

---

# Plot4GMNS

**An open-source academic research tool for visualizing multimodal networks for transportation system modeling and optimization**

Authors: Dr. Junhua Chen, Zanyang Cui

Email : [cjh@bjtu.edu.cn](mailto:cjh@bjtu.edu.cn) , [zanyangcui@outlook.com](mailto:zanyangcui@outlook.com)

## 1. Introduction

To enable rapid transportation modeling and optimization, as railroad management researchers, we provide this free open-source tool for visualizing multimodal networks. Based on GMNS data format by Zepha foundation (<https://zephyrtransport.org/>), plot4gmns is designed for reading and plotting multimodal data sets including transportation network files, demand and agent trace files.

## 2. Data Format

The overarching goal of this package is to offer flexibility in allowing different inputs while closely following the core GMNS specifications. Users can first prepare base network files such as node.csv, link.csv in GMNS format and poi.csv (if available from the osm2gmns package) to display the modeling network. One can also prepare additional data files (such as input\_agent.csv, demand.csv, poi\_trip\_rate.csv, zone.csv from the grid2demand package) to visual network-related information layers. For example, the core input csv files based on GMNS include.

node.csv

name	str
node_id	int
ctrl_type	int (0 or 1)
activity_type	str
x_coord	float

---

y_coord	float
production	float
attraction	float

link.csv

link_id	int
from_node_id	int
to_node_id	int
length	float
lanes	int
free_speed	float
capacity	float
link_type_name	str
geometry	LINESTRING
allowed_uses	str

poi.csv

poi_id	int
building	str
geometry	POLYGON, MULTIPOLYGON
activity_zone_id	int

demand.csv

o_zone_id	int
d_zone_id	int
volume	float
geometry	LINESTRING

input\_agent.csv

agent_id	int
o_zone_id	int

---

d_zone_id	int
geometry	LINESTRING

poi\_trip\_rate.csv

building	str
production_rate1	float
attraction_rate1	float

zone.csv

activity_zone_id	int
centroid_x	float
centroid_y	float
geometry	POLYGON
centroid	POINT
total_poi_count	int
residential_poi_count	int
office_poi_count	int
shopping_poi_count	int
school_poi_count	int
parking_poi_count	int
boundary_node_count	int
total_production	float
total_attraction	float

### 3. Major Functions

function	description
readNetwork()	Reading multimodal data sets from given network files.
showNetByAllMode() showNetByAutoMode()	Visualizing network by a given allowed transportation mode on the link or all modes.

---

showNetByBikeMode()	(auto, bike, walk, rail are supported)
showNetByWalkMode()	
showNetByRailMode()	
showNetByNodeAttr()	Visualizing network by given node attributes and values.
showNetByNodeProduction()	Visualizing network with the size of node circle is proportional to the production value.
showNetByNodeAttraction()	Visualizing network with the size of node circle is proportional to the attraction value.
showNetByLinkAttr()	Visualizing network by given link attributes and values.
showNetByLinkFreeSpeed()	Visualizing network with the width of links is proportional to the free speed value.
showNetByLinkCapacity()	Visualizing network with the width of links is proportional to the capacity value.
showNetByLinkLaneNum()	Visualizing network with the width of links is proportional to the lanes number.
showNetByPOIAttr()	Visualizing network by given POI attributes and values.
showNetByZonePOIAtrractionDensity()	Showing heatmap in terms of POI attraction density.
showNetByZonePOIProductionDensity()	Showing heatmap in terms of POI production density.
showNetByZoneDemandHeatMap()	Showing the heatmap origin demand volume by zone-to-zone demand.
showNetByZoneDemandFlow()	Showing network by zone demand traces.
get_avl_node_attrs() get_avl_link_attrs() get_avl_poi_attrs()	Checking available attributes of nodes, links, and POIs.

---

get_avl_range_of_zone_ids()	Checking available range of zone ids.
get_num_of_nodes_by_attr()	Counting the number of a given node attribute.
get_num_of_links_by_attr()	Counting the number of a given link attribute.
get_num_of_pois_by_attr()	Counting the number of a given POI attribute.
get_range_of_node_attr()	Checking the range of a given node attributes.
get_range_of_link_attr()	Checking the range of a given link attributes.
get_info_of_zone_by_id()	Checking the information of given zone ids.

### 3. Using plot4gmns

#### 3.1 Installation

```
pip install plot4gmns
```

#### 3.2 Building mode network from csv files

(1) import the package

```
import plot4gmns as pg
```

(2) specify the file directory and read csv files

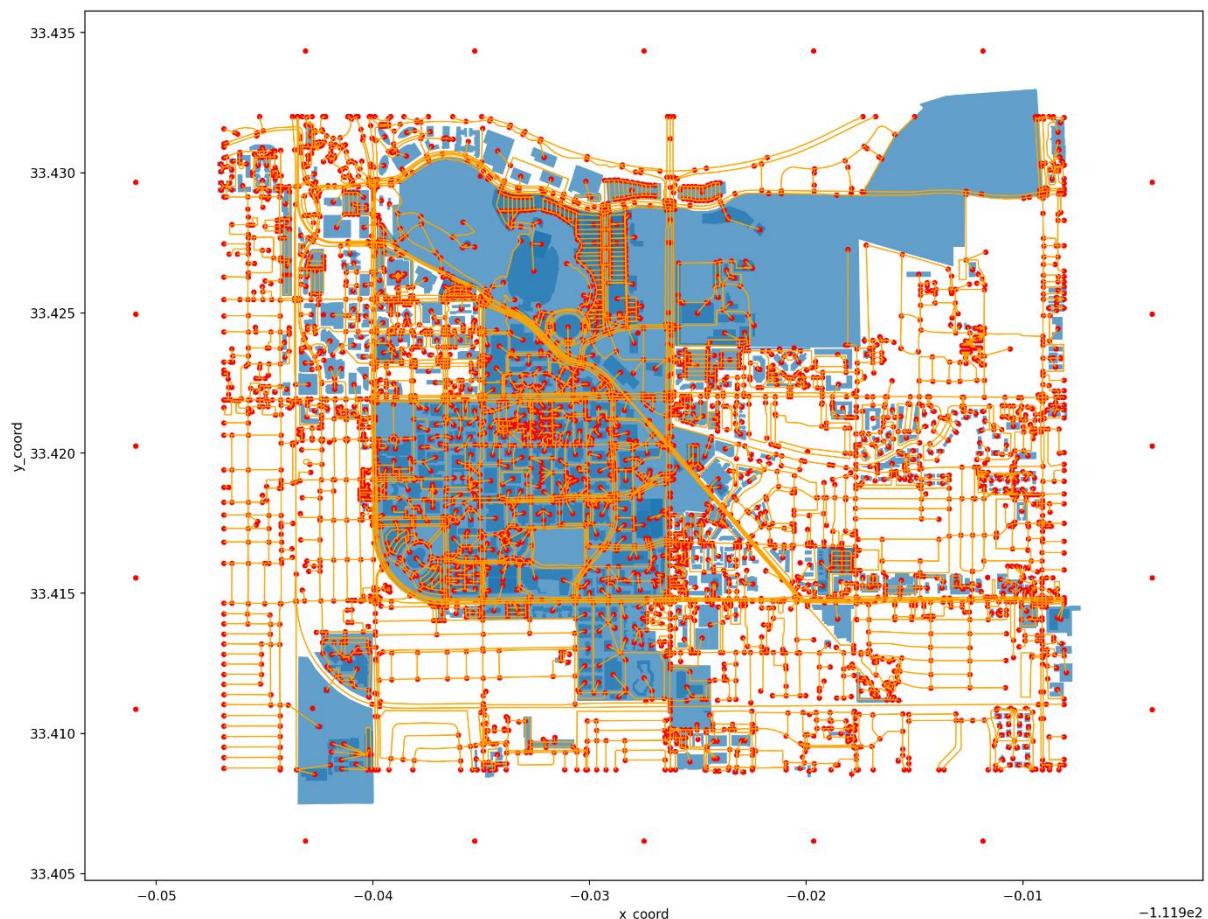
```
net=pg.readNetwork('./data')
```

#### 3.3 Visualizing network of different modes

(1) show network of different modes (all, auto, bike, walk, rail)

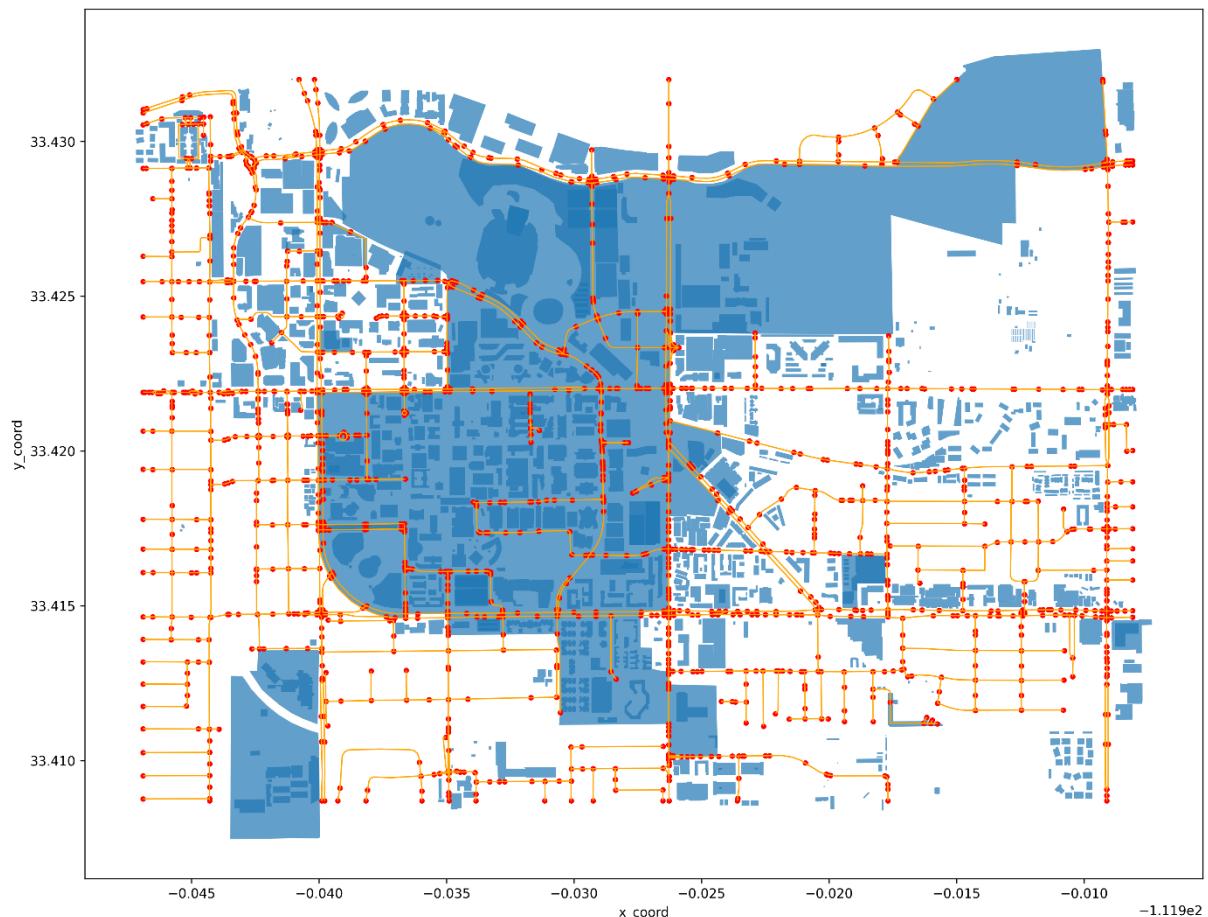
```
pg.showNetByAllMode(net)
```

output:



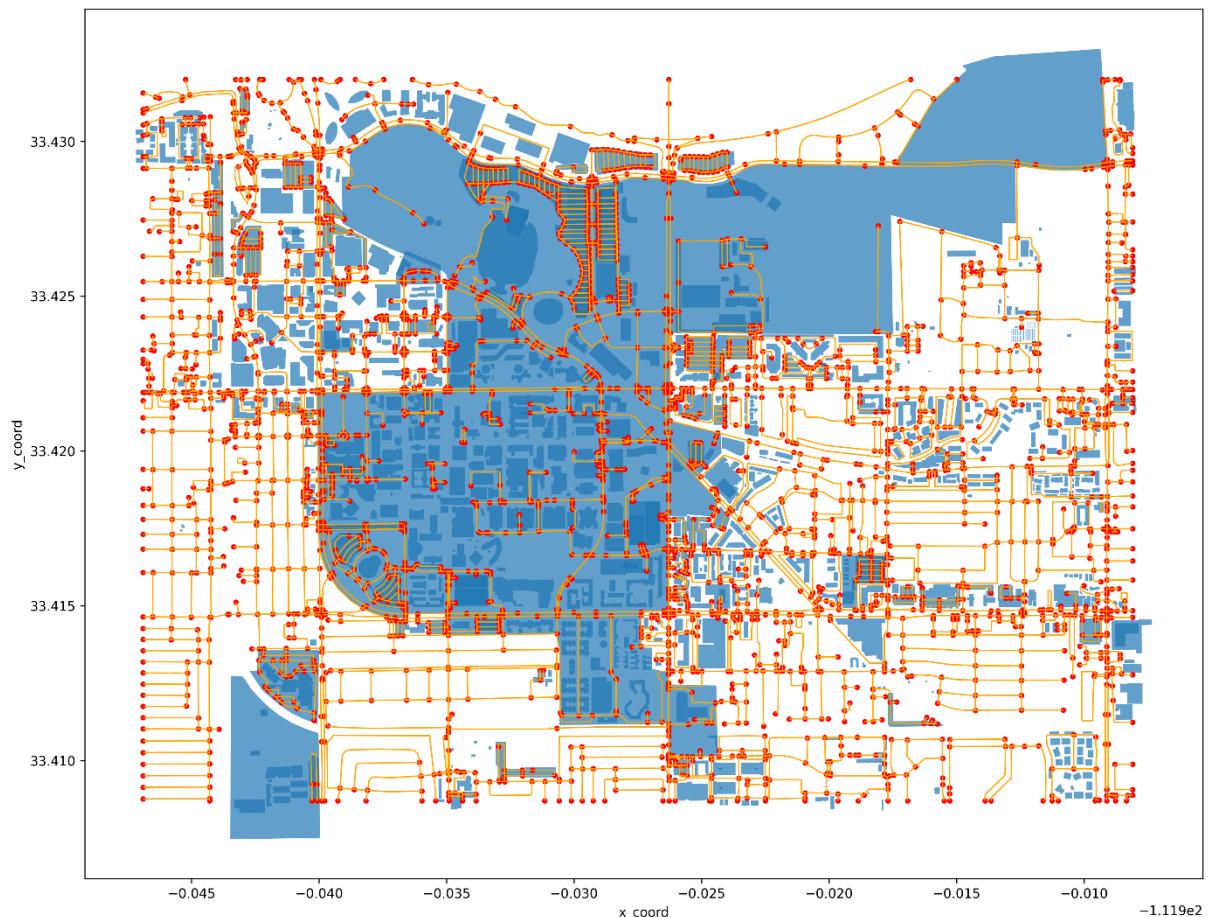
```
pg.showNetByAutoMode(net)
```

output:



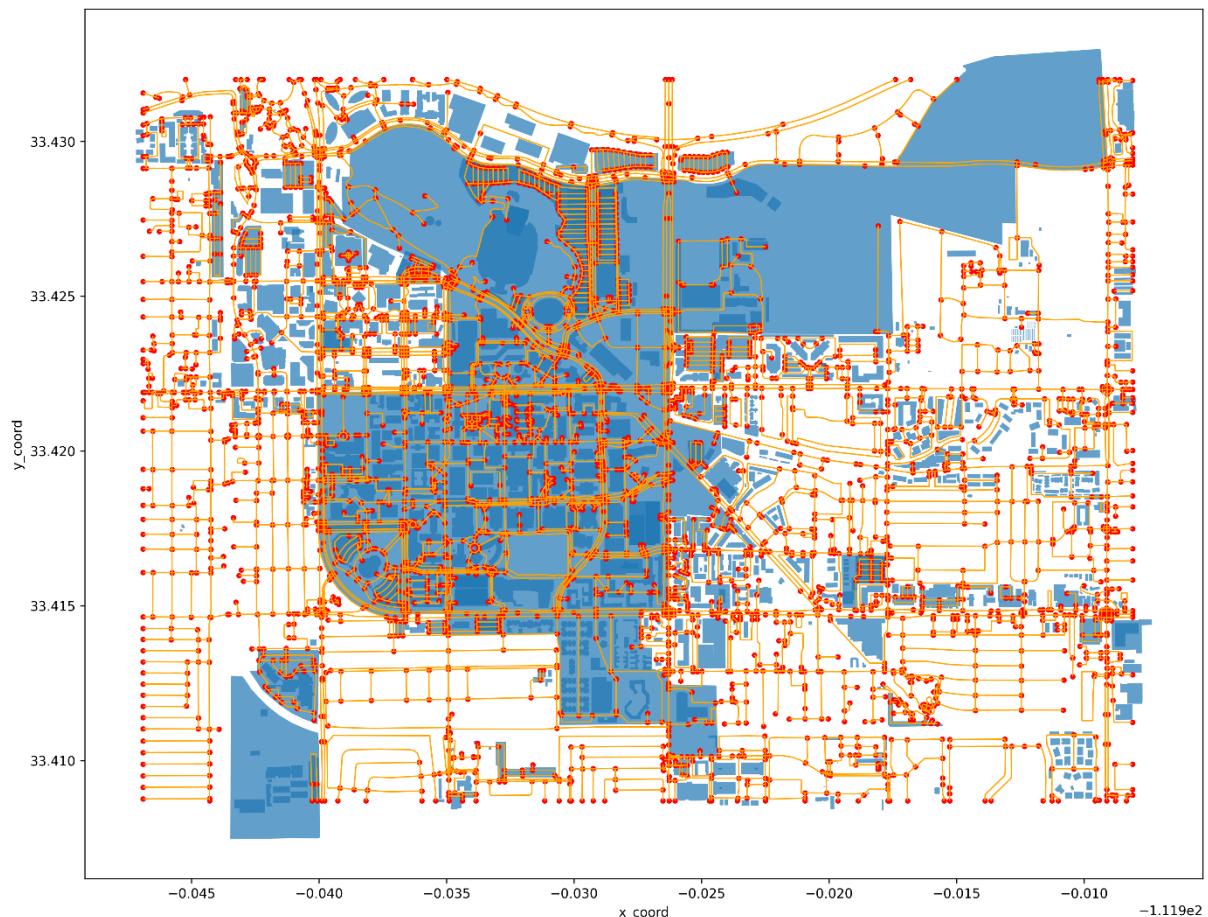
```
pg.showNetByBikeMode(net)
```

output:



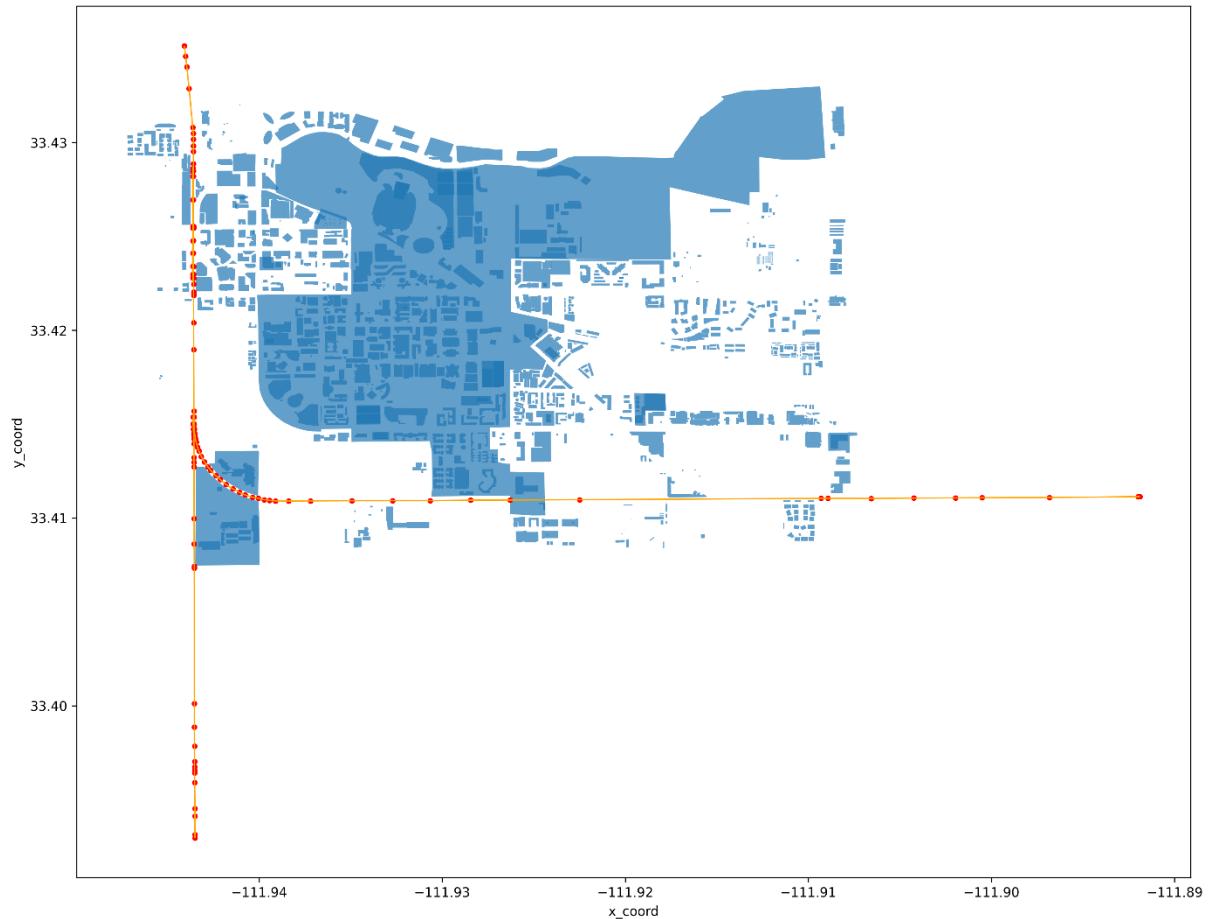
```
pg.showNetByWalkMode(net)
```

output:



```
pg.showNetByRailMode(net)
```

output:



