

## read\_network

读取网络

**evaluate\_accessibility(ui, multimodal=True, mode='p',  
time\_dependent=False, demand\_period\_id=0, output\_dir='.')**

parameter:

multimodal: 如果为 True, 则将对 settings.yml 中定义的所有 agent 进行可访问性评估  
time\_dependent: True:可达性将使用所选时期 link 的自由流旅行时间, 换言之, 可达性只与所选时期有关, 与 agnet\_type 无关

output:

accessibility\_aggregated.csv  
accessibility.csv:

**\_update\_min\_travel\_time(an, at, min\_travel\_times, time\_dependent,  
demand\_period\_id)**

输出各个区域之间的最小旅行时间和其中最大的最小旅行时间 max\_min (用于计算所需的时间区间数)

min\_travel\_times[(zone\_id, to\_zone\_id, at\_str)] = min\_tt, min\_dist

an 的特点

有区域质心, 区域质心与区域内所有点连接

update\_generalized\_link\_cost(at, time\_dependent, demand\_period\_id):

根据 at 和 demand\_period, 更新 link\_cost

$$link_{cost} = link_{fft} + link_{routechoice} + \frac{link_{toll}}{agent_{vot}}$$

$link_{routechoice} - ?$

$link_{toll} - ?$

$agent_{vot} - ?$

interval\_num = \_get\_interval\_id(min(max\_min, MAX\_TIME\_BUDGET)) + 1

计算所需的时间区间数

## output

\_output\_accessibility(min\_travel\_times, zones, mode='p', output\_dir='.')

输出 accessibility\_aggregated.csv 文件

o_zone	d_zone	accessibility	distance	geometry
id 和 name	id 和 name	min_tt	min_dist	坐标

\_output\_accessibility\_aggregated(min\_travel\_times, interval\_num, zones, ats, output\_dir='.')

输出 accessibility\_aggregated.csv 文件

zone_id	geometry	mode	time_budget1	time_budget2	...
	坐标	agent	<MIN_TIME_BUDGET 的区域的数量	MIN_TIME_BUDGET + BUDGET_TIME_INTVL	...

## get\_accessible

get\_accessible\_nodes(self, source\_node\_id, time\_budget, mode, time\_dependent, tau)

tau: demand\_period

返回 lable\_cost<=time\_budget 的 node 集合

get\_accessible\_links(self, source\_node\_id, time\_budget, mode, time\_dependent, tau)

lable\_cost<=time\_budget 的 node 的 pred\_link 的集合