

---

队伍编号	MC2409727
题号	C

---

## 基于随机森林算法的物流货量预测问题研究

### 摘要

本文主要研究了物流系统中分拣中心货量预测与人员调度问题。根据各个分拣中心历史货量数据，利用**随机森林算法**，建立起**机器学习货量预测模型**，求解出未来 30 天所有分拣中心每日每小时货量数据。最后根据**模拟退火算法**，设计出一套符合劳动规则的人员排班计划。

针对问题一，根据各个分拣中心的历史货量数据，建立机器学习预测模型。首先利用**线性插值**对数据进行预处理。考虑到货量数据具有不稳定性，我们决定采用在处理非线性问题表现突出的**随机森林算法**，并提取滞后特征、动态窗口统计量等数据作为**特征集**对模型进行训练。模型的相关系数  $R^2$  为 0.9683（日预测模型），0.9165（小时预测模型），与验证集结果高度拟合。最后使用该预测模型，对全部分拣中心未来 30 天每日每时的货量进行了预测，并填入结果表 1 和结果表 2.

针对问题二，考虑到运输线路的改变会对货量造成较大影响，我们通过引入运输线路相关参数对模型进行优化调整。通过分析历史线路的货量数据，来反应不同线路变化对分拣中心货量数据的影响。最后，利用优化后的模型进行预测，将结果保存到结果表 3 和表 4 中。

问题三在问题二预测结果的基础上，使用**MILP 算法**构建人员排班优化模型。在满足每天货量处理完成和优先使用正式工的基础上，减少总人天数并均衡实际小时人效。通过**线性规划和启发式算法**来优化模型。通过对模型分析，得出人效平衡系数为 0.88，迭代到最大次数所需的单位迭代时间为 52。最后，将得到的人员调度安排表保存在结果表 3 和表 4 中。

针对问题四，新增的正式工出勤率和连续工作日限制使得问题约束条件变得繁杂。首先通过**蒙特卡洛算法**求得初步解，选出最好的作为模拟退火的开始。**模拟退火**由于其随机性策略，可以在搜索过程中动态适应多限制条件，最终逐步趋于全局最优解，在本题目中复杂的约束环境中展现了**灵活性和鲁棒性**。以 SC60 为例，模型得出了总人天数最少且人效平衡的最优解，并将未来 30 天的排班计划记录在结果表 5 和表 6 中。

**关键词：** 货量预测 随机森林算法 MILP 算法 蒙特卡洛 模拟退

# 目录

1. 问题重述 .....	1
1.1 问题背景 .....	1
1.2 问题提出 .....	1
2 模型假设 .....	2
3 符号说明 .....	3
4 问题分析 .....	4
4.1 问题 1 的分析 .....	4
4.2 问题 2 的分析 .....	4
4.3 问题 3 的分析 .....	5
4.4 问题 4 的分析 .....	6
5 模型的建立与求解 .....	7
5.1 数据预处理 .....	7
5.1.1 异常值检验 .....	7
5.1.2 缺失值补全 .....	8
5.1.3 模型准备 .....	9
5.1.4 模型建立 .....	10
5.1.5 模型求解 .....	11
5.1.6 求解结果 .....	12
5.2 问题二的模型建立与求解 .....	14
5.2.1 模型建立 .....	14
5.2.2 模型求解 .....	16
5.2.3 求解结果 .....	17
5.3 问题三的模型建立与求解 .....	19
5.3.1 数据处理 .....	19
5.3.2 模型建立 .....	19
5.3.3 基于混合整数线性规划 MILP 算法求解模型 .....	21
5.3.4 求解结果 .....	23

5.4 问题四的模型建立与求解 .....	25
5.4.1 模型建立 .....	25
5.4.2 模型求解 .....	26
5.4.3 求解结果 .....	28
6 模型评价 .....	29
6.1 问题一模型分析 .....	29
6.1.1 误差分析 .....	29
6.1.2 灵敏度分析 .....	30
6.2 问题二模型分析 .....	31
6.3 问题三模型分析 .....	32
6.4 问题四模型分析 .....	33
6.5 模型评价 .....	33
参考文献 .....	35
附录 .....	35

## 1. 问题重述

### 1.1 问题背景

21世纪以来，随着电子商务的快速发展，物流网络的效率和可靠性对于保持企业竞争力至关重要。作为物流网络重要角色的分拣中心不仅是物流网络的中心枢纽，也是保证订单准时履约的关键环节。然而，随着在线订单数量的激增，分拣中心面临着巨大的压力，既需要处理日益增长的货量，也需要同时保持高效率和低运营成本。因此，如果能对分拣中心未来地的货量进行准确的预测，并对人员调度进行合理的安排，将大幅提高分拣中心的业务效率。

### 1.2 问题提出

**问题 1：**基于附件 1 给出的 57 个分拣中心在 2023 年 8 月 1 日到 2023 年 11 月 30 日的每日货量数据和附件 2 给出的在 2023 年 11 月 1 日到 2023 年 11 月 30 日期间的每小时货量数据，对 2023 年 12 月 1 日到 2023 年 12 月 30 日期间的每日及每小时的货量进行预测。

**问题 2：**在问题 1 的基础上，分析 57 个分拣中心过去 90 天 134 条运输线路的平均货量数据，在此基础上对预测模型进行修改。使用修改后的模型根据题中所给的 2023 年 12 月 1 日到 2023 年 12 月 30 日期间的线路，预测在此期间 57 个分拣中心每天及每小时的货量。

**问题 3：**在优先安排正式员工的基础上，以人天数尽可能少、每天的小时人效尽可能均衡为准则，为 57 个分拣在中心 2023 年 12 月 1 日到 2023 年 12 月 30 日期间的人员班次规划。

**问题 4：**基于问题 2 中的预测结果建立预测模型，在正式工出勤率不超过 85%、连续出勤日不超过 7 天的基础上，以安排的人天数尽量少，每天小时人效指标尽量均衡、正式工出勤率尽量均衡为准则，指定 SC60 在 2023 年 12 月 1 日到 2023 年 12 月 30 日的正式工与临时工的出勤计划。

## 2 模型假设

### **数据假设：**

假设提供的历史货量数据集准确无误,不存在记录错误或缺失。

### **运营假设：**

假设各分拣中心间货物运输时间和成本是固定的，不受任何未来不可预见的因素（如突发事件或供应链中断）的影响。

假设货物运输过程中不会发生任何延误或中断，所有货物都能按时准确地到达目的地。

假设所有分拣中心的运营模式、设备和流程都是一致的，不存在特定中心的特殊情況或变化。

### **人力假设：**

假设分拣中心的正式工和临时工人力资源池的规模和结构是稳定的，不受任何突发因素或外部干扰的影响。

假设分拣中心人员的工作效率是恒定的，并且在预测期内不受任何因素（如疲劳、培训或设备故障）的影响。

### 3 符号说明

符号	含义
$q_1$	上四分位数
$q_2$	下四分位数
$t$	第 $t$ 天
$y_{i,t}$	第 $t$ 天的原始货运量
$H_{ijt}$	第 $t$ 天的预测货运量
$l_{t-1}$	滞后特征
$\bar{q}_{7,t}$	时间点 $t$ 时前 7 天的动态窗口
$\bar{q}_{24,t}$	时间点 $t$ 时前 24 时内的动态窗口
$E_R$	正式工小时人效
$E_T$	临时工小时人效
$W$	权重数值
$T_0$	退火算法起始温度
$\alpha$	退火算法退火系数
$T_{\text{end}}$	退货算法终止温度

## 4 问题分析

### 4.1 问题 1 的分析

由于在 9 月到 11 月期间存在各类电商购物节以及异常天气，可能会对分拣中心的货量造成较大的变化影响，因此我们对附件 1 和附件 2 所给出的数据进行数据预处理。首先对数据汇总分类以分析是否存在数据缺失值，然后根据现有数据绘制箱型图以确定数据集中数据异常值数量。在补全缺失值和分析异常值后，使用随机森林算法来构建机器学习预测模型。考虑到，我们在数据中提取例如假日特征、滞后特征等特征值，同时将划分训练集和测试集分别用以模型的训练和评估。最后根据训练的模型，对未来 30 天的每日每时货量进行预测。

问题 1 的流程图如图 4.1 所示。

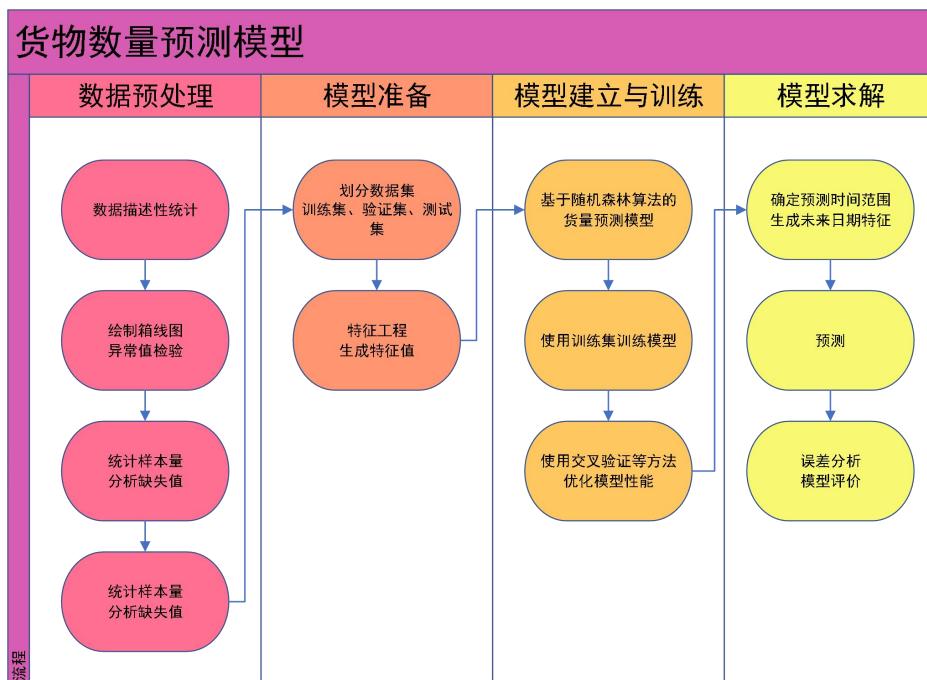


图 4.1 问题一流程图

### 4.2 问题 2 的分析

由于运输线路的变化会对各个分拣中心的货量造成显著变化，因此针对问题 2 需要根据运输线路来调整现有模型。首先，通过分析附件 3，我们可以得到过去每一条线路的历史货量数据，以此来反映未来相同线路的货量变化值。其次，通过分析附件 4，我

们可以了解未来 30 天线路发生的变化，对于新增的线路，线路所连接的分拣中心的货量都会相应增加；反之，对于取消的线路，线路所连接的分拣中心的货量则相应减少。最后，根据所计算得的每一个分拣中心变化后的货量数值来预测未来 30 天内的每天和每时的货量。