### 3.4 Visualizing network by node attributes

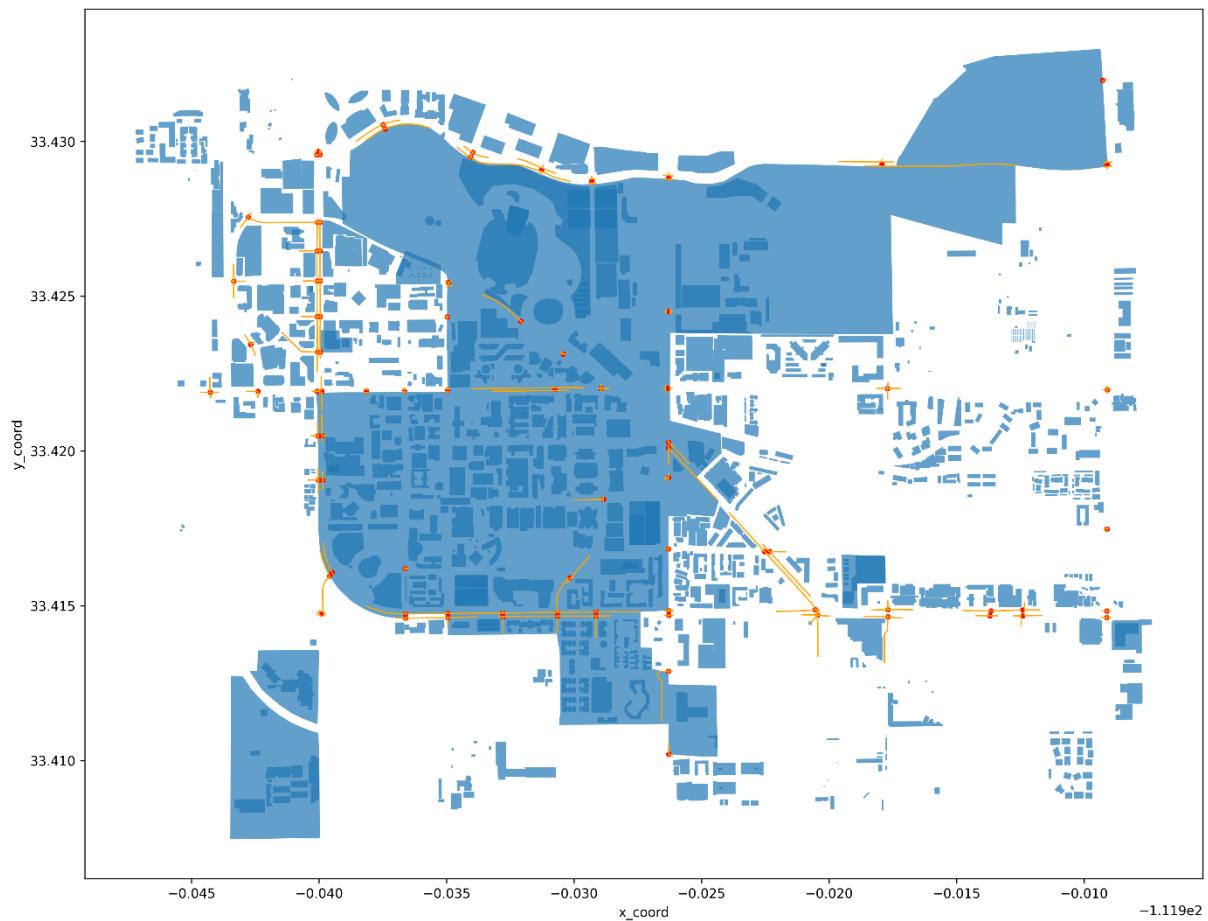
Only available node attributes and values can be displayed. Thus, the first step is to check if the node attribute fields and values exist.

- net.get\_valid\_node\_attr\_list()
- pg.get\_node\_attr\_value\_list(net,attr)

(1) show network by node ‘ctrl\_type’ for 1 as signalized intersection

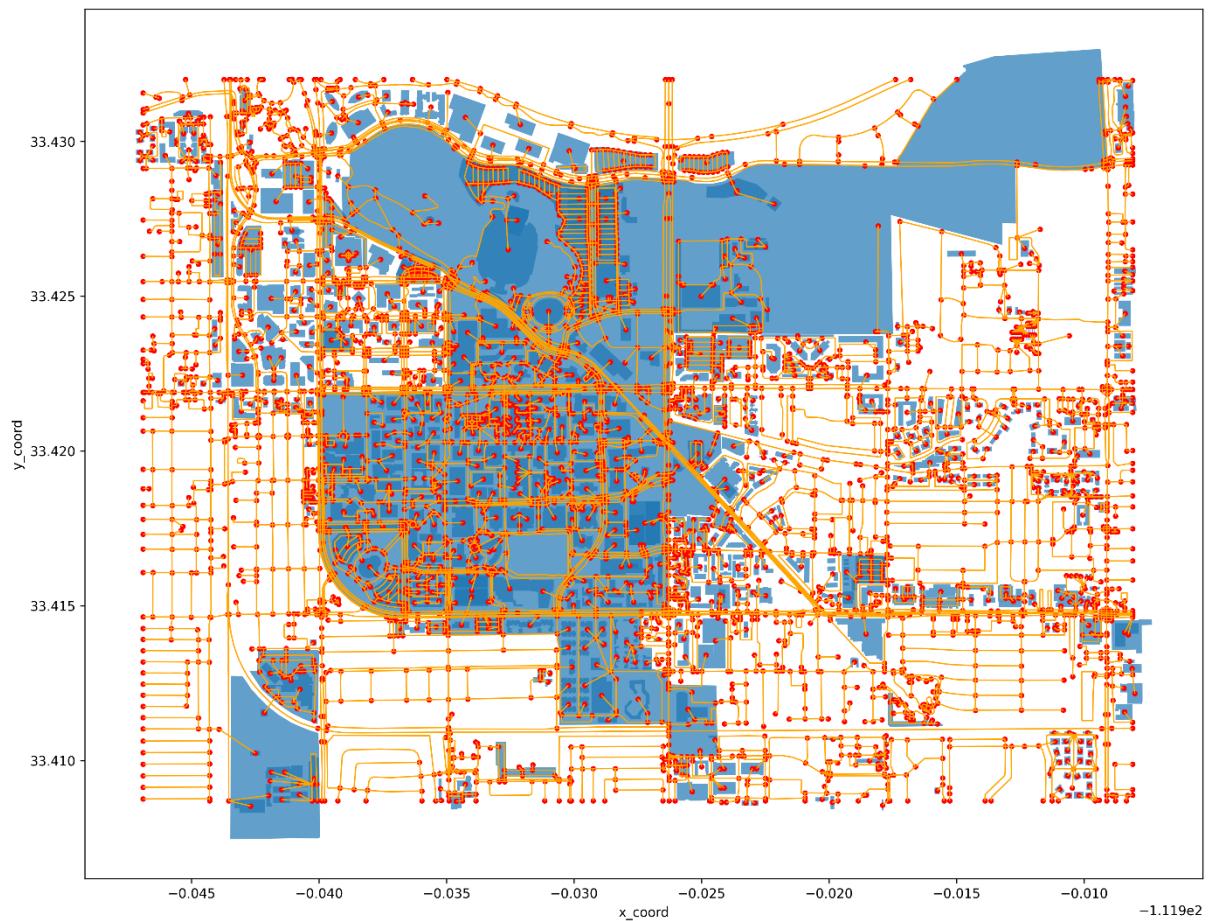
```
pg.showNetByNodeAttr(net,{'ctrl_type':1})
```

output:



```
pg.showNetByNodeAttr(net,{'ctrl_type':(0,1)}) #for signalized and unsignalized intersections
```

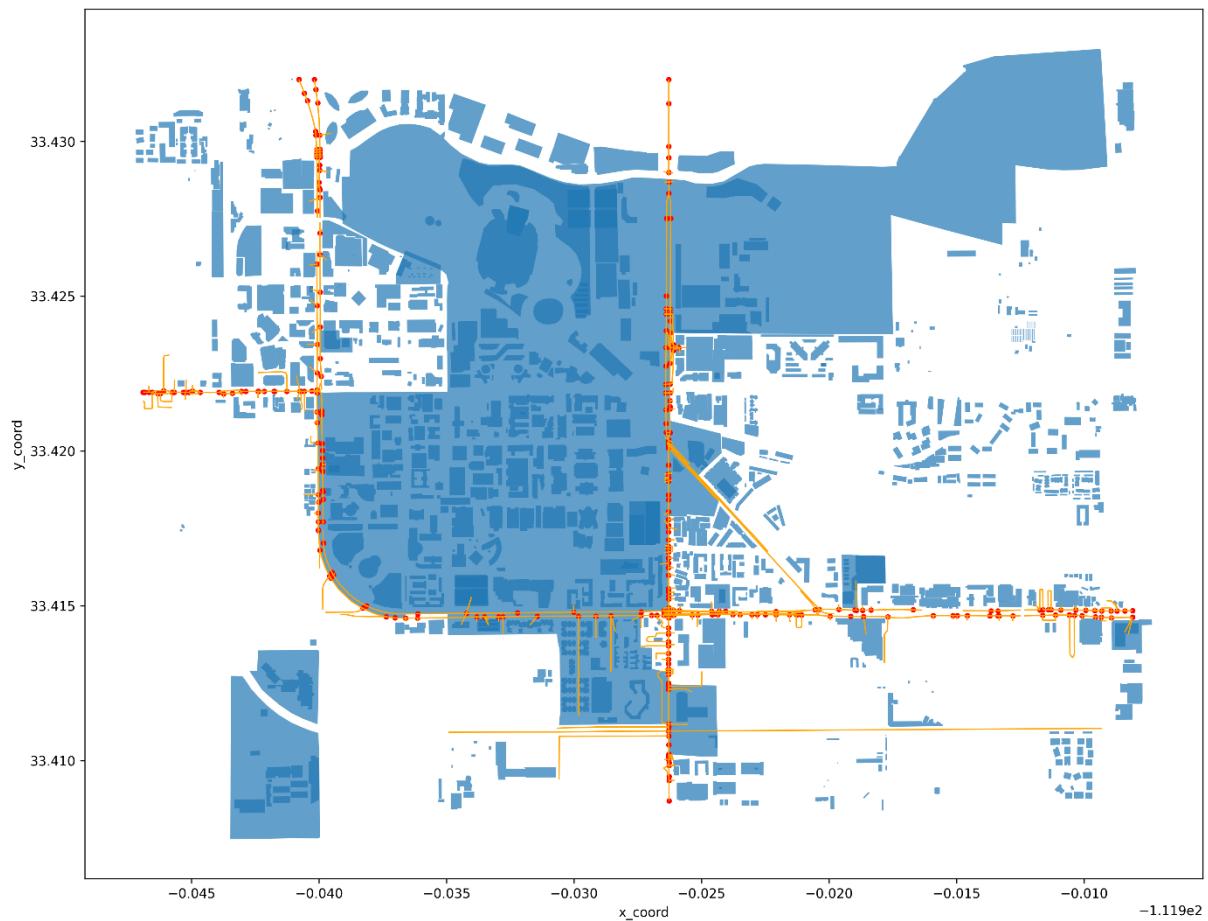
output:



(2) show network by given node activity types

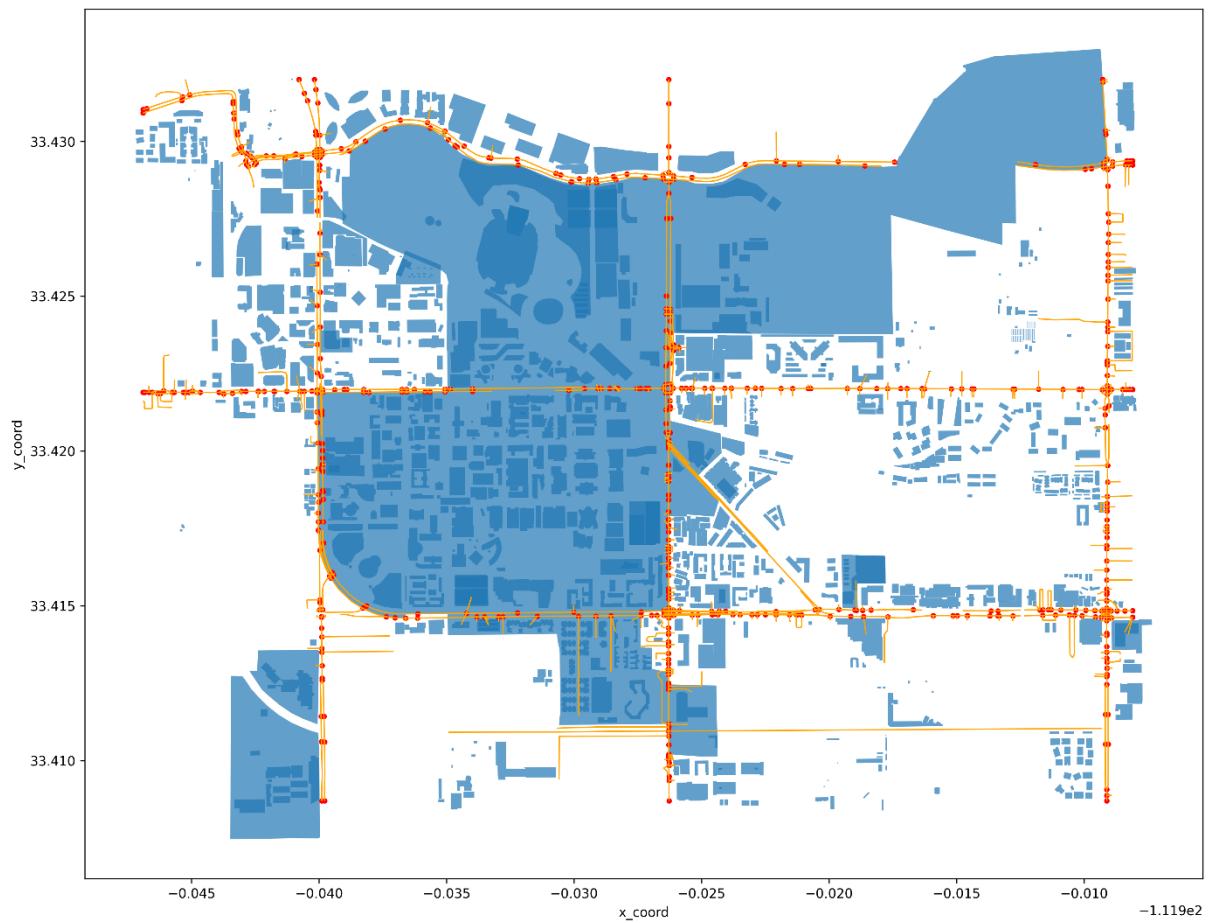
```
pg.showNetByNodeAttr(net,{'activity_type':'primary'})
```

output:



```
pg.showNetByNodeAttr(net,{'activity_type':['primary', 'secondary']})
```

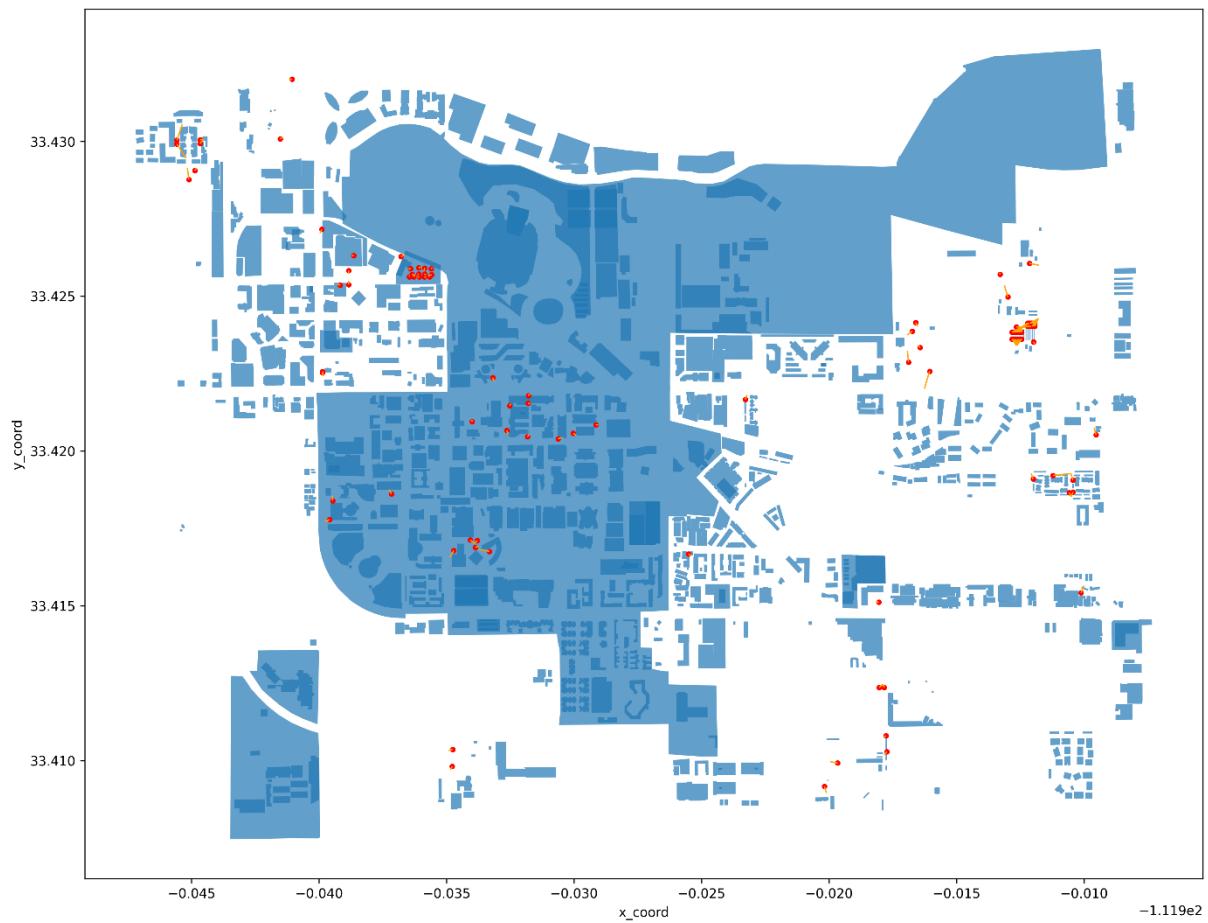
output:



(3) show network by a given range of node attraction volume

```
pg.showNetByNodeAttr(net,{'attraction':(0.001,0.1)})
```

