## 分货算法模型设计

## 一、问题描述

在固定的时间窗口内，对陆续达到的车辆进行批次处理，生成对应装车清单。

在当前时间窗口下进行批次处理之前，获取实时库存。

设当前货物共有 N 件，货物（cargo） 包含属性：品种、重量、规格、优先级、所在仓库、目的地（城市、区县、详细地址）。

设当前时间窗口下达到 M 辆车，车辆（truck） 包含属性：车辆载重、期望运输品种（非必需）、期望运输目的地（非必需）。

根据实时库存数据为每辆车生成装车清单（loading） 即车辆 选取的货物集合。

* **限制条件**：

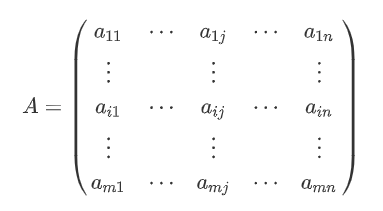
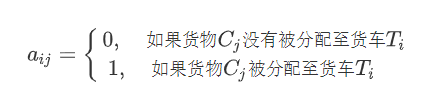
1. 单个车次货物总重量不得超过该车载重；
2. 单个车次货物种类 ；
3. 单个车次货物目的地为同一城市 ；
4. 单个车次取货仓库数目

* **优化目标**：

1. 优先发运客户催货、超期货物；

将客户催货和超期货物转化为优先级数字，其中客户催货对应优先级数字1，超期货物对应优先级数字2，数字越小优先级越高。

1. 货物运载总重量最大；

构造车次和货物的矩阵A，同时同一件货只能被放置一辆车上。

其中 是当前车次的载重限制。

1. 整体甩货单数目最少。

## 二、框架设计

* **生成装车清单候选集**

1. 根据当前车次数量 (M) 为每辆车生成空车次列表，列表长度即为车次数量；
2. 以城市和品种对货物进行分类，根据规则随机挑选货物放入空车次列表，为装车清单候选集元素，记为 ；
3. 利用rank函数计算当前车次列表的价值，该值越高，表示该分货结果效果越好；
4. 交换车次列表与货物列表中的货物 ，得到新的车次列表 ；
5. 对交换货物后的车次列表进行步骤3，反复迭代；
6. 设置迭代次数 ，及装车清单候选集长度 ，当迭代次数达到设置的阈值时，停止迭代，同时根据rank值对所有产生的车次列表排序，取前 个车次列表生成装车清单候选集 ；

* **车货匹配**

1. 以车次和可选候选集为结点，在满足限制条件的情况下，建立车次和候选集的边，以货物放置当前车次后的车次价值 作为边权重，构成二分图结构；
2. 同一件货物只能被放置到一辆车上，使用二分图算法进行匹配，以当前时间窗口下全局 值最高为目标，将对应的候选集与车次结合生成最终匹配结果。

## 三、详细设计

* **rank函数设计**
* **二分图边权重 值设计**

设车辆载重为 ,该车辆对应的第k个候选集的货物重量为 ，其中 。

## 数据

### 输入数据

* **可发库存数据**

实际更新库存时间：20分钟。

该表中数据：0点库存快照，每隔20分钟更新库存后插入的新数据，根据时间间隔对重量进行计算获取粗略的实时库存数据。

装车清单主表：ods\_db\_inter\_lms\_bclp\_loading\_main

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **含义** |
| car\_mark | varchar | 车牌号 |
| main\_prod\_list\_no | varchar | 主清单号 |
| prod\_list\_no | varchar | 清单号 |
| status | varchar | 业务状态 |

* **司机（车辆）数据**

调度司机表：ods\_db\_trans\_t\_plan\_driver 20190801-至今

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **含义** |
| plan\_no | VARCHAR | 调度单号 |
| order\_no | VARCHAR | 委托单号 |
| driver\_id | VARCHAR | 司机 |
| vehicle\_no | VARCHAR | 车牌号 |
| plan\_weight | DECIMAL ( 18, 6 ) | 计划重量（吨） |

司机到达时间近似看成装车清单表生成时间。

### 输出数据

* **装车清单详表**：ods\_db\_inter\_lms\_bclp\_loading\_detail

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **含义** |
| carmark | varchar | 车牌号 |
| main\_prod\_list\_no | varchar | 主清单号 |
| prod\_list\_no | varchar | 清单号 |
| notice\_num | varchar | 发货通知单号 |
| oritem\_num | varchar | 订单项次号 |
| commodity\_name | varchar | 品名 |
| standard | varchar | 规格 |
| weight | varchar | 重量 |
| outstock\_code | varchar | 出库仓库 |
| instock\_code | varchar | 入库仓库 |
| create\_time | varchar | 创建时间 |

1. 库存预处理 车辆信息处理；
2. 为每辆车生成装车清单候选集；
3. 计算每个候选集的score()，依据该值对候选集排序；
4. 对每一辆车，取当前候选集分值最高的货物进行匹配，并更新候选集；

**库存、车辆预处理**：根据品种（大品种）和目的地（省份）为车辆和货物进行第一轮分类。

**生成装车清单候选集**：同时为相同品种和目的地的车辆进行预分货，根据拼货规则为每一辆车生成所有可能的装车清单即为装车清单候选集。

**候选集排序**：利用score() 分别计算每一辆车的候选集里的每一个预分货单的score值，做降序操作。

**车货匹配**：根据车辆到达时间对车次进行排序，先到的车先安排货物，从该车次的预装车清单中选取score值最高的装车清单候选项。

**更新候选集**：一旦有分货即生成预装车清单操作，即更新候选集。具体操作：如果当前打包好的货物中包含已经被分走的货物，那么删除该候选项。

**特殊情况处理**：

限制条件 / 拼货规则：rule()

1. 单个车次货物种类 ；
2. 单个车次货物目的地为同一城市 ；
3. 单个车次取货仓库数目
4. 单车重量范围：0.8 \* 车辆载重 <= 单车重量 <= 车辆载重

Score()：

：优先级为1的货物：+5，优先级为2的货物：+3，优先级为0的货物：+0

：10×q，其中q为当前候选项货物总重量与该车次载重的比值（ ）

：合同号、订单号、发货通知单号三号一致：+2

其中 为手动设置的参数值。

具体的拼货规则问邵鲁玉：品种（大小品种、型钢、规格）

可发件数为0？