



Introduction à **METROPOLIS2**

Lucas Javaudin

LVMT, Institut Polytechnique de
Paris

METROPOLIS2 en bref

- Développé à partir de 2021 (thèse de Lucas Javaudin, direction André de Palma)
- Inspiration : METROPOLIS(1), MATSim
- Objectif : évaluation de politiques publiques

Caractéristiques principales :

- Multi-agents
- Mesoscopique
- Dynamique
- Théorie du choix discret
- Modèle bottleneck
- Processus de convergence itératif

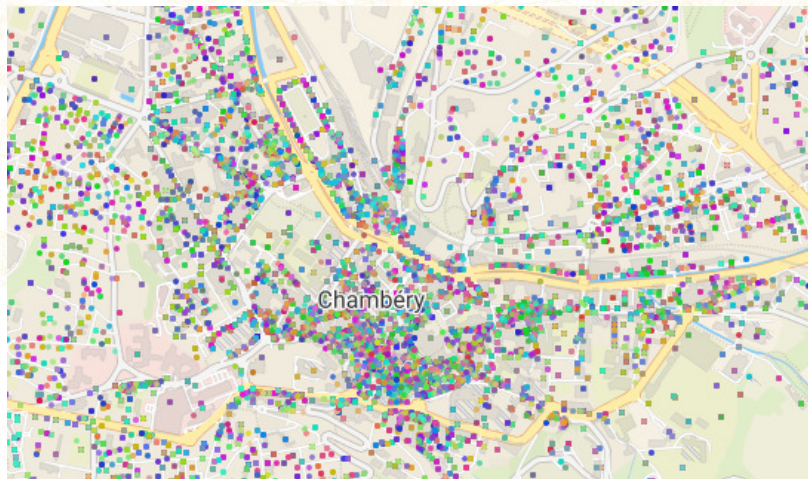
Données entrantes : Population synthétique

- **Population fictive** de ménages et personnes, générés pour être représentatifs d'une population (Hörl & Balac, 2021)
- **Combinaison de données** : recensement, EGT / EMC², SIRET, BDTOPO
- **Exemple** de ménage :
 - ID: 19
 - Nombre de voitures : 1
 - Nombre de vélos : 3
 - Revenu moyen : 11 865 €

Données entrantes : Population synthétique

Personnes

ID	Âge	Genre	Classe professionnelle
34	51	Homme	Employé
35	48	Femme	Cadre
36	17	Homme	Étudiant
37	14	Femme	Étudiant

