

#0 数据格式解析与网络文件读取

数据文件：

settings.yml——包含 agent 的定义、需求时间范围、需求文件的位置等设置信息。

node.csv——包含 node 的 name,id,所在 zone_id,位置坐标等信息。

link.csv——包含 link 的 name,id,起止点,length,capcity 等信息。

demand.csv——包含每个 o_zone 2 z_zone 的需求量。

代码文件：

accessibility.py——可达性评估部分代码。

classes.py——定义了程序中用到的类，如 Node, Link, Agent 等。

colgen.py——列生成算法，用作

consts.py——存储程序中用到的常量。

dtaapi.py——Perform network assignment with DTALite。

path.py——与最短路相关的方法，如 fifo, deque 等。

utils.py——用到的“工具”，如读取网络文件、下载示例数据、转换数据类型等。

pathengine.cpp——C++语言编写的 deque 算法，在寻找最短路功能中会被调用。

文件读取：

首先初始化两个类：

```
assignm=Assignment()
```

```
network=Network()
```

```
read_network(load_demand='true',input_dir='.')——utils.py
```

load_demand——是否加载区域间 demand

input_dir——文件所在目录

子函数：——utils.py

```
read_settings(input_dir, assignm)——读取 settings.yml 中的设置信息，存入 assignm 中。
```

```
read_nodes(input_dir,network.node_list,network.node_id_to_no_dict,network.node_no_t  
o_id_dict,network.zone_to_nodes_dict)——用 DictReader 读取 csv 文件，然后转成 node 类，  
并将每个 node 的 id2no,no2id 存为字典，并保存 zone 与 node 的对应关系。
```

```
read_links(input_dir,network.link_list,network.node_list,network.node_id_to_no_dict,  
network.link_id_dict,assignm.get_agent_type_count(),assignm.get_demand_period_count  
( ),load_demand)——用 DictReader 读取 csv 文件，然后转成 link 类.并给所有 node 加上出弧与入  
弧属性。
```

```
read_demand(input_dir,d.get_file_name(),at,dp,network.zone_to_nodes_dict,assignm.co  
lumn_pool)——读取 odzone 之间的需求量。
```