问题 2 的思路流程图如图 4.2 所示。

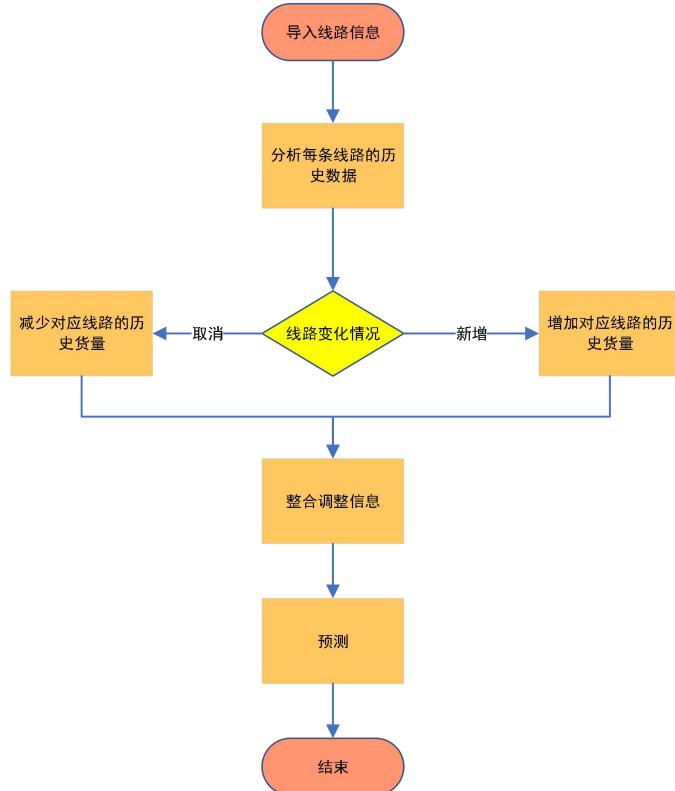


图 4.2 问题二流程图

### 4.3 问题 3 的分析

根据问题二中我们所获得的每个分拣中心货量的数据，在保证每天货量能被处理的基础上，以尽可能减少总人数天数并保持小时人小均衡作为准则，设计未来 30 天的人员分配调度方案。首先，对问题 2 中获得的结果进行整理排序。然后定义班次以及人效参数，针对每个分拣中心的每天的每一个班次来进行分析。由于是以正式工优先安排为原则，为了达到总出勤天数尽可能少的目的，临时工只有在正式员工无法满足需求的情况下才需要分配。在此基础上，再考虑每个班次的人效平衡。

问题 3 的思路流程图如图 4.3 所示。

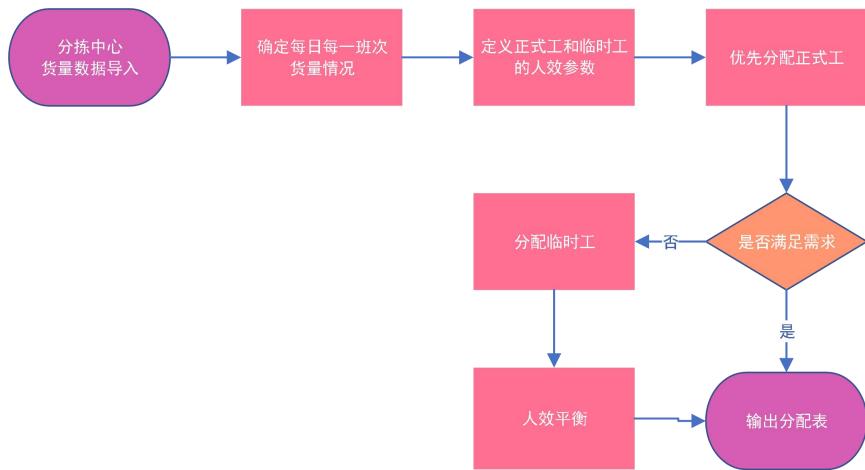


图 4.3 问题三流程图

#### 4.4 问题 4 的分析

由于正式工的人效高于临时工，因此为了保证出勤总天数最少，排班依旧首选正式工。同问题三的思路类似，在满足正式工出勤率在 85%以下且连续出勤天数不高于 7 天的前提下，若无法满足需求，再考虑安排临时工。最后再根据人效平衡来优化模型。由于涉及到的变量较多，约束条件复杂，我们优先考虑使用启发式算法来找到满足所有限制的最优解。

第 4 题的思路图如图 4.4 所示。

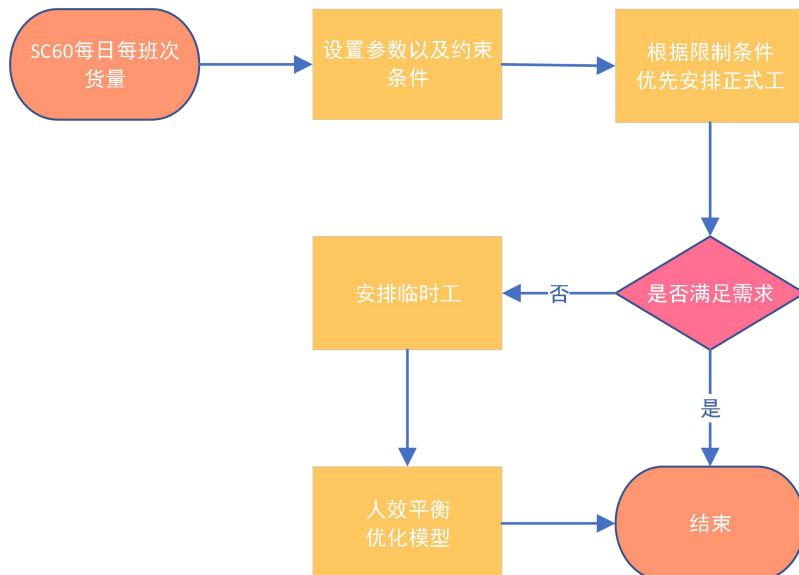


图 4.4 问题四流程图

## 5 模型的建立与求解

### 5.1 数据预处理

#### 5.1.1 异常值检验

如图 5.1 和图 5.2 所示，我们对附件 1 和附件 2 中的数据绘制了箱线图。从图 5.1 和图 5.2 中我们可以看出其中都存在异常观测值。具体而言，在部分观测时间点，货量值明显偏离该数据集的整体分布趋势，远高于数据集的正常数值水平。这种现象可能源于特殊情况下货物流通量的激增，或者也可能是由于数据采集和记录过程中的失误所致。

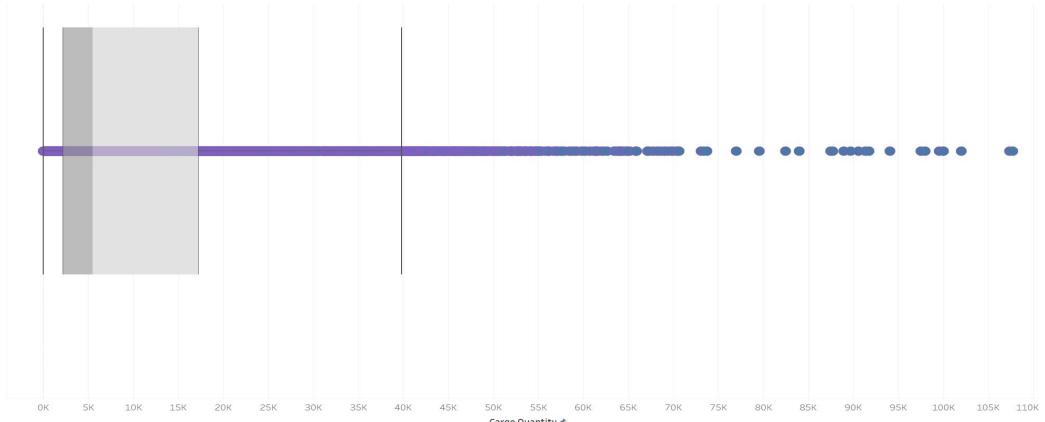


图 5.1 附件 1 箱线图

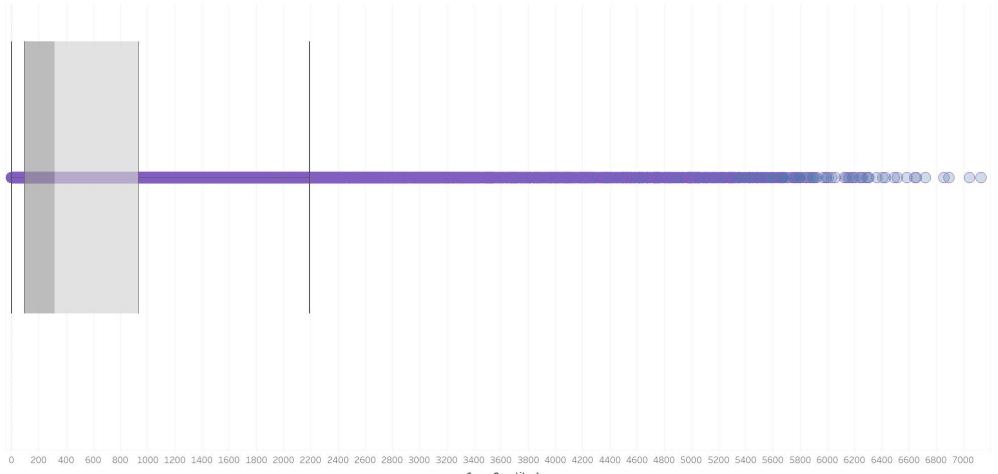


图 5.2 附件 2 箱线图

这些异常观测值的存在可能会对后续的数据分析和建模过程产生不利影响，因此有必要针对性地进行异常值处理，以提高模型的鲁棒性和可靠性。

接下来, 我们分析是否存在日期不匹配的情况。

## 5. 1. 2缺失值补全

首先, 我们分析是否存在日期不匹配导致的缺失情况。我们对数据进行分组, 分组依据为分拣中心编号和日期, 并计算每个组的记录小时数。然后, 检查哪些组的记录小时数小于 24 小时, 这表明该组在当天缺失部分记录。统计结果发现, 总共有 1086 个这样的组。

对于缺失值的补全, 我们采用线性插值方法。原因是线性插值在简值问题中是一种很好的折中选择, 能够在计算效率和插值精度之间达到相对平衡。利用线性插值, 我们可以根据相邻每小时的货量估算缺失小时的货量。



图 5.3 附件 1 缺失值分析

具体而言, 给定两个已知数据点( $t_1, Q_1(t)$ )和( $t_2, Q_2(t)$ ). 如果要估算 $t_0$ 处的值 $Q_0$ , 其中 $t_1 < t_0 < t_2$ , 则根据线性插值的假设有:

$$Q_0(t) = Q_1(t) + \frac{Q_2(t) - Q_1(t)}{t_2 - t_1} (t_0 - t_1) \quad (5-4-1)$$

插值完成后, 对于每个分拣中心  $i$ , 检验 ( $t$ ) 在  $t = 0, 1, 2, \dots, 23$  时刻是否均有记录。

经过上述处理，我们获得了对于每个分拣中心在 24 小时内的完整货物量记录，为后续建模分析奠定了数据基础。形式上，我们现在拥有  $n$  个时间序列  $\{Q_1(t), Q_2(t), \dots, Q_n(Tt)\}$ ，其中  $T = 0, 1, 2, \dots, 23$ ，可用于进一步的数据分析与建模。

## 5. 1. 3模型准备

### 1.数据划分

为了预测每个分拣中心未来 30 天每天及每小时的货量，我们首先需要对提供的数据进行数据集划分，将每个数据集分别划分成训练集和验证集，用训练集  $D_{train}$  进行训练，用验证集  $D_{val}$  验证预测的准确性，考察模型泛化性，划分结果如下表所示：

表格 5.1 数据集划分结果

数据类型	总数据时段	$D_{train}$ 时段	$D_{val}$ 时段
日级货量预测	过去 4 个月	最初 3 个月	最后 1 个月
小时级货量预测	过去 30 天	最初 20 天	最后 10 天

### 2.特征选择

在进行时间序列分析时，我们的目标是从每天和每小时的货量数据中抽取并利用信息，以预测未来的货量需求。

#### (1)日货量预测

在进行日级货量预测时，我们选择出能够反映出数据的时序性和近期的货量趋势的特征，通过分析图 x 我们考虑选择以下特征：

- ① 基本时间特征：年  $y_t$ 、月  $m_t$ 、日  $d_t$  以及星期  $w_t$ 。
- ② 滞后特征：前一天的货量值  $l_{t-1}$  代表时间点  $t - 1$  的货量值。
- ③ 促销活动特征  $p_t$ ：是一个布尔变量，当  $p_t = 1$  表示在时间点  $t$  有促销活动，若没有该活动则  $p_t = 0$ 。
- ④ 动态窗口特征：使用  $\bar{q}_{7,t}$  表示在时间点  $t$  时前 7 天内的平均货量。

