**管厂销售开单分货**

**开发文档**

**京创智汇（上海）物流科技有限公司**

**2020年05月**

修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **作者** | **版本** | **修改原因** | **主要修改内容** |
| 2020.05.19 | 承涛 | 1.0 | 新建 | 框架设计 |
| 2020.05.20 | 邵鲁玉 | 1.1 | 修改 | 内容补充 |
| 2020.05.21 | 林志刚  于萍 | 1.2 | 修改 | 框架结构修改；缺省内容补充 |

**目录**

[1、概述 4](#_Toc41052331)

[2、需求背景 4](#_Toc41052332)

[3、系统功能 4](#_Toc41052333)

[3.1 系统流程 4](#_Toc41052334)

[3.1.1人工流程 4](#_Toc41052335)

[3.1.2系统流程 5](#_Toc41052336)

[3.2系统主要功能描述 6](#_Toc41052337)

[3.2.1 开单推荐 6](#_Toc41052338)

[3.2.2 拼单推荐 7](#_Toc41052339)

[3.2.3 开单确认 7](#_Toc41052340)

[4、系统总体设计 7](#_Toc41052341)

[4.1系统架构 7](#_Toc41052342)

[4.2业务模型设计 8](#_Toc41052343)

[4.2.1数据结构设计 8](#_Toc41052344)

[4.2.2业务逻辑结构设计 11](#_Toc41052345)

[5、系统详细设计 13](#_Toc41052346)

[5.1数据采集 13](#_Toc41052347)

[5.2接口设计 13](#_Toc41052348)

[5.2.1 开单推荐接口 13](#_Toc41052349)

[5.2.2 开单确认接口(对比分析保存差异信息) 14](#_Toc41052350)

[5.2.3拼单推荐接口 15](#_Toc41052351)

[5.3 核心模块详细设计 17](#_Toc41052352)

[5.4 背包算法设计 17](#_Toc41052353)

[5.5代码管理 18](#_Toc41052354)

[附录 19](#_Toc41052355)

[A 工具描述 19](#_Toc41052356)

[B 基类描述 19](#_Toc41052357)

# 概述

本分货模块针对管厂类型，提供分货到车次的功能。由于六个管厂的分货流程基本一致，除各管厂品种类别存在差别外，目前无其他区别。各管厂根据平台录入的用户订单信息，结合销售人员和业务规则，通过算法程序进行分配优化，生成车货分配结果和车次信息。

# 2、需求背景

当前大宗商品产业链中存在物流资源（例如，运输车辆）短缺、仓库库存过多、货物超期严重等现象，这些现象导致了物流的严重阻塞以及物流成本的急剧增长。在物流供需不平衡的场景下，物资如何分配到车队、车辆是平衡物流资源的重要环节。分货策略不仅仅是决定单辆车辆的运载情况的配载策略，而且是对于长期整体物流效率与成本的把控。好的分货策略能够在车载重量、车辆数量限制的情况下，充分利用资源提高发运效率、降低库存。合理的分货策略能够在有限运输资源的条件下降低物流成本、提高服务质量，对整体物流带来经济效益的提升。

# 3、系统功能

## 3.1 系统流程

### 3.1.1人工流程

管厂销售处接到客户的订货需求---销售处根据当前库存进行配货或者安排生产---配货时要考虑到产品的重量、规格、体积，最终划分到发货通知单（车次）---开出发货通知到物流处---物流处按照发货通单上的信息联系合适的车辆。

