variable neighborhood search heuristic for the traveling salesman problem with time windows under completion time minimization. Technical report, CIRRELT, 2019.
- [2] A. Arigliano, G. Ghiani, A. Grieco, E. Guerriero, and I. Plana. Time-dependent asymmetric traveling salesman problem with time windows : Properties and an exact algorithm. *Discrete Applied Mathematics*, (261), 2019.
- [3] G. Lera-Romero, J. J. Miranda-Bront, and F. J. Soulignac. Dynamic programming for the time-dependent traveling salesman problem with time windows. *INFORMS journal on Computing*, 2022.
- [4] C. Pralet. Iterated maximum large neighborhood search for the traveling salesman problem with time windows and its time-dependent version. *To appear in Computer & Operations Research*.