一、Share

Beyond Shortest Paths: Route Recommendations for Ride-sharing

1.方法：

问题目标：

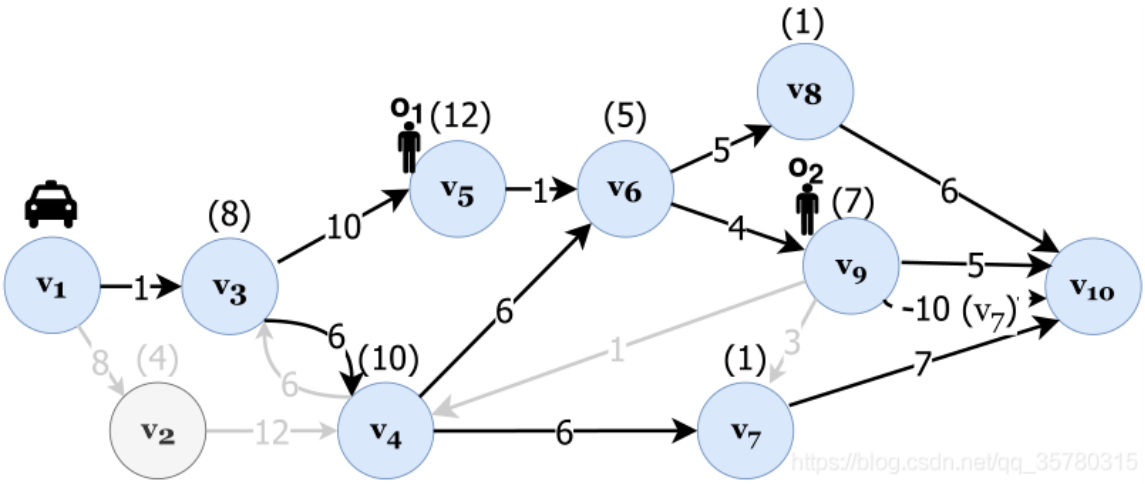
在可接受的范围内(a)，稍微走远拐个弯，同一辆车接载多名客人（4），旅程时间延长，但总收入增加

Can we move beyond shortest paths by recommending a route that is close in length to the shortest path and offers much higher chance of finding a compatible ride-order?