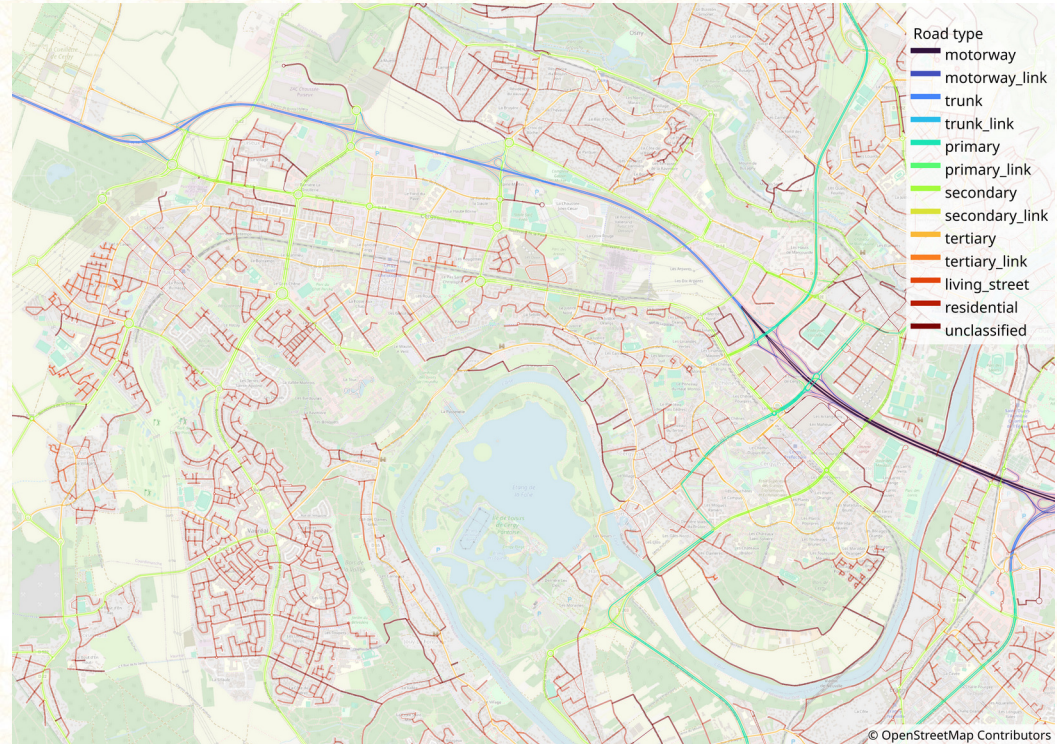


Déplacements

ID	Personne	Motif précédent	Motif suivant	Durée d'activité
133	34	Domicile	Travail	06:35:00
134	34	Travail	Domicile	
135	35	Domicile	Travail	08:03:00
136	35	Travail	Domicile	
137	36	Domicile	Éducation	06:55:00
138	36	Éducation	Domicile	
139	36	Domicile	Loisir	01:25:00
140	36	Loisir	Domicile	
141	37	Domicile	Éducation	07:10:00
142	37	Éducation	Domicile	

Données entrantes : OpenStreetMap & GTFS

- Le réseau routier est importé depuis **OpenStreetMap**
- Les horaires de transports en commun sont lus depuis des fichiers **GTFS**

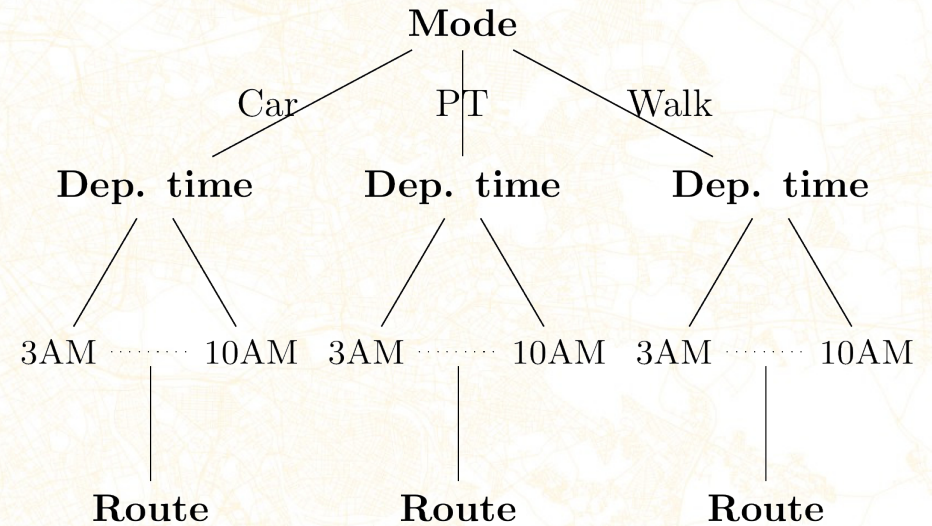


Demande : déplacements et chaînes

- Types de déplacements :
 - **Routiers** (e.g., voiture conducteur, voiture passager, camion)
 - À **temps de trajet constant** (e.g., marche, vélo, transports en commun fiables)
 - **Transports en commun** [à venir]
- **Chaînage** : chaque agent réalise une séquence de déplacements
- **Boucles** : Le choix de mode s'effectue au niveau d'une boucle domicile–domicile