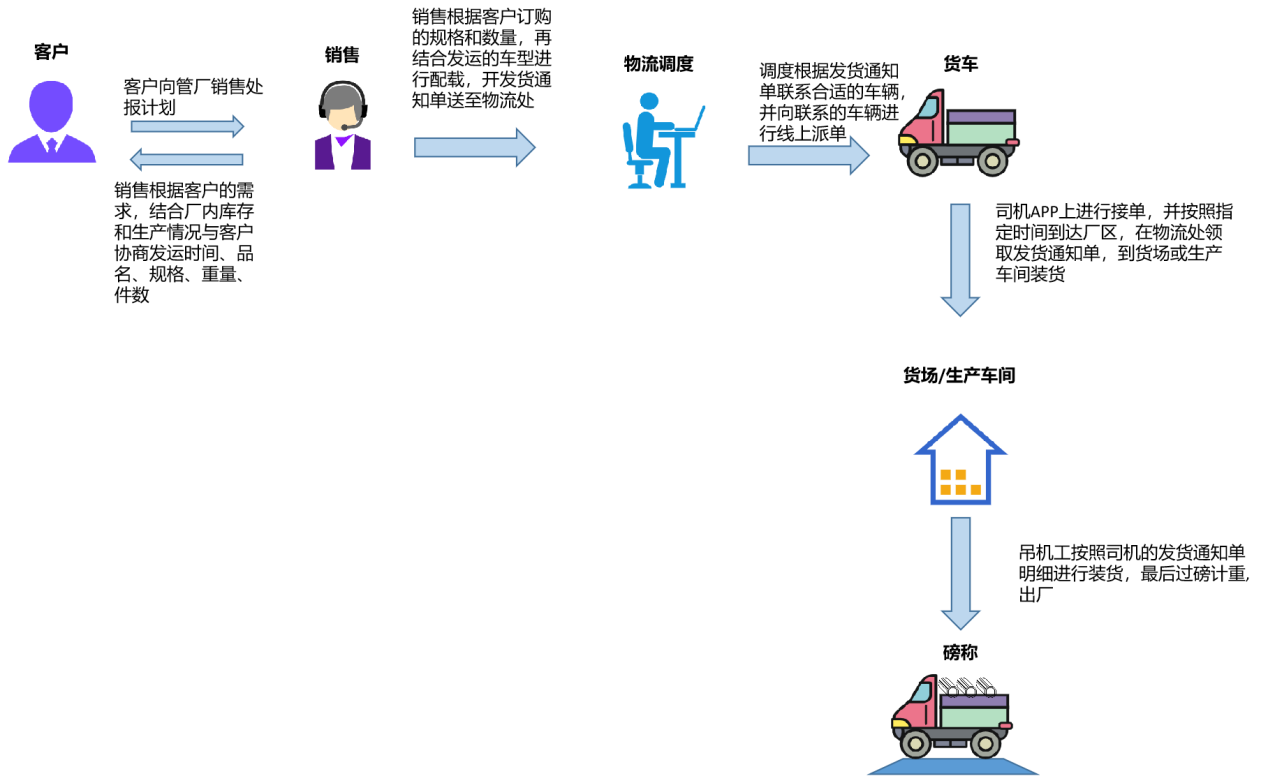


图 1人工分货流程图

### 3.1.2系统流程

程序流程图如下所示。

开单推荐是核心功能，解决的问题是将客户的订单信息，由分货模型过滤得到分货通知单结果。得到结果后由于有未满载车次，可选择使用拼单推荐功能，对车次信息进行检查，对未满载车次进行拼单，拼单模型会将由前端筛选出的可拼单的货物信息装到未满载车次上。另外确认开单功能主要是将人工确认后的订单与系统开单情况进行差异分析对比，收集差异数据方便后期对模型进行优化。

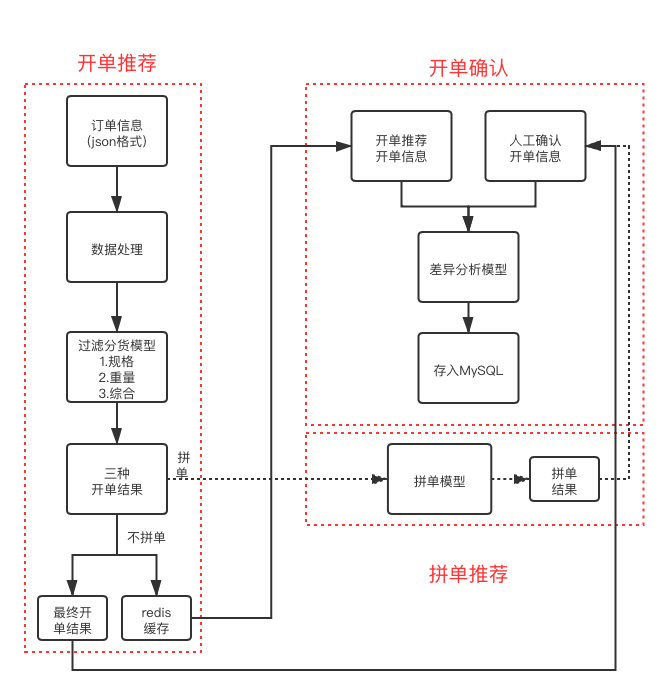


图2 系统流程图

## 3.2系统主要功能描述

智能开单系统主要有三大功能，分别是开单推荐中的过滤功能、拼单推荐拼单模型、开单确认中的分析模型。

### 3.2.1 开单推荐

开单推荐模块的输入是前端传入的客户订单信息的json数据，将json数据通过映射关系转换成系统中的订单实体对象，将处理好的数据由过滤分货模型处理，分货有三种优先级,分别是规格，重量、综合。最后给出对应权重的三种结果，保存至redis。

* 规格优先，假设当客户订单为50件焊管，重量32吨，但是这辆车的容量只够装40件，在进入过滤模型前，这比50件的焊管将拆分成为两单数据,这样的话，同一规格的货物在分货时会优先放在同一辆车上。
* 重量优先是指将订单打散，尽可能使车载接近极限载重，但这样会导致货物很散乱（即同一车会有不同品种，不同规格的货物混杂）。
* 综合分货是指同时考虑重量和规格两种情况，先统一处理大管，将大管尽量平均的放到每个车上，最后用小管补足剩余的载重，即同时考虑重量和规格两种情况，且不至于货物过于散乱。

