

Combiner machine learning et modélisation hiérarchique pour caler les simulateurs multi-agents

Lucas Javaudin

THEMA, CY Cergy Paris Université

lucas.javaudin@cyu.fr

19 juin 2025, RFTM 2025

Objectifs

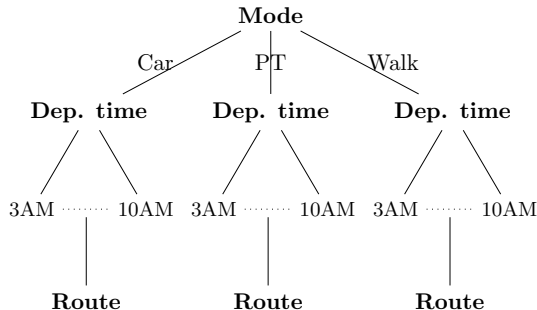
- Calage : ajuster les paramètres du modèle pour répliquer des valeurs observées
- Étape qui peut être difficile (temps de calcul du modèle, dimension des paramètres) mais pourtant crucial
- Exemple dans le cas des simulateurs de transport: CaDyTS (Ziemke et al., 2019), calage à partir de comptages routiers, reposant sur l'algorithme co-évolutionnaire de MATSim

Contributions

- Méthodologie de calage pour les simulateur de transport
- Basé sur des données en accès libre pour la France
- Processus en 4 étapes, reposant sur la hiérarchie des choix (mode puis heure de départ puis route) et le *machine learning* pour optimiser les temps de calcul
- Cas d'étude : Île-de-France ($150 + 12 + 10 + 132 = 294$ paramètres calés)

METROPOLIS2

- Simulateur de transports:
 - ▶ multi-agents
 - ▶ dynamique
 - ▶ mésoscopique
- Congestion simulé par des goulots d'étranglement au niveau arc
- Hiérarchie des choix :
 1. Choix de mode : *Multinomial Logit*
 2. Choix de l'heure de départ : *Continuous Logit*
 3. Choix de route : chemin le plus rapide



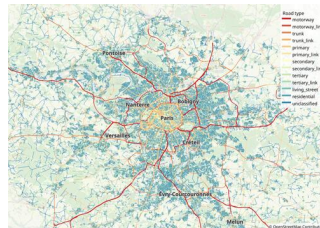
Javaudin L. (2024), *Development of a Dynamic Transport Simulator for Policy Evaluation: Application to Ride-Sharing and Low Emission Zone in Paris*. Ph.D. thesis, CY Cergy Paris Université. Supervision by André de Palma.

METROPOLIS2 : Génération d'une simulation

- Réseau routier → OpenStreetMap
- Réseau transports en commun → fichiers GTFS
- Population synthétique (ménages, personnes, déplacements et activités pour une journée type) → eqasim (Hörl et Balac, 2021)
- Modes : voiture conducteur, voiture passager, transports en commun, marche à pied, vélo

⇒ Île-de-France :

- 610 K tronçons routiers
- 2,452 M agents
- 8,774 M déplacements



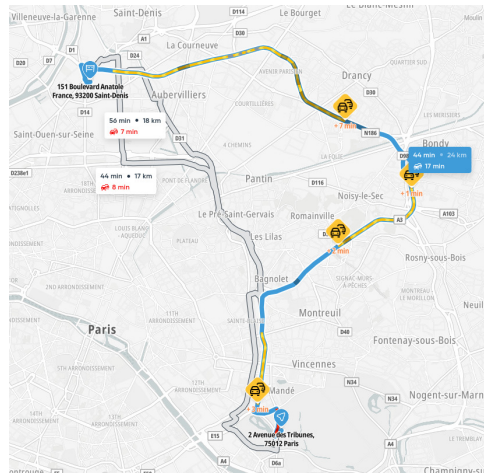
Réseau routier



Localisation des domiciles

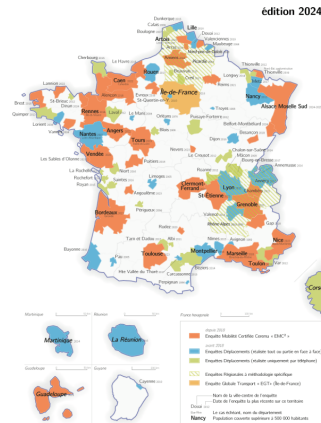
Données 1 : API TomTom

- Requêtes :
 - Entrants (aléatoire) :
 - origine
 - destination
 - heure de départ
 - Sortants :
 - itinéraire le plus rapide
 - temps de trajet prédit avec congestion
 - temps de trajet prédit sans congestion
- Plusieurs centaines de milliers de requêtes en quelques heures (version gratuite)
- Conversion de l'itinéraire de TomTom vers OpenStreetMap (*map matching*)