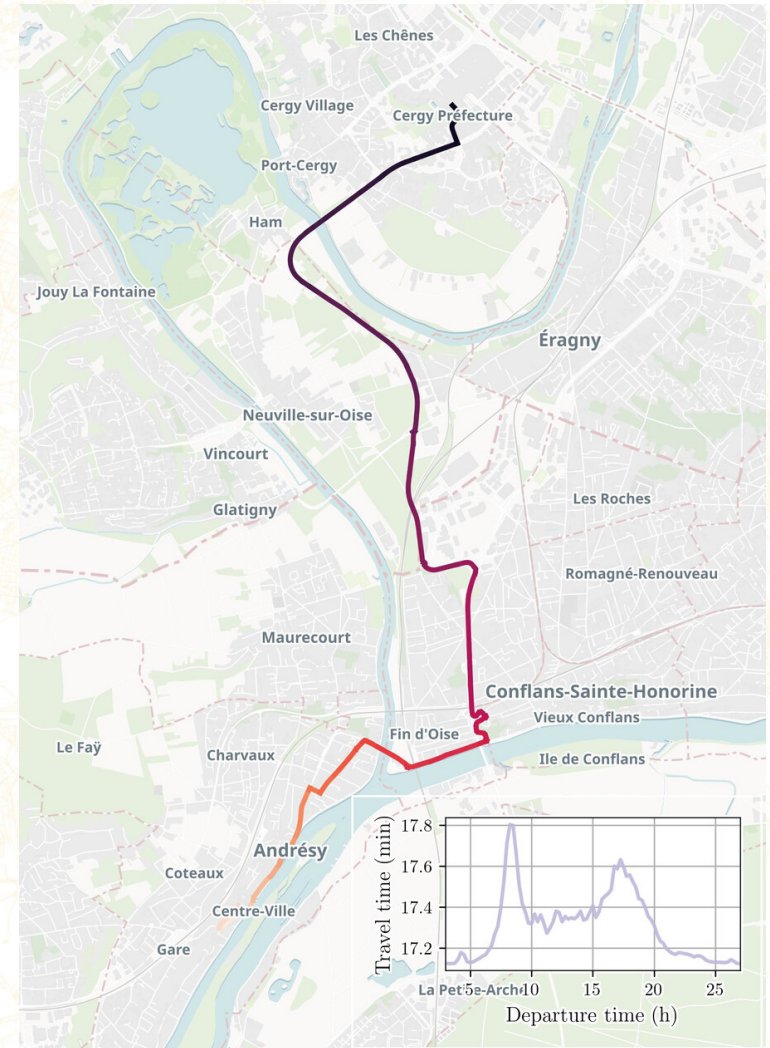
Demande : processus de décision

- Niveau 1 : Les agents choisissent un **mode** (Multinomial Logit, Nested Logit, modèle déterministe, etc.)
- Niveau 2 : Les agents choisissent une **heure de départ** pour leur premier déplacement (Continuous Logit, Multinomial Logit, exogène)
- Niveau 3 : Les agents choisissent leur **itinéraire**



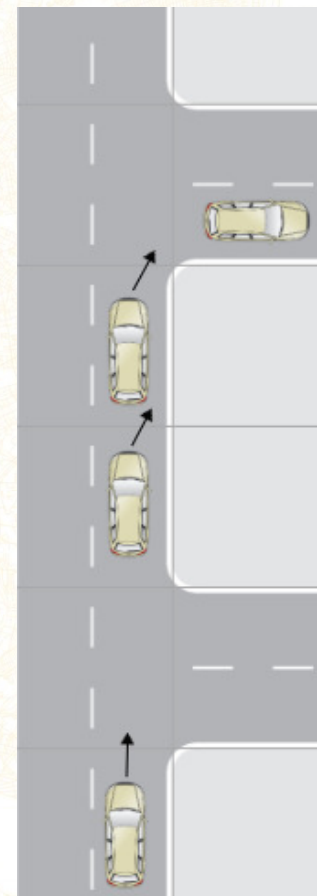
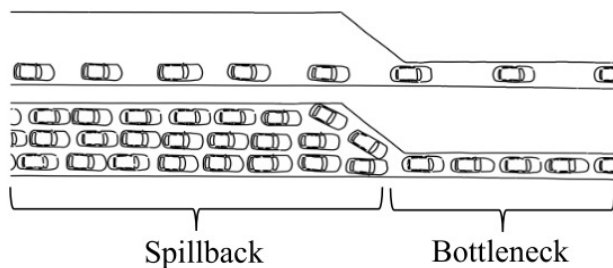
Demande : calcul d'itinéraires

