**日照钢厂调度中心分货**

**测试文档**

**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** |
| 2020.05.22 | 刘伽椰  廖家俊 | 0.1 | 新建 | 框架设计，内容补充 |
| 2020.05.26 | 廖家俊 | 0.2 | 修改 | 内容补充 |

**目录**

[1测试总述 4](#_Toc41507212)

[1.1测试目的 4](#_Toc41507213)

[1.2测试内容 4](#_Toc41507214)

[2模型指标测试 4](#_Toc41507215)

[2.1测试内容 4](#_Toc41507216)

[2.2重量最大化指标测试 4](#_Toc41507217)

[2.2.1 测试用例设计 4](#_Toc41507218)

[2.2.2测试时间与配置 5](#_Toc41507219)

[2.2.3 测试结果 5](#_Toc41507220)

[2.2.4 测试结果分析 6](#_Toc41507221)

[2.3优先发运指标测试 6](#_Toc41507222)

[2.3.1 测试用例设计 6](#_Toc41507223)

[2.3.2测试时间与配置 7](#_Toc41507224)

[2.3.3测试结果 7](#_Toc41507225)

[2.3.4 测试结果分析 8](#_Toc41507226)

[3 功能测试 9](#_Toc41507227)

[3.1测试内容 9](#_Toc41507228)

[3.2 测试甩货数据是否可再分 9](#_Toc41507229)

[3.2.1 测试用例设计 9](#_Toc41507230)

[3.2.2测试时间与配置 9](#_Toc41507231)

[3.2.3测试结果 9](#_Toc41507232)

[3.2.4测试结果分析 10](#_Toc41507233)

[3.3 测试分货结果稳定性 10](#_Toc41507234)

[3.3.1 测试用例设计 10](#_Toc41507235)

[3.2.2测试时间与配置 10](#_Toc41507236)

[3.2.3测试结果 10](#_Toc41507237)

[3.2.4测试结果分析 10](#_Toc41507238)

[4 性能测试 11](#_Toc41507239)

[4.1测试内容 11](#_Toc41507240)

[4.2 程序运行效率测试 11](#_Toc41507241)

# 1测试总述

## 1.1测试目的

本测试主要目的为评测分货模型的业务要求满足程度及进行分货效果评估。

## 1.2测试内容

本测试主要包含模型指标测试、功能测试、性能测试等三个方向的测试。模型指标测试主要是指分货模型的各优化目标是否已经满足。功能测试主要是指分货模型的功能正确性测试。性能测试即对分货程序运行效率的测试。

# 2模型指标测试

## 2.1测试内容

**(1) 重量最大化目标是否满足**

本项测试的测试目标为检测程序是否在分货过程中考虑了车次重量最大化目标，该测试包含两个方面：

* 分货时是否遵循单车重量最大化原则，使每辆车的运力利用达最大化
* 分货时是否遵循整体运载重量最大化原则，即使得分货结果中非甩货货物总重量最大化

**(2) 优先发运机制可用性及其满足程度**

本项测试将验证分货程序的“优先发运机制可用性及其满足程度”，该测试包含两个方面：

* 验证分货程序是否会优先分出在“优先发运”标签有值的货物。
* 在本项目的业务场景中， 具有“客户催货”的优先发运标签的货物是优先于其他货物进行分货的，本项测试将验证这一点是否已实现。

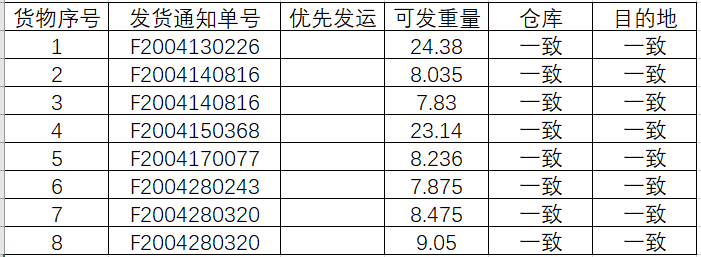
## 2.2重量最大化指标测试

### 2.2.1 测试用例设计

测试用例抽取自5月7日日钢库存明细表，针对测试需要进行了一些修改。测试用例的设计思路如下：

1. 选取相互可拼货的品种，排除测试用例间存在不可拼货情况。
2. 将测试用例的出库仓库、入库仓库、运输终点均设为一致，排除仓库间不可拼货、运输终点跨区县不可拼货情况。
3. 测试用例之间的“优先发运”标签下只存在空值。
4. 测试用例将使得“读入数据顺序优先”与“车次重量最大化优先”。
5. 除此上述提及的修改以外，不对数据做更多的修改。

依此设计思路选取的测试用例如下，省略的部分数据项可根据发货通知单号查询可得。



本次测试的重量最大化原则包含两个层次：一为单车重量最大化，使每辆车的运力利用达最大化；二为整体运载重量最大化，即使得分货结果中非甩货货物总重量最大化。依照此两原则，人工构造的预期分货结果如下：



