

# Simulation Step 2 Generate Delivery Data

March 5, 2024

```
[1]: import matplotlib.pyplot as plt
import math
import random
import numpy as np
```

**Hints for students:** The utility section contains code you need to use *unchanged* to generate the test data required. You may use this code for your final solution, beware not to overwrite function definitions in this section. Otherwise you can ignore the code in this section.

## 1 Utilities

**Note:** Section 1.1-1.4 is identical to the MST example (Week 3). 1.5-1.10 are adjustments of the data structures for the current problem.

### 1.1 Points and Distances

```
[2]: def dist(p1, p2):
    (x1, y1) = p1
    (x2, y2) = p2
    return int(math.sqrt((x1-x2)**2+(y1-y2)**2))
```

### 1.2 Maps

```
[3]: def plotMap(G, T=[], P=[], W=None,
               style='r-o', lw=1, ms=3,
               styleT='go', msT=5,
               styleP='b-o', lwP=3, msP=1,
               stylePT='go', msPT=7,
               styleW='bo', msW=7,
               text=None, grid=False):
    fig = plt.gcf()
    fig.set_size_inches(6, 6)
    V, E = G

    if not grid:
        plt.axis('off')
    plt.plot([ p[0] for p in V ], [ p[1] for p in V ], 'ro', lw=lw, ms=ms)
    for (p, q) in E:
```

```

plt.plot( [ p[0], q[0] ], [ p[1], q[1] ], 'r-o', lw=lw, ms=ms)
for t in T:
    plt.plot( [ t[0] ], [ t[1] ],
              styleT, ms=msT)
plt.plot( [ p[0] for p in P ],
          [ p[1] for p in P ],
          styleP, lw=lwP, ms=msP)
for p in P:
    if p in T:
        plt.plot( [ p[0] ], [ p[1] ],
                  stylePT, ms=msPT)
if W is not None:
    plt.plot( [ W[0] ], [ W[1] ],
              styleW, ms=msW)
if text is not None:
    maxX = max([p[0] for p in V])
    plt.text(0.8*maxX, 0, text)
if grid:
    plt.grid()
plt.show()

```

### 1.3 Add Targets

```

[4]: def addTarget(M, T):
    V, E = M
    E = E.copy()
    V = V.copy()
    for t in T:
        minD = math.inf
        minE = None
        for e in E:
            P, Q = e
            distT = dist(P, t)+dist(t, Q)-dist(P, Q)
            if distT < minD:
                minD = distT
                minE = e
        P, Q = minE
        E.remove( (P, Q) )
        E.append( (P, t) )
        E.append( (t, Q) )
        V.append(t)
    return V, E

```

## 1.4 Generate Warehouse Locations

```
[5]: def generateWarehouseLocation(M):  
      V, _ = M  
      W = random.sample(V, k=1)[0]  
      return W
```

## 2 Loading Sample Data

```
[6]: import pickle  
      with open('data.pickled', 'rb') as f:  
          M, C = pickle.load(f)
```

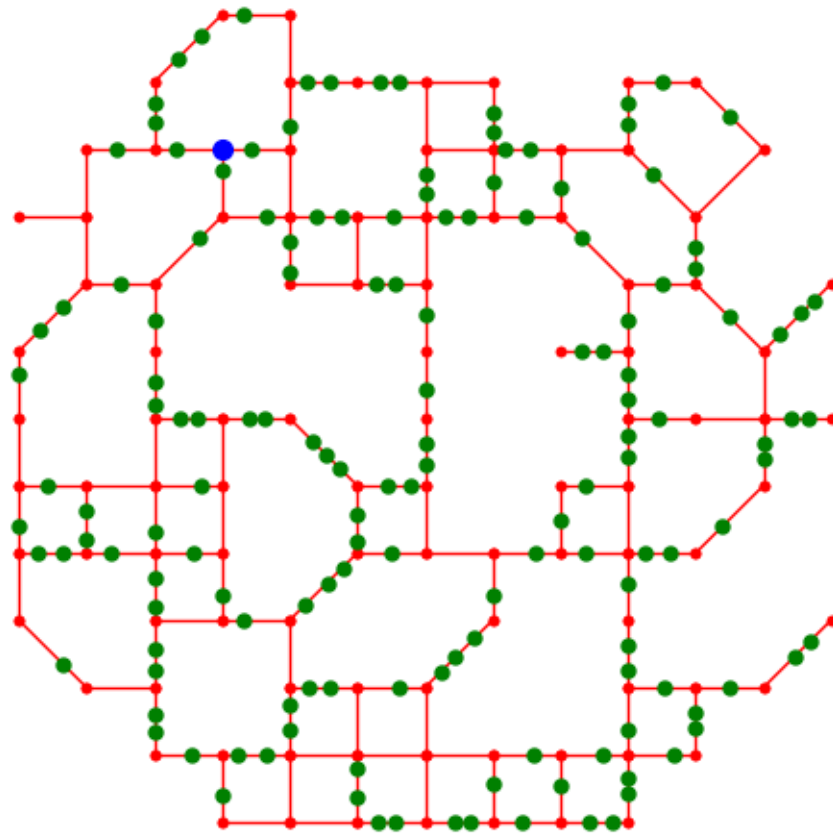
Split the map into Vertices and Edges:

```
[7]: V, E = M
```

Pick a random location for the Warehouse at one of the intersections (vertices) on the map.

```
[8]: W = generateWarehouseLocation(M)
```

```
[9]: plotMap(M, T=C, P=[], W=W, text="seed=9999")
```



seed=9999

### 3 Generating Delivery Data

```
[10]: def generateDeliveryData(p, C, days, seed=0):
    ## p is the average number of parcels per day per customer
    ## C is the number of customers to be served
    ## days is the number of days for which data are to be generated.
    np.random.seed(seed)
    R = np.random.poisson(lam=C*p, size=days)
    D = [ sorted(list(np.random.choice(range(C), size=i))) for i in R ]
    return D
```

#### 3.1 Demo Example

```
[11]: D = generateDeliveryData(0.2, 10, 5)
D
```

```
[11]: [[0, 3, 3], [0, 5], [1, 2, 3, 3, 8], [3], []]
```

```
[12]: sum([len(i) for i in D])
```

```
[12]: 11
```

Over 5 days, a total of 11 parcels are to be delivered:

- On the first day deliver one parcel to customers 0 and two parcels to customer 3.
- On the second day deliver one parcel each to customer 0 and 5.
- On the third day deliver one parcel each to customers 1, 2 and 8, and two parcels to customer 3.
- On the fourth day deliver one parcel to customer 3.
- And on the fiths day there are no parcels to delier.

## 4 Delivery over one month for Sample Map

```
[13]: D = generateDeliveryData(0.23, len(C), 30)
```

```
[14]: sum([len(i) for i in D])
```

```
[14]: 1047
```