### 3.2.2 拼单推荐

从开单推荐的结果得到车次信息（生成的发货通知单最后与车次匹配），根据车次进行拼单选择，通过得到客户的信息，根据客户信息又从数据库得到可拼单的货物列表，通过背包算法得出拼货之后的开单信息。

### 3.2.3 开单确认

通过开单推荐或拼单推荐的开单结果得到人工确认后的车次（生成的发货通知单最后会与车次匹配起来），与redis中缓存的第一次开单推荐结果的车次比较可以得到两次的差异，存入数据库方便后续优化。

# 4、系统总体设计

## 4.1系统架构

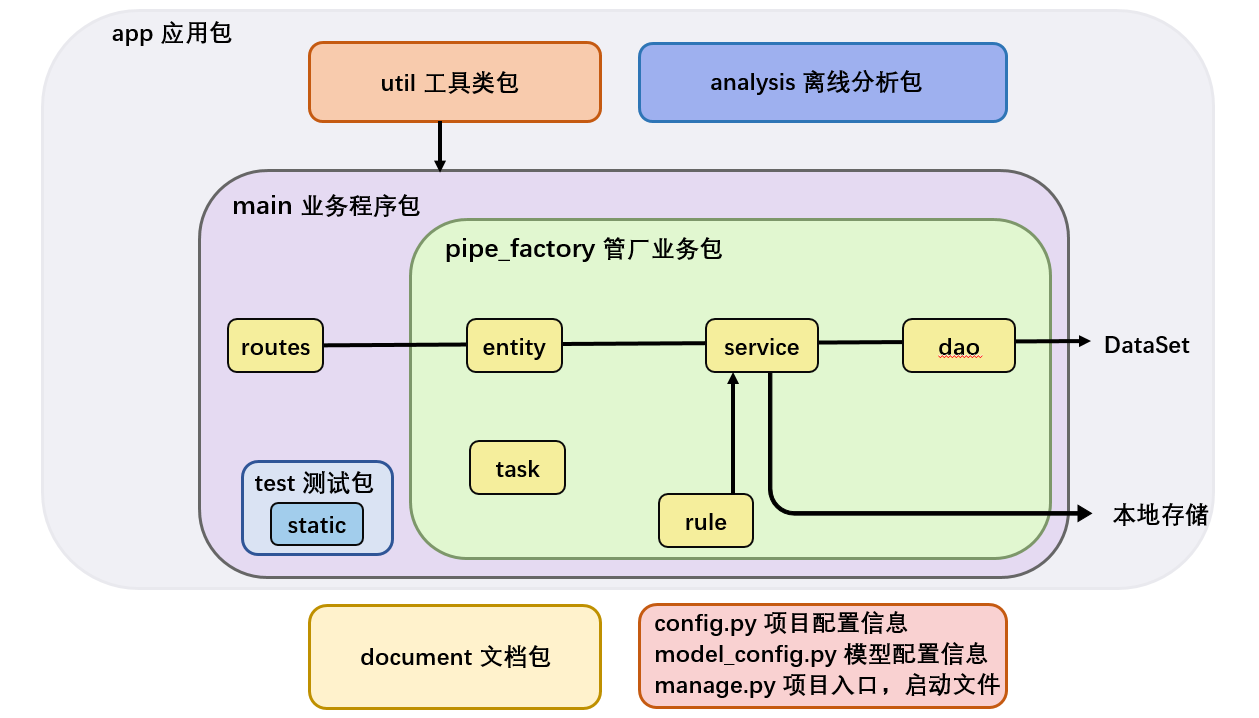
该项目的总体架构如下图所示。

图 3 系统目录结构

analysis离线分析包：暂时未使用。后续会包含本地数据分析、特征分析等任务。

util工具类包：包含各种计算工具、动态规划算法、连接池、基类等。

main业务程序包：业务程序包内按不同的厂区业务划分成不同的业务包，如steel\_factory钢厂业务包。其内部结构是一致的，均含有entity业务实体包（存放数据对象类，如订单、车次等）、dao数据连接包（存放数据库访问逻辑）、service业务逻辑包（存放业务处理逻辑，如取库存预处理与分货等）、task定时任务包、rule规则包（存放重量规则、品类规则、拼货规则等，一类规则对应一个文件）、test测试包（存放测试代码。static包指静态数据，比如测试时读入、输出的excel表）、routes路由包（存放对外输入输出接口，负责把外部数据进行相应的提取操作并转化成项目里的对象数据，项目处理逻辑完结后再由该包负责对外输出）。后续的开发如果涉及到不同的厂，业务逻辑大体不变的情况下，编写不同的rule规则即可，别的类一般不需要修改。

