Transport Network Design Based on Origin/Destination Clustering During the COVID-19 Pandemic Use Case

Matthieu Guillot, Angelo Furno, El-Houssaine Aghezzaf, Nour-Eddin El Faouzi

LABILITY Smart Lab Paris Region

LICIT-ECO7

Université Gustave Eiffel

Université de Lyon

ENTPE
L'école de l'aménagement durable des territoires

Smartlab Lability

The Smartlab

- Temporary laboratory : from January 2021 to January 2023
- Funded Île-De-France region (around Paris) and driven by Gustave Eiffel University
- 9 postdoc researchers
- Multidisciplinary : economy, sociology, applied mathematics, computer science...

Objectives

- Studying the resilience of Île-De-France region during and after the COVID-19 pandemic
- 2 axes : mobility and telework

Context

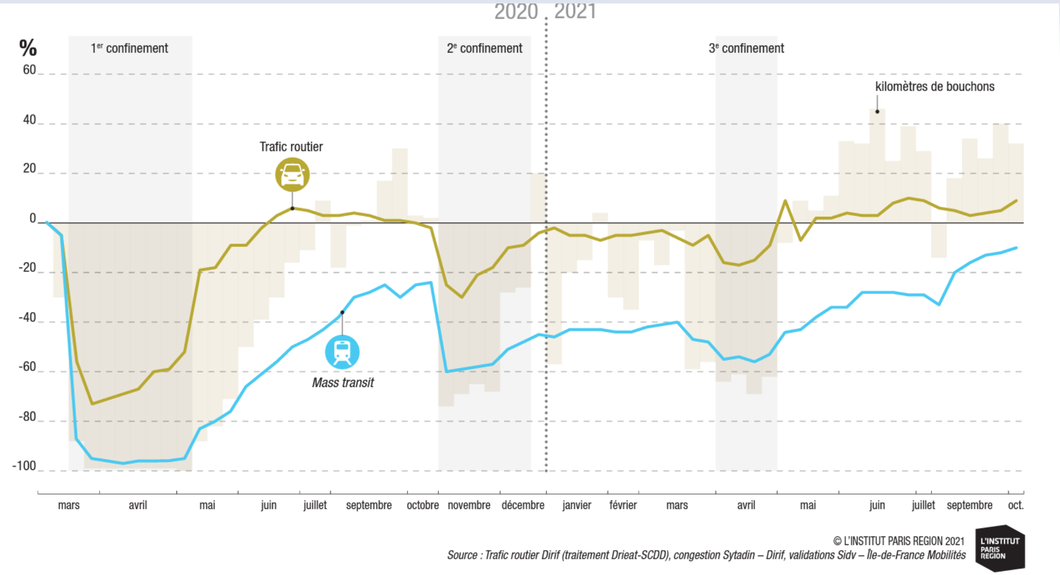
Mobility changes

- 17% of users have changed their mobility habits with COVID-19
- Most of them from collective modes to individual ones

Modes

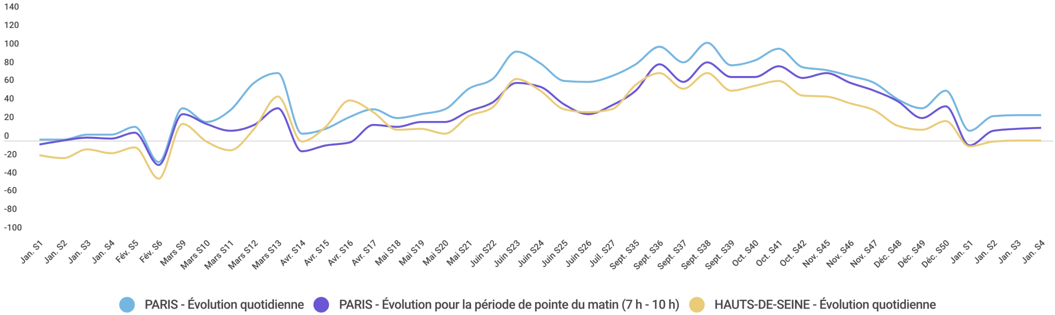
- Decrease of mass transit
- Increase of the use of private cars
- Increase in cycling

