Smallworld

Louis Cohen & Adele Mortier

MPRI 2017

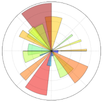
February 25, 2018

Introduction

The aim of the project was to mimic the people's behavior in a big city. It tackles the following problematics :

- What do people do in their everyday life and at what time ?
- In what kind of environment do they live and how do they interact with the infrastructures ?

Tools we used :



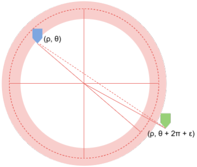
Topology – main steps

We tried to model a subway network similar to the Parisian one.

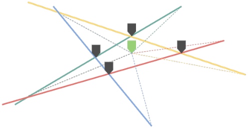
Steps

1. **Terminals** : subway lines modeled as segments.
2. **Intersections** : intersections between lines are glued together if they are close.
3. **Stations** : points at regular intervals (with a little noise).
4. **Hubs** : stations crossed by many lines
5. **Fast lines** : a few lines that connect close hubs together.

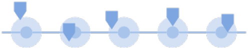
Topology – illustrations



(a) Terminals : originate in the suburbs, go through the center



(b) Intersections : move close intersections to their centroid



(c) Stations : sample the line and add noise

Topology – results

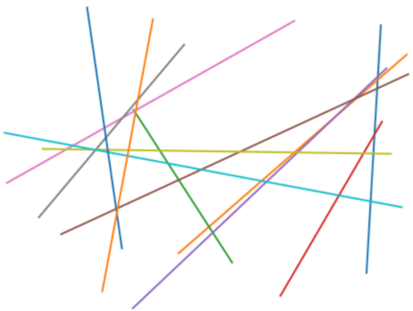


Figure: Terminals generation – subway lines are simple segments

Topology – results

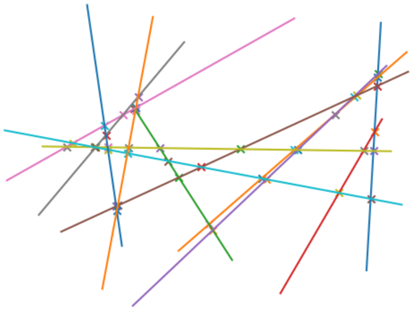


Figure: Intersection resolution – find where the lines cross using SymPy

Topology – results

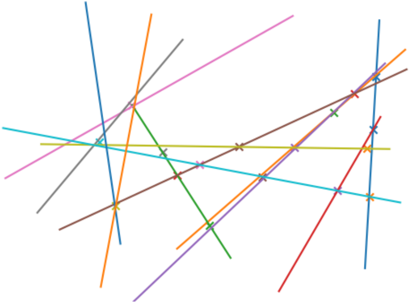


Figure: Intersection gluing – merge close intersections using a clustering algorithm (DBSCAN)

Topology – results

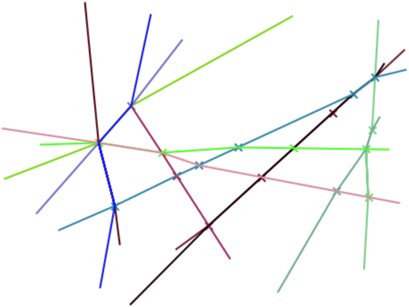


Figure: Line bending – bend te lines such that they cross glued intersections

Topology – results

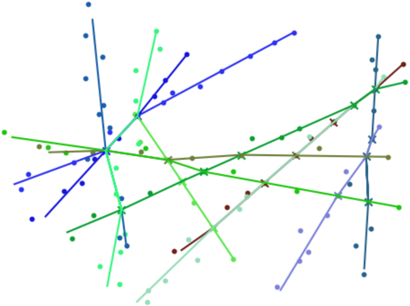


Figure: Stations – put stations at regular intervals plus a little noise

Topology – results

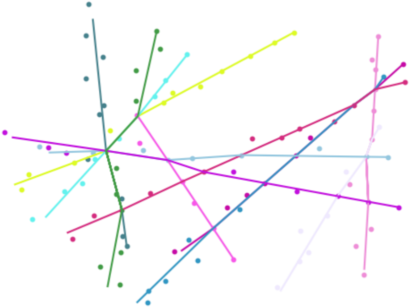


Figure: Stations gluing – merge close intersections with DBSCAN again

Topology – results

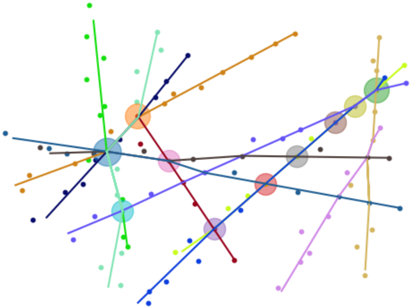


Figure: Hubs – find stations with many lines crossing them and generate fast lines

