

#3 列生成——基于路径的 UE 分配

数据文件：

link_performance.csv——按 Link_ID 记录每条路径的基本信息以及 volume（分配的交通量），travel_time（在该 Link 上所需行驶时间），speed（在该 Link 上的预估行驶速度），VOC（ V_{over_C} ，即每个 Link 的能力利用率）。

代码文件：

colgen.py——列生成算法，用作基于路径的用户均衡交通分配。

三个主函数：

perform_column_generation(assignment_num, column_update_num, network)

output_columns(network)——作用可能同 StepII。

output_link_performance(network)——输出 link_performance，记录分配结果与利用率等。

本节函数几乎都用到两个重要的类，SPNetwork 与 Column。

SPNetwork——“Shortest Path Network”，在 Network 类基础上新增 agent_type, demand_period 属性。

Column 与 ColumnVec——通过 read_Column()读取 agent.csv 生成。

名词：

Voc——“Volume Over Capacity”

Vot——“Value of Time”

Tau—— τ ，代表 DemandPeriod 的 ID。

下设子函数：

_update_link_travel_time_and_cost(links)——更新 Link 的旅行时间&费用。

_reset_and_update_link_vol_based_on_columns(column_pool, links, dps, i, True)
——根据 Column 更新 Link 的流量。

_update_generalized_link_cost(A.get_spnetworks())——更新 Link 的广义费用。定义的广义费用由三部分组成，分别为对应时段走行时间、路段选择成本、路段收费。

_assignment_core(A.get_spnetworks(), column_pool, i)

子函数：_backtrace_shortest_path_tree()——位于 colgen.py #208，它的作用是什么？

_optimize_column_pool(column_pool, links, ats, dps, column_update_num)

用_update_column_gradient_cost_and_flow()方法更新{column_update_num}次 column_pool。

_update_column_travel_time(column_pool, links)——更新 Column 的旅行时间。

_update_column_gradient_cost_and_flow()——更新 Column 的梯度费用与流量。
作用与原理是什么？Gap 的含义是什么？