Context



2020
2021
%
60
40
20
0
-20
-40
-60
-80
-100
1er confinement
2e confinement
3e confinement
kilomètres de bouchons
Trafic routier
Mass transit
mars
avril
mai
juin
juillet
septembre
octobre
novembre
décembre
janvier
février
mars
avril
mai
juin
juillet
septembre
oct.
© L'INSTITUT PARIS REGION 2021
Source : Trafic routier Dirif (traitement Drieat-SCDD), congestion Sytadin – Dirif, validations Sidv – Île-de-France Mobilités

Context


140
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
Jan. S1
Jan. S2
Jan. S3
Jan. S4
Fév. S5
Fév. S6
Mars S9
Mars S10
Mars S11
Mars S12
Mars S13
Avr. S14
Avr. S15
Avr. S16
Avr. S17
Mai S18
Mai S19
Mai S20
Mai S21
Juin S22
Juin S23
Juin S24
Juin S25
Juin S26
Juil. S27
Sept. S35
Sept. S36
Sept. S37
Sept. S38
Sept. S39
Oct. S40
Oct. S41
Oct. S42
Nov. S45
Nov. S46
Nov. S47
Déc. S48
Déc. S49
Déc. S50
Jan. S1
Jan. S2
Jan. S3
Jan. S4
PARIS - Évolution quotidienne
PARIS - Évolution pour la période de pointe du matin (7 h - 10 h)
HAUTS-DE-SEINE - Évolution quotidienne

Consequences

Social

- Increased incivilities between cyclists and car users
- Increased number of accidents involving cyclists
- Increase of traffic jams

Financial

- Revenue decline, shortfalls for public transports operators
- 1 billion (€) for Île-De-France Region

Demand

- Public transportation demand has decreased (even now)
- Possible future changes in the demand

Consequences

Public transport networks must be reconfigured to increase resilience

We want :

- A new network that fits the new (decreasing) demand
  $\rightarrow$ sustainability for network operators
- A good quality of service
- A dynamic solution
- Good properties under sanitary constraints

How to tackle the problem ?

Which mode ?

- Re-use of equipment
- Light infrastructure

  $\rightarrow$ Bus network

Subnetwork in terms of bus stops [1]

- Easier to compute
- Bound the access/egress time
- But the operational cost is not that related with bus stops
- Our approach : work directly on the lines

1. https ://doi.org/10.48550/arXiv.2105.02600

Introduction to the problem

We consider a bus network $G = (V, E)$:

- V : set of bus stops
- E set of edges connecting these stops

Entries :

- $d(v_1, v_2)$ a distance metric between all $v_1 \in V$ and $v_2 \in V$
- $OD(v_1, v_2)$ a demand on the bus stops between all $v_1 \in V$ and $v_2 \in V$

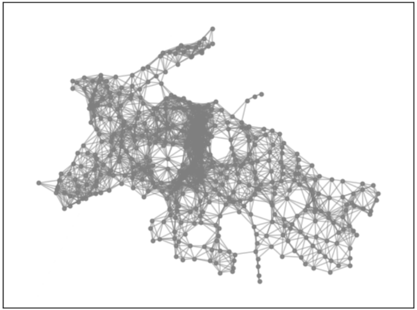


Figure – Bus network of Lyon

Introduction to the problem

We consider a bus network $G = (V, E)$:

- V : set of bus stops
- E set of edges connecting these stops

Output :

- a set of lines on V
- with good properties

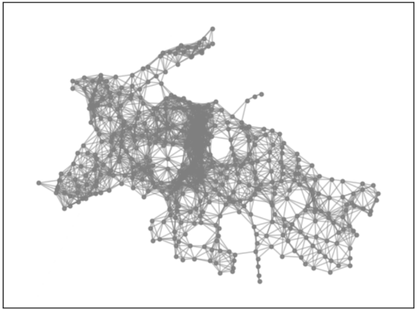


Figure – Bus network of Lyon

Introduction to the problem

We consider a bus network $G = (V, E)$:

- V : set of bus stops
- E set of edges connecting these stops

Output :