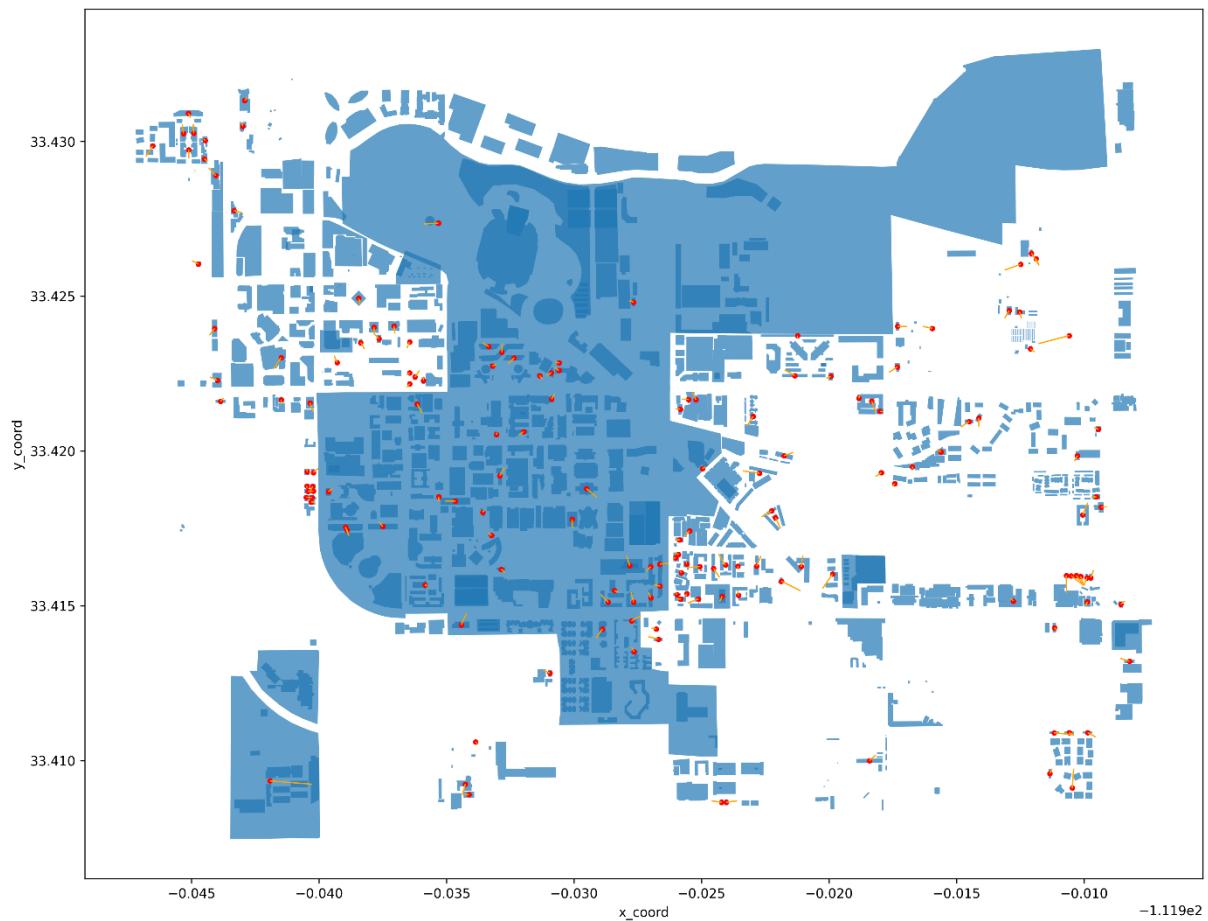
output:



(4) show network by a given range of node production volume

```
pg.showNetByNodeAttr(net,{'production':(1,5)})
```

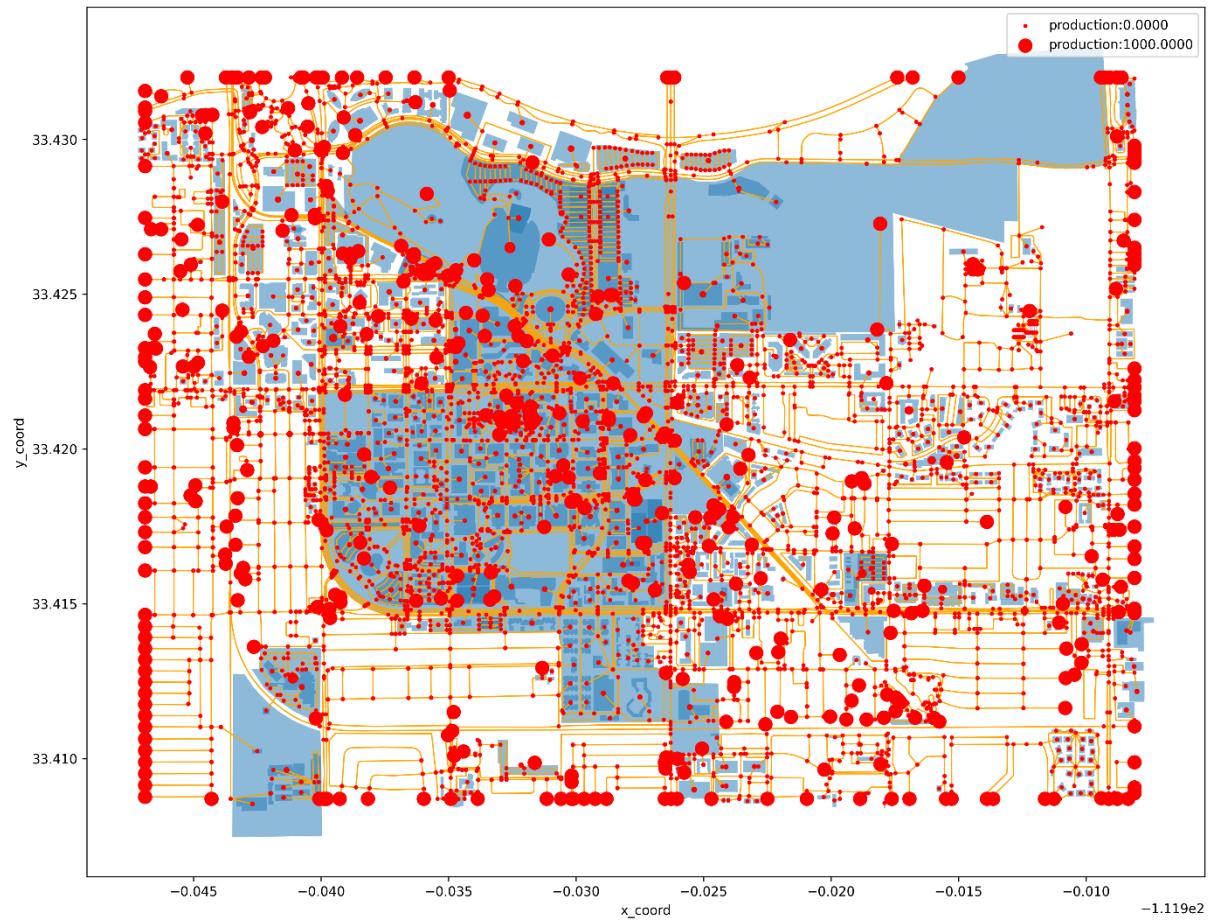
output:



(5) show network with the size of node circle is proportional to the production value

```
pg.showNetByNodeProduction(net)
```

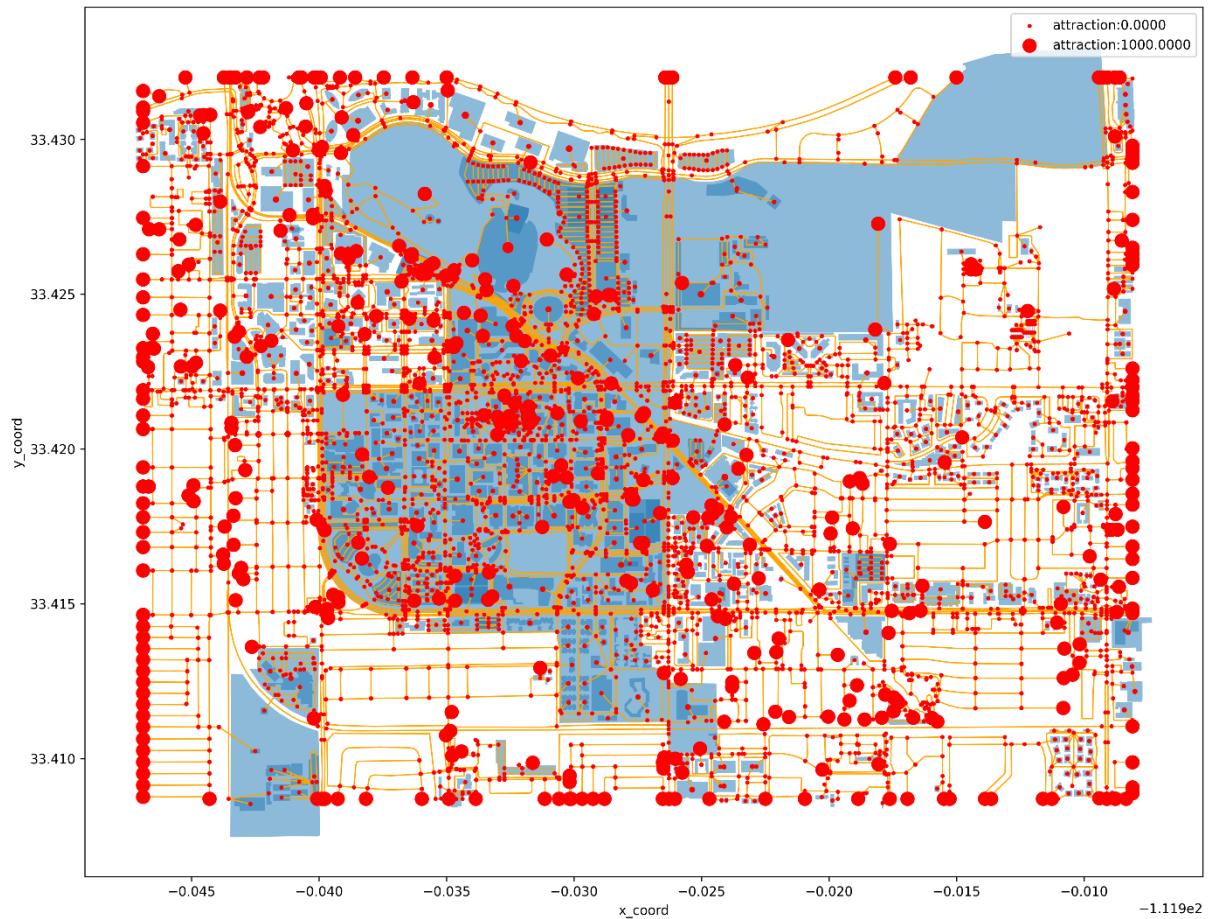
output:



(6) show network with the size of node circles is proportional to node attraction value

```
pg.showNetByNodeAttraction(net)
```

output:



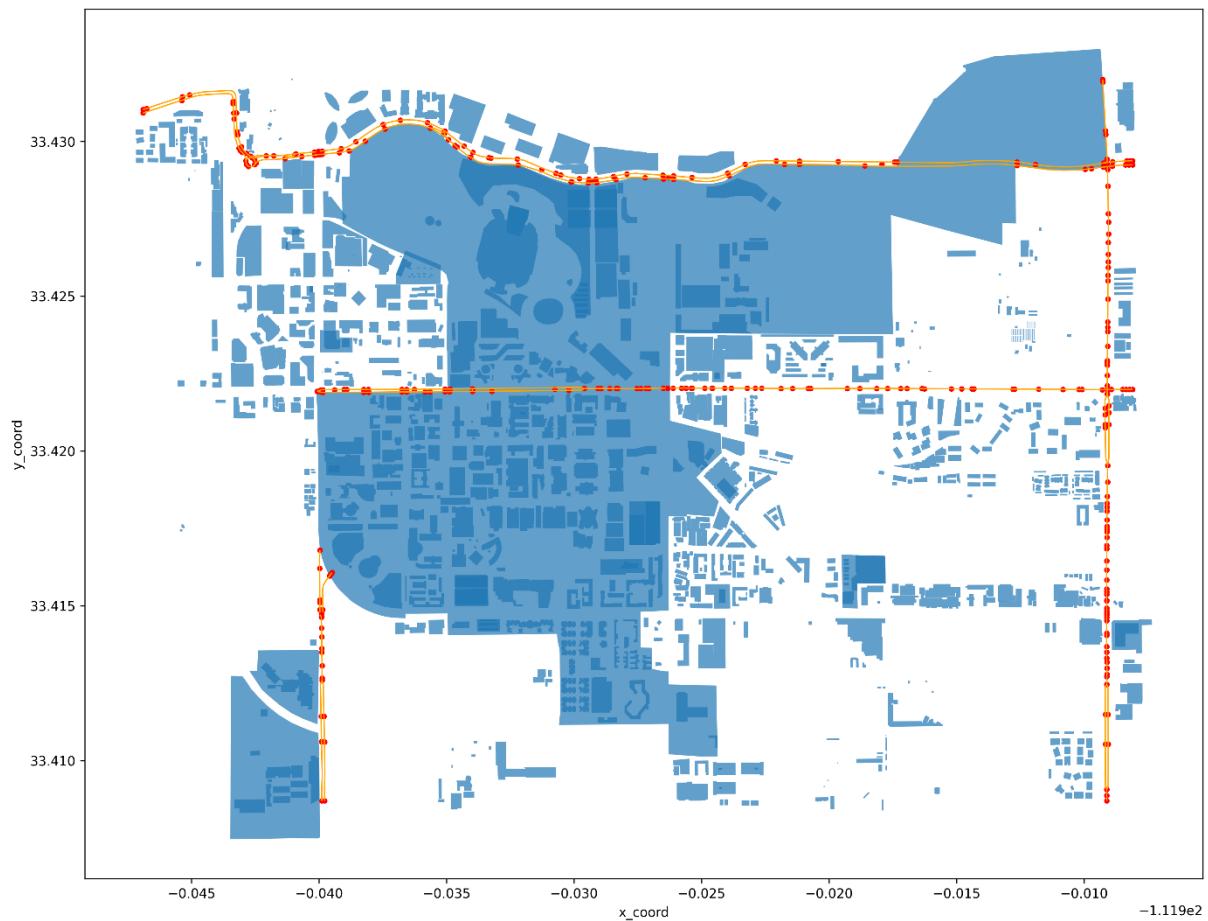
### 3.5 Visualizing network by link attributes

Only available link attributes and values can be displayed. Thus, the first step is to check if the link attribute fields and values exist.

- net.get\_valid\_link\_attr\_list()
  - pg.get\_link\_attr\_value\_list(net,attr)
- (1) show network by given link type names

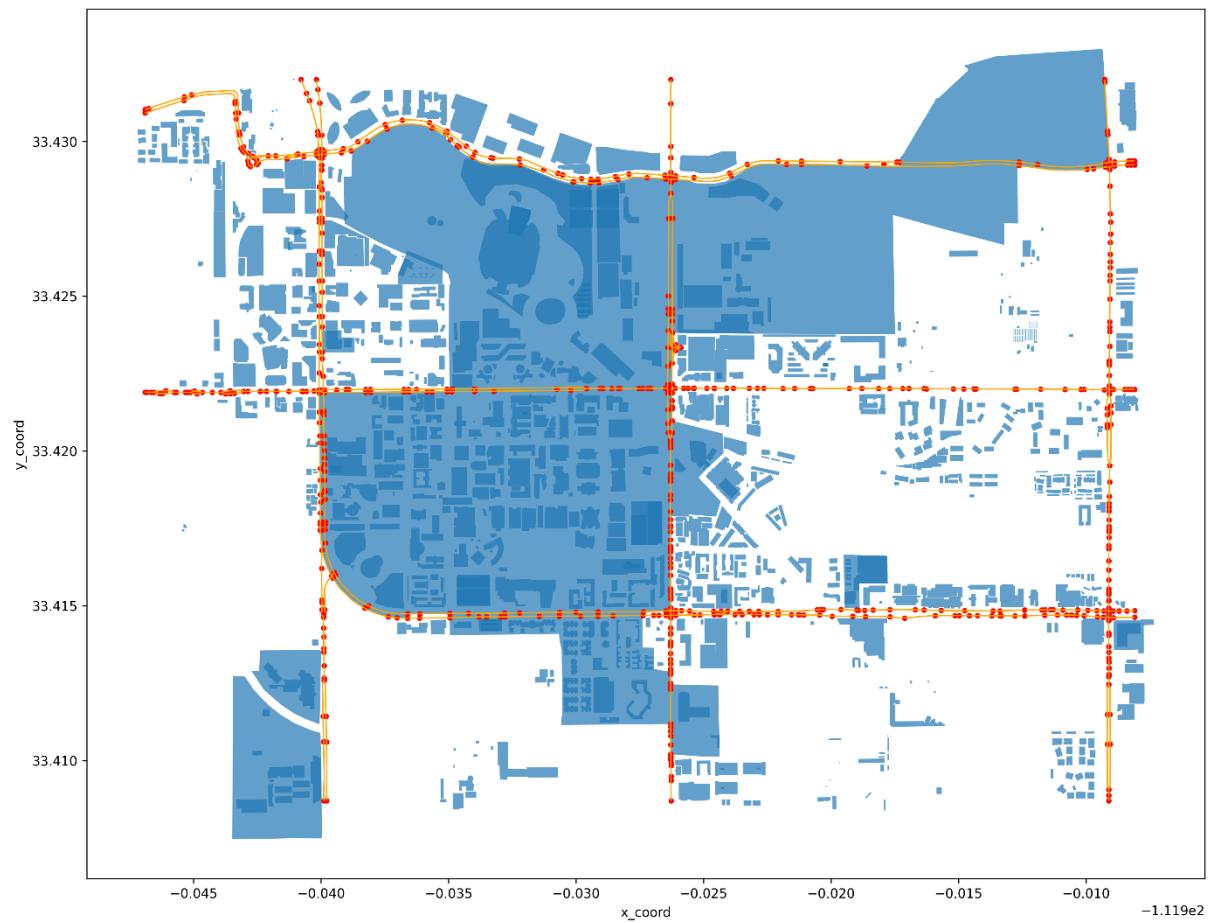
```
pg.showNetByLinkAttr(net,{'link_type_name':'secondary'})
```

output:



```
pg.showNetByLinkAttr(net,{'link_type_name':['secondary','primary']})
```

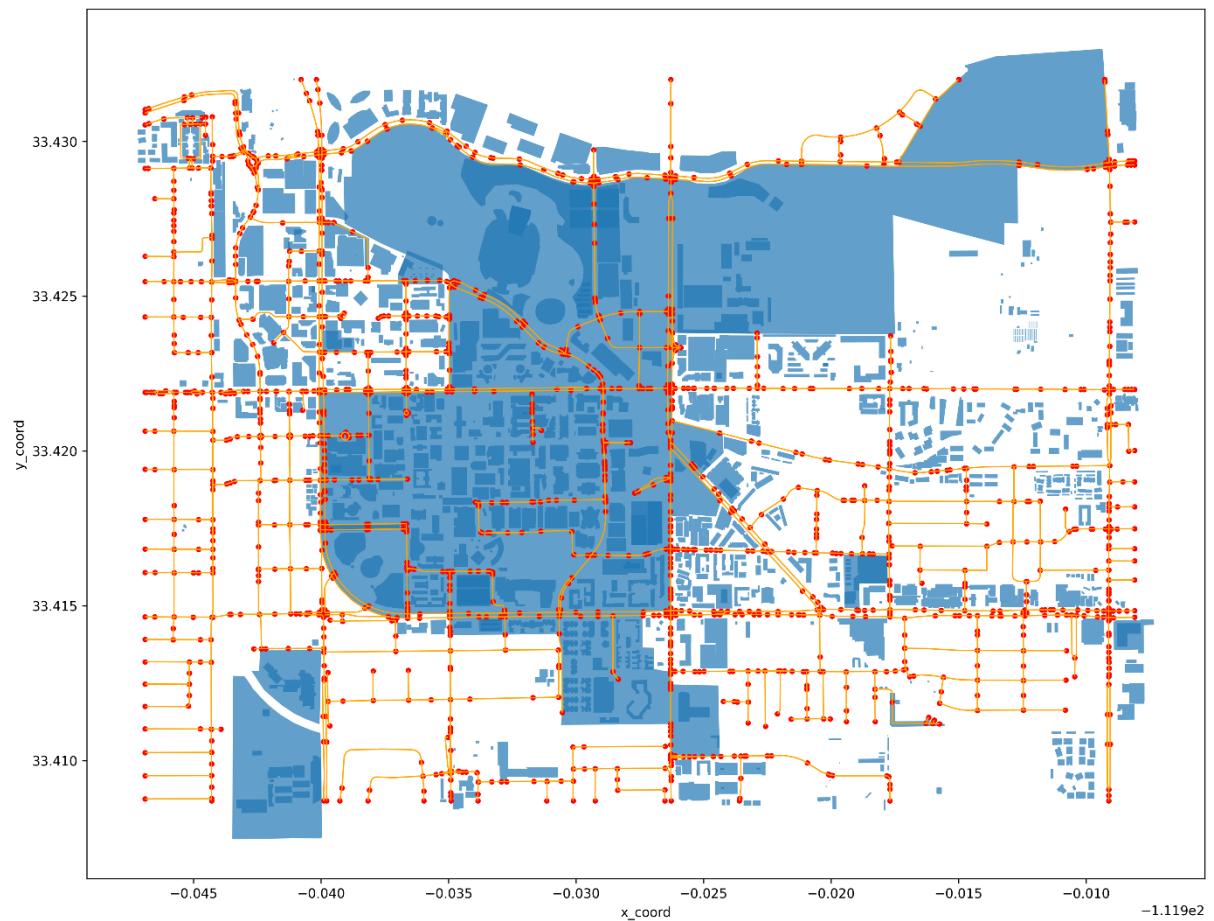
output:



(2) show network by allowed transportation mode on the link

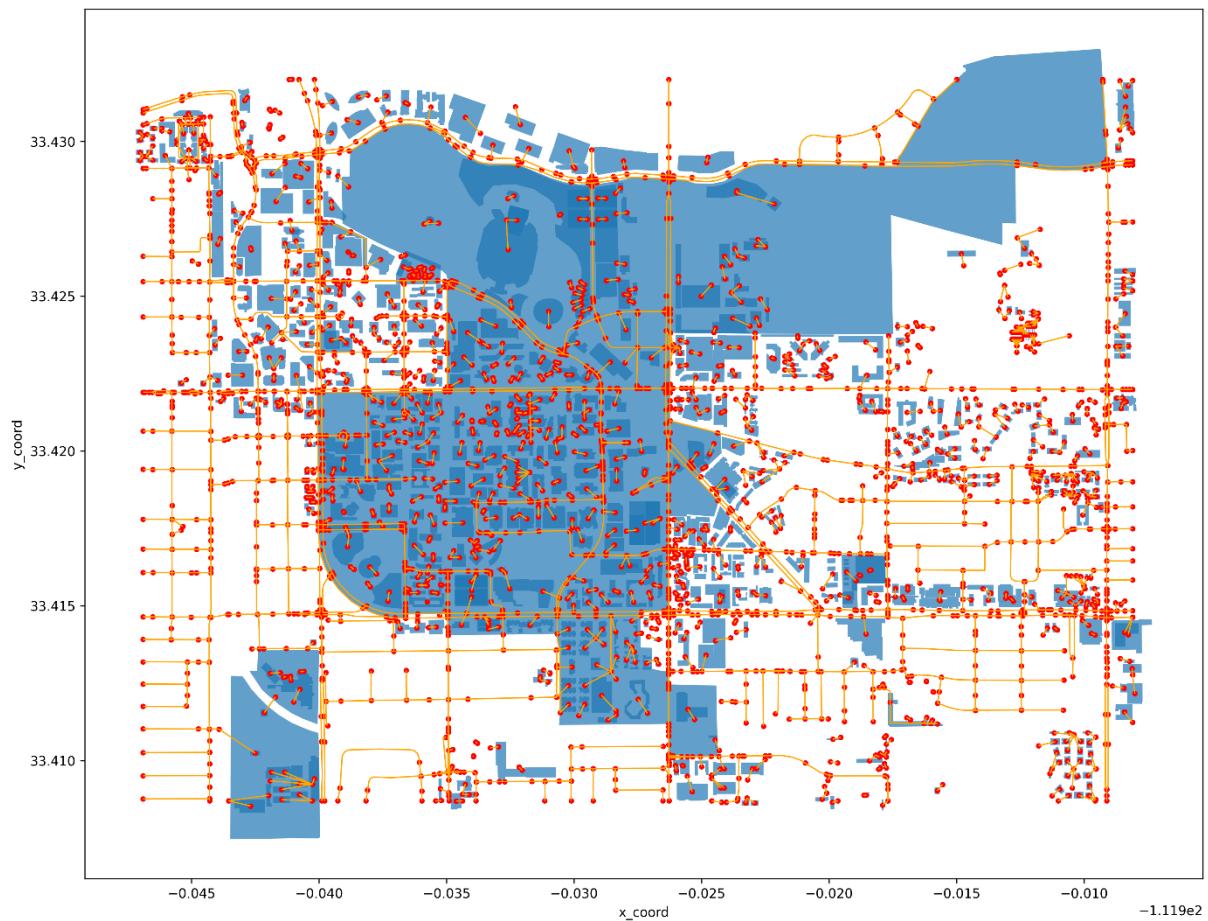
```
pg.showNetByLinkAttr(net,{'allowed_uses':'auto'})
```

output:



```
pg.showNetByLinkAttr(net,{'allowed_uses':['all','auto']})
```

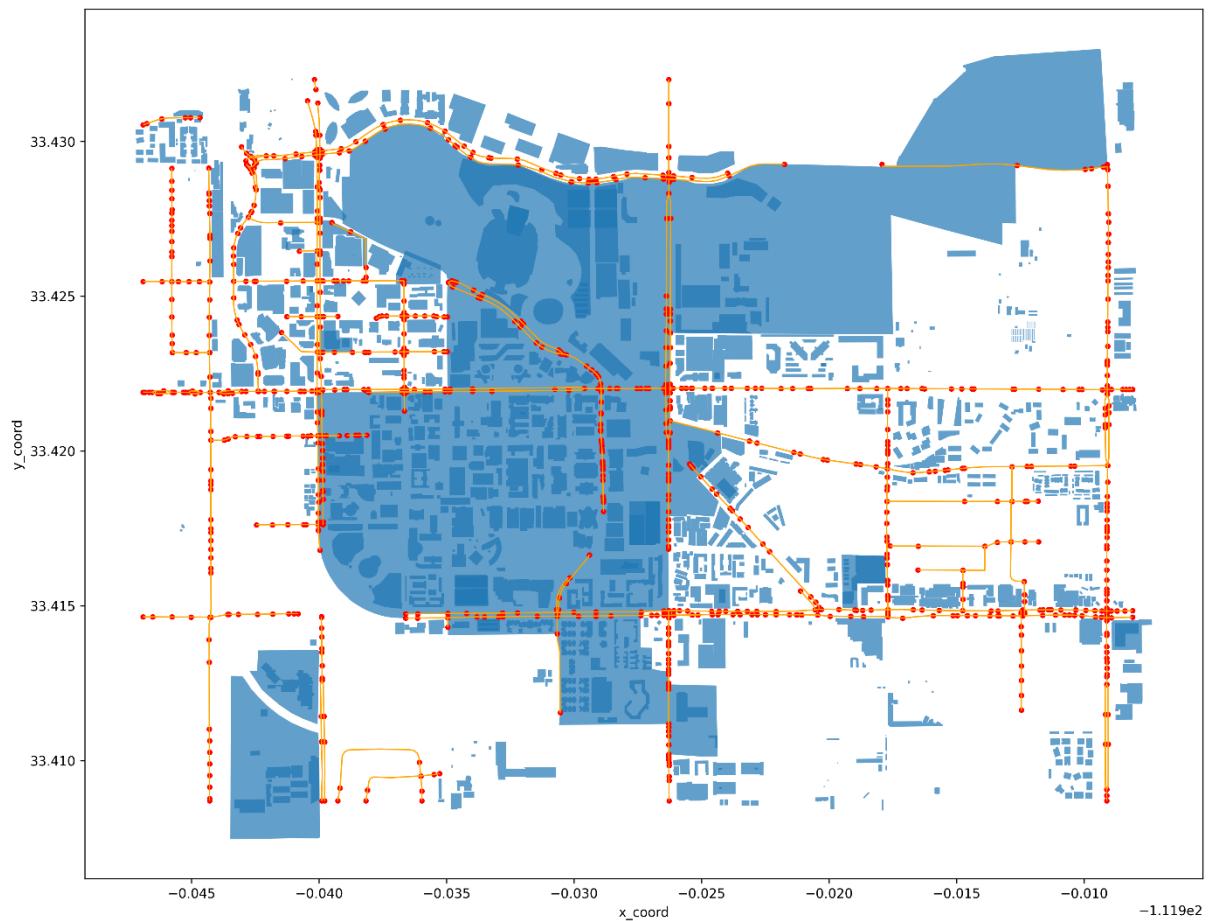
output:



(3) show network by a given range of link free speed

```
pg.showNetByLinkAttr(net,{'free_speed':(20,40)})
```

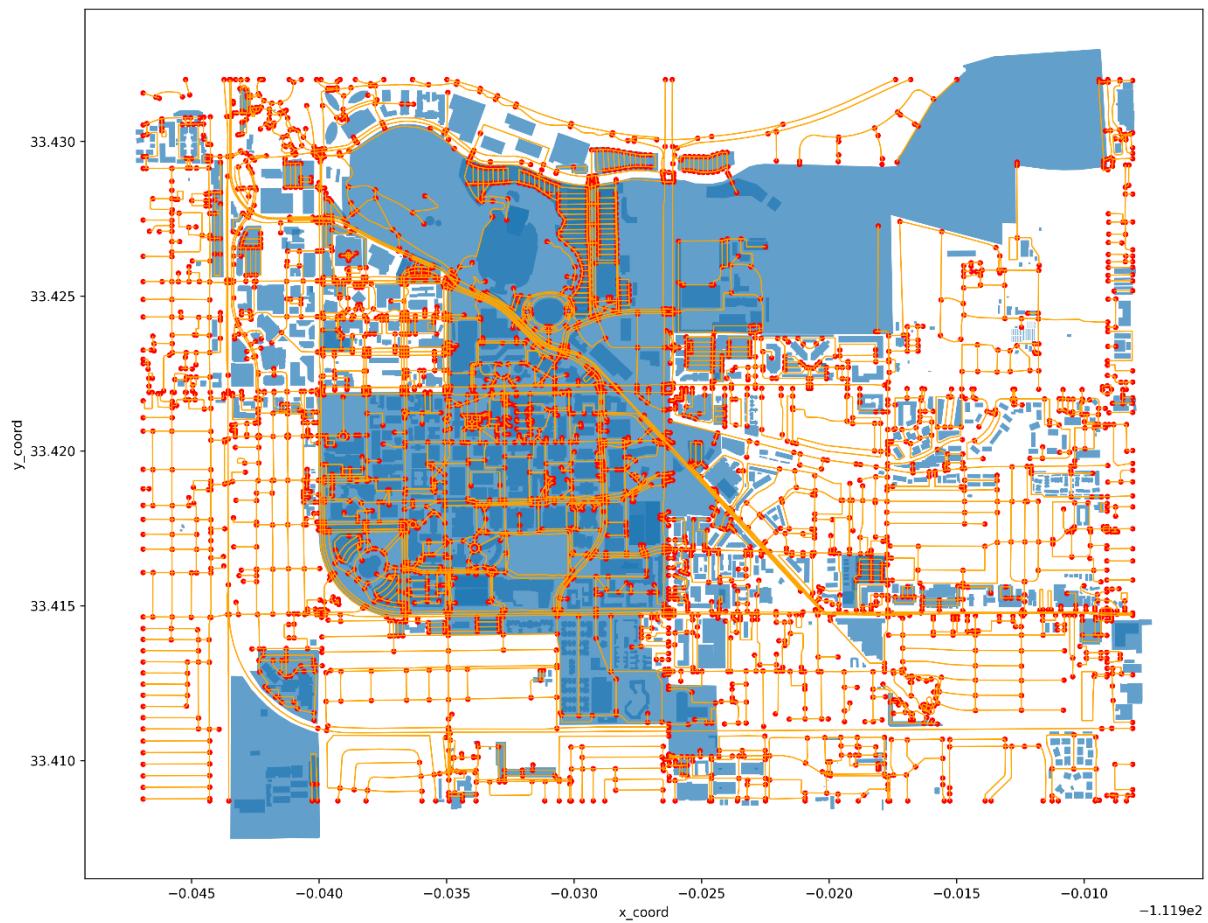
output:



(4) show network by a given range of link capacity

```
pg.showNetByLinkAttr(net,{'capacity':(0,2000)})
```

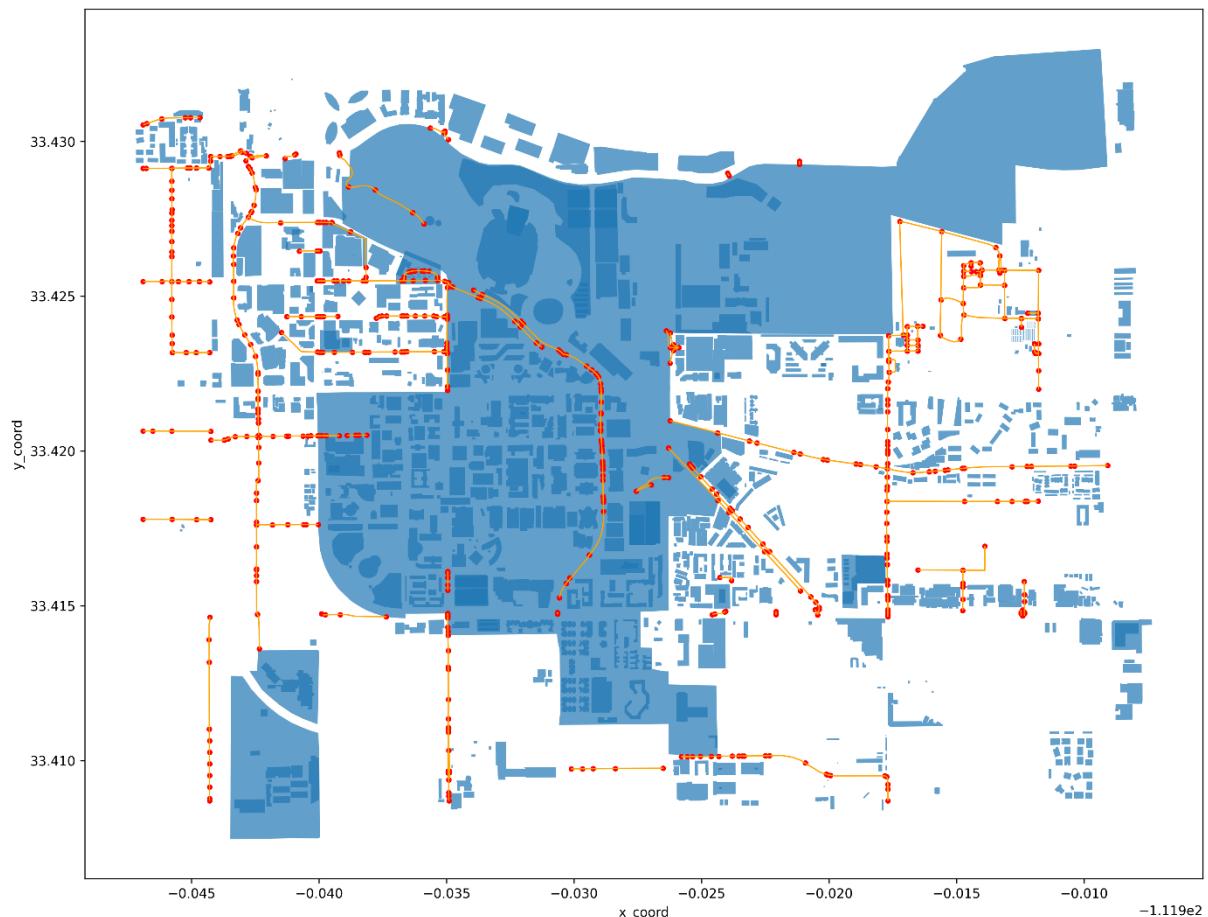
output:



(5) show network by the categories in terms of the number of lanes

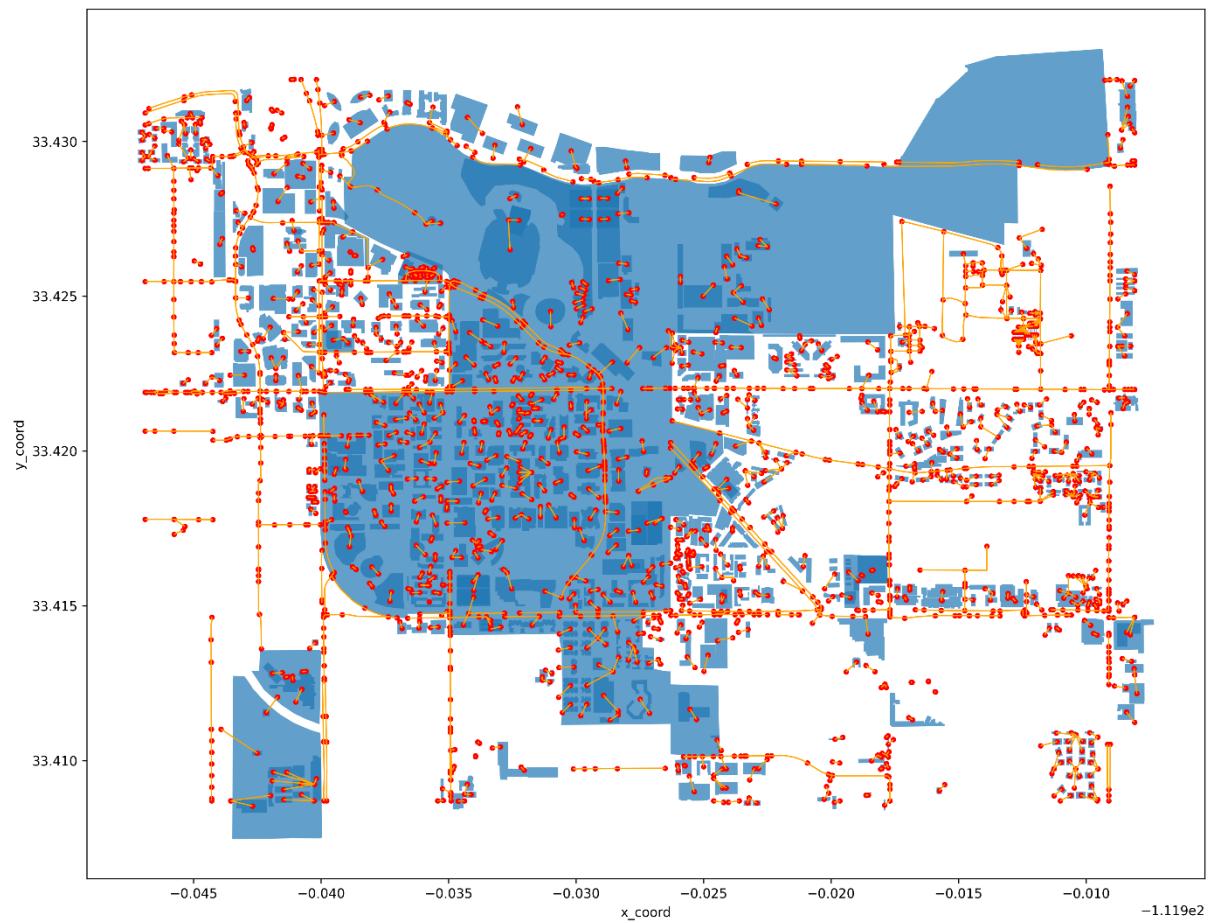
```
pg.showNetByLinkAttr(net,{'lanes':(1,1)})
```

output:



```
pg.showNetByLinkAttr(net,{'lanes':(1,2)})
```

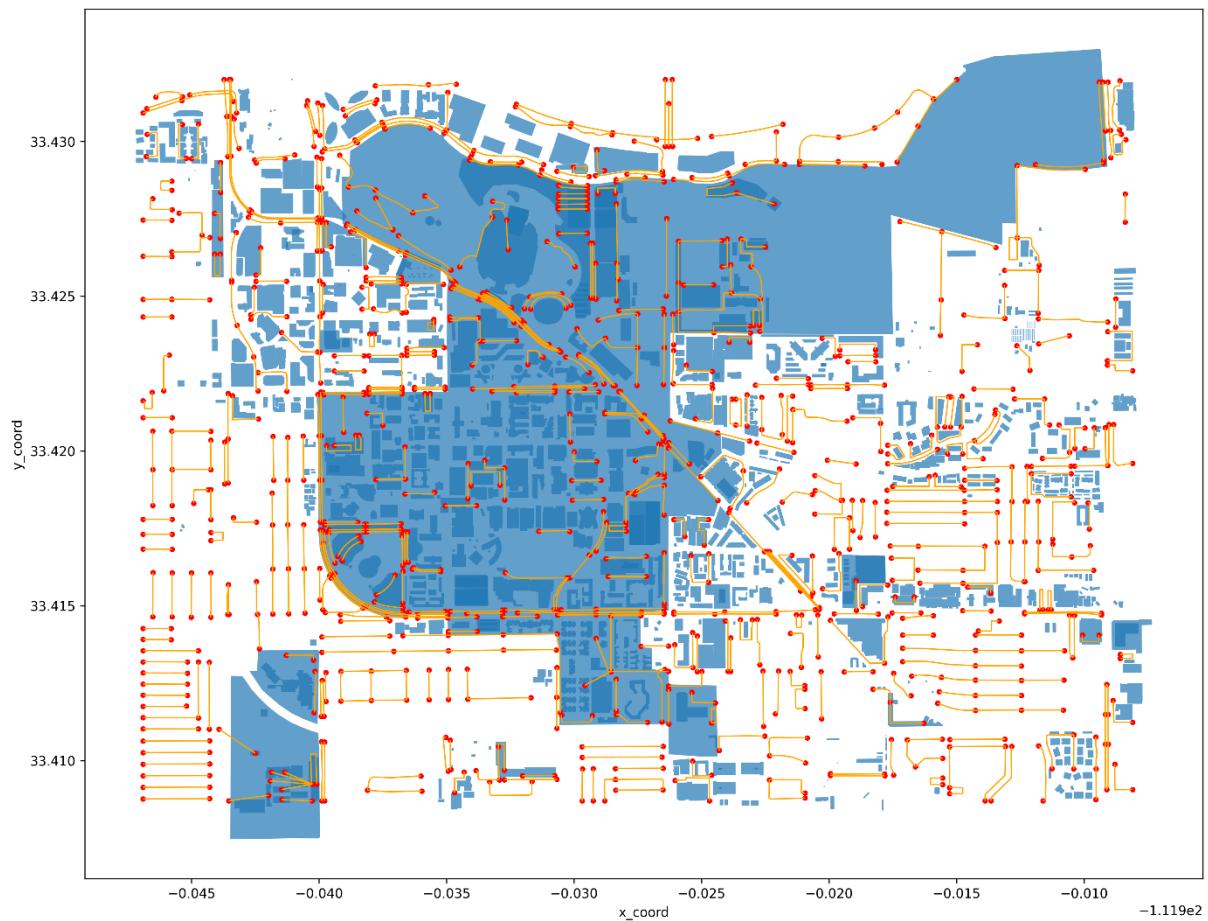
output:



(6) show network by the range of link length

```
pg.showNetByLinkAttr(net,{'length':(100,300)})
```

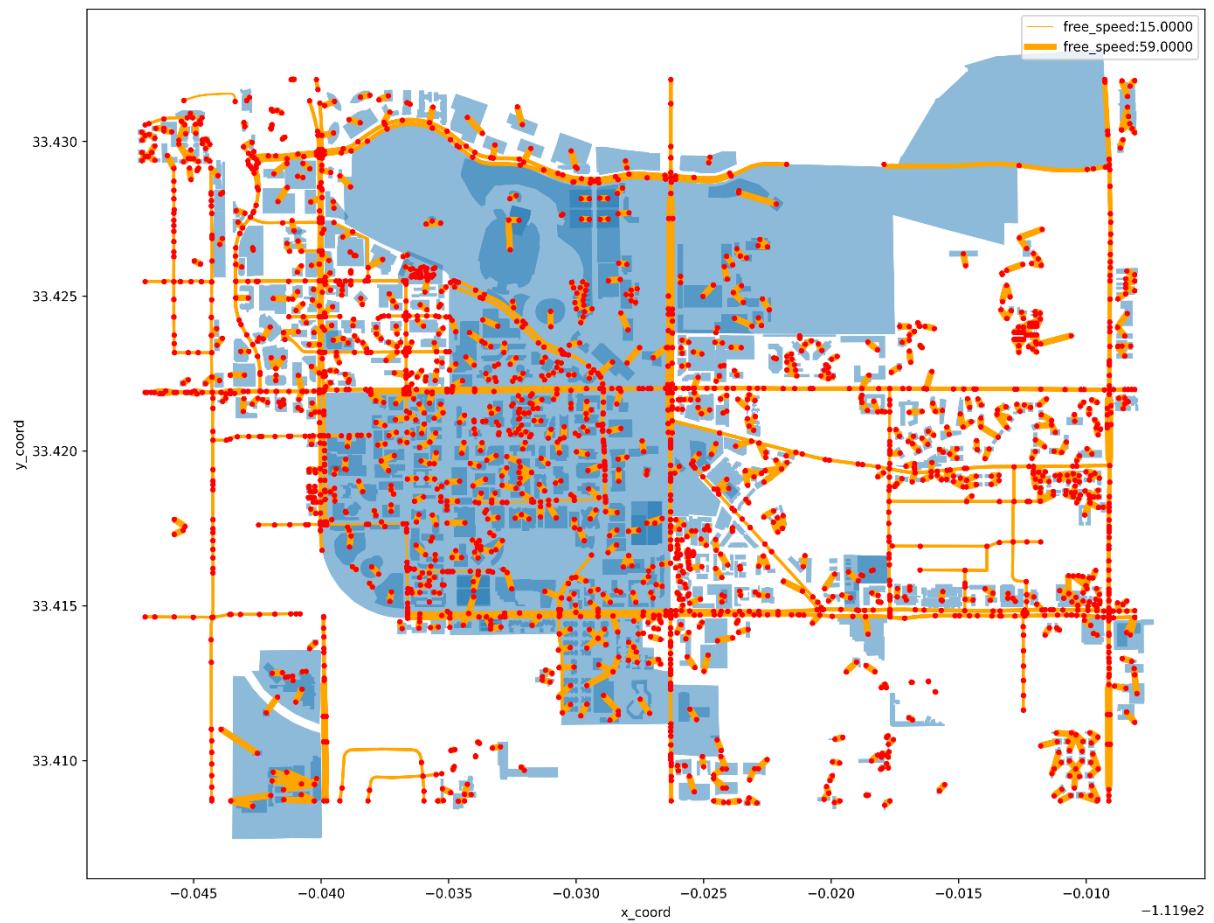
output:



(7) show network with the width of links is proportional to the free speed value

```
pg.showNetByLinkFreeSpeed(net)
```

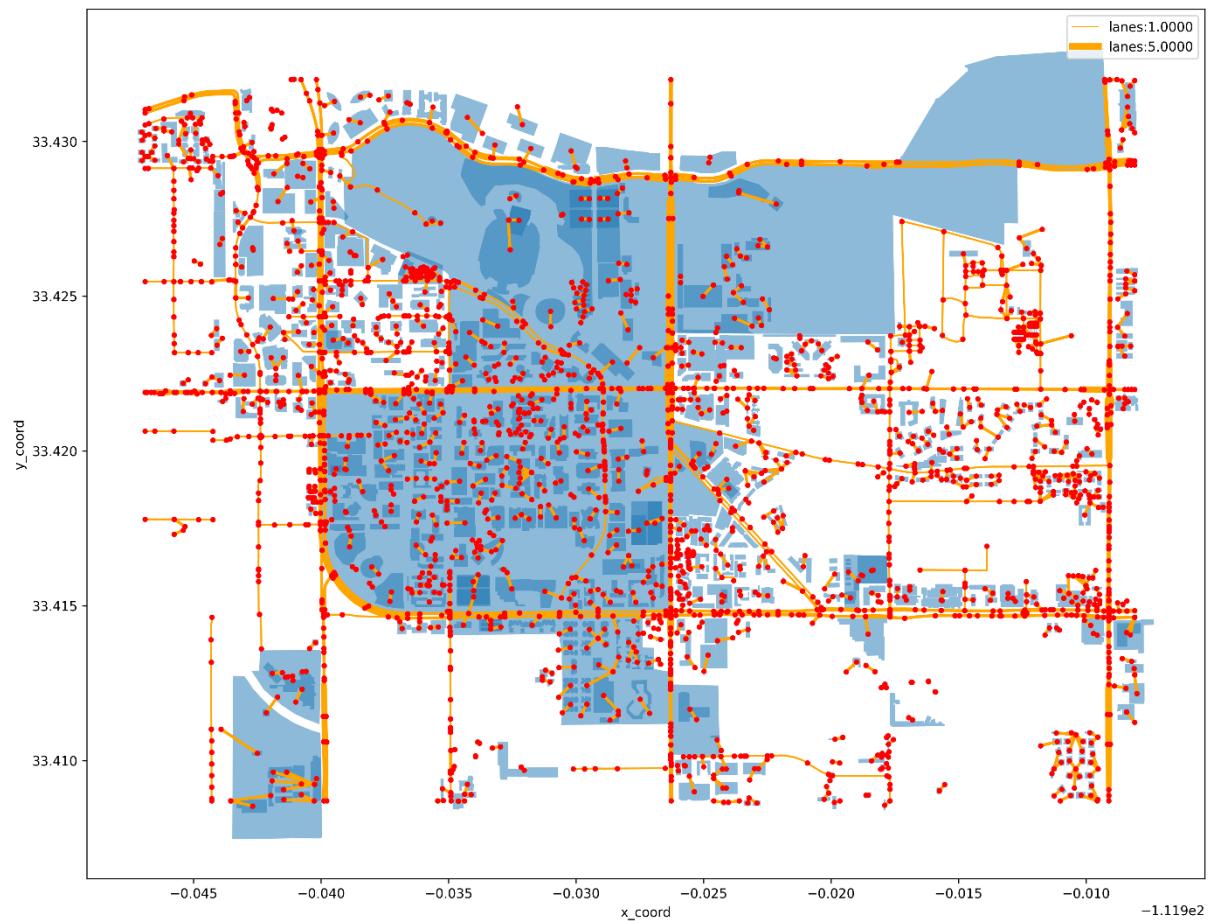
output:



(8) show network with the width of links is proportional to the number of lanes

```
pg.showNetByLinkLaneNum(net)
```

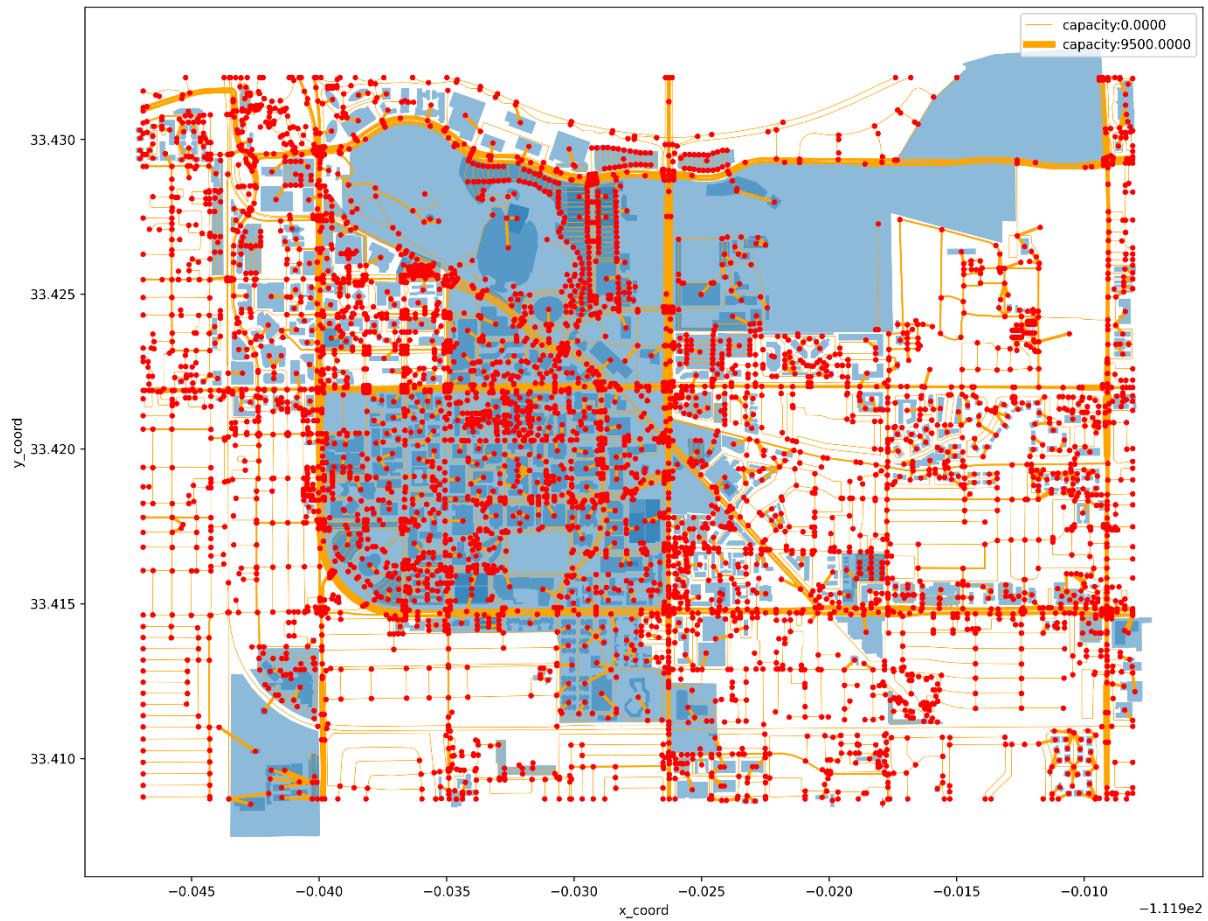
output:



(9) show network with the width of links is proportional to link capacity (vehicles/hour)

```
pg.showNetByLinkCapacity(net)
```

output:



### 3.6 Visualizing network by POI attributes

Only available POI attributes and values can be displayed. Thus, the first step is to check if the POI attribute fields and values exist.

- net.get\_valid\_poi\_attr\_list()
  - pg.get\_poi\_attr\_value\_list(net,attr)
- (1) show network by given POI building types

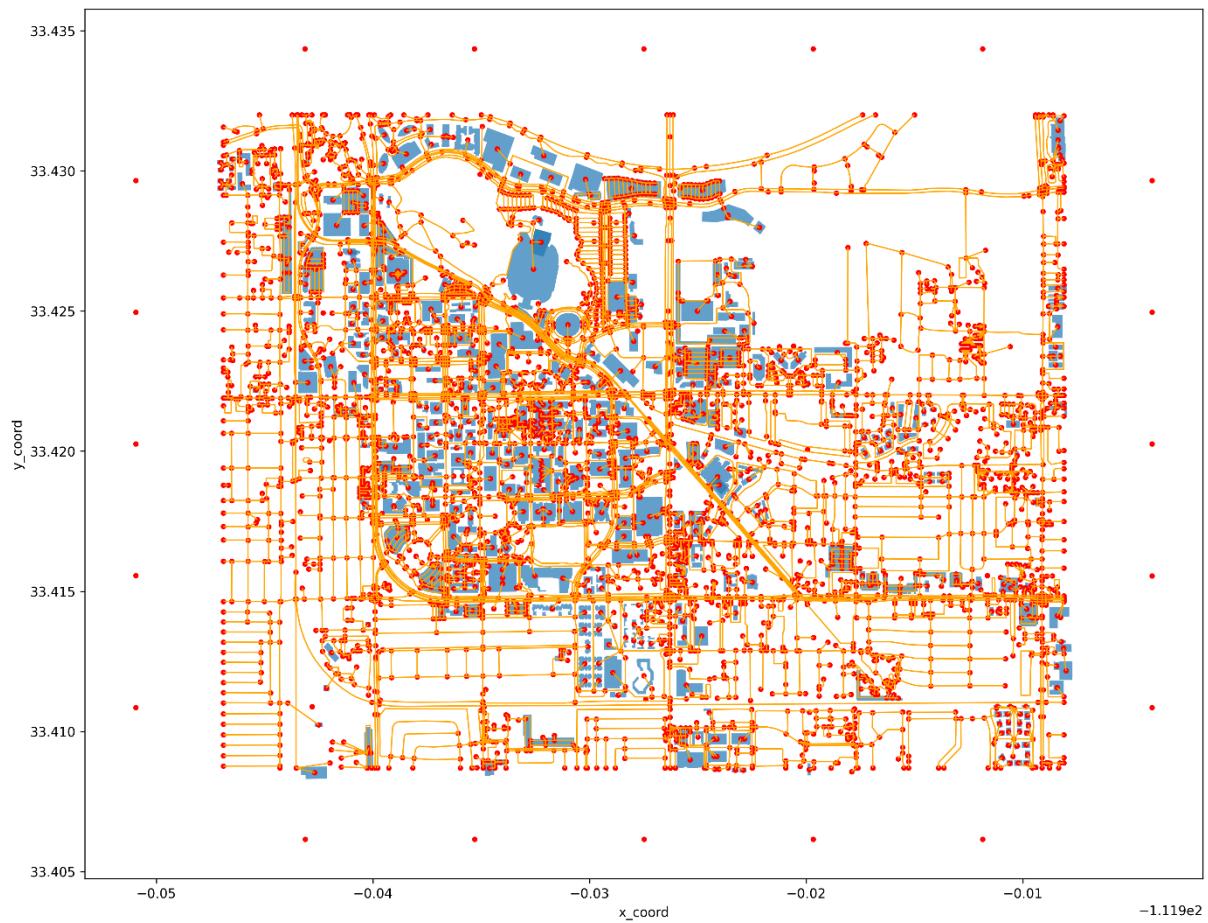
```
pg.showNetByPOIArr(net,{'building':'parking'})
```

output:



```
pg.showNetByPOIAttr(net,{'building':['parking', 'yes']})
```

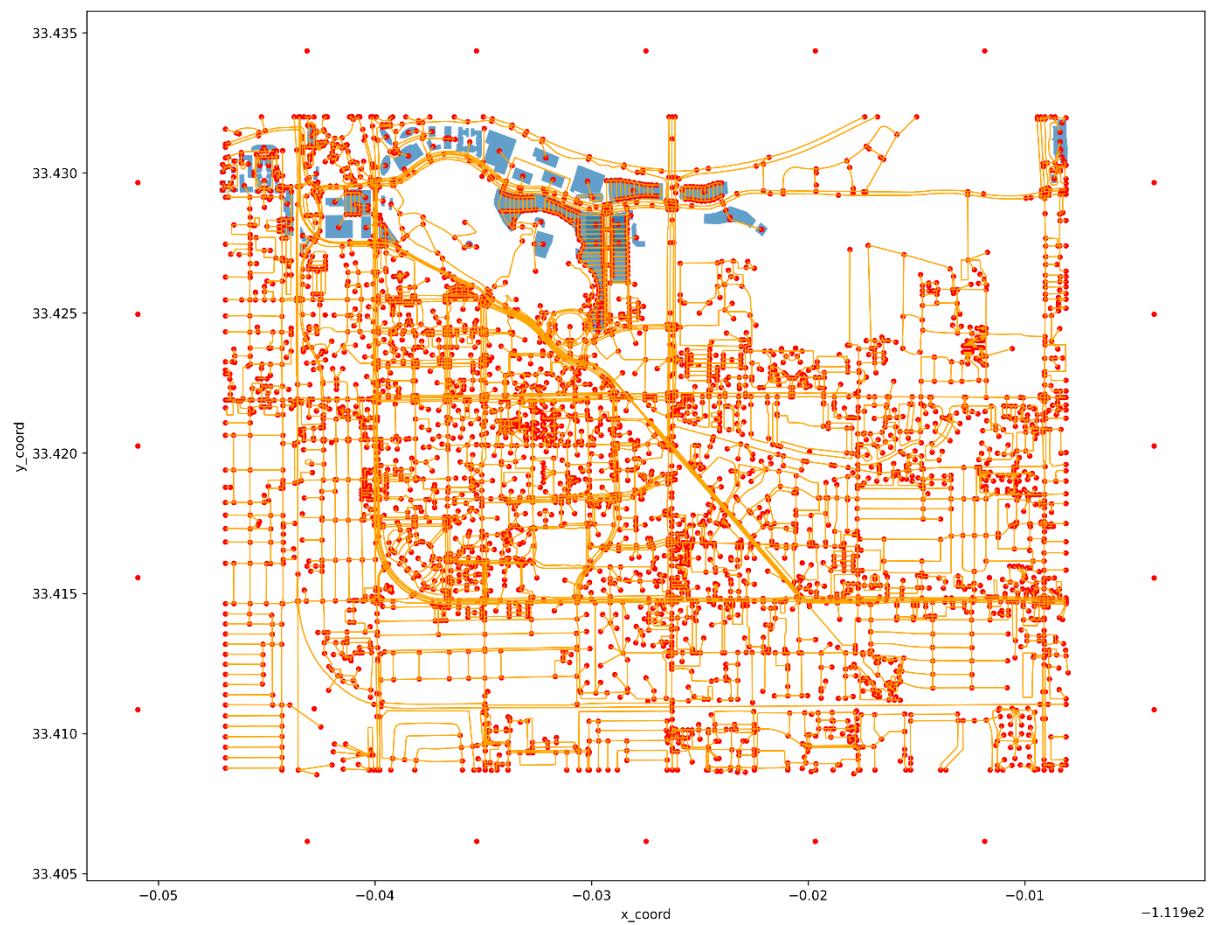
output:



(2) show network only for a given range of POI activity zone ids

```
pg.showNetByPOIArr(net,{'activity_zone_id':(1,5)})
```

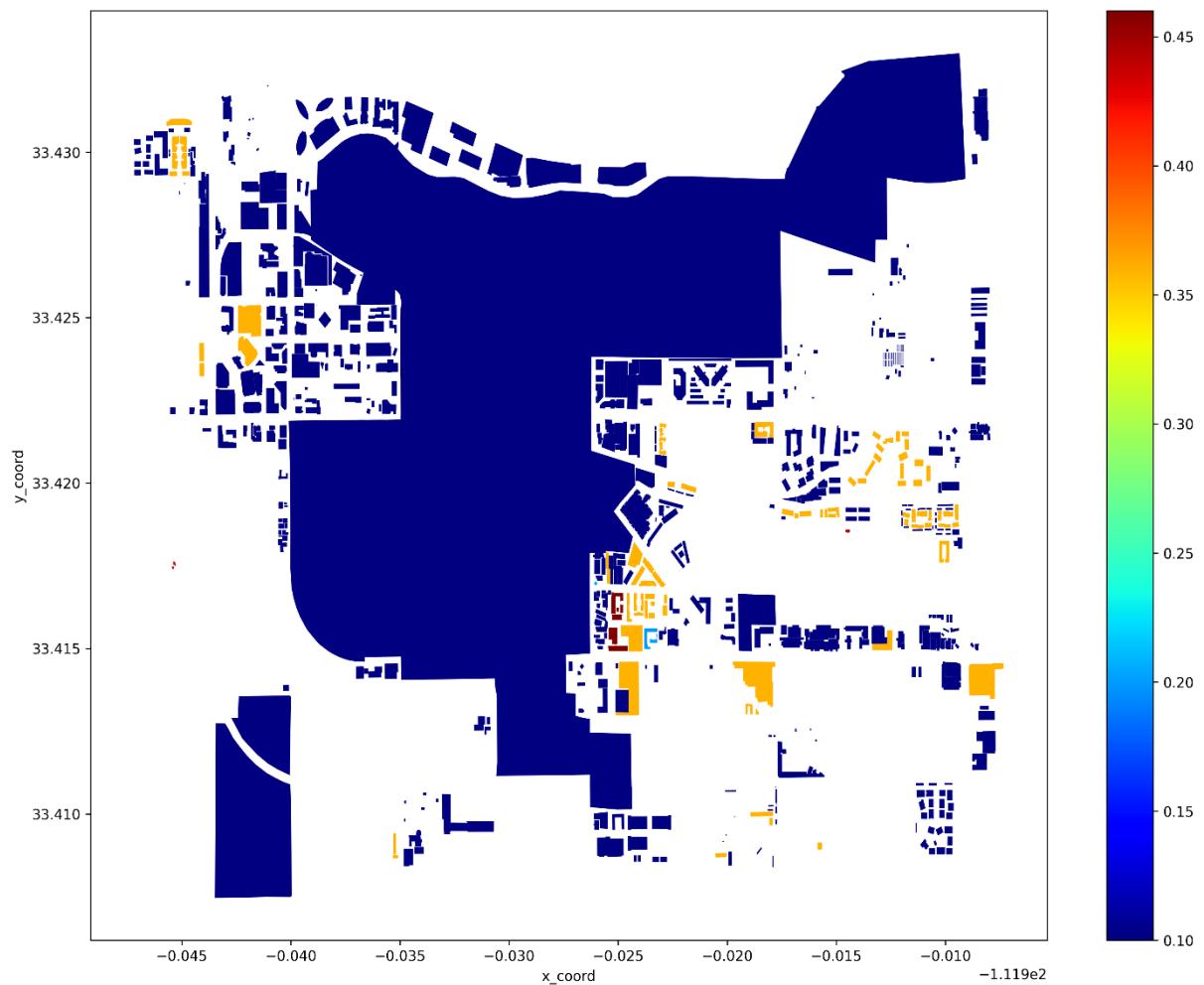
output:



(3) show heatmap in terms of POI attraction density

```
pg.showNetByZonePOIAtrractionDensity(net)
```

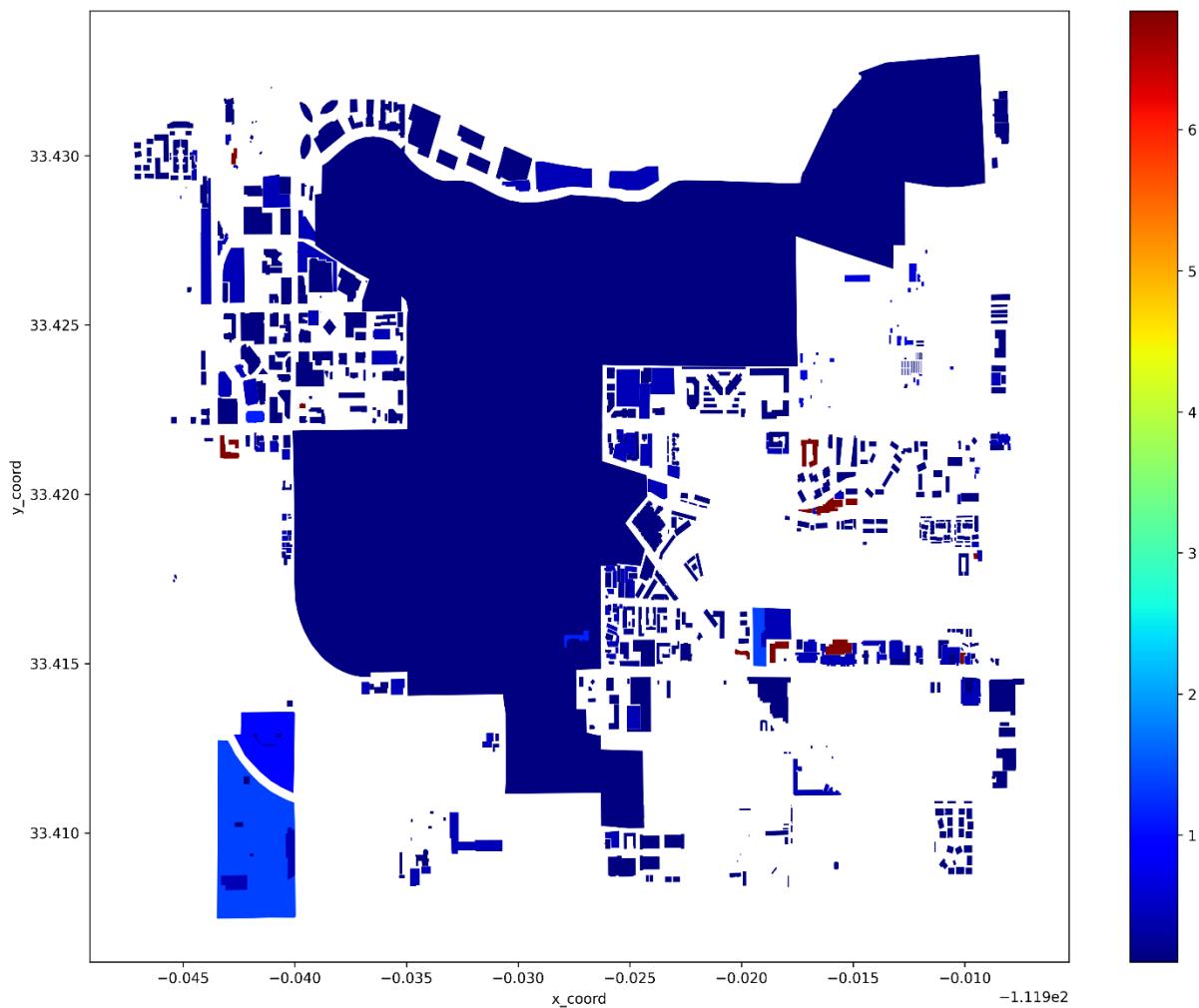
output:



(4) show heatmap in terms of POI production density

```
pg.showNetByZonePOIProductionDensity(net)
```

output:



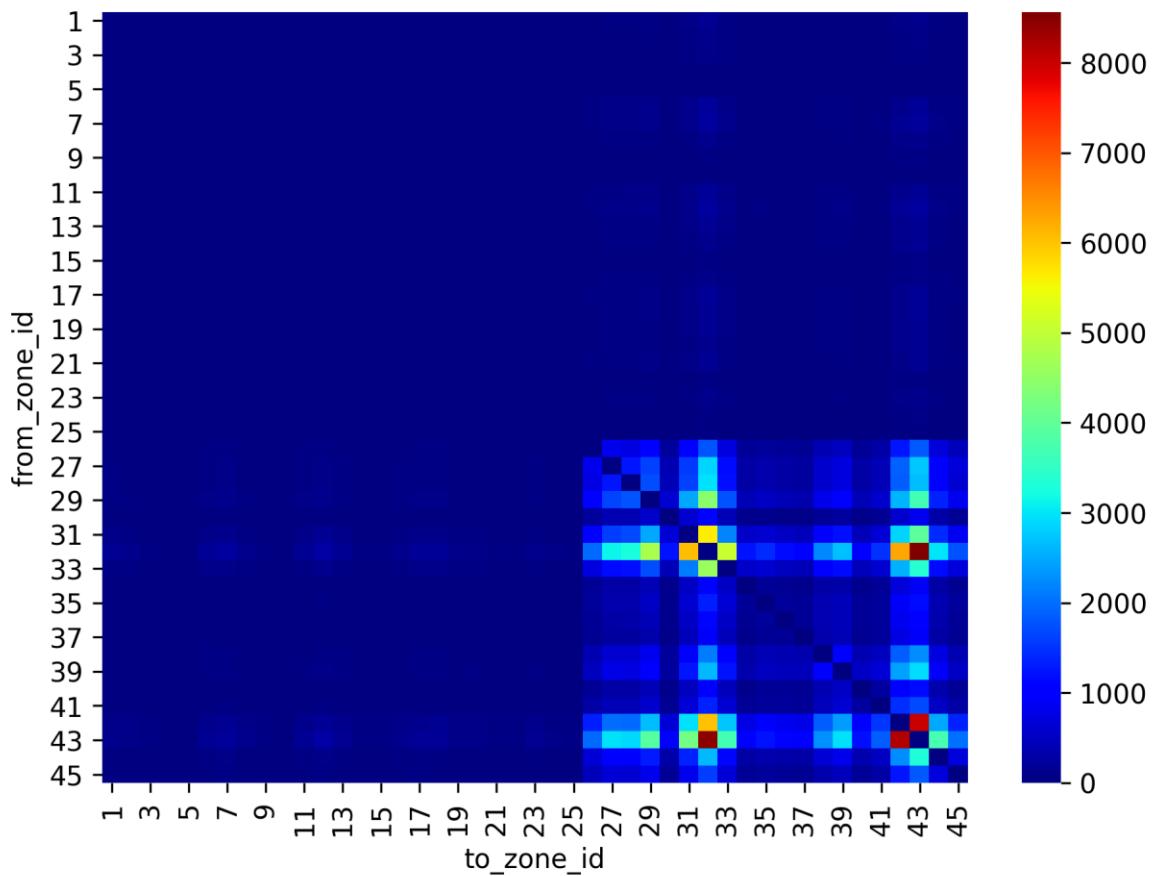
### 3.7 Visualizing network by zone attributes

Only available zone attributes and values can be displayed. Thus, the first step is to check if the zone attribute fields and values exist.

- `net.get_valid_zone_id_list()`
  - `pg.get_zone_id_list(net,zone_id)`
- (1) show the heatmap origin demand volume by zone-to-zone demand

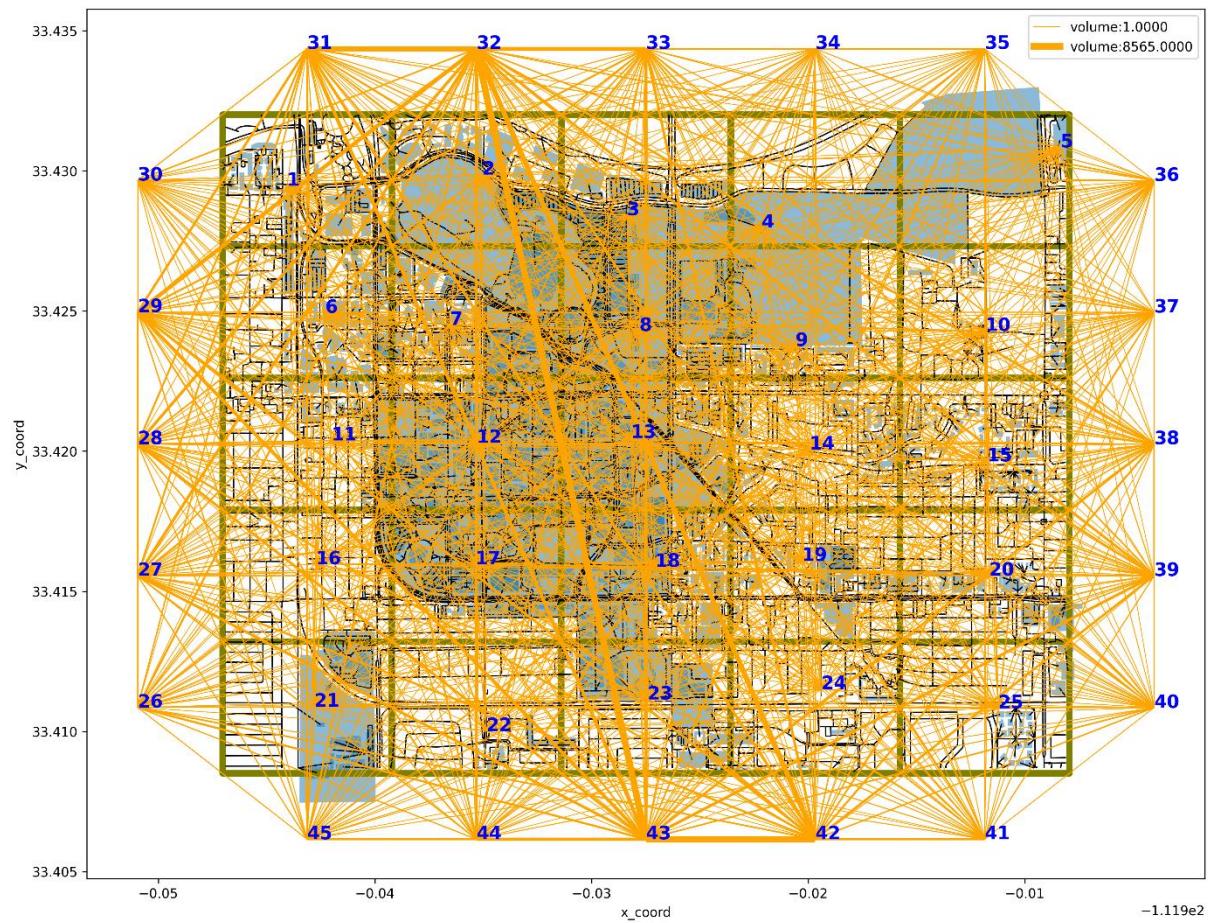
```
pg.showNetByZoneDemandHeat(net,annot=False)
# annot:bool,whether or not show zone-to-zone demand value
```

**output:**



(2) show network by zone demand traces

```
pg.showNetByZoneDemandFlow(net)
```

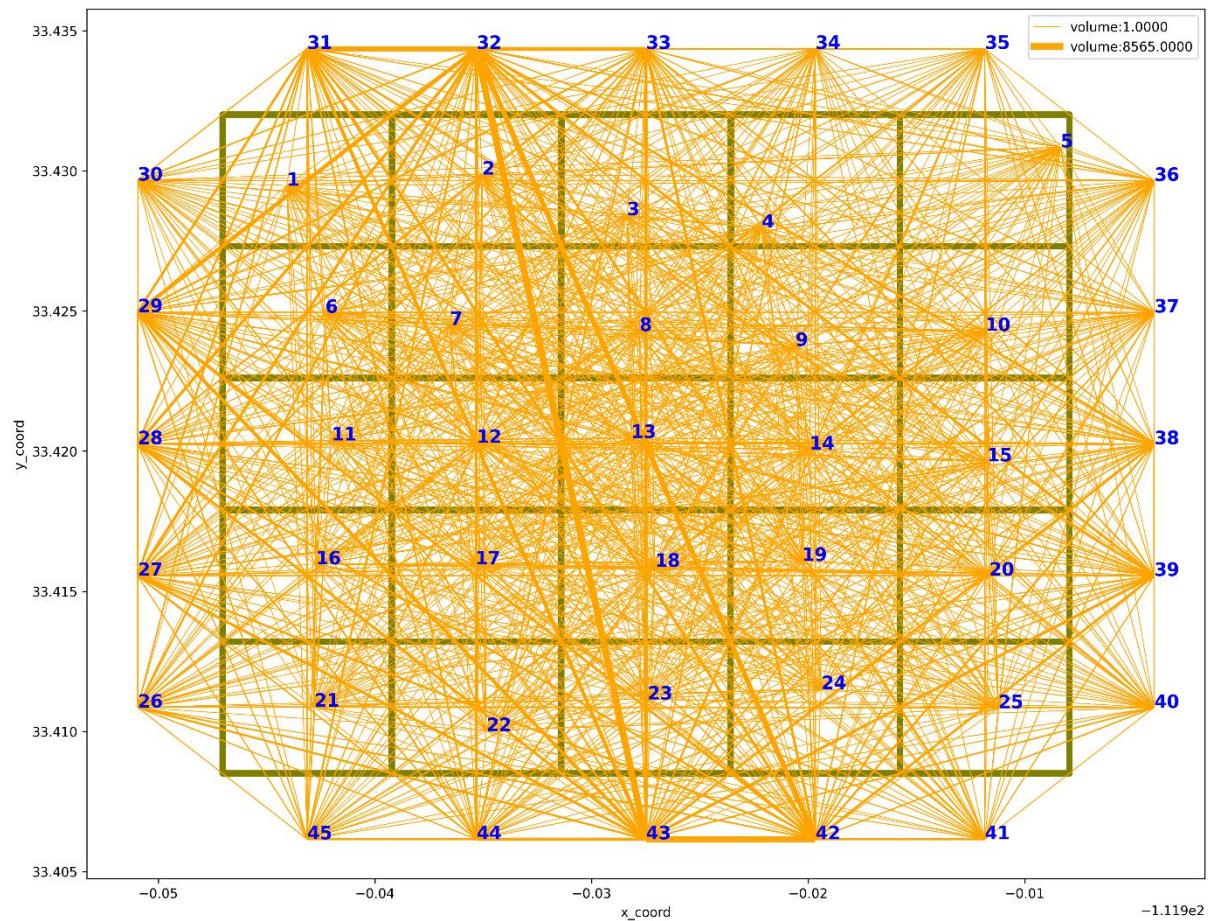


```
pg.showNetByZoneDemandFlow(net,annot=True,bg=False)
```

#annot: bool, whether or not show zone id

#bg: bool, whether or not show network (node, link, and poi)

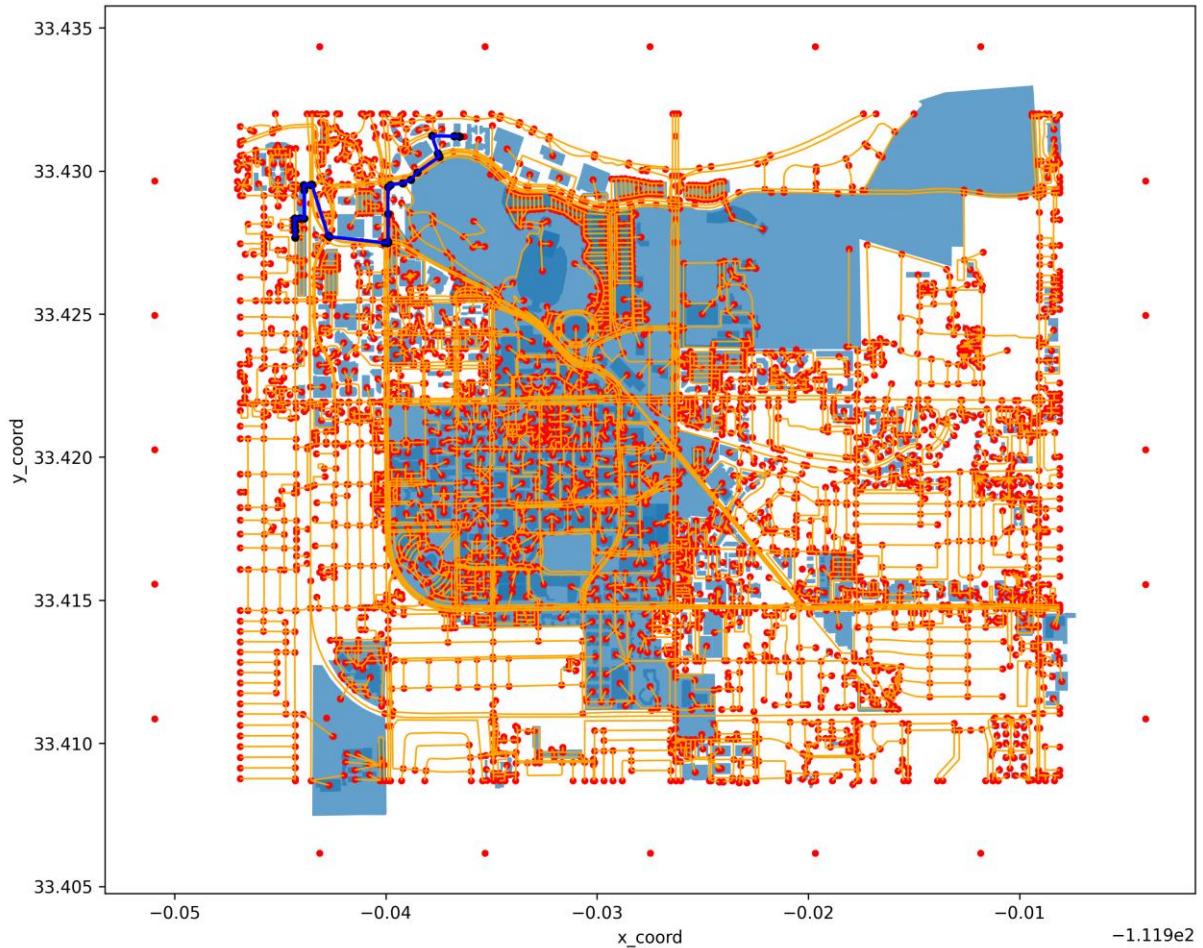
output:



### 3.8 Visualizing network by agent traces

Only available agent's trace can be displayed. Thus, the first step is to check if the agent id and trace exist in the input\_agent.csv.

```
pg.showNetByAgentTrace(net,agent_id=1)
```



### 3.9 Saving figure to a file

You can specify a filename and dpi value in the dict field as below to save the figure when showing network.

```
pg.showNetByXXX(net,savefig={'filename':'net.png','dpi':300})
```

### 3.10 Data availability check

(1) Check availability of node attributes

```
net.get_avl_nodeAttrs()
```

output:

---

attr	type
ctrl_type	int
activity_type	str
production	float
attraction	float

(2) Check availability of link attributes

```
net.get_avl_link_attrs()
```

output:

attr	type
length	float
lanes	int
free_speed	float
capacity	float
link_type_name	str
allowed_uses	str