Données 2 : Enquêtes Mobilités

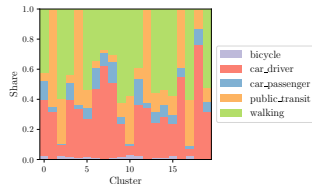
- Diverses enquêtes : EMC², EGT, EMP, etc.
- Pertinence pour le calage : modes et heures de départ par typologie des déplacements (e.g., motif, origine, destination, PCS)
- Standardisation avec MobiSurvStd



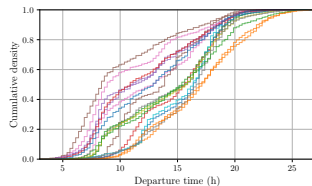
Source : CEREMA

Données 2 : Enquêtes Mobilités

- Les déplacements sont regroupés en *clusters* (algorithme *k-means*) selon :
 - ▶ le motif au départ et à l'arrivée
 - ▶ le département d'origine et de destination
 - ▶ la distance à vol d'oiseau
- Objectif : répliquer les parts modales et heures de départ dans chaque cluster



Parts modales, par cluster (EGT 2010)



Distributions cumulées des heures de départ, par cluster (EGT 2010)

Calage : principe

Sachant un simulateur $f : (\theta_1, \theta_2) \mapsto (x, y)$, l'objectif du calage est de trouver les paramètres θ_1, θ_2 (ex. : capacités, valeurs du temps) qui permettent au simulateur de répliquer aux mieux des valeurs observées x^*, y^* (ex. : temps de trajet, parts modales) :

$$\operatorname{argmin}_{\theta_1, \theta_2} \|f(\theta_1, \theta_2) - (x^*, y^*)\|$$

Difficultés :

- La fonction f (le simulateur) est coûteuse en temps à évaluer
- Les paramètres θ_1, θ_2 peuvent être définis sur un espace de grande dimension

Calage : principe

S'il existe une fonction g telle que $f(\theta_1, \theta_2) = (x, g(\theta_2; x))$, alors le calage peut être effectuée en 2 étapes :

$$\theta_2^{\text{opt}} = \operatorname{argmin}_{\theta_2} \|g(\theta_2; x^*) - y^*\|$$

puis

$$\theta_1^{\text{opt}} = \operatorname{argmin}_{\theta_1} \|f(\theta_1, \theta_2^{\text{opt}}) - (x^*, y^*)\|.$$

Méthodologie : résumé

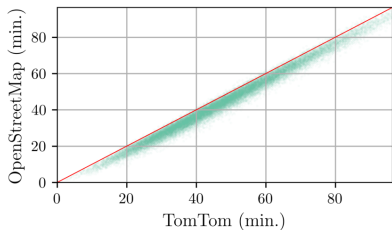
Étape	Paramètres calés	Valeurs cibles	Source des cibles	Méthodologie
1	Vitesses à vide, pénalité de temps de trajet	Temps de trajet sans congestion	API TomTom	Régression Lasso
2	Capacités des arcs	Temps de trajet avec congestion	API TomTom	
3	Pénalités d'avance et de retard	Distribution des heures de départ	EGT	Optimisation Bayésienne avec processus Gaussiens
4	Constantes modales et valeurs du temps	Parts modales	EGT	Régression avec <i>Random Forest</i>

Les étapes pourraient être itérées pour s'assurer de la pertinence de la méthodologie.

Étape 1 : Temps de trajet sans congestion

Temps de trajet moyen sans congestion :

- TomTom : 47' 59"
- OpenStreetMap (limitation de vitesse) : 44' 10"



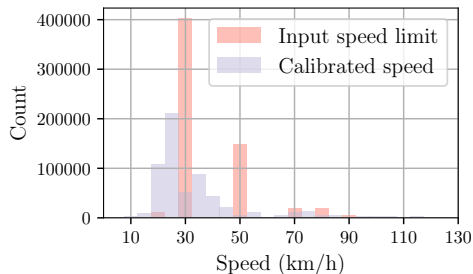
RMSE : 266 secondes

- Paramètres calés : vitesses à vide, pénalités de temps de trajet (*au niveau arc*)
- Valeurs cibles : temps de trajet sans congestion TomTom (*au niveau origine-destination*)
- Méthodologie : régression Lasso des temps de trajet sur les caractéristiques de l'itinéraire (# d'arcs par type, temps de trajet réglementaire par type d'arcs)
- Caractéristiques considérées : feux de signalisation, typologie OSM des arcs, vitesse réglementaire, indicateur urbain / rural, degré entrant et sortant de l'intersection