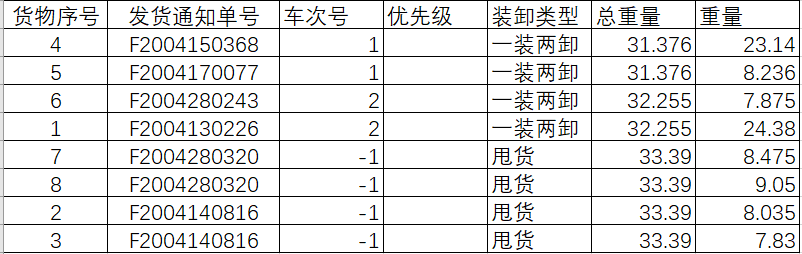
### 2.2.2测试时间与配置

本项测试于2020年5月25日 进行，使用gc-goods-allocation 项目dev\_1.0分支2020年5月23日05:48版本，版本提交SHA值为d148168ca4b8a734c4a8ed8bbb79b3a721aa122b。

### 2.2.3 测试结果

本项测试应达到的效果为：测试所得分货结果与“重量最大化满足优先”下的预期分货结果一致或取得了等同的效果。

下图为本次测试由程序输出的分货结果，省略的部分数据项可根据发货通知单号查询可得。



### 2.2.4 测试结果分析

程序分货结果未能合乎预期，在单车重量最大化与整体运载重量最大化指标上均未能通过测试。

## 2.3优先发运指标测试

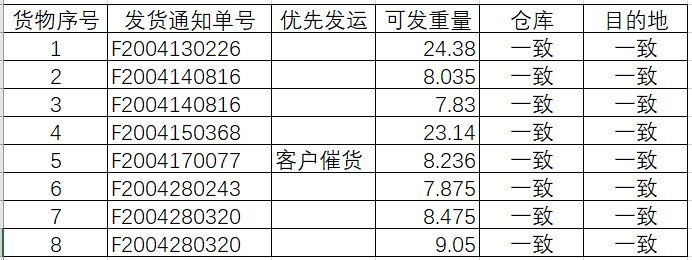
### 2.3.1 测试用例设计

**1）优先发运机制可用性测试**

测试用例抽取自5月7日日钢库存明细表，针对测试需要进行了一些修改。测试用例的设计思路如下：

1. 选取相互可拼货的品种，排除测试用例间存在不可拼货情况。
2. 将测试用例的出库仓库、入库仓库、运输终点均设为一致，排除仓库间不可拼货、运输终点跨区县不可拼货情况。
3. 测试用例之间的“优先发运”标签下只存在有值与空值的区别。
4. 测试用例将使得“读入数据顺序优先”、“车次重量最大化优先”与“优先发运满足优先”的分货结果不同。
5. 除此上述提及的修改以外，不对数据做更多的修改。

依此设计思路选取的测试用例如下，省略的部分数据项可根据发货通知单号查询可得。

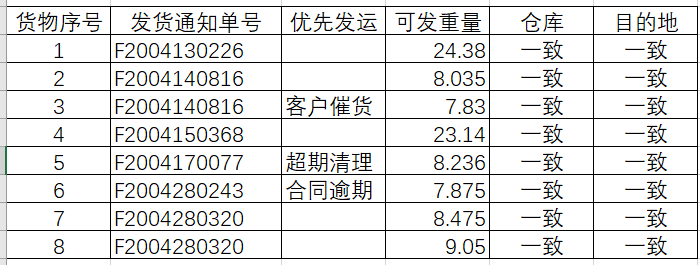


**2） 优先发运机制的业务满足程度测试**

测试用例抽取自5月7日日钢库存明细表，针对测试需要进行了一些修改。测试用例的设计思路如下：

1. 选取相互可拼货的品种，排除测试用例间存在不可拼货情况。
2. 将测试用例的出库仓库、入库仓库、运输终点均设为一致，排除仓库间不可拼货、运输终点跨区县不可拼货情况。
3. 测试用例之间的“优先发运”标签下存在“客户催货”、“超期清理”、“合同逾期”与空值四种情况。
4. 测试用例将使得“读入数据顺序优先”、“车次重量最大化优先”与“优先发运满足优先”的分货结果不同。
5. 除此上述提及的修改以外，不对数据做更多的修改。

依此设计思路选取的测试用例如下，省略的部分数据项可根据发货通知单号查询可得。

****

### 2.3.2测试时间与配置

本项测试于2020年5月21日 进行，使用gc-goods-allocation 项目dev\_1.0分支2020年5月19日03:50版本，版本提交SHA值为a09f18742b1848f586b36e90b1fadbdfa01a5a02。

### 2.3.3测试结果

**1）优先发运机制可用性测试**

本项测试应达到的效果为：测试所得分货结果与“优先发运满足优先”下的预期分货结果一致。

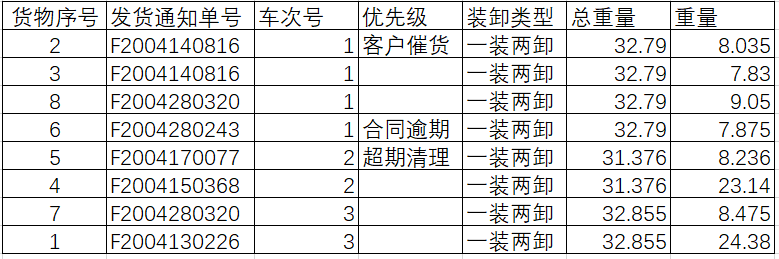
下图为本次测试由程序输出的分货结果，省略的部分数据项可根据发货通知单号查询可得。