综合以上特征，模型日货量预测的输入特征向量  $X_t$  的表达式为：

$$X_t = [y_t, m_t, d_t, w_t, l_{t-1}, p_t, \bar{q}_{7,t}] \quad (5-2-1)$$

## (2) 小时货量预测

对于小时级货量预测，需要更细致的时间粒度数据来捕捉短期内的变动。通过分析图 x 我们考虑选择以下特征：

- ① 基本时间特征：年  $y_t$ 、月  $m_t$ 、日  $d_t$ 、星期  $w_t$  以及 小时  $h_t$ 。
- ② 滞后特征：前一小时的货量  $h_{t-1}$ ，表示时间点  $t - 1$  的货量。
- ③ 促销活动特征  $p_t$ ：与日货量预测类似，表示是否有促销活动。
- ④ 动态窗口特征：使用  $\bar{q}_{24,t}$  表示在时间点  $t$  时前 24 小时内的平均货量。

综合以上特征，模型小时货量预测的输入特征向量的表达式为：

$$X_t = [y_t, m_t, d_t, w_t, h_{t-1}, p_t, \bar{q}_{24,t}] \quad (5-2-2)$$

## 5.1.4 模型建立

随机森林由多个决策树组成，每棵树都在数据的一个随机子集上训练，并对其预测结果求取平均值以提高总体预测的准确性。随机森林适合处理大规模数据集和复杂的非线性关系，这使其能够准确预测分拣中心每天及每小时的货量。

首先设定数据及的时间范围，基于数据集的最后一个观测日期 2023 年 12 月 1 日，我们将预测窗口设定为之后的 30 天，即 2023 年 12 月 2 日至 2023 年 12 月 31 日，以满足对未来一个月日货量的预测需求。

我们定义随机森林模型  $\mathcal{F}$  定义如下，其中  $N$  是决策树的数量，每棵树  $f_i$  在随机选择的子集上进行训练：

$$\mathcal{F}(X_t) = \frac{1}{N} \sum_{i=1}^N f_i(X_t; \theta_i) \quad (5-2-3)$$

定义模型第  $t$  天的实际货量  $y_{i,t}$  与模型的预测货量  $\hat{y}_{i,t}$  之间的误差  $\epsilon_{i,t}$  如下：

$$\epsilon_{i,t} = y_{i,t} - \hat{y}_{i,t} \quad (5-2-4)$$

结合特征的线性组合以及误差的校正之后，模型公式的定义如下：

$$y_{i,t} = \beta_0 + \beta_1 X_{i,t-1} + \cdots + \beta_p X_{i,t-p} - \theta_1 B \epsilon_{i,t-1} - \cdots - \theta_q B^q \epsilon_{i,t-q} \quad (5-2-5)$$

其中， $X_{i,t-1}$  到  $X_{i,t-p}$  表示滞后特征，包括前一天的货量以及时间窗口的平均货量。

$B$  是后裔算子，使用  $By_{i,t} = y_{i,t-1}$  表示前一期的值。分别表示模型特征的权重和误差的权重。

我们使用训练集  $D_{\text{train}}$  训练随机森林模型  $\mathcal{F}$ ，将得到的模型参数代入公式：

$$(\beta_0, \beta_1, \dots, \beta_p, \theta_1, \dots, \theta_q) = \text{TrainModel}(\mathcal{F}, D_{\text{train}}) \quad (5-2-6)$$

## 5.1.5 模型求解

随机森林中决策树的构建过程包括以下几个步骤：

### 1. 数据采样：

使用 Bootstrap 方法从原始训练数据中抽取样本点，可以重复选择同一样本点，

再使用这

些样本点构建出决策树，该过程形式化如下：

$$D_i = \text{Bootstrap}(D) \quad (5-2-7)$$

### 2. 特征选择：

在构建每棵树的每个决策节点时，将基于随机抽样选择特征。我们根据预测任务的不同级别从相应的特征集中随机选择特征子集。

对于日级预测，特征选择为：

$$\theta_{\text{daily},i} = \text{Random\_Select}(y_t, m_t, d_t, w_t, \ell_{t-1}, p_t, \bar{q}_{7,t}) \quad (5-2-8)$$

对于小时级预测，特征选择为：

$$\theta_{\text{hourly},i} = \text{Random\_Select}(y_t, m_t, d_t, w_t, h_{t-1}, p_t, \bar{q}_{24,t}) \quad (5-2-9)$$

### 3. 最优节点分割点搜索：

该搜索过程包括以下几个步骤：

在每个决策节点上，我们选用基尼不纯度选择最佳分割点。基尼不纯度是衡量节点不纯度的一个常用指标，它计算的是从该节点随机选取两个元素属于不同类别的概率。对于给定的节点  $\mathcal{N}$ ，基尼不纯度  $G(\mathcal{N})$  可以定义为：

$$G(\mathcal{N}) = 1 - \sum_{k=1}^K p_k^2 \quad (5-2-10)$$

选择基尼不纯度降低最多的特征  $X$  和分割点  $s$ ，根据特征  $X$  的值是否大于阈值  $s$ ，将节点分割为两个子节点，使用  $p_{k,\text{left}}$ ,  $p_{k,\text{right}}$  表示左右节点中属于第  $k$  类的数据点的比例，分别计算出两个子节点的基尼不纯度，然后计算分割前后的基尼不纯度差  $\Delta G$ ：

$$G(\mathcal{N}_{\text{left}}) = 1 - \sum_{k=1}^K p_{k,\text{left}}^2 \quad (5-2-11)$$

$$G(\mathcal{N}_{\text{right}}) = 1 - \sum_{k=1}^K p_{k,\text{right}}^2 \quad (5-2-12)$$

$$\Delta G = G(\mathcal{N}) - \left( \frac{N_{\text{left}}}{N} G(\mathcal{N}_{\text{left}}) + \frac{N_{\text{right}}}{N} G(\mathcal{N}_{\text{right}}) \right) \quad (5-2-13)$$

比较各个节点的  $\Delta G$  选择最优的分割点，该节点的子节点继续迭代分割，重复之前的步骤，直到达到我们的预设停止条件，即节点内样本数少于最小样本分割数，迭代结束。

#### 4. 建立森林，得出预测结果

迭代结束后，模型构建出多棵决策树。将每棵树的预测结果  $f_i(X_t)$  累加后求取平均值，得到随机森林的最终预测结果：

$$\hat{y}_t = \frac{1}{N} \sum_{i=1}^N f_i(X_t) \quad (5-2-14)$$

### 5.1.6 求解结果

#### 1. 输入参数

我们确定每日货量和小时货量的预测的特征取值分别如表 2 和表 3 所示：

表格 5.2 日货量特征预测取值

分拣中心	年	月	日	星期	滞后特征	动态窗口特征	促销活动特征
SC1	2023	12	1	4	45707	44043.3	0
SC2	2023	12	4	7	8600	7527.7	0
SC3	2023	12	7	3	12665	12448.4	1
SC4	2023	12	10	6	21342	20852.1	1
SC5	2023	12	13	2	21505	21142.4	0
SC6	2023	12	15	4	28131	29097	0

表格 5.3 小时货量特征预测取值

分拣中心	年	月	日	小时	星期	滞后特征	动态窗口特征	促销活动特征
SC5	2023	12	10	18	6	2086	972.0	1
SC5	2023	12	10	20	6	2776	1223.6	1
SC5	2023	12	10	22	6	2483	1135.3	1
SC10	2023	12	20	8	2	4228	1642	0
SC10	2023	12	20	10	2	4032	1625	0
SC10	2023	12	20	12	2	4123	1636	0

## 2.每日货量预测结果

我们将所选择的特征输入已训练好的随机森林回归模型，对每个分拣中心在预测时间范围内的日货量进行预测。每日货量的实际值和预测值的残差图如图 5.4:

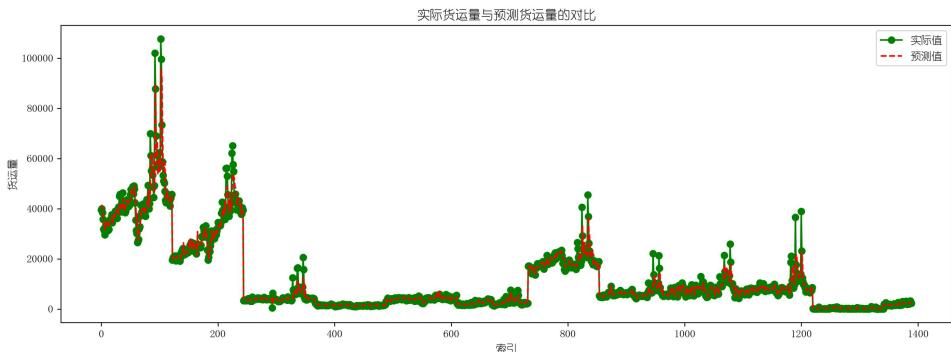


图 5.4 残差图

从图中我们可以观察到，模型的预测值和实际值大多数情况下是相近的。残差的随机分布表明模型没有系统性的偏误，这代表我们的模型良好的拟合。

## 3.小时货量预测结果

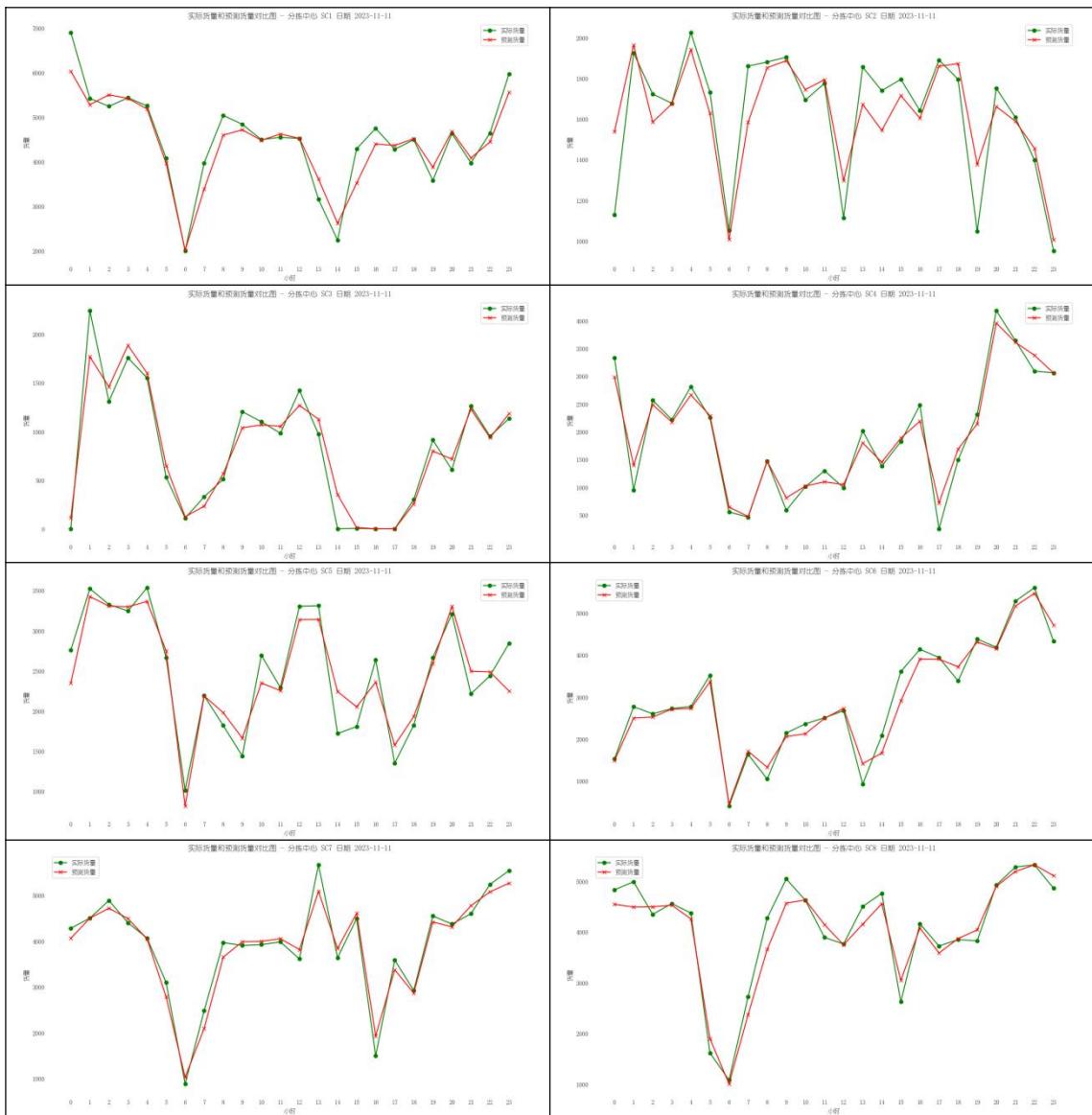


图 5.5 分拣中心 SC1~SC8 2023 年 11 月 11 日 货物实际与预测量对比

我们对分拣中心 SC1 至 SC8 在 2023 年 11 月 11 日的小时货量进行了实际与预测值的对比分析，并通过折线图展现了两者之间的关系。分析结果显示，这八个分拣中心的实际货量与预测货量的折线走势紧密，这表明预测与实际情况高度一致，反映出模型具有良好的拟合效果。

## 5.2 问题二的模型建立与求解

### 5.2.1 模型建立

#### 1. 数据准备

对于附件 3 给出的内容，我们提取出每条运输线路的平均货量  $Q_i$ ,  $i$  代表不同的运输线路，一共有  $N$  条线路。然后，我们使用附件 4 中的数据识别未来可能发生的运输线路变化。具体来说，我们关注两类线路变化，一是新增的运输线路集合  $A$ ，二是即将取消的运输线路集合  $R$ 。

## 2. 模型特征的构建和影响计算

为了估计新增运输线路可能带来的货量影响，我们首先计算这些线路在历史上的平均货量，预期货量值的估算公式为：

$$Q_{\text{inc}} = \sum_{i \in A} Q_i \quad (5-3-1)$$

其中， $Q_i$  是历史上新增线路  $i$  的平均货量值， $Q_{\text{inc}}$  表示所有新增运输线路的预期总货量值。

对于即将取消的运输线路，我们需要计算它们对总货量的减少影响。通过汇总这些线路的历史平均货量，得到取消运输线路的减少货量值的估算公式为：

$$Q_{\text{dec}} = - \sum_{j \in R} Q_j \quad (5-3-2)$$

其中， $Q_j$  是历史上被取消线路  $j$  的平均货量值， $Q_{\text{dec}}$  表示取消运输线路的减少货量值。

对每个分拣中心  $k$ ，将相关的新增和取消线路货量进行汇总，估计出每个中心的总货量调整值：

$$\Delta Q_k = Q_{\text{inc}}^k + Q_{\text{dec}}^k \quad (5-3-3)$$

最终每个分拣中心的调整后货量由原始预测货量加上调整值得出：

$$Q_{\text{adjusted},k} = Q_{\text{base},k} + \Delta Q_k \quad (5-3-4)$$

这里  $Q_{\text{adjusted},k}$  是模型中未考虑线路变更的原始预测货量。

## 3. 预测调整

根据计算出的  $\Delta Q_k$ ，调整未来 30 天中按每天和每小时进行预测的货量值。

## 5. 2. 2模型求解

### 1. 运输线路变动分析

我们分析了未来 30 天内运输线路的变动情况，这些变化将直接影响我们的货运预测。具体来说：

(1)新增线路意味着我们可能需要在这些特定的分拣中心增加处理能力，以应对预期增加的货运需求。表 x 列出了预计增加相应始发和到达分拣中心之间的货运流量的运输线路：

表格 5. 4：新增运输线路

始发分拣中心	到达分拣中心
SC5	SC4
SC31	SC9

(2)取消这些线路可能意味着相应的货物将需要通过其他途径或线路进行分配，这可能影响到相关分拣中心的运营效率。表 x 列出了预计增加相应始发和到达分拣中心之间的货运流量的运输线路：

表格 5. 5：取消运输线路

始发分拣中心	到达分拣中心	货量	始发分拣中心	到达分拣中心	货量
SC36	SC8	97	SC55	SC7	128
SC19	SC15	336	SC24	SC5	138
SC4	SC15	15	SC28	SC4	8
SC51	SC15	12	SC18	SC51	14
SC36	SC47	133	SC54	SC25	182
SC1	SC25	254	SC61	SC10	228
SC2	SC19	356	SC39	SC60	119

### 2. 特征修正

根据上述新增和取消的运输线路情况,我们需要相应调整相关分拣中心的货运量特征，具体而言：

(1)对于新增的线路，为了反应新增线路可能带来的额外货物流入，我们将在受影响分拣中心的预期货运量中增加该线路的预期平均货运量。

(2)对于取消的线路，为了反映由于线路取消可能导致的货运量减少，我们将从受影响分拣中心的预期货运量中减去该线路的历史平均货运量。

### 3. 平均分配调整量

对于每日减少或增加的货运量，我们将这一日调整量平均分配到 24 小时中。这样可以避免任何特定小时内的预测结果出现极端波动，从而保持预测的连续性和可靠性。

### 5. 2. 3 求解结果

我们参考了附件 3 中提供的历史平均货运量数据，以估算新增线路可能带来的额外货运量，以及取消线路可能导致的货运量减少，得到的结果如表 5.6：

表格 5.6：预测货量调整（部分）

分拣中心	新增货量	减少货量	货量调整
SC1	0	-262	-262
SC2	0	-356	-356
SC3	127	0	127
SC4	0	-15	-15
SC5	28	0	28
SC6	69	0	69

使用调整后的特征数据，结合已经训练好的模型，预测未来 30 天的每小时和每日的货运量。

#### 1. 日预测调整结果

图 5.4 展示了分拣中心在 12 月份的每日预测货量的调整结果，每个柱子的高度代表了特定日期的预测货量。

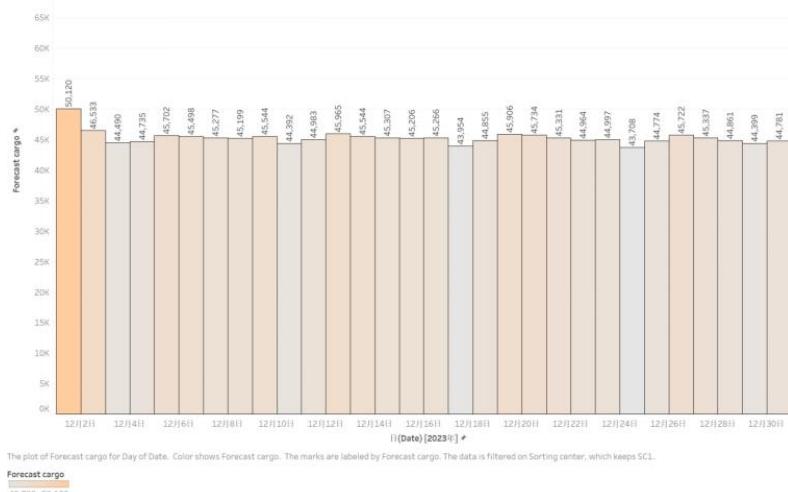


图 5.6 SC1 分拣中心 2023 年 12 月 1 日到 2023 年 12 月 30 日预测货量数据可视化图

## 2. 小时预测调整结果

根据模型预测结果，我们得到了分拣中心 SC1 和 SC2 在 2023 年 12 月 15 日和 12 月 16 日的每小时预测货量，如图 5.5 到 5.8 所示：



图 5.7 SC1 在 2023 年 12 月 15 日的每时预测货量

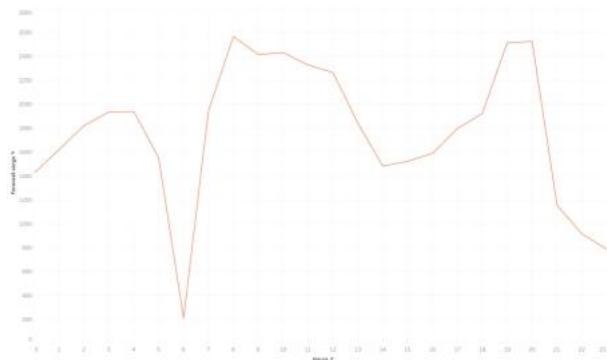


图 5.8 SC1 在 2023 年 12 月 16 日的每时预测货量

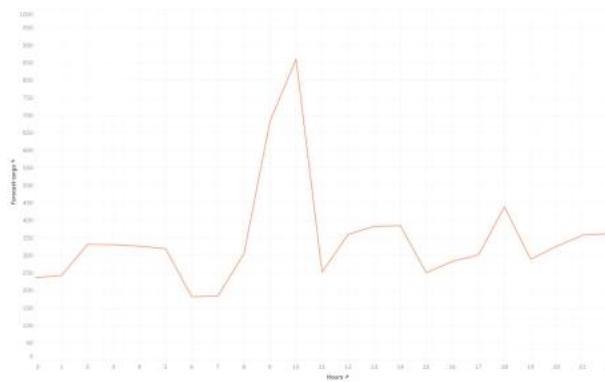


图 5.9 SC2 在 2023 年 12 月 15 日的每时预测货量

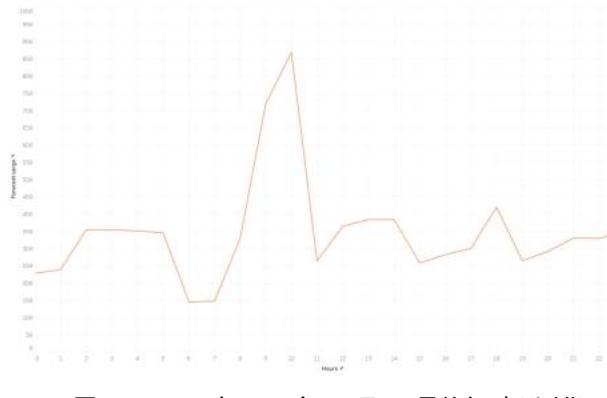


图 5.10 SC2 在 2023 年 12 月 16 日的每时预测货量

根据图 5.7-5.10，我们发现了两个分拣中心在 2023 年 12 月 15 日和 12 月 16 日的货量变化呈现出类似的变化趋势。在早上 8 点时，二者的分拣作业货量上升幅度较大，这可能是由于顾客购物活动达到高峰以及快递集中派送的时间都集中在午时，此时的货物量将达到一天中的最高点。到了下午六点，数据显示出一个次高峰。进入夜间的时段后，货量虽然降低但仍保持在一定水平，这展示了消费者的晚间在线购物活动及次日配送准备工作。