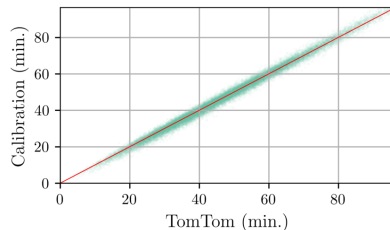
Étape 1 : Temps de trajet sans congestion

Paramètres calés

- Pénalité pour les arcs avec des feux de signalisation : 9,1 secondes (7,9 en milieu rural)



Distance aux valeurs cibles



RMSE : 104 secondes

Étape 2 : Temps de trajet avec congestion

- Paramètres calés : capacités des arcs par catégorie (typologie OSM + ronds-points + feux de signalisation)
- Valeurs cibles : temps de trajet avec congestion TomTom (*au niveau origine-destination*)
- Méthodologie :
 1. Générer des déplacements voitures représentatifs
 2. Initialiser les capacités à des valeurs par défaut
 3. Lancer une simulation de METROPOLIS2 (choix de route uniquement)
 4. Comparer les temps de trajet simulé sur les itinéraires TomTom avec les temps de trajet TomTom
 5. Ajuster les capacités par catégorie d'arc (optimisation manuelle pour le moment)

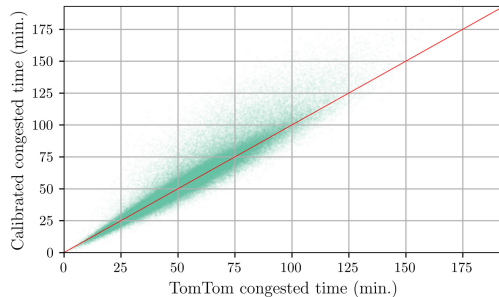
Étape 2 : Temps de trajet avec congestion

Paramètres calés

Typologie OSM	Capacité calée (PCE / h)
motorway	1700
motorway_link	1350
trunk	1850
trunk_link	1350
primary	2150
secondary	1700
tertiary	1400
residential	1300
unclassified	1000
living_street	900

- Multiplicateur ronds-points : 40 %
- Multiplicateur feux de signalisation : 70 %

Distance aux valeurs cibles

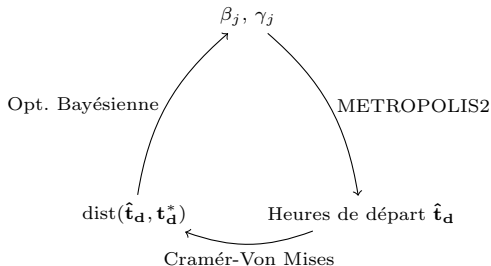


RMSE : 523 secondes

Corrélation : 94.48 %

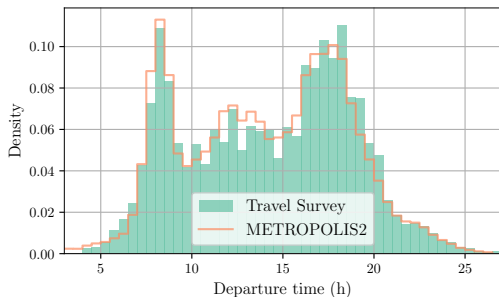
Étape 3

- Paramètres calés : pénalités d'avance β ou de retard γ par rapport à l'heure désirée d'arrivée, pour chaque motif de déplacement
- Valeurs cibles : distribution des heures de départ par cluster (EGT 2010)
- Méthodologie :
 - ▶ Génération des heures désirées d'arrivée
 - ▶ Simulation des heures de départ avec METROPOLIS2 (temps de trajet exogènes)
 - ▶ Minimisation de la distance Cramér-Von Mises, au niveau cluster
 - ▶ Optimisation Bayésienne avec processus Gaussiens