Toponymy – main steps

Generate a “realistic” name for each station, like “Place Edith Piaf”, “Rue de la Chine” or “Saint Marcel”

Steps

1. **Data collection** : collect names from databases (country names, first names) or manually (famous people).
2. **Combine elements together** : use link words (“Place de la/le”, “Saint(e)”, “-”...) appropriately.
3. **Do some tricks** : avoid things like “Place d’Arc” or “Avenue de Maupassant”...

“Best-of” : “Avenue Johnny Hallyday”, “Gare Nabilla”, “Rue du Swaziland”...

Toponymy

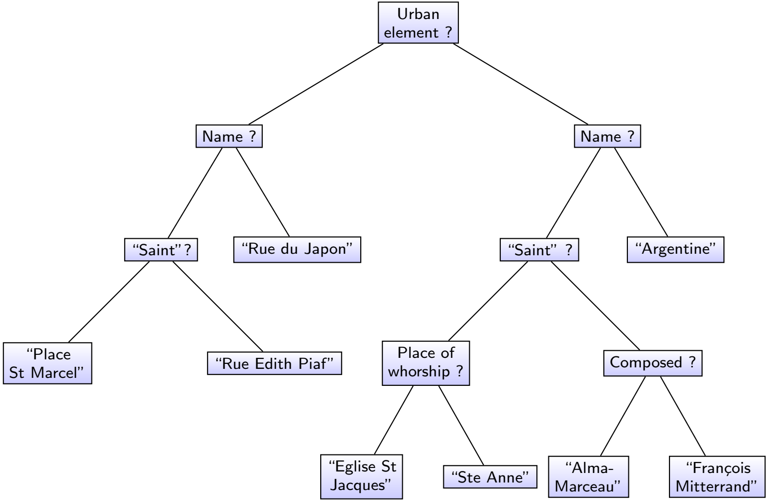


Figure: Simplified binary tree for name generation

Schedule – main steps

Generate a schedule for each station and deduce point to point travel times.

Steps

1. Compute travel times between stations, depending on the line speed and the distances between stations.
2. Generate departure times from the terminals with a frequency that varies during the day.
3. Propagate the departure times along the line using the values computed at first step.
4. Use the schedule to compute a shortest path that is sensitive to de day/hour of departure (**not implemented**)

Schedule – illustration

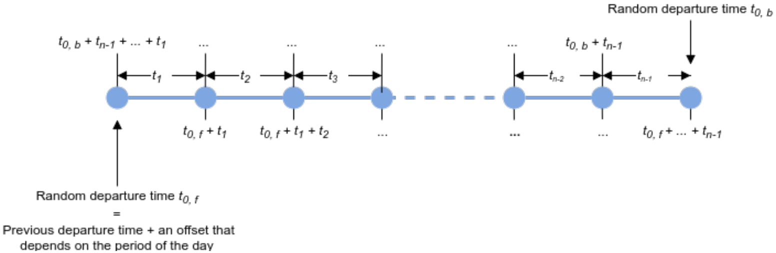

Random departure time $t_{0,b}$
$t_{0,b} + t_{n-1} + ... + t_1$
...
...
...
...
$t_{0,b} + t_{n-1}$
t_1
t_2
t_3
t_{n-2}
t_{n-1}
$t_{0,f} + t_1$
$t_{0,f} + t_1 + t_2$
...
...
...
$t_{0,f} + ... + t_{n-1}$
Random departure time $t_{0,f}$
=
Previous departure time + an offset that
depends on the period of the day

Figure: Computation of a line of the schedule for one subway line

Key ideas

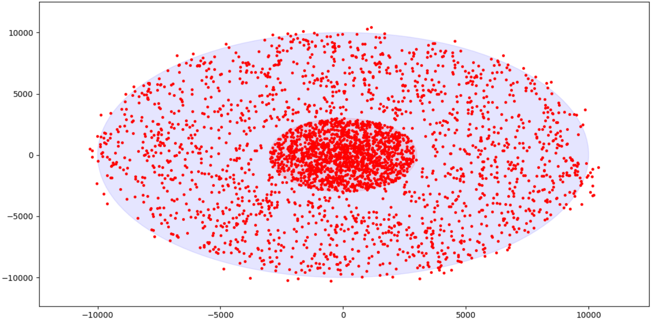
- Families
- Work
- Activities
- Home
- Deplacement between those points !

