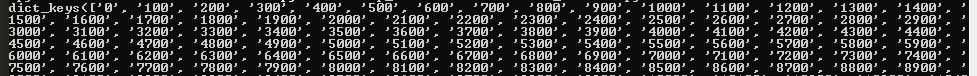
Data format

1. Grid network related
2. A NxN adjacency matrix for grids
3. A Nx2 numpy array to store centroid lng and lat of each zone

N is the number of zones

1. Order related
2. all\_request.pickle

a dict shown as below



Each entry of the dict is another list, which contains orders as below:

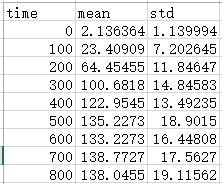


Here, each entry respectively represents order\_id, t\_start, t\_end, origin\_lng, origin\_lat, dest\_lng, dest\_lat, immediate\_reward, dest\_grid\_id, (origin\_grid\_id will be provided in Siyuan’s experiment)

To note: the time interval here can be changed with flexibility

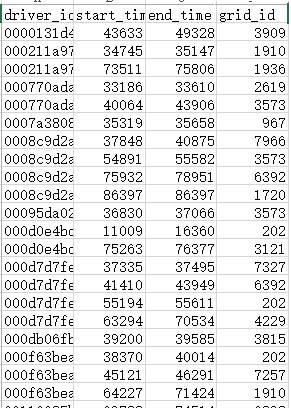
1. demand\_stat

a csv file shown as below; time represents true timestamp (counted in seconds); mean and std represents the statistical measurement for the total demand within each time interval. The length of time interval can be changed with flexibility



1. Driver related
2. Driver\_info.csv

A csv file shown as below; start\_time and end\_time represent the time when the driver becomes online or offline; grid\_id represents the depot grid for the driver



1. driver\_stat.csv

mean and std are utilized to generate the total number of drivers for a whole day



Simulator logic

Process for simulator

1. 每10 min预测一次未来订单，更新predicted\_pool
2. Order dispatching (先用段的 or 用km)，考虑long-term reward (extracted from value\_table)以及预测订单(设置两种方式：全部展开以及基于权重), 更新predicted\_pool
3. collect transitions (st, at, st+1, r) for long-term reward（先简单参照kdd的格式）, update transition\_buffer
4. 更新wait\_request/matched\_request, order cancellation based on waiting time/pickup distance
5. Conduct reposition
6. New order generation
7. Online/offline update
8. Update next state （纯idle车辆暂时先停在原地； reposition的车辆位置实时更新；需要update predicted\_pool）
9. Update time, output transition\_buffer

更新对象

1. predicted\_pool

an OD matrix; each entry value represents the predicted od flow from zone i to zone j within 10 minutes

1. driver\_table

a pandas DataFrame object; column names are as follows

|  |  |
| --- | --- |
| status | 0 is idle; 1 is pick-up; 2 is on-trip; 3 is repositioning |
| start\_time | The time stamp when the current status begins |
| end\_time | The time stamp when the current status ends |
| lng | The longitude where the driver is at |
| lat | The latitude where the driver is at |
| grid\_id | The grid id where the driver is currently located at |
| target\_grid\_id | The grid id where the current task will finish at |
| target\_loc\_lng | The longitude where the current task will finish at |
| target\_loc\_lat | The latitude where the current task will finish at |
| remaining\_time | The remaining time to finish the current task |
| matched\_order\_id | The python str object that represents the order id matched with the driver; the str object “None” is used if no orders matched with the driver |
| total\_idle\_time | The total of idling time since the last time when the driver status turns to idling. 0 when the status is not idling. |

1. wait\_requests

|  |  |
| --- | --- |
| 'order\_id' | A str object used to identify each order |
| 'trip\_time' | Trip time from the origin to the destination for each order |
| 'origin\_lng' | Longitude for origin |
| 'origin\_lat' | Latitude for origin |
| ‘origin\_grid\_id’ | Grid id for origin |
| 'dest\_lng' | Longitude for destination |
| 'dest\_lat' | Latitude for destination |
| 'dest\_grid\_id' | Grid id for destination |
| 'immediate\_reward' | Real reward |
| 't\_start' | The timestamp when the order coming into the system |
| 't\_matched' | The timestamp when the order matched with a driver |
| 'pickup\_time' | Time required for pick-up after matching |
| 'wait\_time' | Waiting time the order has experienced till now |
| 't\_end' | The timestamp when the order arrive its destination |
| 'status' | 0 is not matched; 1 is matched |
| 'driver\_id' | The driver id matched with the order |
| 'maximum\_wait\_time' | The maximum of waiting time that the order allows |
| 'cancel\_prob' | The probability for cancellation of the order based on pickup distance |
| 'pickup\_distance' | Pickup distance counted in meter |

1. matched\_requests

The format is the same as wait\_requests

1. transition\_buffer

a list with transitions. Each transition is in the form of [st, at, st+1, reward]. St is an numpy array of time stamp and grid id; at can be ignored here; reward is a float value.

1. value\_table

required definition from Duan

Noted:

1. All the time-related object is counted in seconds, while all the distance-related object is counted in meter
2. All the six objects will be stored as a property of the simulator class. “self.xxx” is necessary when utilizing the objects.