document文档包：包含项目的文档资料

config.py项目配置信息：包含项目运行所需要的程序参数，如数据库连接的端口信息等。

model\_config.py 模型配置信息：包含模型处理时会用到的参数，比如重量上下限，可合并的品类等参数

manage.py 项目启动入口

## 4.2业务模型设计

### 4.2.1数据结构设计

1. delivery\_sheet

DeliverySheet类，继承BaseEntity类，发货通知单主表。

一个DeliverySheet对象就是一个发货通知单。

成员变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 含义 | 初始值 | 示例 |
| rowid | string | 主键id | None | 394 |
| load\_task\_id | string | 所属车次号 | None | CC00000482 |
| car\_mark | string | 车牌号 | None | 苏B50817 |
| city | string | 城市 | None | 潍坊市 |
| delivery\_no | string | 发货通知单 | None | t3-20010227 |
| company\_id | string | 公司id | None | C000000878 |
| batch\_no | string | 批次号 | None | ba00000 |
| status | string | 状态，  FHZT00：预留，  FHZT10：已生成，  FHZT20：打印，  FHZT30：已结算 | None | FHZT30 |
| customer\_id | string | 客户id | None | cdwcnmjxyxgs |
| salesman\_id | string | 业务员id | None | 2d |
| total\_pcs | string | 总根数 | None | 8135 |
| weight | string | 重量 | None | 56257 |
| type | string | 类型 | None | 规格优先 |
| volume | string | 所占体积 | None | 0.8 |
| items | list | 发货通知单子项 | [] | [delivery\_item,...] |
| create\_time | string | 创建时间 | None | 2020-01-13 10:31:38 |
| update\_time | string | 更新时间 | None | 2020-01-13 10:31:38 |

1. delivery\_item

DeliveryItem类，继承BaseEntity类，发货通知单子项。

一个发货通知单主表可能关联多个发货通知单子项

成员变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 含义 | 初始值 | 示例 |
| rowid | string | 主键id | None | 846 |
| delivery\_no | string | 发货通知单主单号 | None | t3-20010227 |
| delivery\_item\_no | string | 子单号 | None | 1 |
| product\_type | string | 产品类型 | None | 焊管 |
| company\_id | string | 公司id | None | C000000878 |
| spec | string | 产品规格 | None | 021X2.0X6000 |
| item\_id | string | 物资代码 | None | 010021\*2.0\*6000 |
| f\_whs | string | 仓库 | None | rdzyk1 |
| max\_quantity | string | 体积限制的物资装载最大件数 | None | 20 |
| volume | string | 所占体积 | None | 0.8 |
| f\_loc | string | 垛号 | None | 2 |
| material | string | 材质 | None | Q235 |
| weight | string | 产品重量 | None | 13415 |
| quantity | string | 件数 | None | 11 |
| free\_pcs | string | 散根数 | None | 0 |
| total\_pcs | string | 总根数 | None | 2387 |
| one\_quantity\_weight | string | 单件重量 | None | 1219.55 |
| one\_free\_pcs\_weight | string | 单根重量 | None | 5.62 |
| create\_time | string | 创建时间 | None | 2020-01-13 10:31:38 |
| update\_time | string | 更新时间 | None | 2020-01-13 10:31:38 |

1. delivery\_log

DeliveryLog类，继承BaseEntity类，用来记录开单推荐时的车次结果与确认开单时的车次结果的差异。

成员变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 含义 | 初始值 | 示例 |
| rid | string | 主键id | None | 58 |
| delivery\_no | string | 发货通知单主单号 | None | 提货单4-1 |
| delivery\_item\_no | string | 子单号 | None | di\_d0910f02334f11ea95e6 |
| company\_id | string | 公司id | None | C000000878 |
| op | string | 操作标记00：删除，01：插入，02：更新 | None | 0 |
| quantity\_before | int | 修改前数量 | None | 1 |
| quantity\_after | int | 修改后数量 | None | 0 |
| free\_pcs\_before | int | 修改前散根数 | None | 0 |
| free\_pcs\_after | int | 修改手散根数 | None | 0 |
| create\_time | string | 创建时间 | None | 2020-01-10 10:22:40 |
| update\_time | string | 更新时间 | None | 2020-01-10 10:22:40 |

1. order

Order类，继承BaseEntity类，订单主表。

成员变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 含义 | 初始值 | 示例 |
| rowid | string | 主键id | None | 184 |
| order\_no | string | 订单号 | None | order\_7a85a45e964811eaa1d0 |
| items | string | 订单子项 | [] | [roder\_item,...] |
| company\_id | string | 公司id | None | C000000878 |
| customer\_id | string | 客户id | None | cdshjwzyxgs |
| salesman\_id | string | 销售员id | None | 0 |
| truck\_weight | string | 载重自定义 | None | 33000 |
| create\_time | string | 创建时间 | None | 2020-03-20 15:51:26 |
| update\_time | string | 更新时间 | None | 2020-03-20 15:51:26 |