(3) Check availability of poi attributes

```
net.get_avl_poiAttrs()
```

output:

attr	type
building	str
activity_zone_id	int

(4) Check range of zone ids

```
net.get_avl_range_of_zone_ids()
```

output:

min zone id	max zone id
1	45

---

(5) Count the number of nodes for a given field

```
pg.get_num_of_nodes_by_attr(net,'ctrl_type')
```

output:

ctrl_type	number
0	5144
1	84

```
pg.get_num_of_nodes_by_attr(net,'activity_type')
```

output:

activity_type	number
residential	672
primary	291
tertiary	233
secondary	263
service	1640
rail	17

(6) Check range of a given node field (similar to pivot table)

```
pg.get_range_of_node_attr(net,'production')
```

output:

attr	min value	max value
production	0.0	1000.0

```
pg.get_range_of_node_attr(net,'attraction')
```

output:

attr	min value	max value
attraction	0.0	1000.0

(7) Count the number of links for a given field

```
pg.get_num_of_links_by_attr(net,'link_type_name')
```

output:

link_type_name	number
primary	455
residential	1468
unclassified	90
secondary	452
tertiary	474

```
pg.get_num_of_links_by_attr(net,'allowed_uses')
```

output:

allowed_uses	number
bike	7493
auto	2948
walk	9195
unclassified	229
all	1868

(8) Check the range of a given link field

```
pg.get_range_of_link_attr(net,'length')
```

output:

attr	min value	max value
length	0.0961	1576.9808

```
pg.get_range_of_link_attr(net,'lanes')
```

output:

attr	min value	max value
lanes	1	5

```
pg.get_link_attr_value_list(net,'capacity')
```

---

output:

attr	min value	max value
capacity	0.0000	9500.0000

(9) Count the number of POIs for a given field

```
pg.get_num_of_pois_by_attr(net,'building')
```

output:

building	number
parking	122
yes	396
police	2
hospital	2
community_centre	1
library	2
school	11

```
pg.get_num_of_pois_by_attr(net,'activity_zone_id')
```

output:

activity_zone_id	number
8	34
7	57
17	50
12	93
11	36
21	17
13	50

(10) Check the information of one or all zones

```
pg.get_info_of_zone_by_id(net,zone_id=1)
```

---

output:

zone_id	name	total_poi_count	total_production	total_attraction
1	A1	51.0	845.6797	714.5002

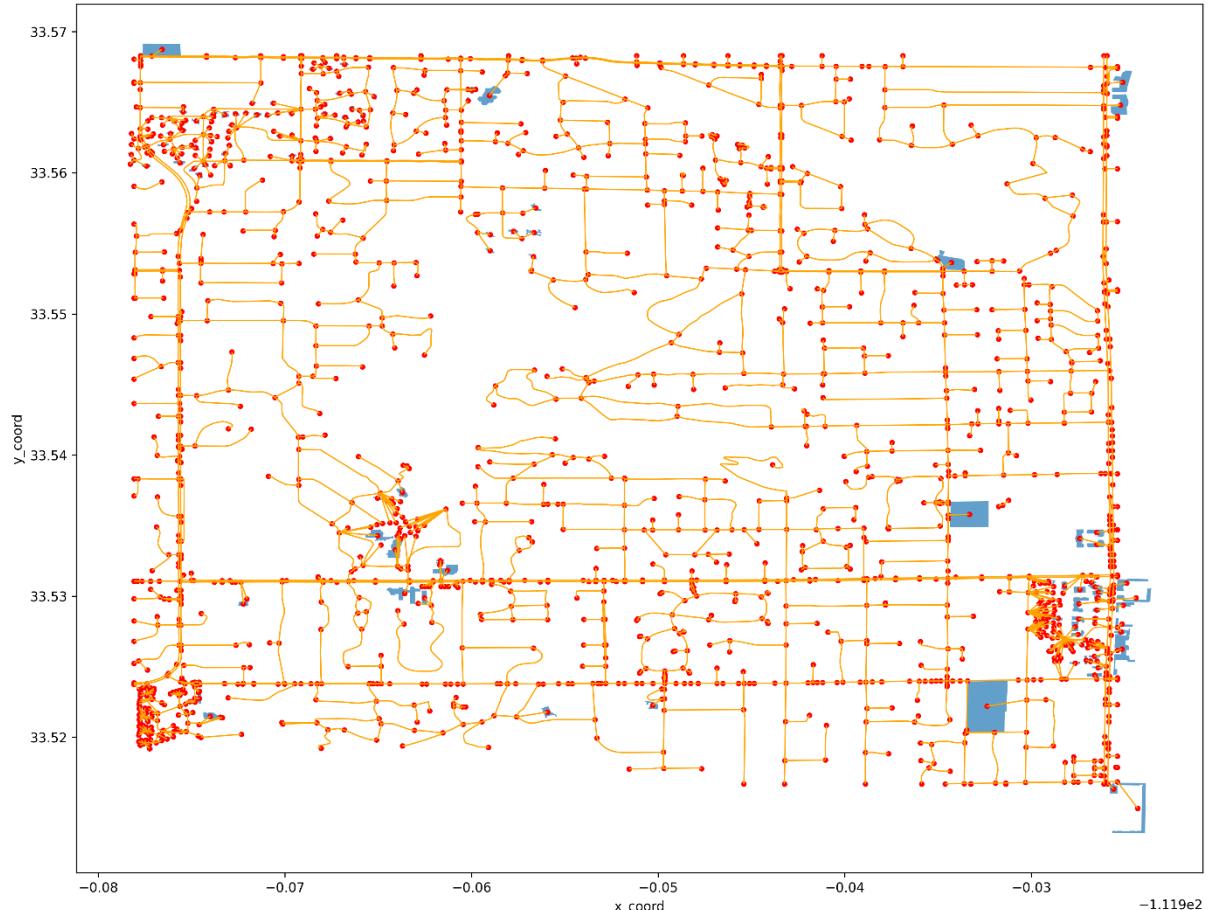
pg.get\_info\_of\_zone\_by\_id(net)

output:

zone_id	name	total_poi_count	total_production	total_attraction
1	A1	51.0	845.6797	714.5002
2	A2	14.0	634.2504	626.9926
3	A3	12.0	486.4889	328.7245
4	A4	1.0	16.8531	16.8531
5	A5	7.0	97.9670	97.9670
6	B1	36.0	1389.0082	1067.9394

## 4. Sample Networks

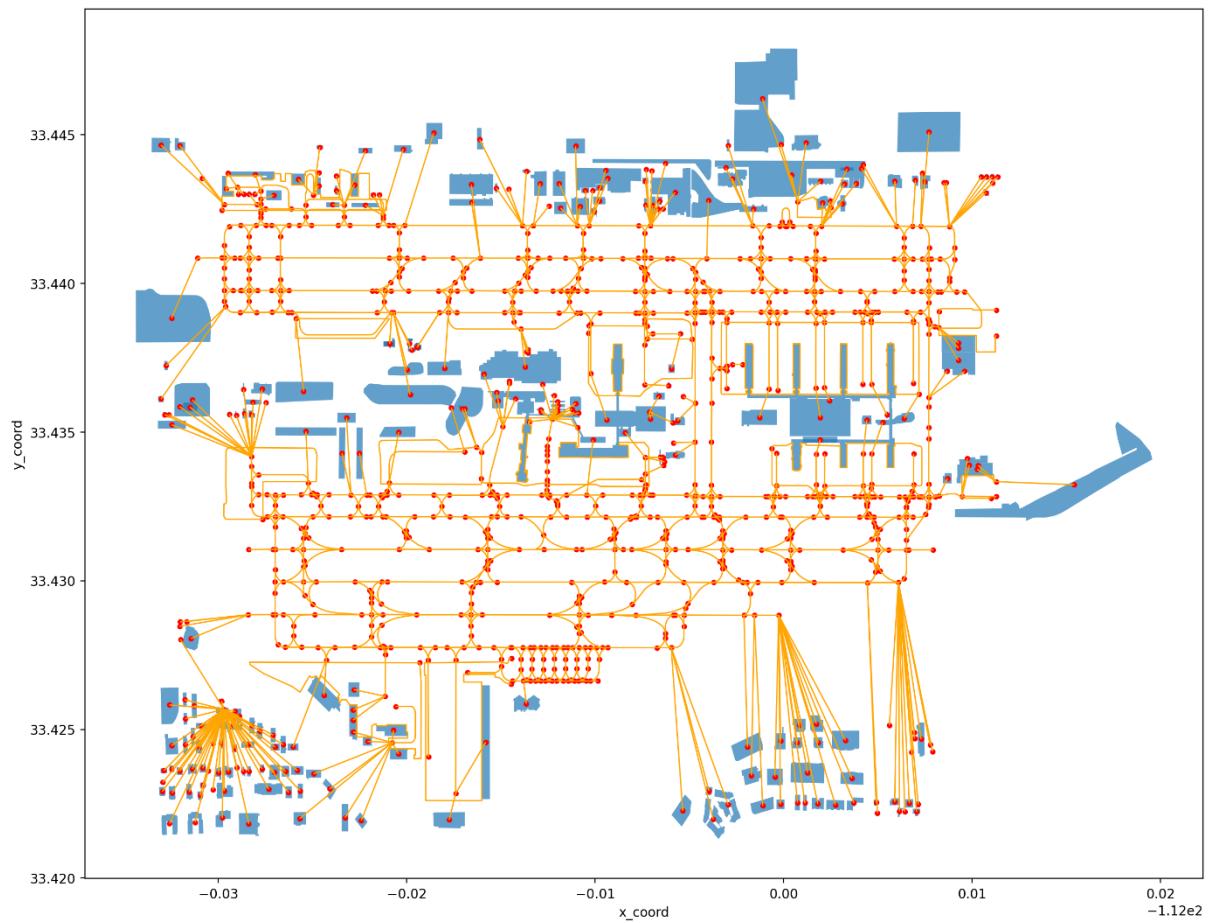
### 4.1 Example of Paradise Valley, USA



Paradise Valley

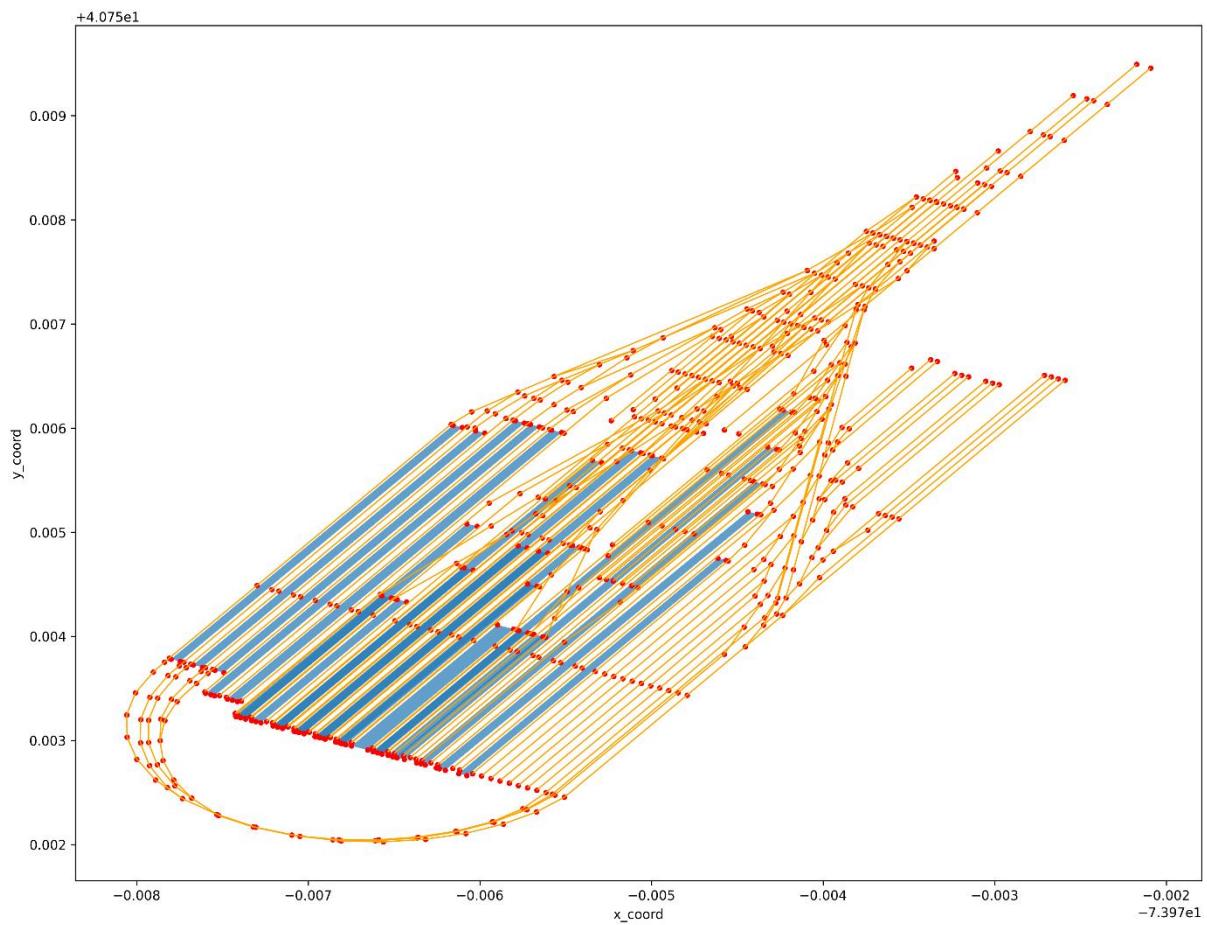
---

## 4.2 Example of [Phoenix Sky Harbor International Airport, USA](#)



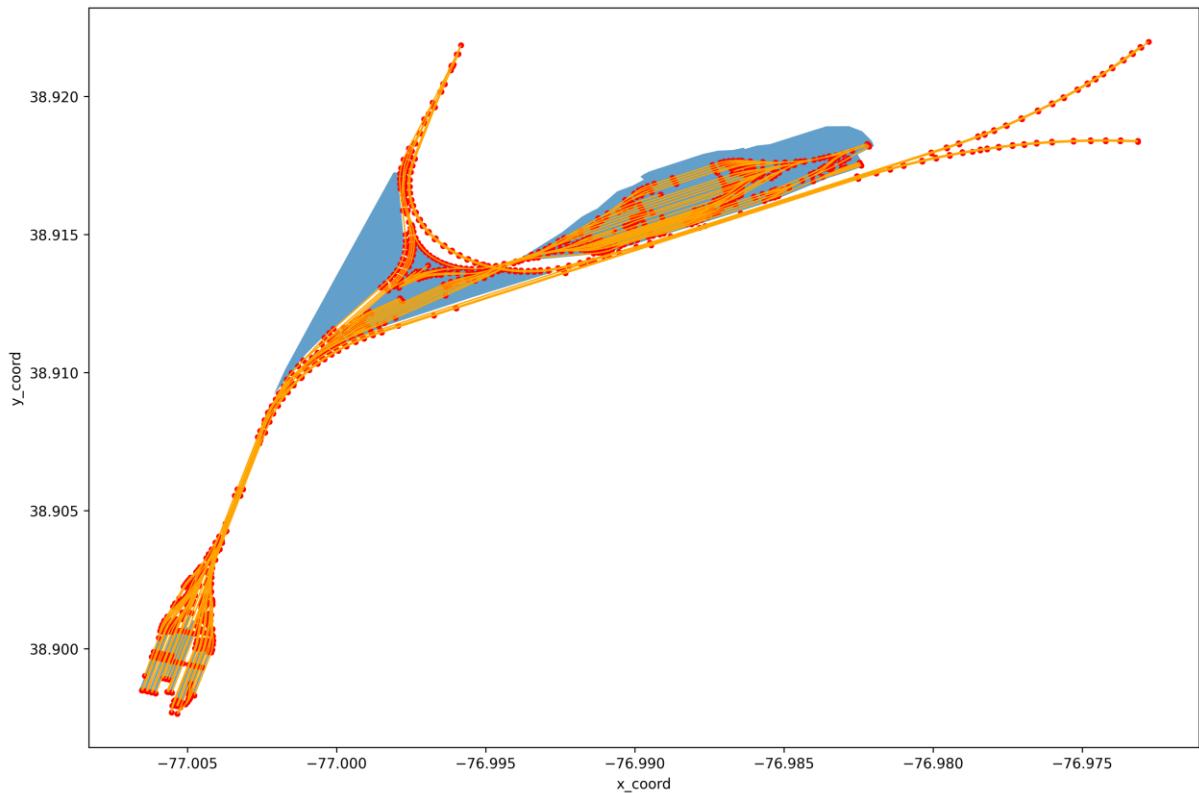
---

### 4.3 Example of Grand Central Terminal, USA



---

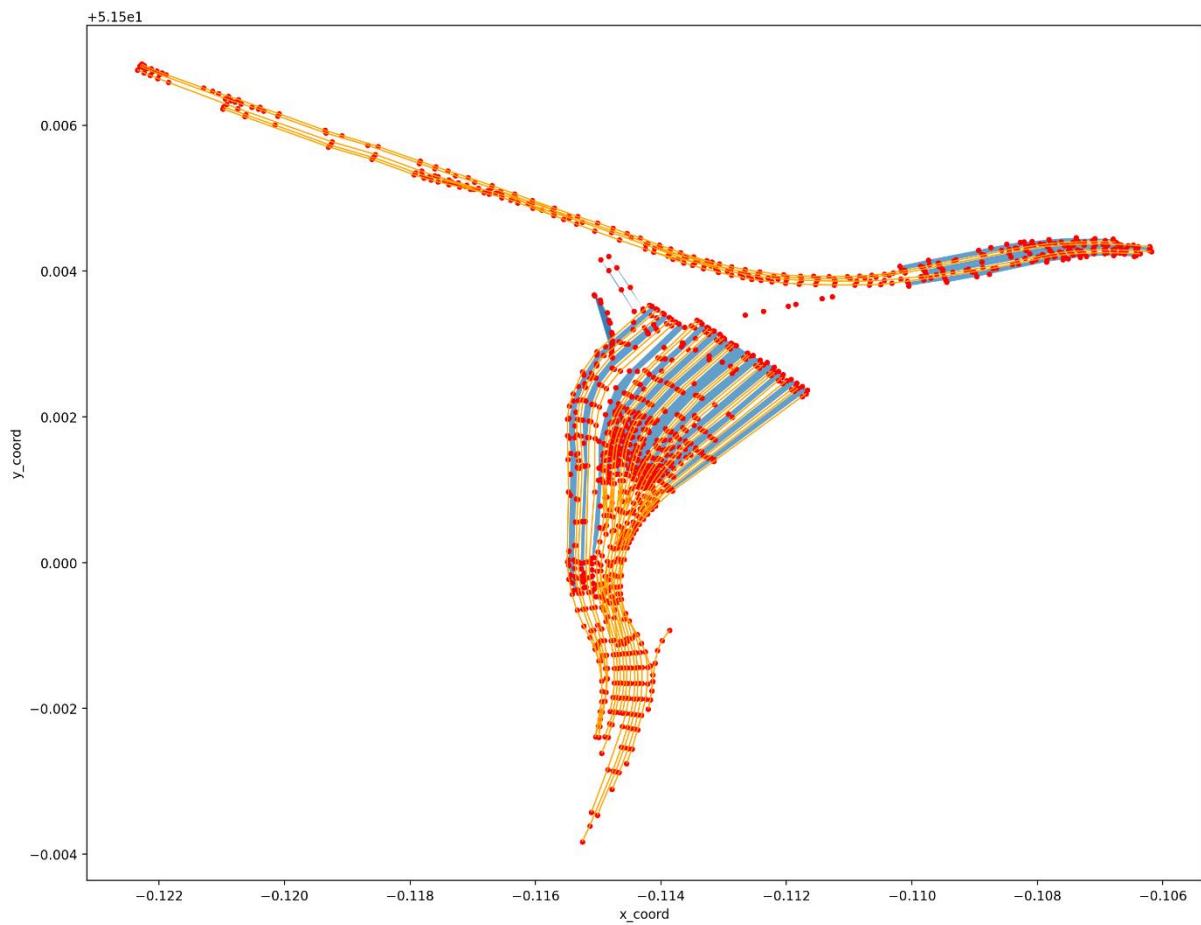
## 4.5 Example of [Near Washington Union Station, USA](#)