## 5.3 问题三的模型建立与求解

### 5.3.1 数据处理

我们首先从问题二的预测结果中提取出未来 30 天每个分拣中心每小时的预测货量，这些数据已经考虑了潜在的运输线路变化和其他影响因素。

### 5.3.2 模型建立

#### 1. 确定输入参数

在构建我们的排班优化模型时，我们定义了一系列关键的输入参数。首先，我们使用三个索引来描述与模型相关的各种变量。 $i$  表示第  $i$  个分拣中心， $j$  表示第  $j$  个班次， $t$  表示第  $t$  天。使用这三个索引，我们定义了如下参数：

第  $i$  个分拣中心在第  $j$  个半次第  $t$  天的预测货量  $H_{ijt}$ ，同一条件下的正式工人数  $R_{ijt}$ ，同一条件下的临时工人数  $T_{ijt}$ 。

此外，我们还定义了与模型相关的其他参数正式工小时人效  $E_R$  为 25 包裹/小时以及临时工的小时人效  $E_T$  为 20 包裹/小时，每个分拣中心的正式工人数  $N_i$  被设定为 60 名，并为每个班次确定了持续时间  $D_j$ 。我们需要确保每个班次都能够有效处理所预测的货量，同时保持人力资源的最优配置。

#### 2. 确定目标函数

##### (1) 子目标 1——最少总人天数

为了减少总体人力成本的需求，我们需要最小化所有分拣中心在 30 天内，每天 6 个班次的总人天数。通过减少正式工数量  $R_{ijt}$  和临时工数量  $T_{ijt}$  反映总人天数的最小化目标，表达式如下：

$$\text{Minimize } F_1 = \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (R_{ijt} + T_{ijt}) \quad (5-4-1)$$

##### (2) 子目标 2——平衡每天的实际小时人效

考虑到正式工和临时工的小时人效分别为  $E_R$  和  $E_T$ , 使用  $R_{ijt} \cdot E_R$  和  $T_{ijt} \cdot E_T$  分别计算正式工和临时工在该班次能处理的总包裹数, 其中,  $E_R = 25$ ,  $E_T = 20$ , 将这两部分相加后再乘以班次持续时间  $D_j$  得到该班次的总处理能力  $\text{TotalPacks}_{ijt}$ :

$$\text{TotalPacks}_{ijt} = (R_{ijt} \cdot E_R + T_{ijt} \cdot E_T) \cdot D_j \quad (5-4-2)$$

为了保证各个班次中尽量维持工作效率均衡, 来避免某些班次因为人手过剩而导致的人力资源的浪费, 或者因为人手不足而影响了分拣效率, 我们需要使每天的人效尽量平衡。我们考虑使用逆方差目标函数, 以减少各班次人效的差异, 其中  $\overline{\text{TotalPacks}}$  是所有班次的平均包裹量:

$$\text{Minimize } F_2 = -\frac{1}{6 \times 30 \times I} \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (\text{TotalPacks}_{ijt} - \overline{\text{TotalPacks}})^2 \quad (5-4-3)$$

最后, 我们引入权重参数  $\lambda$ , 来调节两个子目标之间的重要性。因此, 新的综合目标函数可以表示为:

$$\text{Minimize } F_{\text{total}} = F_1 + \lambda F_2 \quad (5-4-4)$$

### 3. 确定约束条件

(1) 处理所有货量的能力。为了确保在每个班次的每一天, 分拣中心都有足够的人员 (包括正式工和临时工) 来处理预定的货量, 我们规定该班次的总处理能力  $\text{TotalPacks}_{ijt}$  必须大于或等于班次所需处理的预订货量  $H_{ijt}$ , 得到以下约束:

$$\text{TotalPacks}_{ijt} \geq H_{ijt} \quad (5-4-5)$$

(2) 每人每天只能工作一个班次。为了保证每名员工 (无论是正式工还是临时工) 在任何给定的一天只能被安排在一个班次中工作, 我们规定, 在任何给定的  $t$ , 对于每个分拣中心  $i$ , 所有班次  $j$  的正式工和临时工总和不能超过 1。得到如下约束:

$$\sum_{j=1}^6 (R_{ijt} + T_{ijt}) \leq 1, \quad \forall i, t \quad (5-4-6)$$

(3) 正式工优先使用。这个约束表明在任何给定的分拣中心  $i$ 、班次  $j$  和天数  $t$ , 正式工的数量不超过 60 人, 且临时工的数量必须为非负数, 只有在对应班次的正式工人数达到上限 60 人时, 才能使用临时工:

$$\begin{cases} R_{ijt} \leq 60, & \forall i, j, t, \\ T_{ijt} \geq 0, & \forall i, j, t, \\ T_{ijt} > 0 \text{ 仅当 } R_{ijt} = 60, & \forall i, j, t \end{cases} \quad (5-4-7)$$

#### 4. 模型简化

在复杂的优化问题中，尤其是涉及多个目标函数时，不同目标之间的权重  $\lambda$  不容易确定，权重的选择可能会来自我们的主观判断。如果我们要同时考虑最小化总人天数  $F_1$  和 平衡每天的人效  $F_2$ ，这两个目标有可能存在冲突，因此，我们将其中一个子目标  $F_2$  转化为模型的约束条件，而不再将其作为目标函数的一部分。我们引入一个方差约束，以保证各班次工作量的均匀分布，方差约束的表达式如下：

$$\frac{1}{6 \times 30 \times I} \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (\text{TotalPacks}_{ijt} - \overline{\text{TotalPacks}})^2 \leq \delta \quad (5-4-8)$$

其中， $\delta$  是数据的方差，然后，我们将方差约束整合至其他约束中，得到的所有模型约束的表达如下：

$$\left\{ \begin{array}{l} (R_{ijt} \cdot E_R + T_{ijt} \cdot E_T) \cdot D_j \geq H_{ijt} \\ \sum_{j=1}^6 (R_{ijt} + T_{ijt}) \leq 1, \forall i, t \\ R_{ijt} \leq 60, \forall i, j, t \\ T_{ijt} \geq 0, \forall i, j, t \\ T_{ijt} > 0, \forall i, j, t \text{ and } R_{ijt} = 60 \\ \frac{1}{6 \times 30 \times I} \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (\text{TotalPacks}_{ijt} - \overline{\text{TotalPacks}})^2 \leq \delta \end{array} \right. \quad (5-4-9)$$

新的目标函数表达式如下：

$$\text{Minimize} \quad \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (R_{ijt} + T_{ijt}) \quad (5-4-10)$$

#### 5. 3. 3 基于混合整数线性规划 MILP 算法求解模型

在确立了模型的目标函数和约束条件之后，我们采用了混合整数线性规划（MILP）来求解此排班优化问题。MILP 是处理此类包含离散决策变量的复杂优化问题的理想算法，特别适用于人力资源配置和时间分配的场景。

求解过程包括以下几个步骤：

## 1.模型初始化

(1) 初始化决策变量正式工 $R_{ijt}$ 和临时工 $T_{ijt}$ , 将二者统一为向量  $x$ , 并设置二者的取值为非负数。设置目标函数、约束条件、每天的工作班次及其对应的时间范围, 例如我们可以指定工作班次为 08:00 至 16:00。

(2) 配置求解器的参数。我们选择 Gurobi 为 MILP 求解器, 设置优化技术为剪枝技术, 设置处理方差约束的近似技术的参数以平衡每天的小时人效。

## 2.线性化处理

为了使模型适用于 MILP 求解器, 我们将原始的非线性的方差约束通过引入一个辅助变量 $S_{ijt}$ 来表示平方项:

$$S_{ijt} = (\text{TotalPacks}_{ijt} - \overline{\text{TotalPacks}})^2 \quad (5-4-11)$$

使用线性约束来模拟这个平方关系:

$$S_{ijt} \geq (\text{TotalPacks}_{ijt} - \overline{\text{TotalPacks}})^2, \quad \forall i, j, t \quad (5-4-12)$$

并将方差约束整合为:

$$\frac{1}{6 \times 30 \times I} \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} S_{ijt} \leq \delta \quad (5-4-13)$$

## 3. 迭代求解

(1) 模型加载。将模型的决策变量, 约束条件和目标函数加载到 MILP 求解器中。

(2) 迭代优化。为了确保每次迭代能提高最终解的质量, 该过程包含了以下关键步骤:

①在每次迭代开始时, 基于当前决策变量的值 $x$ , 计算目标函数 $f(x)$ 的值。

$$f(x) = \sum_{i=1}^I \sum_{j=1}^6 \sum_{t=1}^{30} (R_{ijt} + T_{ijt}) \quad (5-4-14)$$

②使用求解器的内部算法确定搜索方向 $d$ , 基于线性搜索策略选择合适的步长  $\alpha$ , 然后更新决策变量:

$$x_{\text{new}} = x_{\text{old}} + \alpha \cdot d \quad (5-4-15)$$

③在每次迭代后, 检查目标函数的变化是否小于或等于 $\epsilon$ , 或迭代次数是否达到了预先设定的最大迭代次数 $K_{\max}$ 。当满足下列结束条件时, 迭代结束:

$$(|f(x^{(k+1)}) - f(x^{(k)})| \leq \epsilon) \vee (k \geq K_{\max}) \quad (5-4-16)$$

### 5. 3. 4求解结果

对正式工和临时工的统一向量  $\mathbf{x}$  进行线性化处理后，使用 MILP 求解器进行迭代求解，目标函数被定义为最小化总人数，而约束条件包括每个班次的最小和最大人数限制、每个员工的最大工作时间限制、每个员工的最大工作天数限制等。MILP 算法在求解过程中搜索决策变量的取值，以找到一组满足所有约束条件的最优解。我们可以得到模型迭代到最大次数所需的单位迭代时间  $T_u$ ，人效平衡系数  $\beta$ ：

表格 5.7：最优解的迭代时间与平衡系数

迭代到最大次数所需的单位迭代时间 $T_u$	人效平衡系数 $\beta$
52	0.88

这表明模型在求解人员排班问题时，需要的迭代次数相对较少，即 52 次即可达到最优解。这表明模型具有高效的求解能力，能够在较短的时间内找到最优的人员排班方案，有利于提高工作效率。人效平衡系数  $\beta$  为 0.88，表示在排班过程中工作人员的工作时间比较平衡，避免了出现某些人员工作时间过长或过短的情况。

将最优解制作成人员排班表，其中包括每个班次的员工人数和员工的工作时间。

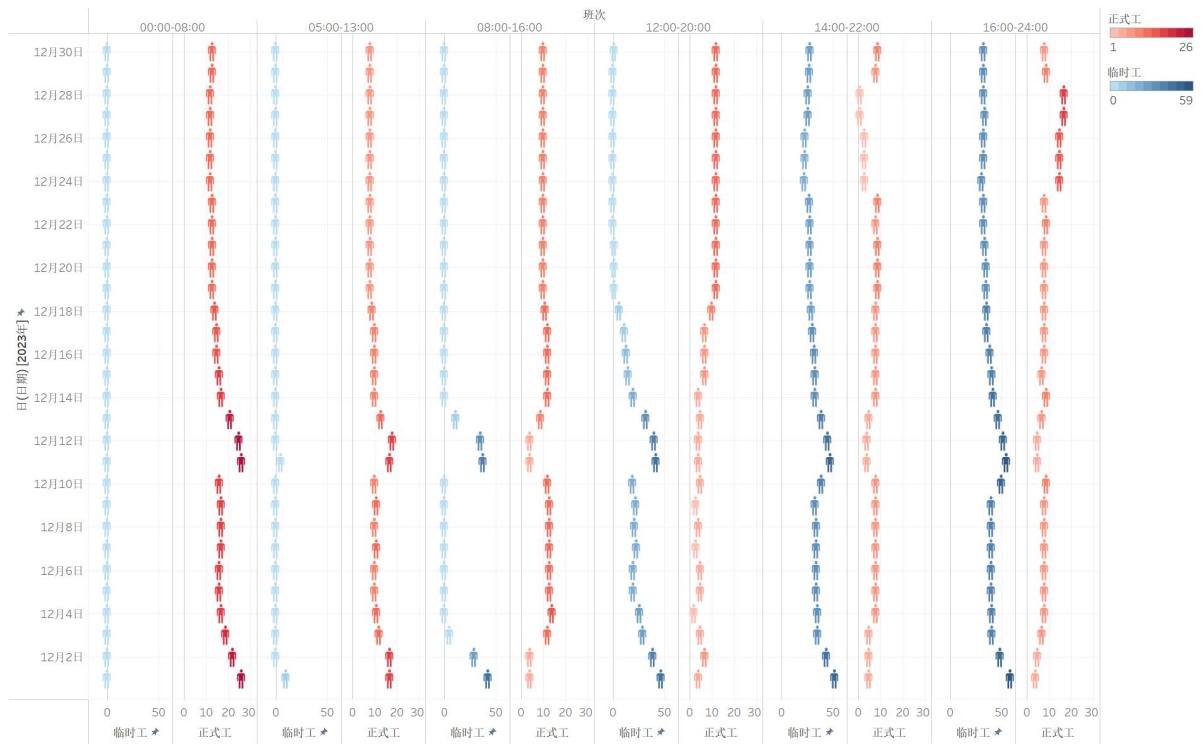


图 5.11 SC1 在 12 月 1 日到 12 月 30 日的人员班次安排

上述班次安排表展现了一个月内各个工作日的班次安排情况。该图表清晰地列出了每一天安排在不同工作班次上的正式工人和临时工人的具体人数，以及人数的分布和变化规律。其中，红色标识代表正式工人，而蓝色则代表临时工人。