如何构图实现：前向网络

前向边：**这条边的终点**离最终**目的地**的距离大于**这条边的起点**距离最终目的地的距离

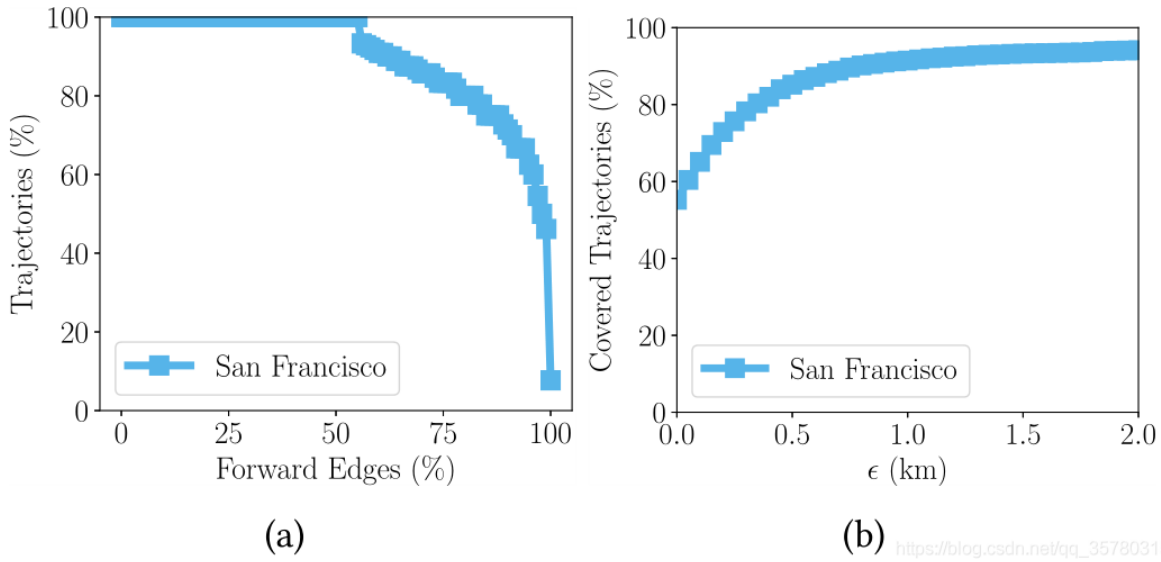
前向网络由前向边构成



数据分析：

1、图a显示的是SF市**前向边**所占的比例，可以看到有50%的轨迹都只包含前向边，而多达80%的轨迹数据中至少有80%也全是前向边。

2、图b描述的则是额外距离（也就是可在后向网络里绕路打转的距离）对实际路径的覆盖度。可以看到当绕路距离保持在0.5km时，85%左右的实际轨迹都可被覆盖。



说明：

1、出租车的轨迹数据大多数都是一直趋向目的地的，说明了前向网络的有效性；

2、适当绕路有助于获取更多的兼容乘客。

2、如何在DAG前向自环图中找到在a约束下的最优路径：DP

3、实验

数据集：

Table 2: Summary of the datasets used

City	# nodes	# edges	# ride-orders
New York	61,298	141,373	14.77 million
Singapore	52,653	86,410	12.57 million
San Francisco	3,527	7,964	0.372 million

基于三点评估

- quality
- 路线规划
- 效率

baseline

- shortest path
- route planning
- optimal path

metrics

- Percentage of orders without ride-sharing
- Passengers per kilometer (km)
- Average waiting time
- Rejection rate

Share的代码实现：

问题：C++GNU运行出了些问题

https://github.com/idea-iitd/Ride-Sharing

This project can be run on any city map and we have experimented and testes on the data for the cities: Singapore, New York and San Francisco

Steps to run the project

1. Download the Data Set for the city for which you want to run the simulation from the following link: https://drive.google.com/drive/folders/1DiVSOqANI3Ww0jHSKMw5VcxH04aIJsHC?usp=sharing
2. Keep all the data files in the 'main' folder as that of all the other code.
3. To compile the Project run:

   ```
   g++ -O3 -std=c++11 realtimesimulation.cpp Hungarian.cpp
   ```

   This will generate the 'a.out' file and then execute it with parameters.
4. To run : Usage:

   ```
   ./a.out [location={SF|NY|SG}] [alpha=1.3] [route={dij|dag|dex}]
   [maxDepth=0.2] [assign={hun|pxa}] [maxCab=2000] [cabCapacity={2|3}]
   [Optional: starttime endtime]
   ```

   The parameters can be explained as follows:

   - **Location:** SF|NY|SG represents the city in which you are running the simulation. SF= San Francisco, NY= New York , SG= Singapore
   - **Alpha:** Percentage Amount of extra distance that we allow the passenger to travel as compared to the shortest path from its source to its destination.
   - **Route:** This parameter chooses the type of route recommendation algorithm that we run.

     ```
     'dij' = Dijkstra (Shortest Path)

     'dag' = Use the Dynamic Programming algorithm after converting
     the graph into a DAG.

     'dex' = Use the Dynamic Programming algorithm after converting
     the graph into a DAG but along with back edges of
     length equal to Max Depth.
     ```

   - **Max Depth:** It is the maximum length of the reverse(back) edges allowed in the DAG.
   - **Assign:** This represents the assignment algorithm that is used for assignment of free cabs to passengers.

```
'hun' = Hungarian Algorithm for empty cab assignment

'pxa' = Greedy Price assignment algorithm used for comparison
with the VLDB paper.
```

- **MaxCab:** Maximum number of cabs running in the city
- **cabCapacity:** This represents the maximum number of passengers in the cab.

Optional:

- **starttime:** The start time of the simulation in the day
- **endtime:** The end time of the simulation in the day

Thus, a sample command would look like:

```
./a.out NY 1.2 dex 0.3 hun 4000 3 480 570
```

Here, the definitions for each variable have been mentioned in the paper.

二、Effective and Efficient Reuse of Past Travel Behavior for Route Recommendation KDD 19（确定没有源码）

三、meng Qu 的邮件没有收到回复（KDD14）

四、Profitable Taxi Travel Route Recommendation Based on Big Taxi Trajectory Data智能交通系统 （找不到源码）