- Des **algorithmes de calcul d'itinéraires** sont utilisés pour le choix de l'heure de départ (temps de trajet en fonction de l'heure de la journée pour une paire OD) et le choix d'itinéraire (itinéraire le plus rapide pour une origine, destination et heure de départ données)
- METROPOLIS2 utilise des algorithmes de **Contraction Hierarchies (time-dependent)**, permettant de calculer des milliers de requêtes par secondes sur grands réseaux
- **OpenTripPlanner** est utilisé pour calculer les itinéraires en transports en commun



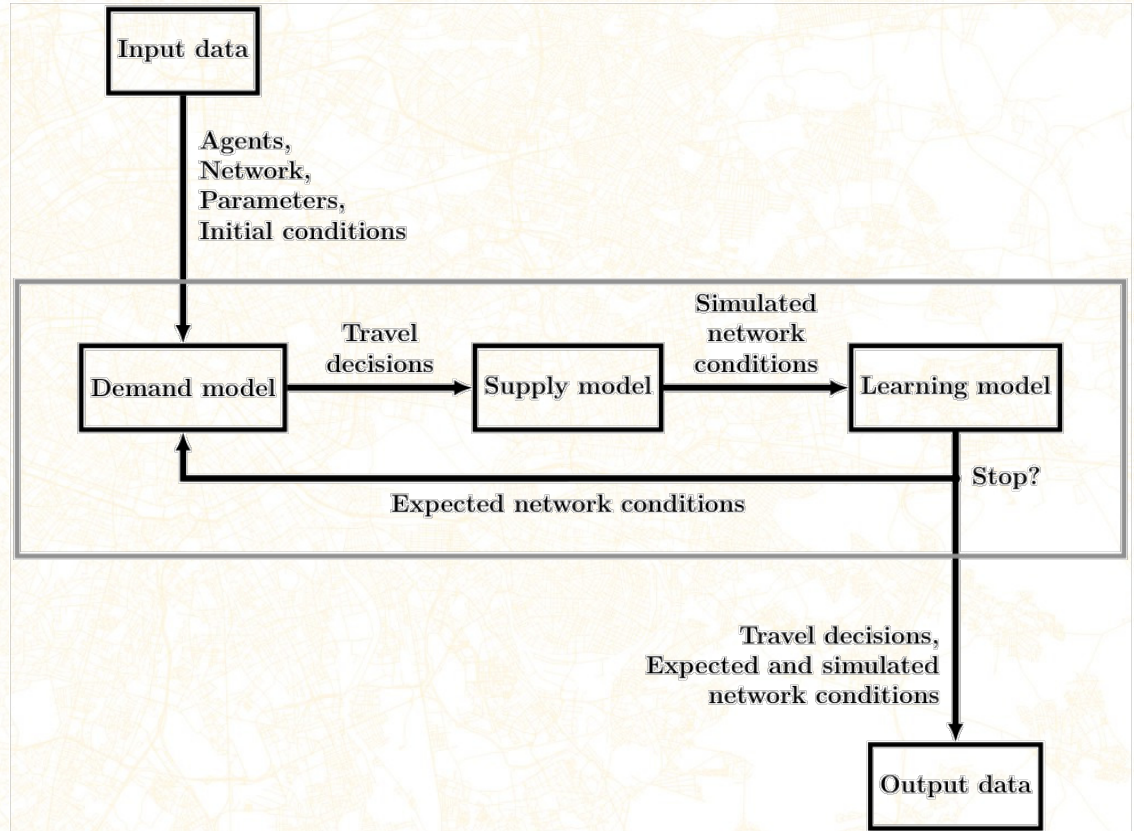
Offre : modélisation de la congestion

- **Goulots d'étranglements (bottleneck)** : le flux de véhicule (dynamique) est limité par la capacité
- **Propagation des files (spillback)** : le flux de véhicule est limité par la capacité de stockage du tronçon (# véhicules qui peuvent s'y diffuser)