- a set of lines on V
- with good properties

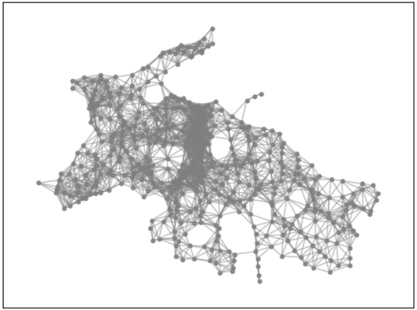
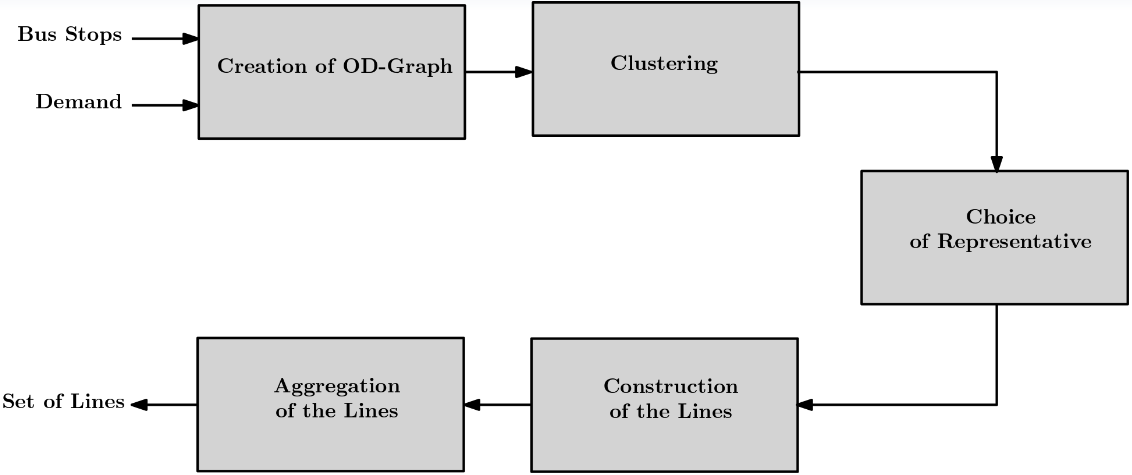


Figure – Bus network of Lyon

Main idea : group origin/destination pairs that are close

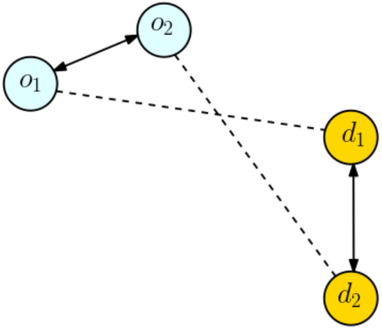
Outline


Bus Stops
Demand
Creation of OD-Graph
Clustering
Choice
of Representative
Construction
of the Lines
Aggregation
of the Lines
Set of Lines

Creation of OD-Graph

We define $G' = (V', E')$ the OD-graph as :

- V' : set of OD pairs
- E' set of edges that connect close ODs
- d_{walk} a acceptable distance by walk



$$V' = \{v' = (o, d) \in V | d(o, d) > d_{walk}\}$$

$$E' = \{(v'_1 = (o_1, d_1), v'_2 = (o_2, d_2)) \in V' | d(o_1, o_2) + d(d_1, d_2) < d_{walk}\}$$

Clustering

We apply a clustering algorithm on G'

- Louvain, spectral algorithm, ...
- c : the number of clusters

$\mathcal{C} = \{C_1, C_2, ..., C_c\}$ the set of clusters of ODs

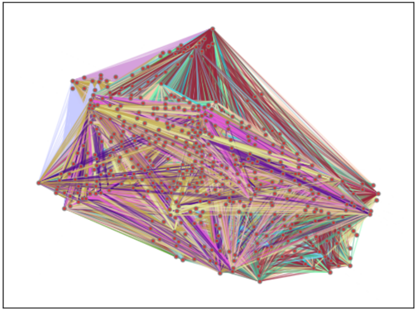


Figure – $c = 32$

Choice of representatives

For each $C_i \in \mathcal{C}$, we compute a representative (a node $r(C_i) \in C_i$) :

- central
- demand-dependant

$$D^{C_i} = \min_{v \in C_i} \sum_{u \in C_i} D(u,v) OD(u)$$

$$r(C_i) = \arg\min_{v \in C_i} D^{C_i}$$

D : minimal distance in G' (in term of # of edges)

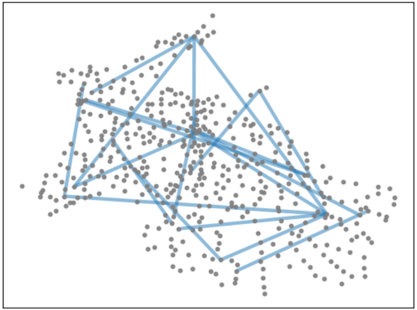


Figure – representatives for $c = 32$

Construction of the lines

For each $C_i \in \mathcal{C}$, we build a bus line from $r(C_i)$

- back in G, $r(C_i) = (o_{C_i}, d_{C_i})$
- we build a line from o_{C_i} to d_{C_i}