1. order\_item

OrderItem类，继承BaseEntity类，订单子项，订单的详细信息。

成员变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 含义 | 初始值 | 示例 |
| rid | string | 主键id | None | 803 |
| order\_no | string | 主订单id | None | order\_9a342dd86a7f11ea9d3a |
| product\_type | string | 产品类型 | None | 焊管 |
| spec | string | 产品规格 | None | 060X3.0X6000 |
| quantity | string | 数量（件数） | None | 120 |
| free\_pcs | string | 散根数 | None | 0 |
| item\_id | string | 产品id | None | 015060\*3.0\*6000 |
| f\_whs | string | 仓库号 | None | hgk |
| f\_loc | string | 垛号 | None | n |
| material | string | 材质 | None | Q235BG |
| create\_time | string | 创建时间 | None | 2020-03-20 15:51:26 |
| update\_time | string | 更新时间 | None | 2020-03-20 15:51:26 |

### 4.2.2业务逻辑结构设计

业务工程实现的逻辑结构如下图所示，管厂分货业务根据实际情况可分为开单推荐、拼货推荐、开单确认三个部分。其中拼货推荐为可选业务项。所有的工程逻辑都存储在service 业务逻辑包和rules 管厂分货规则包中。

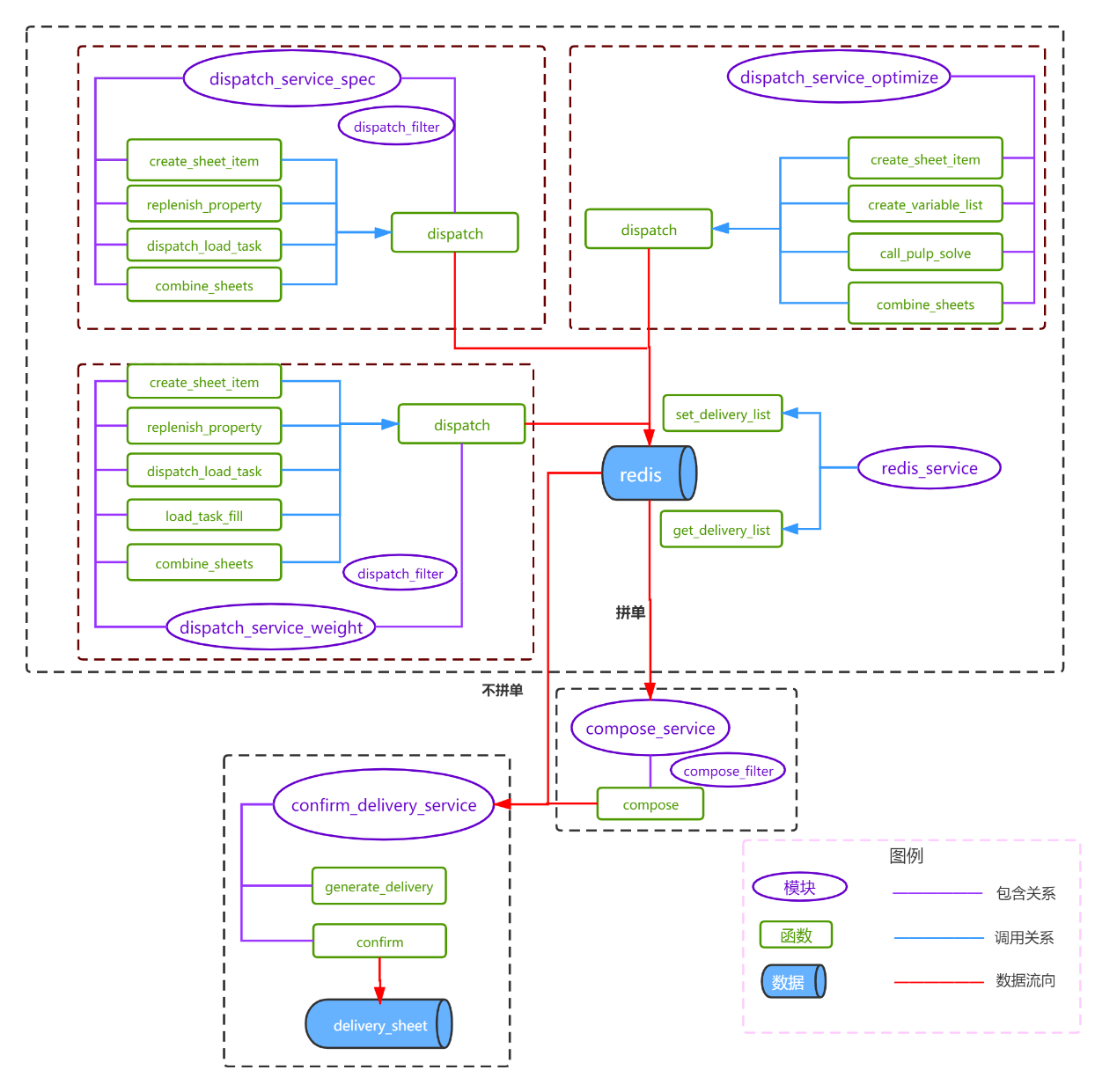
在开单推荐业务中，根据实现的功能分为三个模块：dispatch\_service\_spec、dispatch\_service\_weight、dispatch\_service\_optimize。