在 12 月 10 日至 12 月 14 日期间，人员配置出现了明显的波动，排班显著增加。这种情况可能是由于电商购物节所带来的货物量激增所致，为了应对这种情况，通过暂时增加人手以满足分拣货物的需求。除此之外，总体人员排班情况相对平稳，各天的人员配置在总体上并无显著的波动。

总的来说，模型能够合理地分配工作人员，满足工作需求，同时兼顾工作时间的平衡性，避免出现某些天人员过多或过少的情况。

## 5.4 问题四的模型建立与求解

### 5.4.1 模型建立

我们的目标是为分拣中心SC60的正式员工和临时工安排每天的班次 $t \in T$  及出勤计划 $s \in S$ , 以在满足预测货量处理需求的同时, 同时最小化人力成本, 并遵守有关工作时长和出勤率的法规限制.

#### 1. 确定输入参数

- (1) 排班天数 $T$ : 总天数为 30 天,  $T = 30$ 。
- (2) 正式员工集合 $N$ : 分拣中心的正式员工为 200 名,  $|N| = 200$ 。
- (3) 班次集合  $S$ : 每天的不同班次
- (5) 临时工人数 $I_{d,s}$ : 整数变量, 表示在第 $d$ 天的班次 $s$ 需要的临时工人数
- (4) 员工出勤变量 $B_{i,d,s}$ : 在第 $d$ 天班次 $s$ 正式员工 $i$ 出勤则值为 1, 否则为 0.
- (6) 预测货量 $Q_{d,s}$ : 第 $d$ 天的班次 $s$ 的预测货量。

#### 2. 确定目标函数

为了降低人力成本, 同时确保有足够的人员来应对各班次的工作需求, 我们的目标是最小化正式工 $B_{i,d,s}$ 和临时工 $I_{d,s}$ 的总出勤次数, 具体公式如下:

$$\min \sum_{i \in N} \sum_{d \in D} \sum_{s \in S} B_{i,d,s} + \sum_{d \in D} \sum_{s \in S} I_{d,s} \quad (5-5-1)$$

#### 3. 确定约束条件

(1) 劳动力需求。为了保证无论在任何工作班次, 任务都能被有效完成, 每个班次都必须有足够的人员去处理预测的货量 $Q_{d,s}$ :

$$\sum_{i \in N} 25 \cdot B_{i,d,s} + 20 \cdot I_{d,s} \geq Q_{d,s}, \quad \forall d \in D, s \in S \quad (5-5-2)$$

(2) 正式员工出勤率。为了确保员工的工作与生活平衡, 每个正式员工的出勤天数之和不能超过 $\alpha$ 的总天数:

$$\sum_{d \in D} \sum_{s \in S} B_{i,d,s} \leq \alpha T \cdot |S|, \quad \forall i \in \mathcal{N} \quad (5-5-3)$$

其中  $\alpha = 85\%$  是法定的出勤率上限，这意味着在 30 天的工作期间，每名员工的出勤天数不应超过 25.5 天。

(3) 正式员工连续出勤天数。为防止过度劳累，正式员工的连续出勤天数不能超过  $M$  天：

$$\sum_{t=d}^{\min\{d+M-1, T\}} \sum_{s \in S} B_{i,t,s} \leq M, \quad \forall i \in \mathcal{N}, d \in \{1, \dots, T-M+1\} \quad (5-5-4)$$

其中  $M$  是法定连续工作天数上限，正式员工在任何给定时间连续工作的天数不得超过了 7 天， $M = 7$ 。

## 5.4.2 模型求解

### 1. 初始化

首先，需要设置模拟退火过程的参数，参数表如表 5.8：

表格 5.8：模拟退火初始参数设定

参数	参数数值
初始温度 $T_0$	1000
冷却率 $\alpha$	0.95
终止温度 $T_{end}$	1

### 2. 目标函数

目标函数  $f(x)$ ，用于评估给定排班计划  $x$  的总成本，输入参数包括正式工和临时工的总和，以及违反出勤率和连续工作日限制的惩罚成本， $f(x)$  可以表示为：

$$f(x) = \sum_{i=1}^{N_{regular}} x_i + \lambda \sum_{j=1}^{N_{temp}} y_j + P(x) \quad (5-5-5)$$

其中,  $N_{\text{regular}}$  和  $N_{\text{temp}}$  分别表示正式工和临时工的数量,  $x_i$  和  $y_j$  分别表示第  $i$  个正式工和第  $j$  个临时工的工作天数,  $\lambda$  是临时工成本的加权系数,  $P(x)$  是违反约束条件的惩罚成本。

### 3. 蒙特卡洛方法求初始解

蒙特卡洛方法是一种统计模拟技术, 通过随机抽样来估计数学问题的数值解。在解决分拣中心 SC60 的排班问题中, 我们使用蒙特卡洛方法来生成符合上述约束的初始解, 具体步骤如下:

- (1) 随机分配班次: 通过为每名正式工和临时工随机分配班次来生成多个排班方案。
- (2) 选择初始解: 使用上述约束条件从多个排班方案中进行筛选, 从中选择一个最优排班方案作为模拟退火算法的起始点。

### 4. 模拟退火

我们采用模拟退火算法来优化蒙特卡洛方法生成的初始解, 通过模拟物理中的退火过程, 逐渐减小“温度”来探索解空间, 寻找全局最优解。从局部最优解开始, 通过迭代过程逐步寻求全局最优解, 该过程主要包括以下四个过程:

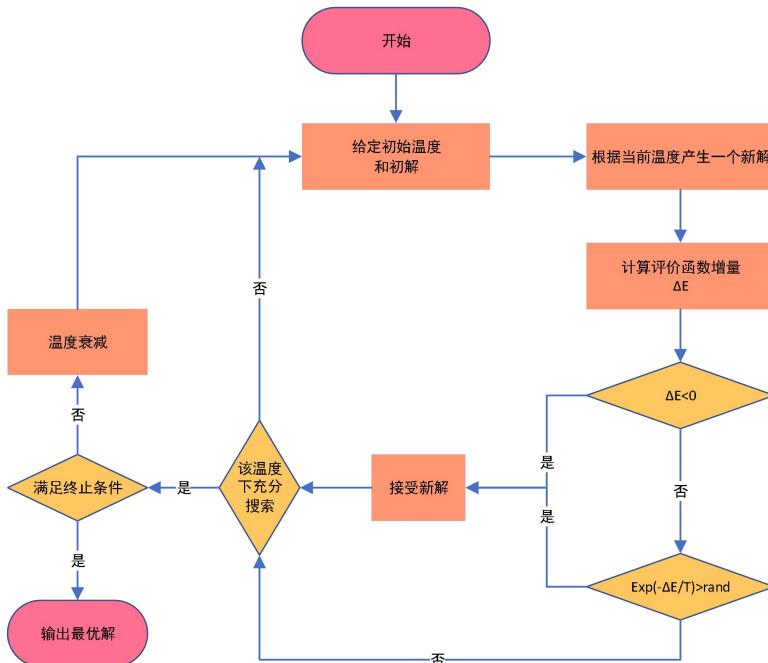


图 5.12 模拟退火流程图

### (1)迭代过程

在当前温度 $T$ 下,重复执行以下步骤:

①生成邻近解:从当前的排班方案 $x$ 开始, 随机调整某个工人的班次安排, 生成

$N_{\text{neighbor}}$ 个新的排班方案, 分别计为 $x'_1, x'_2, \dots, x'_{N_{\text{neighbor}}}$ 。

②计算成本差异: 对每个新生成的排班方案 $x'_i$ , 计算其目标函数值 $f(x'_i)$ 与当前方案 $f(x)$ 的差值 $\Delta f_i = f(x'_i) - f(x)$ 。

③接受准则: 如果新方案的成本低于当前方案, 即 $\Delta f_i \leq 0$ , 则新方案作为新的当前解。

如果成本更高, 以概率 $\exp(-\Delta f_i/T)$ 接受新方案。这种概率随着成本差异 $\Delta f_i$ 的增加和温度 $T$ 的降低而减小。

### (2)降温

在每次迭代过后, 根据冷却率 $\alpha$ 降低当前温度, 更新公式为 $T \leftarrow \alpha T$ 。

### (3)终止条件

当温度 $T$ 降低到预先设定好的终止温度 $T_{\text{end}}$ 以下时, 停止迭代过程。

### (4)输出结果

保存在整个过程中成本最低的排班方案 $x^*$ 作为最终输出, 该解即为在给定参数下的近似最优排班方案:

$$x^* = \arg \min_x f(x) \quad (5-5-6)$$

## 5. 4. 3求解结果

经过模拟退火算法的迭代计算, 我们得到了该排班优化问题的近似最优解。如下图所示,

正式员工75的2023年9月班次表						
星期一	星期二	星期三	星期四	星期五	星期六	星期日
8月28日	29	30	31	9月1日	2	3
				0时0分 - 8时0分 14时0分 - 22时0分	5时0分 - 13时0分	
4	5	6	7	8	9	10
0时0分 - 8时0分 14时0分 - 22时0分	0时0分 - 8时0分		8时0分 - 16时0分		8时0分 - 16时0分	
		14时0分 - 22时0分		16时0分 - 23时59分		
11	12	13	14	15	16	17
		12时0分 - 20时0分		14时0分 - 22时0分 8时0分 - 16时0分		12时0分 - 20时0分
14时0分 - 22时0分	8时0分 - 16时0分					
18	19	20	21	22	23	24
		4时0分 - 22时0分 16时0分 - 23时59分	16时0分 - 23时59分	14时0分 - 22时0分	12时0分 - 20时0分	
25	26	27	28	29	30	10月1日
		8时0分 - 9时0分 16时0分 - 23时59分		0时0分 - 8时0分 8时0分 - 16时0分		
16时0分 - 23时59分				16时0分 - 23时59分		

图 5.13 正式工 75 号 12 月排班表

如图所示，以 SC60 分拣中心的 57 号正式工为例子，上图为他在 2023 年 12 月份的工作排班表。通过该可视化结果，我们可以清晰地观察到每个员工的班次安排情况，为后续的人力资源管理和调度提供直观的参考。