We compute the shortest path in G between o_{C_i} and d_{C_i}

$$l_{C_i} = (o_{C_i}, .., d_{C_i}) = SP_G(o_{C_i}, d_{C_i})$$

Construction of the lines

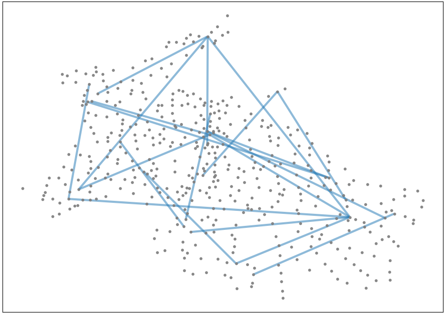


Figure – Representatives for $c = 32$

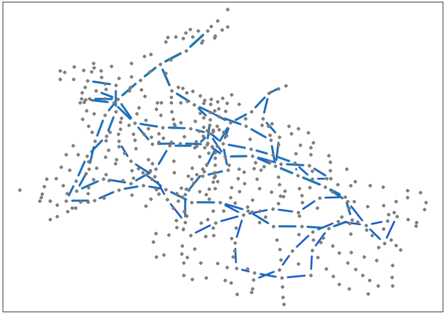


Figure – Corresponding lines

Aggregation of the lines

Issue : some lines are built from different "size" of OD. We choose p_{aggreg}, and define :

$$p_{com}(l_1, l_2) = \frac{|l_1 \cap l_2|}{\min(|l_1|, |l_2|)}$$

We merge the lines until no pair l_1, l_2 such that $p_{com}(l_1, l_2) > p_{aggreg}$ exists

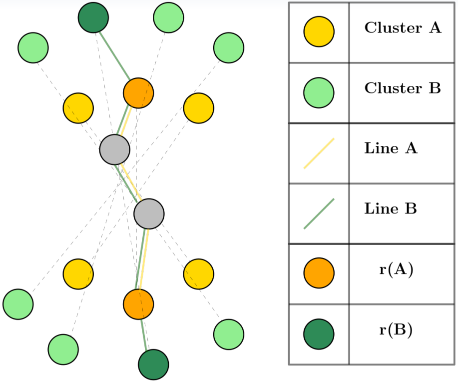


Figure – aggregation of the lines

Aggregation of the lines

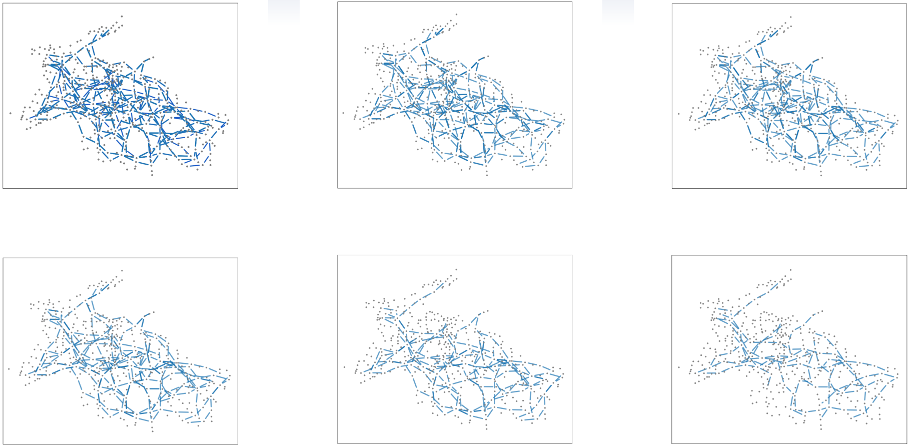
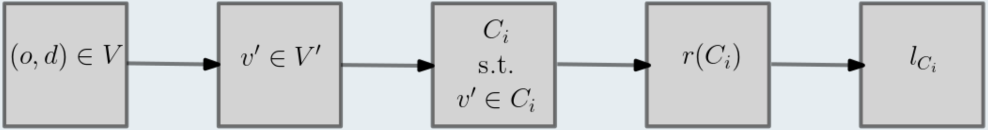