**2） 优先发运机制的业务满足程度测试**

本项测试应达到的效果为：测试所得分货结果与“优先发运满足优先”下的预期分货结果一致，且“客户催货”标签分货优先度大于其余标签。

下图为本次测试由程序输出的分货结果，省略的部分数据项可根据发货通知单号查询可得。



### 2.3.4 测试结果分析

**1）优先发运机制可用性测试**

针对该分货结果我们可以作出如下分析：

1. 分货结果并非遵循“读入数据顺序优先”规则，例如序号1货物与序号2货物并没有拼到一起
2. 分货结果不完全遵循“车次重量最大化优先”规则，例如车次1中，并没有遵循车次重量最大化规则。
3. 分货结果遵循了“优先发运满足优先”规则。具有“客户催货”优先级的5号货物被最先考虑并完成了分货。

本项测试的结论为：程序的优先发运机制是可用的。

**2） 优先发运机制的业务满足程度测试**

针对该分货结果我们可以作出如下分析：

1. 分货结果并非遵循“读入数据顺序优先”规则，例如序号1货物与序号2货物并没有拼到一起
2. 分货结果不完全遵循“车次重量最大化优先”规则，例如车次1中，并没有遵循车次重量最大化规则（2号货物与5号货物对换，可将车次1重量最大化）。
3. 分货结果遵循了“优先发运满足优先”规则。具有“客户催货”优先级的2号货物被最先考虑并完成了分货，“合同逾期”与“超期清理”也优先于无优先级货物完成分货。。

本项测试的结论为：程序的优先发运机制是满足业务要求的。

# 3 功能测试

## 3.1测试内容

**(1) 测试甩货数据是否可再分**

本测试将对程序分货结果中的甩货数据进行正确性确认，确认其甩货货物中不存在可继续分货的数据。

**(2) 测试分货结果稳定性**

本项测试将进行分货结果稳定性测试。其具体内容为提取程序分货的结果，将该分货结果转为库存信息，重新进行分货，比对两次运行程序所得的分货结果是否产生了较大差异。

## 3.2 测试甩货数据是否可再分

### 3.2.1 测试用例设计

测试用例来自5月7日日钢库存明细表分货结果中的甩货数据。数据均未作修改。

### 3.2.2测试时间与配置

本项测试于2020年5月25日 进行，使用gc-goods-allocation 项目dev\_1.0分支2020年5月23日05:48版本，版本提交SHA值为d148168ca4b8a734c4a8ed8bbb79b3a721aa122b。

### 3.2.3测试结果

程序运行过程中返回了如下错误：





### 3.2.4测试结果分析

程序返回的该错误表明本次分货结果中没有可进行分货的数据。本项测试结果为甩货数据中不存在可再进行分货的数据。

## 3.3 测试分货结果稳定性

### 3.3.1 测试用例设计

本项测试的测试用例为提取程序分货的结果，将该分货结果转为库存信息以作测试用例。

### 3.2.2测试时间与配置

本项测试于2020年5月26日 进行，使用gc-goods-allocation 项目dev\_1.0分支2020年5月23日05:48版本，版本提交SHA值为d148168ca4b8a734c4a8ed8bbb79b3a721aa122b。

### 3.2.3测试结果

原分货结果中分出货物总重量56009吨，甩货16649吨。

将该结果转为库存信息并进行重新分货，分出货物总重量56553吨，甩货16104吨。

### 3.2.4测试结果分析

两次分货结果的差别不大，可认为该分货程序输出的分货结果具备稳定性。

# 4 性能测试

### 4.1测试内容

**（1）程序运行效率测试（库存信息存放于本地）**

本项测试将进行分货程序的运行效率测试，本项测试所用的库存信息存放在本地。本项测试也将用于作为后续库存信息存放于数据仓库相关效率测试的比较基准。效率测试主要关注程序在业务场景正常数据量下的单次运行时间。

**（2）程序运行效率测试（库存信息存放于数据仓库）**

本项测试将进行分货程序的运行效率测试，本项测试所用的库存信息存放在数据仓库。本项测试的结果将与库存信息放置在本地的测试结果作对比，以评估数据仓库接口的可用性及其对程序执行效率的影响。效率测试主要关注程序在业务场景正常数据量下的单次运行时间。

**（3）数据仓库真实数据效果测试**

本项测试将进行数据仓库中真实数据的效果效率测试。进行本测试的原因为数据仓库中的数据格式与目前工程所需要输入的数据格式不完全一致，我们将在这项测试中评估本工程对数据仓库中真实数据的分货效果，分析内容与2.2.1.(3)中的相同。

### 4.2 程序运行效率测试