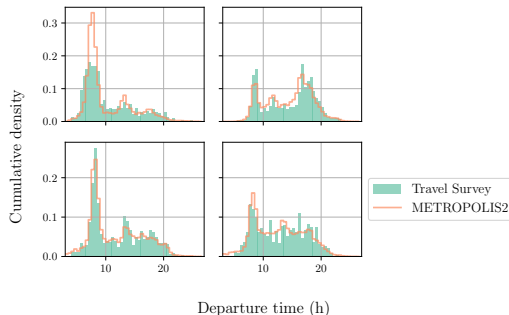


Étape 3 : Distribution des heures de départ

Distribution agrégée

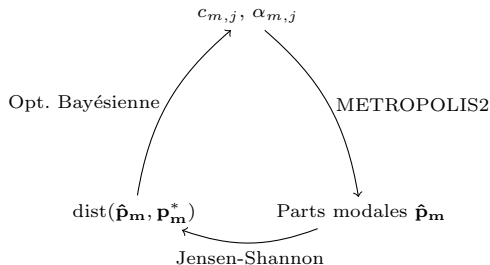


Distribution par cluster



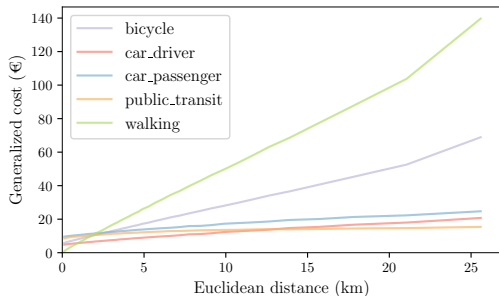
Étape 4: Parts modales

- Paramètres calés : constante et valeur du temps par mode, pour chaque genre et PCS
- Valeurs cibles : parts modales par cluster (EGT 2010)
- Méthodologie :
 - ▶ Minimisation de la distance Jensen-Shannon, au niveau cluster
 - ▶ Optimisation Bayésienne et régression avec *Random Forest*
 - ▶ Simulation des modes avec METROPOLIS2 (temps de trajet exogènes)

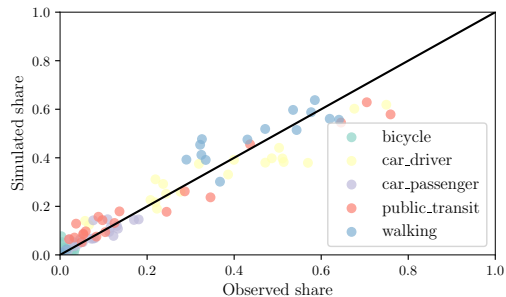


Étape 4 : Parts modales

Coût généralisé du déplacement, selon le mode (hommes, employés)

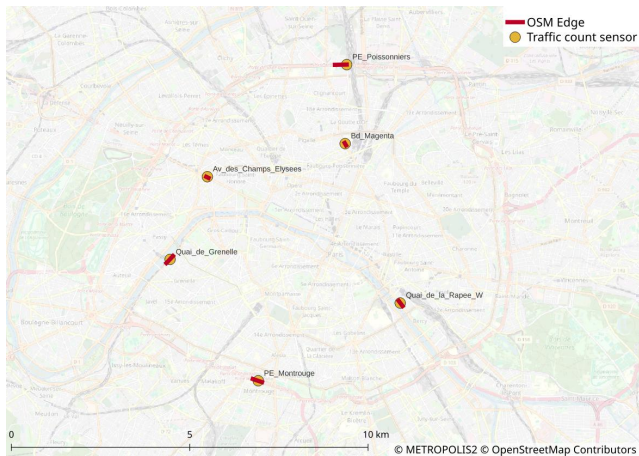


Parts modales par cluster

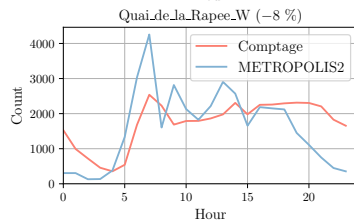
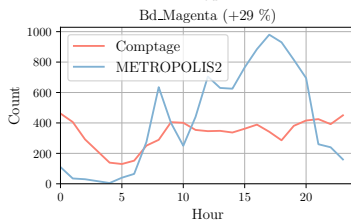
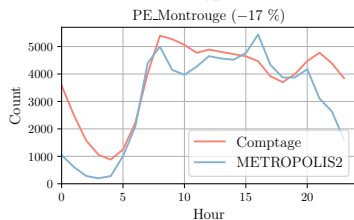
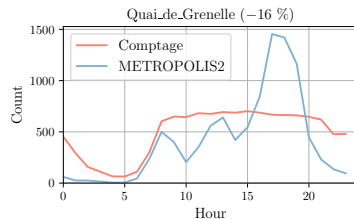
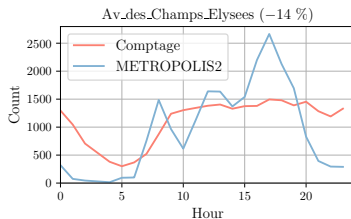
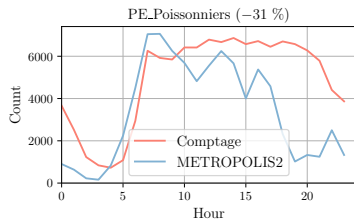


Validation

- Comptages routiers de la mairie de Paris (open data)
- Données non utilisées lors du calage
- Comparaison sur 6 capteurs
- Moyenne horaire sur 5 jours (décembre 2023)



Validation



Résumé et travaux futurs

- Méthodologie innovante de calage en 4 étapes, combinant modélisation hiérarchique et *machine learning*
- Utilisation exclusive de données en open data pour la France
- Simulations adaptées à l'évaluation de politiques publiques (ex. : Zones à Faibles Émissions, chapitre 3 de ma thèse)
- Travaux futurs : application à d'autres territoires et automatisation complète du processus