Figure – Procedure of aggregation for p_{aggreg} from 0.7 to 0.3

Network properties

One only bus line

- for any trip (o, d), one can take only one bus line
- we can tell which one to take



Sanitary constraints

- only one bus to take, no transfer
- decrease of risks of contamination

Network properties

Capacity and frequency of the buses

We can easily define the bus capacities and frequencies since only one line is taken per trip $\rightarrow$ sanitary constraints

Walking time

We can also bound the additional walking time for $v' \in C_i$:

$$WT(v') \leq D(v', r(C_i)) \times d_{walk}$$

Geographical equity

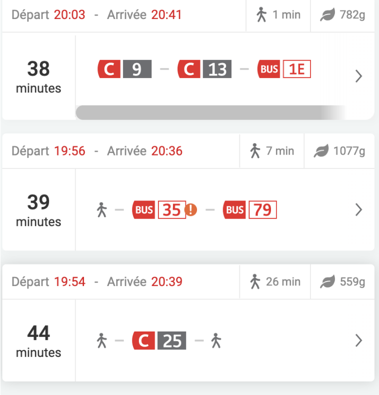
- the demand is taken into account only in the choice of representative
- no spacial preferential treatment
- suburb $\rightarrow$ suburb trips are not penalized

Evaluation of the network

p_{eggreg}	# lines	% covered stops
1	150	55%
0.7	76	53%
0.6	69	52%
0.5	56	50%
0.4	35	44%
0.3	20	33%

Travel time : Toy example

Trip : Guillotière → Mermoz



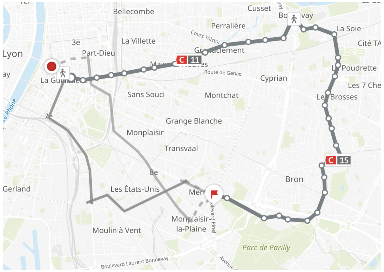
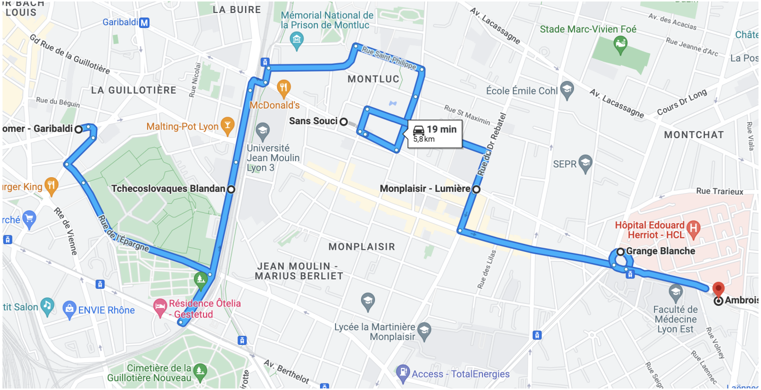


Figure – Advised trip from TCL

Travel time : Toy example

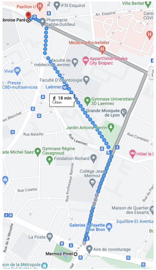
Trip : Guillotière $\rightarrow$ Mermoz

Our line : Domer $\rightarrow$ Ambroise Paré



Travel time : Toy example

Access and egress time : 4 and 18 minutes



Total : 40 minutes + waiting time at stops <u>without transfer</u>

Conclusion & Perspectives

Work done

- a methodology to build bus lines from existing bus stops
- the lines have some interesting properties
- the methodology is modular

Perspectives

- evaluation of the network with travel time metrics
- evaluation of the increase of access time
- comparison with existing network
- test other choices for the different steps of the methodology

Thank you for your attention !

Algorithm 1 Aggregate lines

Require: a set of lines $\mathcal{L} = l_1, .., l_c$

Ensure: a subset $\mathcal{L}' \subseteq \mathcal{L}$ such that $p_{com}(l, l') < p_{aggreg}$ for all pair $(l, l') \in \mathcal{L}'$

```
continue ← True
L_curr ← L
while continue do
   continue ← FALSE
   for l ∈ L_curr do
      for l' ∈ L_curr do
         if l ≠ l' AND p_com(l, l') > p_aggreg then
            if |l| < |l'| then
               Delete l from L_curr
            else
               Delete l' from L_curr
            end if
            continue ← TRUE
         end if
      end for
   end for
end while
```