* dispatch\_service\_spec实现根据品种优先来进行货物分配。首先调用create\_sheet\_item() 方法将订单项转为发货通知单子单，然后利用模型过滤器dispatch\_filter 来生成发货通知单，之后调用 replenish\_property() 方法来补充发货单属性，然后通过 dispatch\_load\_task() 方法为发货单分配车次，最后调用 combine\_sheets() 方法将车次提货单按类合并，将合并后的数据存在redis缓存中，存为sheet1。
* dispatch\_service\_weight实现根据重量优先来进行货物分配。首先调用 create\_sheet\_item() 方法将订单转为发货通知子单，为了方便数据的处理，在这之后调用 create\_variable\_list() 方法进行格式转换，这里通过 call\_pulp\_solve() 调用 pulp\_solve() 中的动态规划算法，得到按重量来进行分配货物的订单列表，最后通过 combine\_sheets() 归类合并，然后将该推荐发货通知单存入redis 缓存中，存为sheet2。
* dispatch\_service\_optimize 实现根据综合情况考虑来进行分货。首先调用 create\_sheet\_item() 方法将订单转为发货通知子单，然后利用模型过滤器dispatch\_filter 来生成发货通知单，之后调用 replenish\_property() 方法来补充发货单属性，然后通过 dispatch\_load\_task() 方法为发货单分配车次，这之后为了更好的分配车辆装载的货物，调用load\_task\_fill() 方法将小管装填到大管的车次，最后调用 combine\_sheets() 方法将车次提货单按类合并，将合并后的数据存在redis缓存中，存为sheet3。

最后将根据这三种不同优先级的分货方式得到的推荐发货通知单即sheet1，sheet2，sheet3一起返回到销售处，由销售人员定夺，选择采用这三者之中的某一个方案，安排装车清单，或者销售人员有更好的方案，选择进行拼单。

如果销售人员选择进行拼单，那么进入管厂分货业务的第二个模块，即拼货推荐。在拼货推荐业务中，利用compose模块里的方法进行拼单。compose() 方法调用了过滤器compose\_filter() 根据货物的体积重量品种信息进行拼单，尽可能优化分货结果。最后将拼货的发货通知单输出给销售处。

销售处如果选择平台推荐的任一种发货通知单，那么根据选中的分货结果作为最终的发货通知单，并将数据存入数据库。如果销售人员认为该笔订单不适合分配车次，则放弃推荐的发货通知单，此时该笔订单留滞成为尾货。

图 4 业务逻辑结构图

# 5、系统详细设计

## 5.1数据采集

本模块的数据直接来自平台，销售通过输入客户所需要的货物信息，前端通过json的形式传递给分货模块，由于不同管厂品种存在差异，根据目前三家管厂的调研结果，可以得知：

|  |  |
| --- | --- |
| 管厂 | 品种 |
| 成都管厂 | 焊管、热镀方矩管、黑方矩管、螺旋焊管 |
| 郑州管厂 | 焊管、热镀锌 |
| 唐山管厂 | 焊管、热镀方矩管、黑方矩管、螺旋焊管、热镀钢塑、燃气管 |

在开单推荐过程中，我们会采集到的每次输入的订单信息并存入生产数据库db\_trans\_plan的t\_ga\_delivery\_sheet和t\_ga\_delivery\_item表中,而发货通知单信息装换成的order信息也存入db\_trans\_plan的t\_ga\_order和t\_ga\_order\_item表中；在此过程中我们还会得到用户输入的车辆上限设置或者是系统的默认设置。

在拼单推荐过程中，我们会得到开单推荐的结果和客户公司信息，得到可参与拼单的货物信息，给推荐出拼单结果。

在确认开单时，对于开单推荐和最后确认有差异的情况，模块同样会收集差异数据，t\_ga\_delivery\_log表中，用来优化分货。

以下是三个管厂的品种规格。

## 5.2接口设计

### 5.2.1 开单推荐接口

1. 测试接口链接:[http://192.168.1.27:9238/order](http://192.168.1.44/#/system/goods-allocation/goods-allocation/billOfManage)
2. 功能说明：对客户需求列表上所有单子进行总量、价格的计算，同时利用模型对开单列表数据根据各物资的重量与车载重自动匹配，生成对应的车次和批次。
3. 接口说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **请求协议** | **请求方式** | **编码格式** | **返回格式** |
| HTTP | POST | UTF-8 | JSON |