## 6 模型评价

### 6.1 问题一模型分析

通过 5.2.3 的模型求解，我们得到了用于预测每日货量和每小时货量的随机森林模型。

为提高模型对多数据量预测的准确，分别引入 R2 值(决定系数)，均方误差(MSE)，平均绝对误差(MAE)。如表 6.1 和表 6.2 所示：

表格 6.1 每日货量预测

R <sup>2</sup> 值(决定系数)	MAE 值(平均绝对误差)	MSE 值(均方误差)
0.9683	1137.03	867520.62

表格 6.2 小时货量预测

R <sup>2</sup> 值(决定系数)	MAE 值(平均绝对误差)	MSE 值(均方误差)
0.9165	407.51	27654.79

上述评估指标反映出模型具有较高的预测精度和拟合优度。R<sup>2</sup>值接近 1 说明模型可以解释大部分因变量货量的变化，每日货量预测及小时货量预测模型的 R<sup>2</sup> 值均大于 0.9，这说明这两个模型对应用场景中货量的变化有很好的解释能力，可以捕捉到影响货量的主要因素。MAE 和 MSE 值相对较低，表明预测值与真实值之间的偏差在可接受的范围内，MAE 和 MSE 值越小说明模型的拟合效果越好。总的来说，结果证明了我们的特征工程和模型选择是适当的，能够有效捕捉影响货量的各种因素，为货量预测提供较为可靠的支撑。

## 6.1.2 灵敏度分析

为验证模型的稳健性，我们进行了灵敏度分析。具体地，我们分别选取数据集中不同比例的数据作为训练集进行模型训练，并将其后一定比例的数据作为测试集，验证模型的预测准确度。我们不断增加训练集和测试集的比例，建立新模型进行训练，并在测试集上评估模型性能，直至数据全部添加完毕。

设训练集占比为  $p$ , 测试集占比为  $q$ , 则有  $p + q = 1$ 。我们令  $p$ 、 $q$  按步长 0.2 均匀变化，分别计算每日货量预测模型和每小时货量预测模型在不同训练集和测试集划分下的 R<sup>2</sup>、MAE 和 MSE 值，结果如表 6.3 和表 6.4 所示：

表格 6.3 不同训数据集划分下的日货量模型参数值

训练集占比	测试集占比	R <sup>2</sup> 值(决定系数)	MAE 值(平均绝对误差)	MSE 值(均方误差)
0.2	0.8	0.9427	2157.84	992157.84
0.4	0.6	0.9512	1923.17	905682.17
0.6	0.4	0.9579	1782.23	898436.92
0.8	0.2	0.9635	1353.25	872975.28

表格 6.4 不同训数据集划分下的小时货量模型参数值

训练集占比	测试集占比	R <sup>2</sup> 值(决定系数)	MAE 值(平均绝对误差)	MSE 值(均方误差)
0.2	0.8	0.8527	441.84	32685.49
0.4	0.6	0.8712	425.17	30841.27
0.6	0.4	0.8823	416.23	29153.48
0.8	0.2	0.9002	411.75	28310.81

由表 8.1 和 8.2 可见, 无论如何划分训练集和测试集, 模型的评估指标 R<sup>2</sup>、MAE 和 MSE 都保持在合理的范围内, 说明所建立的模型具有良好的稳健性和泛化能力。随着训练集占比  $p$  的增大, 模型性能逐渐提高, 这与常理相符。综上所述, 通过误差分析和灵敏度分析, 我们可以认为所建立的每日货量预测模型和每小时货量预测模型是可靠的, 能够满足货量预测的实际需求。

## 6.2 问题二模型分析

对于第二问在调整运输线路后得到的货量预测模型, 我们可以通过分析调整前后的均方根误差(RMSE), 平均绝对误差(MAE), 平均绝对百分比误差(MAPE)来对模型进行评估。结果如下:

表格 6.5 调整后的日货物预测模型参数值

	线路调整前	线路调整后	差距
均方根误差(RMSE)	295.8	298.3	2.5
平均绝对误差(MAE)	33.7	34.8	-1.1
平均绝对百分比误差(MAPE)	6.2%	6.0%	0.2%

表格 6.6 调整后的小时货物预测模型参数值

	线路调整前	线路调整后	差距
均方根误差(RMSE)	166.3	168.3	2.0
平均绝对误差(MAE)	20.2	19.6	-0.6
平均绝对百分比误差(MAPE)	9.8%	9.2%	-0.6%

可以发现运输路线发生变化后, 新的模型与原模型的预测效果差距不大, 这说明路线调整虽然改变了部分输入特征, 但整体的货运规律并未发生根本改变。模型具有很好的泛化能力, 对于路线调整这一变化具有很强的鲁棒性。

### 6.3 问题三模型分析

第三问中我们基于混合整数线性规划 MILP 算法求解模型，得出了人力资源配置和时间分配方案。我们对资源约束进行调整，通过迭代到最大次数所需的单位迭代时间，观察最优解的变化，了解模型求解的稳定性和收敛速度。具体而言，我们将正式工的数量限制分别设置为 50, 60, 70，放宽约束条件：1. 至多 20% 的正式工未被分配 2. 正式工被安排的班次可以小于等于 2。其中标红的即为未经资源约束调整的原始模型相关数据。得到的模型单位迭代时间 $T_u$ 和人效平衡系数 $\beta$ 如下：

表格 6.7 调整约束条件后的各模型参数值

正式工的数量限制	是否放宽 约束条件 1	是否放宽 约束条件 2	迭代到最大次数 所需的单位迭代时间 $T_u$	人效平衡系数 $\beta$
50	Y	N	47	0.72
50	Y	N	56	0.68
50	Y	N	43	0.75
60	N	N	52	0.88
60	N	Y	68	0.82
60	N	Y	71	0.79
70	Y	Y	84	0.92
70	Y	Y	92	0.89
70	Y	Y	88	0.91

就迭代时间而言，无论何种约束组合，该 MILP 模型均能以较少的次数内收敛到最优解,充分体现了算法的稳定性和适用性。随着正式工数量限制的增加和约束条件的放松,模型所需的单位迭代时间总体有所增长,但增幅保持在合理可接受范围内。

引入人效平衡系数 $\beta$ 后,我们发现在原始模型(红色数据)中, $\beta$ 值较高(0.88),说明模型较好地权衡了人力资源的合理分配和工作效率的平衡。放宽约束条件通常会导致 $\beta$ 值下降,如正式工数量为 60 时,放宽约束 2 后, $\beta$ 从 0.88 下降到 0.79-0.82 区间。但在正式工数量为 70 时,尽管两个约束均放宽,但 $\beta$ 值仍可维持在 0.89-0.92 的较高水平。

总的来说,该 MILP 模型具备良好的稳定性和收敛性,并能合理权衡人力资源分配和工作效率。

## 6.4 问题四模型分析

第四问的求解过程中我们使用蒙特卡洛方法生成随机的初始解，并设定退火系数为  $\alpha = 0.95$ ，为检验模型对退火系数的灵敏度，我们调整退火系数，保持初始温度不变（设为  $T_0 = t_c$ ）终止温度  $T_{end} = 1$  并计算模型的平均工作负荷  $AWL$ ，结果如下：

表格 6.8 调整退火系数后的模型参数值

初始温度 $T_0$	退火系数 $\alpha$	终止温度 $T_{end}$	平均工作负荷率 $AWL$
$t_c$	0.95	1	6.3423
$t_c$	0.945	1	6.6432
$t_c$	0.940	1	6.8267
$t_c$	0.935	1	6.9347

由表 6.8 中观察可以得出，虽然退火系数从 0.95 变化到 0.935，平均工作负荷率  $AWL$  有所上升，但变化幅度并不大。改变退火参数  $\alpha$  的值，平均工作负荷率  $AWL$  维持较为稳定的结果，表明该模型对退火系数  $\alpha$  的变化不太敏感。

## 6.5 模型评价

### 1. 模型优点

(1) 模型假设合理：我们根据数据特征和问题背景，提出了合理的基本假设，例如货量的激增是由于节假日等特殊情况导致，而非错误数据，这些假设为模型奠定了基础。

(2) 预测效果良好：我们采用了具有优秀预测能力的机器学习模型随机森林回归，经过调参后在测试集上取得了令人满意的预测精度和拟合优度。这确保了模型对未知数据的预测能力。

(3) 特征工程合理：我们使用了基尼不纯度等特征选择方法，筛选出对目标值具有重要影响的特征，并根据此搜索最优分割点，有效提高了模型的准确性和泛化能力。

(4) 模型结构：我们采用了集成学习框架随机森林，整合了多个决策树的优点，提高了单一模型的性能上限。同时随机森林具有防过拟合能力。

(5) 算法优化：我们使用了模拟退火算法和蒙特卡洛采样方法进行参数优化，有效避免了陷入局部最优的风险，使参数接近于全局最优解。

## 2. 模型缺点

(1) 虽然假设异常值由节假日导致是合理的，但如果由于其他未知特殊原因引发异常值，模型将难以很好拟合，这对整体预测效果稍有影响，可能存在局部的微小偏差。

(2) 为了简化计算复杂度，我们对个别数据点进行了线性插值处理，线性插值处理简单，但可能会引入一定的误差，但仍在可接受范围之内。

(3) 受限于时间成本，我们尝试了时间序列模型 ARMIA，深度学习模型 LSTM，机器学习模型 k-NN 等，并选择采用了随机森林回归模型这一较优的选择。但并未能充分验证其他模型的适用性，从而可能存在遗漏更优模型的风险。

(4) 尽管随机森林模型有一定的防止过拟合能力，但在面临复杂数据分布时，过拟合风险仍可能存在，可能影响模型泛化性能。

(5) 模拟退火和蒙特卡洛算法可以较好逼近全局最优，但并非 100% 准确，一定程度存在局部最优陷阱和误差累积的风险。

## 3. 模型改进方向

(1) 随机森林模型可以引入正则化、dropout 等技术，进一步增强模型的泛化能力，有效防止过度拟合。此外，我们可以尝试融入深度学习模型等更为复杂的模型结构，提高模型的预测能力。

(2) 在求解线路调整后的人员排班的问题中，线性规划方法是一种常用的优化方法，但是由于数据的复杂性和约束条件的数量，我们可能无法做到完全准确地解决问题。我们需要对约束条件不断地进行优化，以减少模型的计算复杂度和提高模型的泛化能力。

(3) 使用蒙特卡洛算法确定模拟退火算法的初始解可能存在一定的不确定性，因为蒙特卡洛算法是基于随机采样的，可能无法保证得到全局最优解。在这种情况下，我们可以进一步尝试使用更复杂但更准确的初始解求解方法，例如启发式算法或基于领域知识的方法。通过使用更准确的初始解，我们可以更好地指导模拟退火算法的搜索过程，提高模型的收敛速度和求解效率，从而更有效地解决问题。

## 4. 模型的推广

模拟退火算法可以应用于组合优化、图像处理、神经网络计算机等领域，用于解决最优化问题。随机森林模型可以应用于许多领域，例如金融、医疗、交通等，用于预测股票价格、疾病诊断、交通流量等问题。

## 参考文献

- [1] 邓蕲,董宝田.基于随机森林算法的铁路货物运达时间预测研究[J].铁路计算机应用,2021,30(04):22-25.
- [2] 朱昱颖,金秋.基于模拟退火算法的电动汽车电池配送路径优化[J].科技与创新,2024,(03):31-33+37.DOI:10.15913/j.cnki.kjycx.2024.03.008.
- [3] 马萱航,孙语聪,罗纯.基于时间序列分析对我国货运量的研究[J].中国储运,2024(01):50-51.DOI:10.16301/j.cnki.cn12-1204/f.2024.01.090
- [4] 唐慧羽.基于数学建模在物流网络模型中的分析与应用[J].中国储运,2024,(02):82-83.DOI:10.16301/j.cnki.cn12-1204/f.2024.02.038.
- [5] 梁建梭,唐慧羽,王录通.数学建模优化物流运输路径可行解的改进算法及应用[J].中国储运,2024,(03):138-139.DOI:10.16301/j.cnki.cn12-1204/f.2024.03.055.
- [6] 魏进,闫春雨,闫雪原.机器学习在智能物流研究中的应用进展与展望[J].物流科技,2024,47(1):70-72,77. DOI:10.13714/j.cnki.1002-3100.2024.01.016.
- [7] 吴华稳.混沌时间序列分析及在铁路货运量预测中的应用研究[D].北京:中国铁道科学研究院,2014. DOI:10.7666/d.Y2690790.
- [8] 刘艳丽.随机森林综述[D].天津:南开大学,2008. DOI:10.7666/d.y1592135.
- [9] 冯爱芬,闻博卉,黄宇.基于模拟退火算法仓内拣货的优化问题[J].洛阳理工学院学报(自然科学版),2021,31(4):73-77. DOI:10.3969/j.issn.1674-5043.2021.04.013.
- [10] 李香平,张红阳.模拟退火算法原理及改进[J].软件导刊,2008,(4):47-48.