Localisation ideas

Center : working
Subburbs : housing
(all probabilistic)

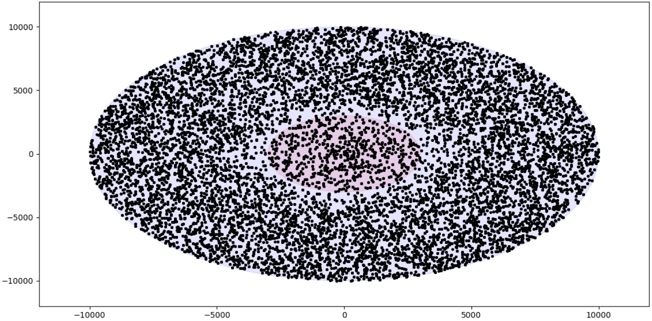
Localisation ideas

Center : working
Subburbs : housing
(all probabilistic)



Localisation ideas

Center : working
Subburbs : housing
(all probabilistic)



Activities

For now : uniform generation + some more on the hubs

A person

- Id
- Age
- Work type and informations
- Work location
- Home location
- Family
- Typical activities
- Planning

Persons models

How to differentiate

Typical activities
Days worked
Work location distribution

Persons type

- Students : work near home location, ludic activities, student weak
- White collar : working in center, groceries, different work shedule
- Unenmployed, outdoor work . . .

Generating families

- one or two parents
- several childs

modularity

Proba of being single
Number of child proba repartition
Monoparental families...

Planning

Being at work during worktime
Random activity on day activity
Going back to home to sleep

demonstration

Modularity

Depending on other people, family
More activities
Special activities only on some locations
Special events (worldcup etc...)

Deplacement from x to y :
s_x, s_y the stations
Shortest_path(s_x, s_y)

Use of the different lines

How many persons use it ?
Shedules
Breakdown sensibility

To go further

Special events
Adding other way of deplacement
Feeding it as a blackbox to other learning algorithm

Merger of the two projects – roadmap

What we got

- Pairwise shortest paths between stations.
- Series of travels from one point to another.

What we want

- Retrieve the itinerary for each travel !

Merger of the two projects – illustration

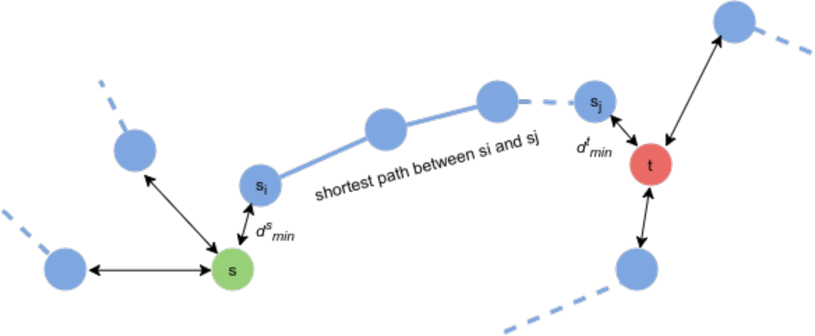


Figure: Computation of point-to-point shortest paths

Merger of the two projects – result

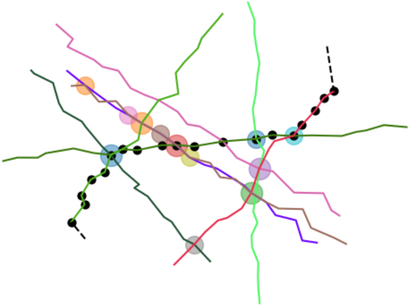


Figure: Shortest path – compute point to point shortest path using SymPy and NetworkX

Conclusion

- Complex and quite “realistic” city with many adjustable parameters.
- But define “realistic” ? Which metric could we use ?
- Possible refinements : animated viz’, wider range of public transport, wider range of socio-professional categories...