1. 接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名字** | **类型** | **是否必须** | **默认值** | **说明** |
| company\_id | 字符型 | 是 | 无 | 公司id |
| customer\_id | 字符型 | 是 | 无 | 客户id |
| salesman\_id | 字符型 | 是 | 无 | 业务员id |
| product\_type | 字符型 | 是 | 无 | 品名代码 |
| spec | 字符型 | 是 | 无 | 规格 |
| quantity | 整型 | 是 | 无 | 件数 |
| free\_pcs | 整型 | 是 | 0 | 散根数 |
| item\_id | 字符 | 是 | 无 | 子项id |
| material | 字符 | 是 | 无 | 材质 |
| f\_loc | 字符 | 是 | 无 | 垛位 |
| f\_whs | 字符 | 是 | 无 | 仓库 |

1. 返回结果：

|  |  |  |
| --- | --- | --- |
| **返回字段** | **类型** | **说明** |
| result.code | 字符型 | 返回状态码 |
| result.message | 字符型 | 返回提示信息(成功、失败) |
| sheets | 对象 | 推荐的发货通知单 |

### 5.2.2 开单确认接口(对比分析保存差异信息)

1. 测试接口连接:[[http://192.168.1.27:9238/](http://192.168.1.44/#/system/goods-allocation/goods-allocation/billOfManage)confirm](http://192.168.1.44/#/system/goods-allocation/goods-allocation/billOfManage)
2. 功能说明：对模型计算完成的发货通知单进行手动调整重量、件数、批次，生成新的提货单或进行修改，并对比分析保存差异信息。
3. 接口说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **请求协议** | **请求方式** | **编码格式** | **返回格式** |
| HTTP | POST | UTF-8 | JSON |

(4)接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名字** | **类型** | **是否必须** | **默认值** | **说明** |
| delivery\_no | 字符型 | 是 | 无 | 发货通知单主单号 |
| delivery\_item\_no | 字符型 | 是 | 无 | 子单号 |
| product\_type | 字符型 | 是 | 无 | 产品类型 |
| company\_id | 字符型 | 是 | 无 | 公司id |
| spec | 字符型 | 是 | 无 | 产品规格 |
| item\_id | 字符型 | 是 | 无 | 物资代码 |
| f\_whs | 字符型 | 是 | 无 | 仓库 |
| max\_quantity | 字符型 | 是 | 无 | 体积限制的物资装载最大件数 |
| volume | 字符型 | 是 | 无 | 所占体积 |
| f\_loc | 字符型 | 是 | 无 | 垛号 |
| material | 字符型 | 是 | 无 | 材质 |
| weight | 整型 | 是 | 无 | 产品重量 |
| quantity | 整型 | 是 | 无 | 件数 |
| free\_pcs | 整型 | 是 | 0 | 散根数 |
| total\_pcs | 整型 | 是 | 无 | 总根数 |

(5)返回结果：

|  |  |  |
| --- | --- | --- |
| **返回字段** | **类型** | **说明** |
| result.code | 字符型 | 返回状态码 |
| result.message | 字符型 | 返回提示信息(成功、失败) |

### 5.2.3拼单推荐接口

(1)测试接口连接:[[http://192.168.1.27:9238/](http://192.168.1.44/#/system/goods-allocation/goods-allocation/billOfManage)compose](http://192.168.1.44/#/system/goods-allocation/goods-allocation/billOfManage)

(2)功能说明：根据在开单列表中选中的数据选择该客户的暂存历史通知单，利用模型会自动将核实的通知单重新进行推荐，生成新的重量、价格、和车次号，同时支持业务人员人工进行调整。

(3)接口说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **请求协议** | **请求方式** | **编码格式** | **返回格式** |
| HTTP | POST | UTF-8 | JSON |

(4)接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名字** | **类型** | **是否必须** | **默认值** | **说明** |
| delivery\_no | 字符型 | 是 | 无 | 发货通知单主单号 |
| delivery\_item\_no | 字符型 | 是 | 无 | 子单号 |
| product\_type | 字符型 | 是 | 无 | 产品类型 |
| company\_id | 字符型 | 是 | 无 | 公司id |
| spec | 字符型 | 是 | 无 | 产品规格 |
| item\_id | 字符型 | 是 | 无 | 物资代码 |
| f\_whs | 字符型 | 是 | 无 | 仓库 |
| max\_quantity | 字符型 | 是 | 无 | 体积限制的物资装载最大件数 |
| volume | 字符型 | 是 | 无 | 所占体积 |
| f\_loc | 字符型 | 是 | 无 | 垛号 |
| material | 字符型 | 是 | 无 | 材质 |
| weight | 整型 | 是 | 无 | 产品重量 |
| quantity | 整型 | 是 | 无 | 件数 |
| free\_pcs | 整型 | 是 | 0 | 散根数 |
| total\_pcs | 整型 | 是 | 无 | 总根数 |

(5)返回结果：

|  |  |  |
| --- | --- | --- |
| **返回字段** | **类型** | **说明** |
| result.code | 字符型 | 返回状态码 |
| result.message | 字符型 | 返回提示信息(成功、失败) |
| delivery\_list | 对象 | 推荐拼单的发货通知单 |