Near Washington Union Station

---

## 4.6 Example of London Waterloo Station, UK



---

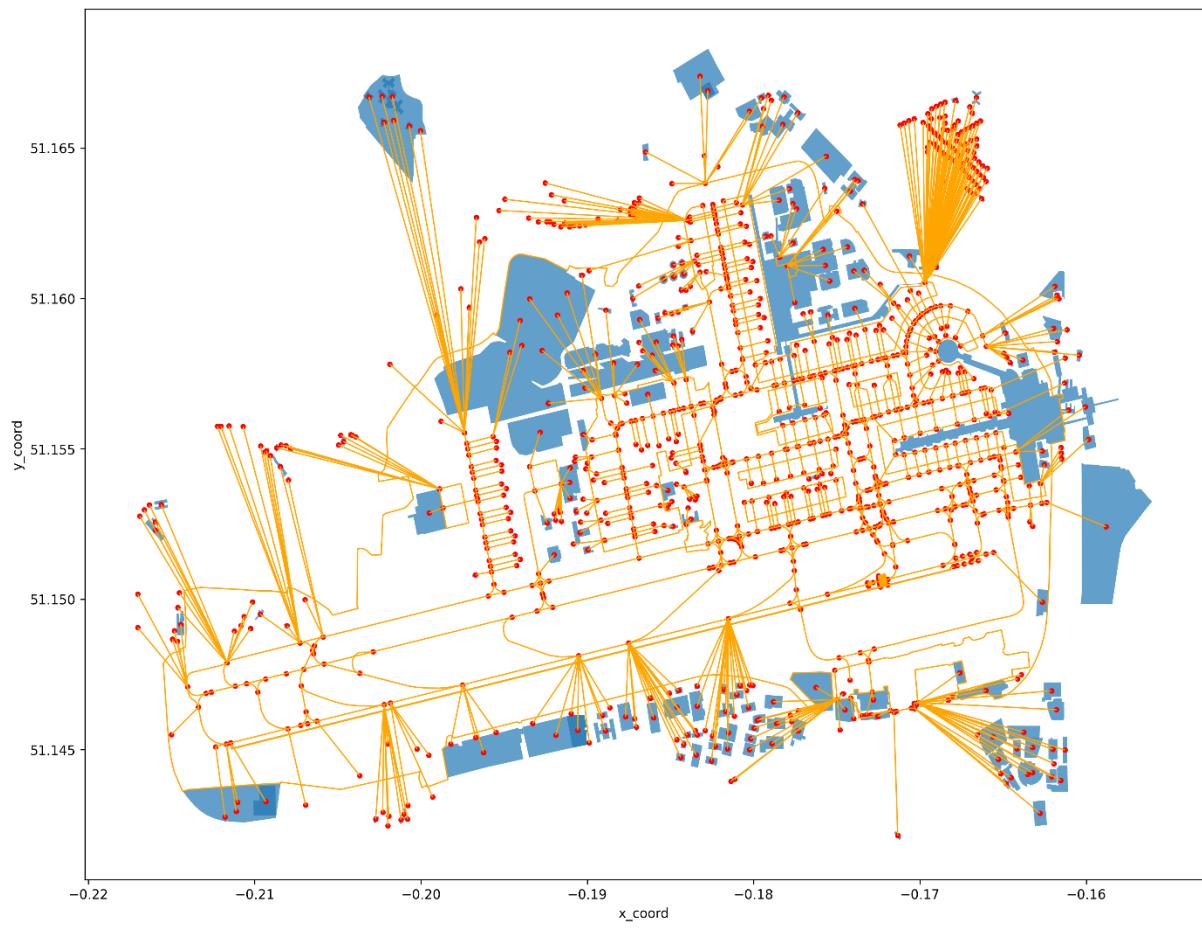
#### 4.7 Example of University of Cambridge, UK



University of Cambridge

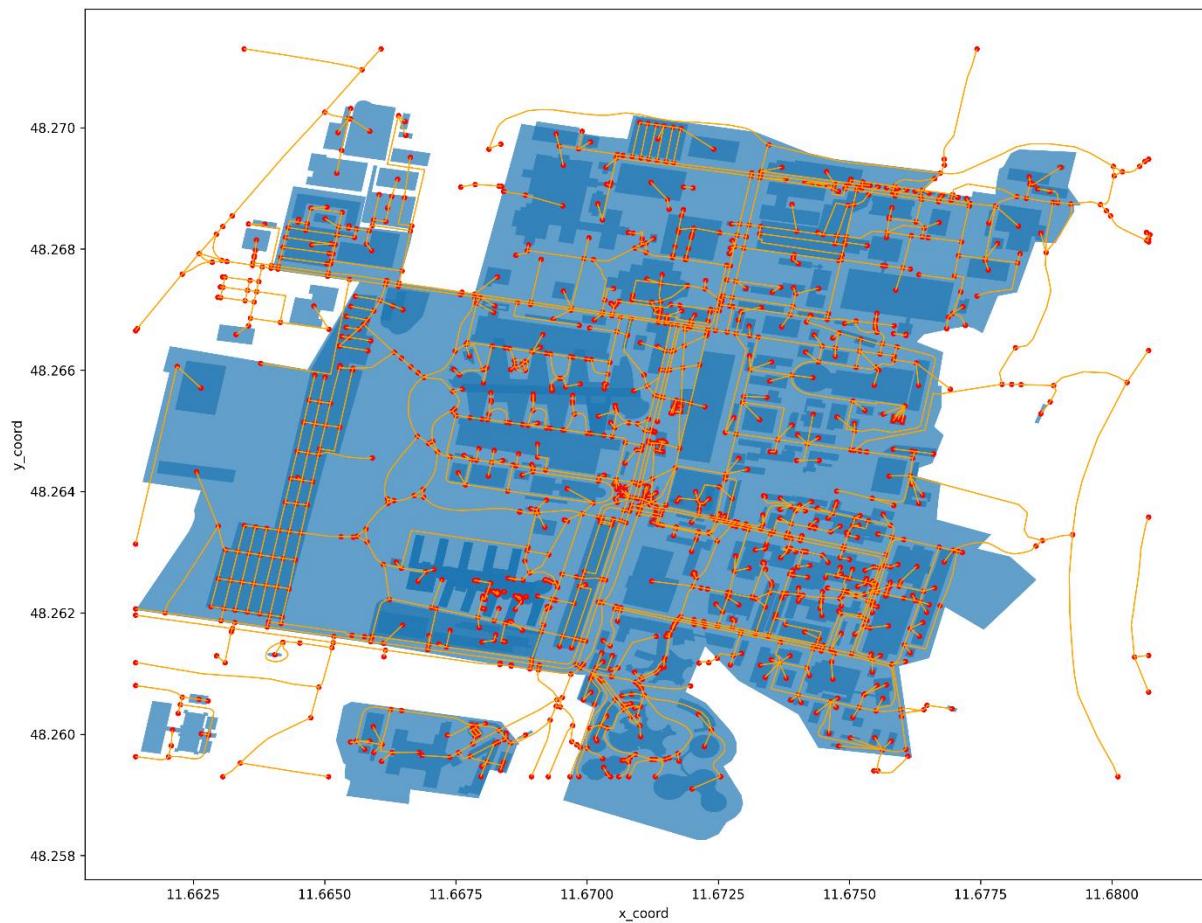
---

#### 4.8 Example of Gatwick Airport, UK



---

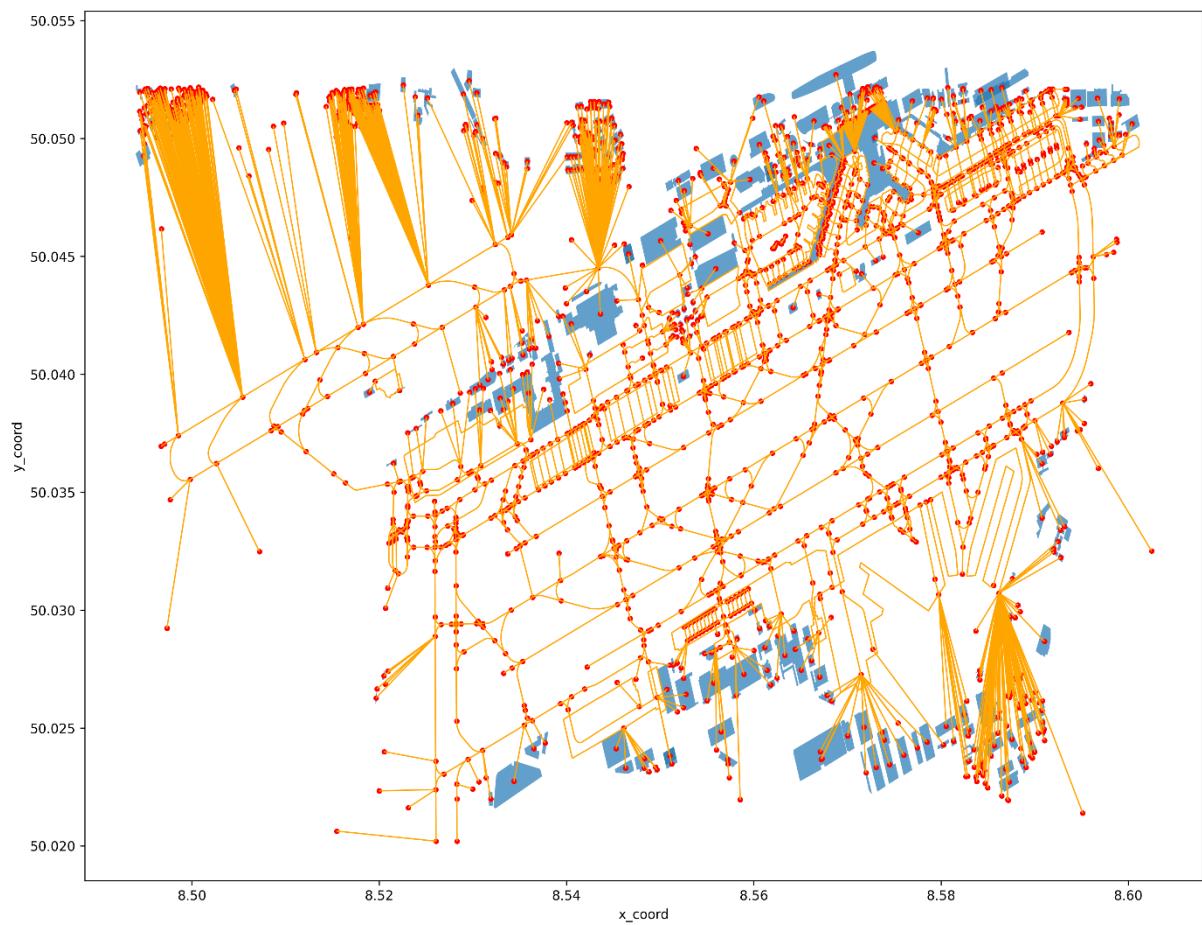
## 4.9 Example of [Technische Universitat Munchen](#), Germany



Technische Universitat Munchen

---

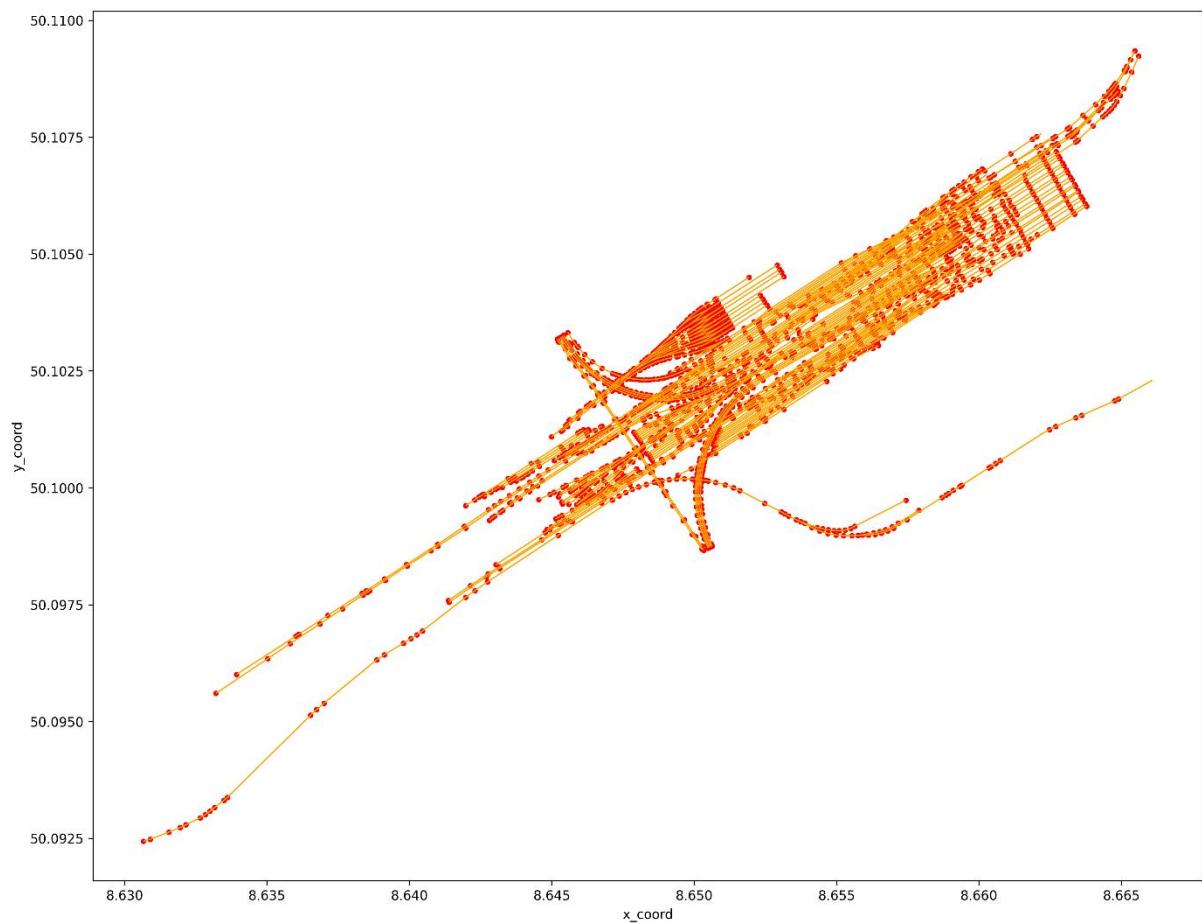
#### 4.10 Example of [Frankfurt International Airport](#), Germany



Frankfurt International Airport

---

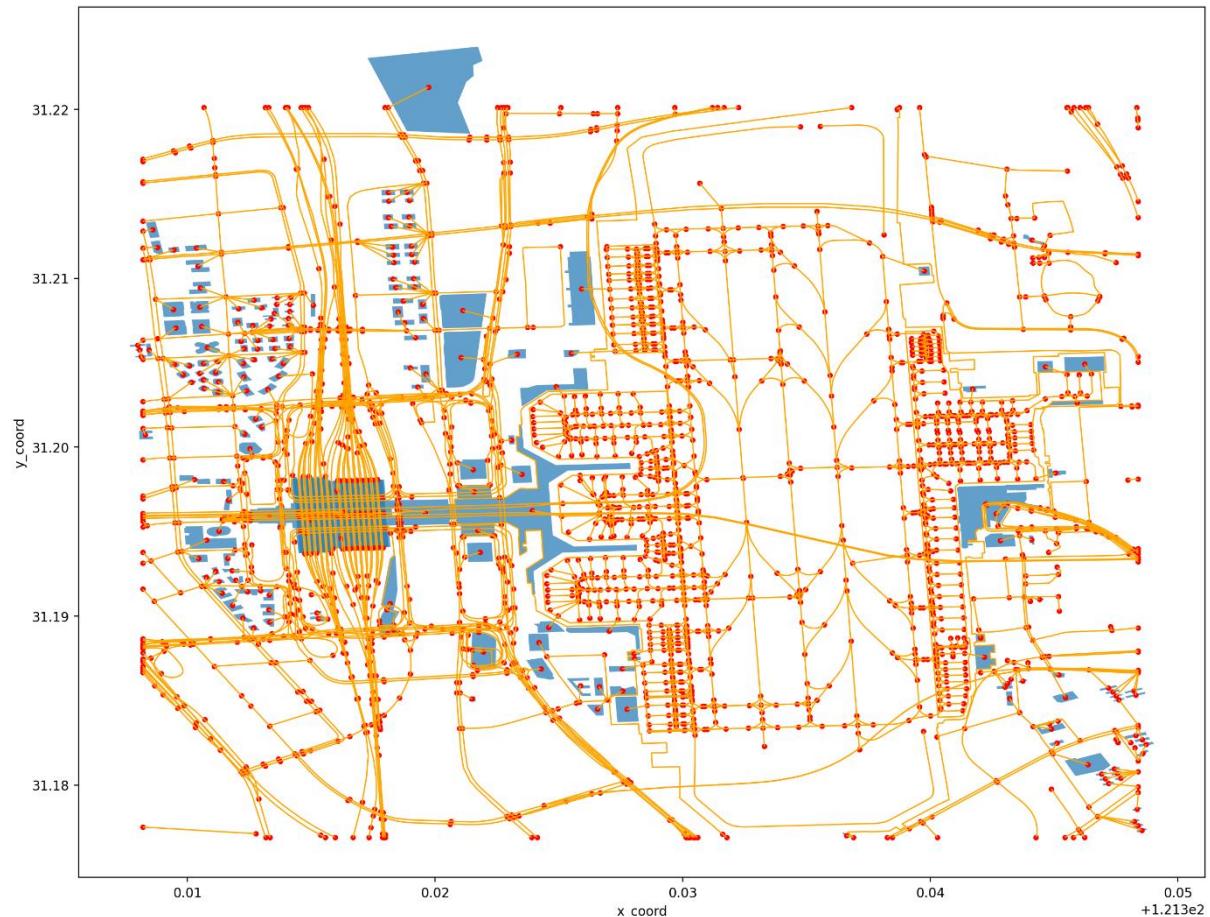
#### 4.11 Example of [Frankfurt Main Hauptbahnhof](#), Germany



Frankfurt Main Hauptbahnhof

---

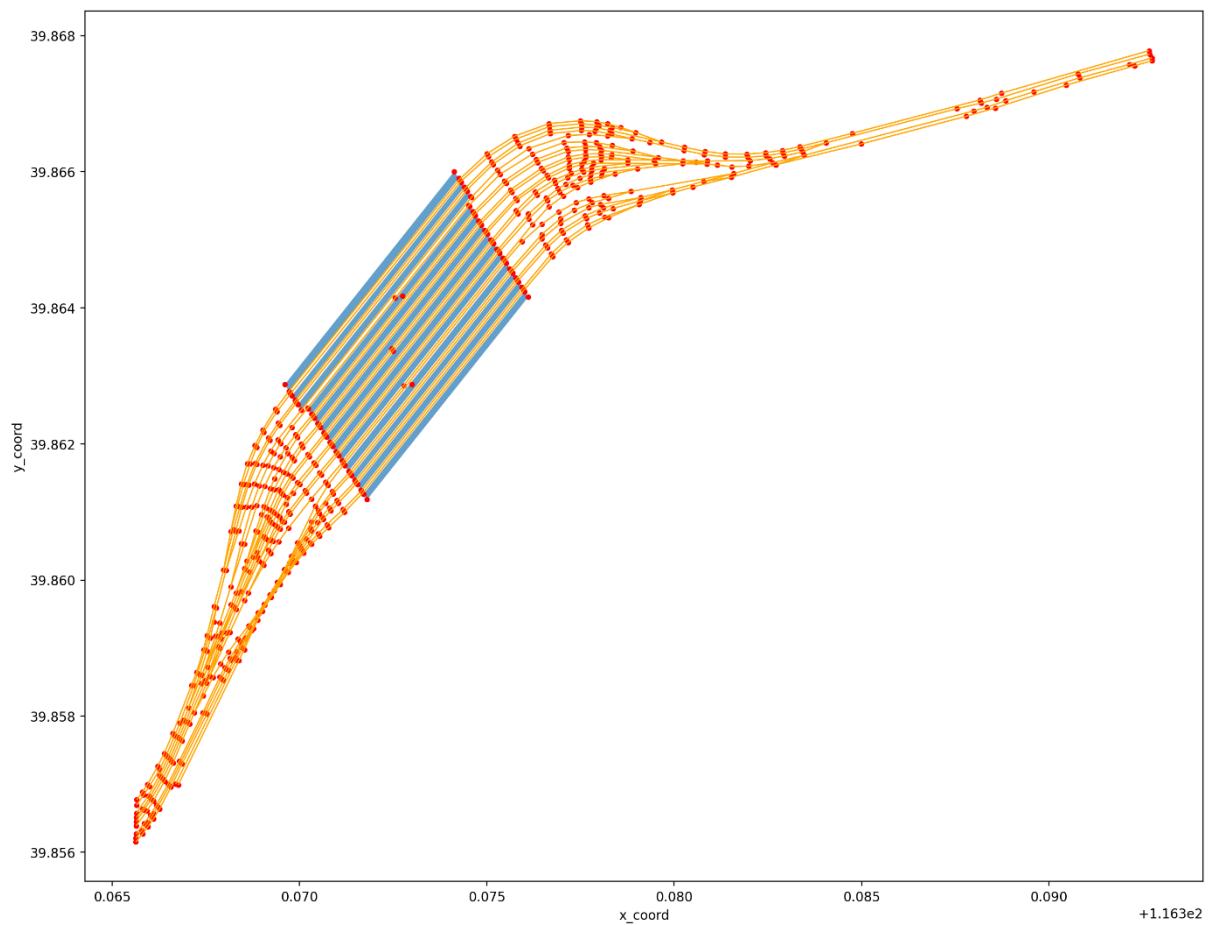
**4.12 Example of Shanghai-Hongqiao integrated railway station and airport facility, China**



Shanghai-Hongqiao integrated railway station and airport facility

---

#### 4.13 Example of Beijing South Railway Station, China



Beijing South Railway Station

---

#### 4.14 Example of Beijing EMU Depot, China

