

# Générer des instances réalistes grâce à l’OpenData

Philippe Lacomme<sup>3</sup>, Gwénaél Rault<sup>1,2</sup>, Marc Sevaux<sup>2</sup>

<sup>1</sup> Mapotempo, Groupe Octime, Bordeaux

[gwenael@mapotempo.com](mailto:gwenael@mapotempo.com)

<sup>2</sup> Université Bretagne Sud, Lab-STICC, UMR 6285, CNRS, Lorient

[marc.sevaux@univ-ubs.fr](mailto:marc.sevaux@univ-ubs.fr)

<sup>3</sup> Université de Clermont-Ferrand, LIMOS

[placomme@isima.fr](mailto:placomme@isima.fr)

**Mots-clés :** *Vehicle Routing, Instance generation, Open Data, OpenStreetMap*

## 1 Introduction

Depuis la description des problèmes de tournées de véhicules par Dantzig et Ramser[3], de nombreuses variantes ont été introduites : elles ont toutes fait l’objet de nombreuses études soit par des méthodes exactes de résolution soit par des méthodes approchées. Nous nous intéressons, ici, uniquement aux problèmes statiques, excluant ainsi la prise en compte dynamique du trafic. On peut constater depuis plusieurs années les efforts importants de la communauté pour fournir des comparaisons “intelligentes” des méthodes de résolution en prenant en compte le type de machine utilisée pour les calculs, le langage et le système d’exploitation. Ces comparaisons passent par l’utilisation de jeux de données qui sont reconnus par la communauté et qui servent de référence à l’ensemble des méthodes. Sur le HVRP par exemple, nous avons sélectionné 5 jeux de données représentatifs dont le plus ancien date de 1984 et le plus récent (celui proposé par Duhamel et al.) date de 2011.

Auteurs	Distance	Config	Noeud	Demande	Capacité/coût
Golden et al.[6]	2D Eucl	5-6	Concentrique	10 ou 30	$r^2$
Taillard[7]	2D Eucl	3-5	Aléatoire	Aléatoire	1, r, $r^2$
Cordeau and Laporte[2]	2D Eucl	4-7	Clusters	Aléatoire	1
Brandão[1]	2D Eucl	4-6	Repris	Aléatoire	r
Duhamel et al.[4]	routières	2-8	Code postal	Aléatoire	Unfair

TAB. 1 – Instances HVRP et variantes

Malgré une évolution très importante, les jeux de données utilisés (TAB. 1) restent encore très éloignés de jeux de données “réalistes”. Tout d’abord, dans la grande majorité des jeux de données, il s’agit de distances euclidiennes, indépendamment du type de véhicule assurant le trajet de  $i$  vers  $j$ . Les matrices de distance sont alors symétriques et sont assez éloignées de ce que l’on connaît dans la plupart des grandes villes européennes.

Les nœuds sont positionnés de manière aléatoire ou autour de centroïdes, dans certains cas les points sont placés de manière symétrique autour du dépôt avec une répartition concentrique. Les demandes, quant à elles, sont soit tirées dans un ensemble de valeurs prédéfinies ou déterminées de manière aléatoire avec une distribution uniforme. Le ratio entre la capacité des véhicules et leur coût associé croît de manière linéaire ou quadratique, lorsque le coût n’est pas simplement fixé à une valeur unique. Autre constatation, les flottes de véhicules sont générées de manière à avoir une répartition équitable dans l’utilisation des véhicules, soit par un ratio entre quantité et capacité, soit par un ajustement du coût pour favoriser certains types

de véhicules. Dans la grande majorité des cas, les instances sont obtenues par modification d'instances classiques d'autres problèmes de tournées de véhicules.

Dans l'échantillon ici représenté, seul l'ensemble d'instances proposé par Duhamel et al.[4] propose des distances issues du réseau routier, et les distances sont de "vraies" distances kilométriques. Toutefois, les matrices de distances sont symétriques et le plus court chemin ne dépend pas du véhicule utilisé. Ces matrices ne respectent pas l'inégalité triangulaire, car il s'agit de distances kilométriques obtenues en interrogeant les services de routage de Google avec un plus court chemin minimisant le temps de transport.

Les remarques précédentes restent vraies pour de nombreux problèmes de tournées dont par exemple le LRP (Location and Routing Problem). Néanmoins, il est notable de souligner la contribution de De Smet[5] pour le CVRP (Capacitated Vehicle Routing problem) qui a proposé quelques ensembles d'instances avec un calcul de matrices via le réseau routier depuis des données issues de OpenStreetMap et un calculateur d'itinéraire Open Source.

En conclusion, on peut dire que l'utilisation d'instances plus réalistes correspond à un besoin de la communauté mais que la création de ces instances est un travail important nécessitant à la fois de disposer de distances réalistes, et de volumes à transporter qui soient représentatifs des moyens de transport et de stockage actuel. Les problèmes de tournées où les temps de transport évoluent en fonction des heures de la journée sont aussi concernés par les remarques précédentes.

## 2 Proposition

Nous proposons un système de génération d'instances "réalistes" basé sur OpenStreetMap et qui se présente sous la forme d'une API et d'une web application. Ce système est basé sur trois composants principaux que sont la collecte de données relatives aux clients, la définition des caractéristiques de la flotte de véhicules et le calcul d'itinéraires à partir d'un graphe routier.

Les demandes des clients sont localisées sur des nœuds répartis différemment selon le cas d'usage ciblé, de manière uniforme le long des axes principaux avec des quantités similaires pour modéliser l'entretien d'arrêts de bus, ou par cluster dans les zones commerciales avec des quantités variables pour modéliser l'approvisionnement des magasins. La génération des caractéristiques de la flotte de véhicule permet de générer des flottes hétérogènes avec des types différents en terme de coûts fixes, de coûts variables et de vitesse moyenne avec des restrictions d'accès à certains arcs. L'outil de calcul d'itinéraire manipule des graphes routiers qui sont la base de l'ensemble des problèmes de tournées de véhicules avec des arêtes soumises à des vitesses qui peuvent être différentes en fonction du type de véhicule (on peut ainsi faire la différence entre un VL et un PL).

## Références

- [1] J. Brandão. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers Operations Research*, 38(1) :140 – 151, 2011. Project Management and Scheduling.
- [2] J.-F. Cordeau and G. Laporte. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR : Information Systems and Operational Research*, 39(3) :292–298, 2001.
- [3] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1) :80–91, 1959.
- [4] C. Duhamel, P. Lacomme, and C. Prodhon. A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 25(2) :345 – 358, 2012. Special Section : Local Search Algorithms for Real-World Scheduling and Planning.
- [5] G. D. S. et al. *OptaPlanner VRP examples : Belgium 2017 dataset*. Red Hat and the community. OptaPlanner is an open source constraint satisfaction solver in Java.
- [6] B. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers Operations Research*, 11(1) :49 – 66, 1984.
- [7] Taillard, E. D. A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO-Oper. Res.*, 33(1) :1–14, 1999.