## 5.3 核心模块详细设计

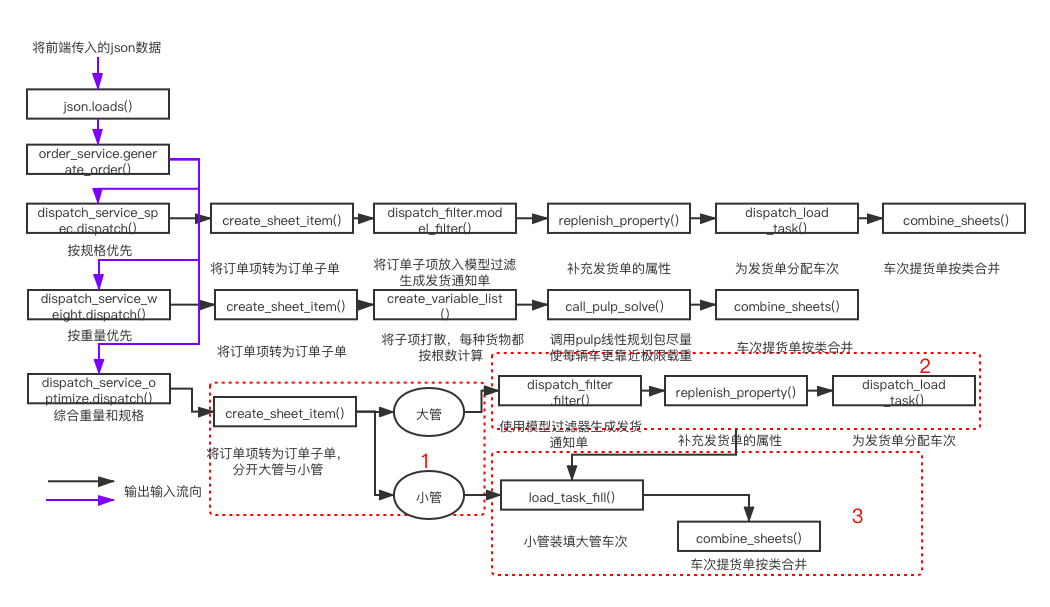
****开单推荐模块为该系统的核心模块，该模块利用不同的规则和算法通过销售给出的用户订单生成三种发货通知单，交由销售人员决定最终结果。该模块详细设计如下图所示。

图 5开单推荐模块逻辑图

## 5.4 背包算法设计

管厂分货采用动态规划中的0-1背包实现的求解算法，流程大致分为：

* 进行（重量+体积）检测，拆分录入的超重或超体积的订单项；
* 将所有订单子项进行动态规划中的二维0-1背包计算，设背包最大体积为，最大载重为，放入背包的每个子项体积为，重量为，根据需求，最终背包的价值可看成总重量。体积占比约束是重量占比约束是,目标函数是。

其中主函数为dynamic\_programming(number,capacity,volume,weight\_cost),其中number表示子项数量，capacity表示背包重量上限，volume表示背包体积上限，weight\_cost表示（子项重量，子项体积，子项价值），最终返回结果序列，放入背包标记为1，不放入背包则标记为0，根据下标取出选中的子项组合，该算法demo如下图所示。

volume = 1.18 #车的剩余空间  
number = 6 #子项娄量  
capacity = 35 #最大载重  
  
# 每个tuple表示(重量，体积，价值=重量)  
weight\_cost = [(16, 0.4, 16), (9, 0.4, 9), (3, 0.2, 3), (1, 0.2, 1), (8, 0.2, 8), (9, 0.4, 9)]  
  
#bestvalue是最终拼完的重量，result 是个list,标志哪项被拼了,这个例子中为[1, 1, 0, 0, 1, 0],即1,2,5被选中  
bestvalue, result = dynamic\_programming(number, capacity, volume, weight\_cost)  
print(bestvalue)  
print(result)

图 6背包算法demo

## 5.5代码管理

一、项目Git地址：<https://git.jczh56.com/models/gc-goods-allocation>

二、分支说明:

1. dev\_1.0为项目主发布分支，在该分支上构建开发分支

2. dev\_steel为钢厂开发分支，

3. dev\_pipe为管厂开发分支

三、开发分支管理规范

1. 开发新功能时选择钢厂或管厂分支拉到本地进行开发；

2. 在开发分支进行功能开发和测试，同步提交相关文档；

3. 测试完成后准备发布时将代码合并到dev\_1.0并打上对应的版本tag；

# 附录

## A 工具描述

util包db\_pool.py数据库连接池

connection（pymsql中的数据库连接对象）统一存放在该工具模块中

## B 基类描述

BaseEntity类

位于app.main.entity.base\_entity.py，实体类的基类，子类继承可调用或重写以下通用方法，当前工程中类继承关系如下图。



包含方法as\_dict、get\_attr\_str、set\_attr

as\_dict将对象（成员变量）转换为dict数据类型返回

get\_attr\_str读取类成员变量，将其统一转换为string数据类型

set\_attr根据传入的对象设置成员变量，将传入的dict数据类型转换为成员变量