Processus de convergence

Un processus **itératif** est utilisé pour ajuster les décisions des agents jusqu'à ce que la congestion anticipée coïncide avec la congestion simulée



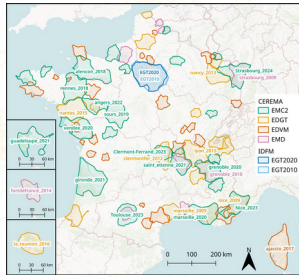
Calibration

Calibration semi-automatique des paramètres de la simulation par des approches de *Machine Learning*

Étape	Paramètres calibrés	Valeurs cibles	Données	Méthodologie
1	Vitesse et pénalité par tronçon	Temps de trajet sans congestion	TomTom API	Régression LASSO
2	Capacités des tronçons	Temps de trajet avec congestion	TomTom API	Régression OLS
3	Pénalités de des-horage	Distribution des heures de départ par cluster	EGT / EMC ²	Opt. bayésienne avec processus gaussiens
4	Valeurs du temps et constantes par mode	Parts modales par cluster	EGT / EMC ²	Régression Random Forest

Pipeline Pymetropolis

1. Collector



2. Configurer

```

1 main_directory = "chambery/output"
2
3 crs = "epsg:2154"
4
5 osm_file = "chambery/data/rhone-alpes-250101.osm.pbf"
6
7 [simulation_area]
8 # Simulation area is the Chambéry Aire d'attraction.
9 aav_name = "Chambéry"
10
11 [road_network]
12 min_lanes = 0.5
13 min_speed_limit = 10
14 remove_duplicates = false
15 ensure_connected = true
16 secondary_types = [
17     "living_street",
18     "unclassified",
19     "residential",
20 ]
21 ensure_primary_connected = true
22
23 [mode_choice]
24 modes = ["car_driver", "public_transit", "walking", "bicycle"]
25 model = "Logit"
26 mu = 1.0
27
28 [simulation]
29 period = ["03:00:00", "27:00:00"]
30 recording_interval = 900
31 learning_factor = 0.01
32 nb_iterations = 100
33 spillback = true
34 max_pending_duration = 60
35
36 [metropolis_ops]

```

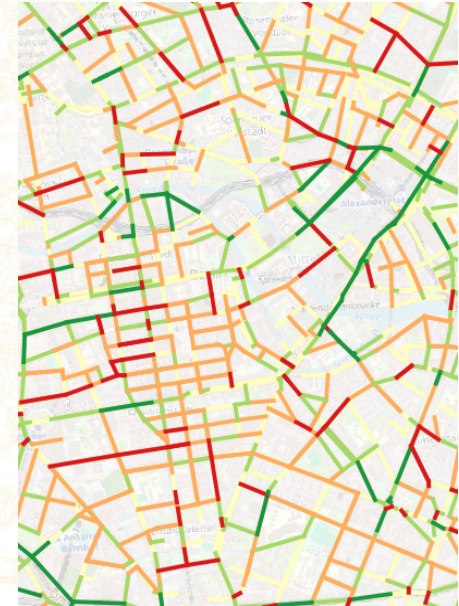
3. Exécuter

```

> uv run pymetropolis config-bottleneck.toml
1. GridNetworkStep
2. PostprocessRoadNetworkStep
3. ExogenousCapacitiesStep
4. EdgesFreeFlowTravelTimesStep
5. WriteMetroVehicleTypesStep
6. ODMatrixEachStep
7. RoadNetworkPrimaryEdgesStep
8. WriteMetroEdgesStep
9. TripsCarFreeFlowTravelTimesStep
10. GenericPopulationStep
11. CarAccessEgressStep
12. CarDriverPreferencesStep
13. LinearScheduleStep
14. WriteMetroAgentsStep
15. HomogeneousTstarStep
16. UniformDrawsStep
17. WriteMetroTripsStep
18. WriteMetroAlternativesStep
19. WriteMetroParametersStep
20. RunSimulationStep
21. RoadNetworkCongestionFunctionPlotsStep
22. IterationResultsStep
23. TripResultsStep
24. TripModeSharesStep
25. TripDepartureTimeDistributionStep
26. ConvergencePlotStep

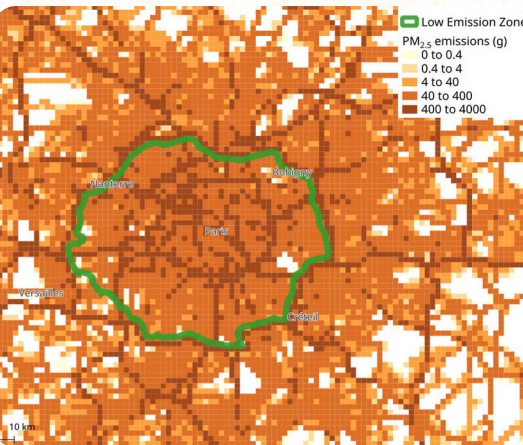
```