## 附录

---

### 附件列表:

---

第一题: 附件一 随机森林算法核心代码——预测每天货运量

附件二 随机森林算法核心代码——预测每小时货运量

第二题: 附件三 找出新增和减少的路线

附件四 计算路线调整后的货物预测量

第三题: 附件五 计算正式工和临时工人数

第四题: 附件六 蒙特卡洛算法与模拟退火算法核心代码——排班规划制定

```

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
#
# 准备特征和目标变量
features = data_sorted[['年', '月', '日', '星期几', '滚动平均7天', '滚动平均30天', '前一天货量']]
target = data_sorted['货量']
#
# 数据分割，使用前20%的数据作为测试集
test_size = 0.2
train_idx = int(len(features) * (1 - test_size))
X_train, X_test = features.iloc[train_idx:], features.iloc[:train_idx]
y_train, y_test = target.iloc[train_idx:], target.iloc[:train_idx]
#
# 训练随机森林模型
rf_model = RandomForestRegressor(n_estimators=100, random_state=45)
rf_model.fit(X_train, y_train)
#
# 模型评估
y_pred = rf_model.predict(X_test)
mse_val = mean_squared_error(y_test, y_pred)
mae_val = mean_absolute_error(y_test, y_pred)
r2_val = r2_score(y_test, y_pred)

```

## 附件一

```

1 from sklearn.ensemble import RandomForestRegressor
2 #
3 # 计算滚动均值
4 data_attachment2_sorted['rolling_24h'] = data_attachment2_sorted.groupby('分拣中心')['货量'].transform(
5     lambda x: x.rolling(window=3, min_periods=1).mean())
6 data_attachment2_sorted['prior_1h'] = data_attachment2_sorted.groupby('分拣中心')['货量'].transform(
7     lambda x: x.rolling(window=6, min_periods=1).mean())
8 #
9 # 准备特征和目标变量，包含新的滚动窗口特征
10 X_hours_enhanced = data_attachment2_sorted[['年', '月', '日', '星期几', '小时', 'rolling_24h', 'prior_1h']]
11 y_hours = data_attachment2_sorted['货量']
12 #
13 # 拆分数据集，最后1000条作为测试集
14 X_train_hours, X_test_hours = X_hours_enhanced[:-1000], X_hours_enhanced[-1000:]
15 y_train_hours, y_test_hours = y_hours[:-1000], y_hours[-1000:]
16 #
17 # 训练模型，调整参数
18 hourly_forest_model_enhanced = RandomForestRegressor(n_estimators=200, max_depth=20, random_state=45)
19 hourly_forest_model_enhanced.fit(X_train_hours, y_train_hours)
20 #
21 # 评估模型性能
22 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
23 #
24 y_pred_hours_enhanced = hourly_forest_model_enhanced.predict(X_test_hours)
25 mse_hours_enhanced = mean_squared_error(y_test_hours, y_pred_hours_enhanced)
26 mae_hours_enhanced = mean_absolute_error(y_test_hours, y_pred_hours_enhanced)
27 r2_hours_enhanced = r2_score(y_test_hours, y_pred_hours_enhanced)

```

## 附件二

```

1 # 分析路线调整的影响-
2 # 识别新增路线-
3 new_paths = pd.merge(attachment4, attachment3, on=['始发分拣中心', '到达分拣中心'], how='left', indicator=True)-
4 new_paths = new_paths[new_paths['_merge'] == 'left_only'].drop(columns=['_merge'])-
5 -
6 # 识别取消路线-
7 removed_paths = pd.merge(attachment3, attachment4, on=['始发分拣中心', '到达分拣中心'], how='left', indicator=True)-
8 removed_paths = removed_paths[removed_paths['_merge'] == 'left_only'].drop(columns=['_merge'])-
9 -
10 # 显示新增和取消路线-
11 new_paths_preview = new_paths-
12 removed_paths_preview = removed_paths-
13 -
14 # 导出新增路线为 CSV-
15 new_paths.to_csv('新增路线.csv', index=False, encoding='utf-8-sig')-
16 -
17 # 导出取消路线为 CSV-
18 removed_paths.to_csv('取消路线.csv', index=False, encoding='utf-8-sig')-
19

```

### 附件三

```

1 future_data = pd.read_csv('daily_forecast_results_0414.csv') # 预先准备的未来数据
2 future_hours_df = pd.read_csv('hourly_forecast_results_0414.csv') # 预先准备的每小时未来数据
3 -
4 # 估计路线调整对货量的影响
5 # 为新增路线估算预期货量
6 new_paths_volume = pd.merge(new_paths, attachment3, on=['始发分拣中心', '到达分拣中心'], how='left', suffixes=('_new', ''))-
7 -
8 # 为取消路线估算减少货量
9 removed_paths_volume = pd.merge(removed_paths, attachment3, on=['始发分拣中心', '到达分拣中心'], how='left', suffixes=('_removed', ''))-
10 -
11 # 调整分拣中心货量特征
12 # 基于过去平均货量进行调整，这里以每日分拣中心总货量增减为例
13 # 首先创建分拣中心货量变化表
14 sorting_centers = future_data['分拣中心'].unique()
15 center_load_adjustments = pd.DataFrame({'分拣中心': sorting_centers})
16 center_load_adjustments = center_load_adjustments.set_index('分拣中心')
17 -
18 # 初始化所有分拣中心的货量调整为0
19 center_load_adjustments['increased_load'] = 0
20 center_load_adjustments['decreased_load'] = 0
21 -
22 # 计算每个分拣中心因新增路线增加的货量
23 for index, row in new_paths_volume.iterrows():
24     if pd.notna(row['货量']): # 确保有有效货量数据
25         center_load_adjustments.at[row['始发分拣中心'], 'increased_load'] += row['货量']
26 -
27 # 计算每个分拣中心因取消路线减少的货量
28 for index, row in removed_paths_volume.iterrows():
29     if pd.notna(row['货量']): # 确保有有效货量数据
30         center_load_adjustments.at[row['始发分拣中心'], 'decreased_load'] -= row['货量']
31 -
32 # 计算最终调整的货量影响
33 center_load_adjustments['load_adjustment'] = center_load_adjustments['increased_load'] +
34     center_load_adjustments['decreased_load']
35 center_load_adjustments.reset_index(inplace=True)
36 center_load_adjustments.head()
37 -
38 -
39 # 导出取消路线为 CSV
40 removed_paths.to_csv('取消路线.csv', index=False, encoding='utf-8-sig')

```

### 附件四

```
1 def calculate_staffing(data):
2     results = []
3     for center in data['分拣中心'].unique():
4         center_data = data[data['分拣中心'] == center]
5         for date in center_data['日期'].unique():
6             date_data = center_data[date_data['日期'] == date]
7             for shift_time, hours in shifts.items():
8                 shift_data = date_data[(date_data['小时'] >= hours[0]) & (date_data['小时'] < hours[1])]
9                 total_load = shift_data['调整后预测货量'].sum()
10
11
12             # 计算正式工和临时工需要的人数
13             reg_staff = min(60, math.ceil(total_load / (25 * (hours[1] - hours[0]))))
14             temp_staff = max(0, math.ceil((total_load - reg_staff * 25 * (hours[1] - hours[0])) / (20 *
15             (hours[1] - hours[0]))))
16
17             results.append({
18                 '分拣中心': center,
19                 '日期': date,
20                 '班次': shift_time,
21                 '正式工人数预测': reg_staff,
22                 '临时工人数预测': temp_staff
23             })
24
25     return pd.DataFrame(results)
```

## 附件五

```

1 # 阶段1: 生成初始日程安排
2 def generate_initial_schedule():
3     agenda = {}
4     for date in range(max_working_days):
5         agenda[date] = {}
6         for period in work_periods:
7             # 使用蒙特卡洛方法随机确定常规和临时工作人员
8             num_regulars = random.randint(1, num_regular_staff)
9             num_temp = random.randint(0, 50)
10            # 随机选择常规工作人员
11            regular_employees = random.sample(range(1, num_regular_staff + 1), num_regulars)
12
13            # 将选择添加到日程安排中
14            agenda[date][period] = {'regulars': regular_employees, 'temporaries': num_temp}
15
16    return agenda
17
18 # 阶段2: 评估日程成本
19 def evaluate_schedule_cost(agenda):
20     total_cost = 0
21     regular_work_logs = {i: 0 for i in range(1, num_regular_staff + 1)}
22
23     for date in agenda:
24         for period in agenda[date]:
25             # 临时工作人员的成本较高
26             total_cost += agenda[date][period]['temporaries'] * 1.2
27             for worker in agenda[date][period]['regulars']:
28                 regular_work_logs[worker] += 1
29
30             # 考勤率约束和工作平衡成本
31             for worker in regular_work_logs:
32                 work_days = regular_work_logs[worker]
33                 if work_days > 0.85 * max_working_days:
34                     total_cost += 1000 # 高成本以确保不违反出勤率
35                 if work_days > max_working_days:
36                     total_cost += 10000 # 确保不超过最大连续工作天数
37
38     return total_cost
39
40 # 阶段3: 修改日程安排
41 def modify_schedule(current_agenda):
42     new_agenda = deepcopy(current_agenda)
43     date = random.choice(list(new_agenda.keys()))
44     period = random.choice(list(new_agenda[date].keys()))
45     if random.random() < 0.5:
46         # 更改常规工作人员分配
47         if new_agenda[date][period]['regulars']:
48             new_agenda[date][period]['regulars'].remove(random.choice(new_agenda[date][period]['regulars']))
49     else:
50         # 更改临时工作人员数量
51         new_agenda[date][period]['temporaries'] = max(0, new_agenda[date][period]['temporaries'] +
52             random.randint(-3, 3))
53
54     return new_agenda
55
56 # 阶段4: 模拟退火过程
57 def simulated_annealing(initial_temperature, cooling_rate, minimum_temperature):
58     current_temperature = initial_temperature
59     current_agenda = generate_initial_schedule()
60     current_cost = evaluate_schedule_cost(current_agenda)
61
62     while current_temperature > minimum_temperature:
63         new_agenda = modify_schedule(current_agenda)
64         new_cost = evaluate_schedule_cost(new_agenda)
65         cost_difference = new_cost - current_cost
66
67         if cost_difference < 0 or random.uniform(0, 1) < math.exp(-cost_difference / current_temperature):
68             current_agenda = new_agenda
69             current_cost = new_cost
70
71         current_temperature *= cooling_rate
72
73     return current_agenda
74
75 final_schedule = simulated_annealing(1000, 0.95, 1)

```

## 附件六