

说明文档

问题设置

给定快递公司一个城市区域内的网点集合，每个网点有发往其他网点的快递和对应的时间约束，求以最小的成本（成本计算逻辑下文阐述）安排车辆执行所有快递的运输任务。

假设快递公司使用的是同一车型的车辆，车辆数量不限，每辆车有对应的容量限制、固定成本和时间相关的变动成本，每辆车无固定起始场站、目的场站限制。

此外，部分网点可以进行快递的转运操作，以达到快递量的聚集，进一步降低运营成本。本赛题限制一个快递最多转运一次。

考虑实际操作要求，同一批次快递的运输线路一致（同样的始发网点、目的网点、开始时间、结束时间的快递为一个批次，在输入数据中对应一行），不允许拆分运输。

输入字段解释

输入数据包含多个算例。每一个算例的输入数据包括三张表，网络节点表，时间矩阵表，需求表。且所有算例共用同一份参数表。

注意，输入数据中坐标、快递票数、行驶时间、作业时间、成本等参数均经过转换处理，不体现公司或行业实际运营信息。

网络节点表（node_*.csv）

- name: 网点名称
- x,y: 网点横纵坐标（注：网点间时间单独提供，坐标信息仅用于可视化展现）
- transfer: 网点是否可以进行转运操作，True表示可以，False表示否

时间矩阵表（time_matrix_*.csv）

- name_x: 起始网点名称
- name_y: 目的网点名称
- time: 行驶时间

需求表（demand_*.csv）

- id: 快递编号
- origin: 快递起始网点

- destination: 快递目的网点
- start_time: 快递在起始网点的最早装车时间
- end_time: 快递在目的网点的最晚卸车时间
- quantity: 快递对应票数

参数表 (parameter.csv)

- vehicle_capacity: 车辆装载容量上限
- vehicle_service_time: 车辆在一个网点的装卸货时长
- parcel_transfer_time: 快递在转运网点的处理时长
- vehicle_fixed_cost: 单辆车的固定成本
- vehicle_unit_travel_cost: 车辆单位行驶时间的成本
- parcel_transfer_unit_cost: 快递转运的单位操作成本（每票）

计算逻辑

计算逻辑包括两个部分：成本（即目标函数）和约束。

成本

总成本 = 车辆数 * 车辆固定成本 (vehicle_fixed_cost) + 车辆总行驶时长 * 单位行驶成本 (vehicle_unit_travel_cost) + 转运快递票数 * 单位转运成本 (parcel_transfer_unit_cost)

注，车辆行驶时长计算不包括装卸货时长和在某个网点的等待时间。

约束

- 车辆停留时长约束：车辆在每个网点的到达时间和出发时间至少间隔装卸货时长 (vehicle_service_time)，即车辆从网点出发时间 - 车辆到达网点时间 \geq 装卸时长。注意，此处假设约束与以下两个因素无关：(1) 车辆在该网点的装卸作业类型（只装货不卸货，只卸货不装货，既装货又卸货），(2) 具体装卸快递票数
- 车辆装载容量约束：车辆在运行过程中，车上装载快递票数不超过车辆容量上限
- 快递时间窗约束：快递在始发网点可装车的车辆需要满足车辆离开网点时间 \geq 快递最早装车时间 (start_time) + 装卸货时长 (vehicle_service_time)；快递随车辆到达目的网点时，需要满足 车辆到达网点时间 + 车辆装卸货时长 (vehicle_service_time) \leq 快递最晚卸车时间 (end_time)
- 快递转运时长约束：快递如果在某网点发生转运操作，则对应装载此快递到达和发出该网点的车辆需要满足：装载此快递从该网点发出车辆的出发时间 - 装载此快递到达该网点车辆的到达时间 \geq 快递在网点的处理时长 (parcel_transfer_time)

提交要求

提交内容包含文档及输出结果（进决赛队伍需提供源代码）。根文件夹名为 {参赛队伍名称}_城市配送系统，文件夹中包含 文档 文件，和 输出结果 文件夹。

```
{参赛队伍名称}_城市配送系统
├── 文档.(md/docs/pdf)
└── 输出结果
    ├── vehicle_{id}.csv
    └── parcel_{id}.csv
```

文档应至少包括以下章节：问题描述、数学建模、算法设计、结果展示分析和团队分工。

输出结果每一个算例包含两张表：车辆结果表和快递结果表。

- 多个算例的结果放在同一个文件夹中，输出文件的编码为utf-8。
- 同一个算例的输入数据有一个共用的编号id，如id=30时，输入文件为node_30.csv, time_matrix_30.csv, demand_30.csv。对应的输出文件命名规范为vehicle_{id}.csv和parcel_{id}.csv，当id=30时，输出文件为vehicle_30.csv和parcel_30.csv。

车辆结果表（vehicle_*.csv）

- vehicle_id: 车辆编号，数字，从0开始计数
- node_name: 停靠网点，与输入数据中节点表的name字段一致
- visit_no: 停靠顺序，数字，每辆车从0开始计数
- arrival_time: 到达网点时间，数字
- departure_time: 离开网点时间，数字
- loaded_parcel: 该网点装上的快递对应编号，多个快递以英文分号（;）区隔，不存在则为空值
- unloaded_parcel: 该网点卸下的快递对应编号，多个快递以英文分号（;）区隔，不存在则为空值

快递结果表（parcel_*.csv）

- parcel_id: 快递编号，与输入数据需求表的id字段一致
- is_transfer: 是否经过转运，填True/False（不小写不限）
- vehicle_id: 车辆编号，与车辆表中的vehicle_id字段一致
- node_loaded: 装车网点，与输入数据中节点表的name字段一致
- node_unloaded: 卸车网点，与输入数据中节点表的name字段一致

注，如果是转运的快递，则需要拆成两行填写（第一行为始发网点至转运网点的信息，第二行为转运网点至目的网点的信息）。