4. Analyser

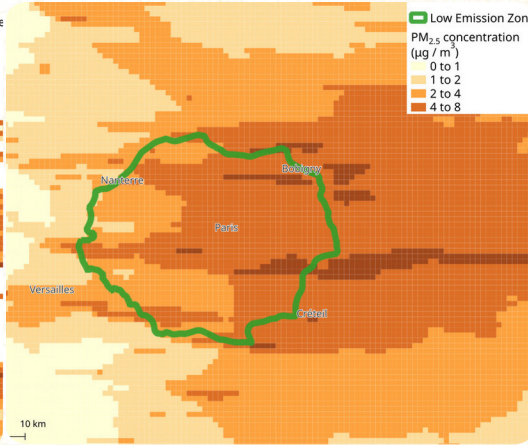


METRO-TRACE

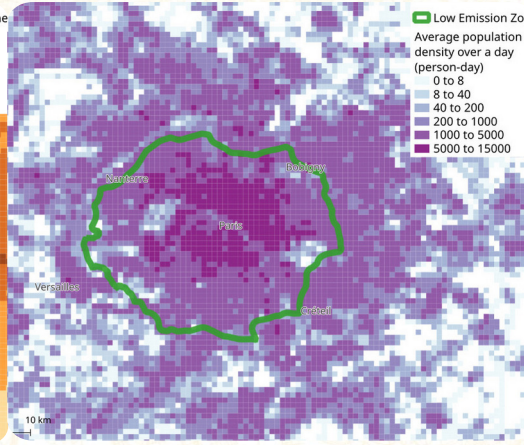
1. Émissions



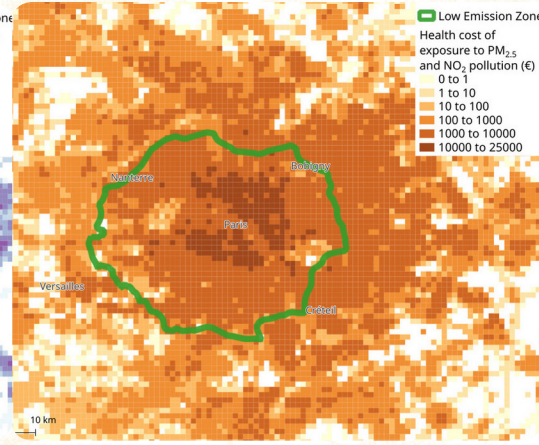
2. Dispersion



3. Densité population



4. Exposition



Questions ?

Lucas Javaudin

lucas.javaudin@enpc.fr

<https://lucasjavaudin.com>

<https://metropolis2.org>