

队伍编号	MCB2400772
赛道	B

## 基于混合预测与多目标优化的电商大促期间仓储与库存管理优化模型研究

### 摘要

随着电子商务的快速发展，大型促销活动（如“双十一”和“黑色星期五”等）已成为物流行业面临的重要挑战。这些活动通常伴随着销量激增、库存波动及仓储压力，进而对供应链管理提出了更高的要求。为应对这些挑战，本研究提出了一种**综合模型**，旨在通过**结合时间序列预测与优化算法**，提升电商大促期间的库存管理与仓储分配效率。具体而言，本研究聚焦于两个核心问题：一是如何精确预测促销期间各类商品的销量与库存变动；二是如何优化仓储分配策略，以实现物流成本的降低并减少仓储爆仓的风险。

针对问题一，本研究提出了一种**结合时间序列分析与机器学习的混合预测模型**。具体而言，对于平稳变化趋势的数据，采用了**ARIMAX (ARIMA + ETS)**模型进行处理；而对于波动剧烈的数据，则选用了**Prophet**模型。为了提高预测精度，本研究构建了多维度特征集，并运用**LightGBM**模型对数据中的非线性关系进行建模。**创新之处在于深度融合时间序列分析与机器学习方法，从多个维度全面捕捉销量与库存的趋势性、季节性及复杂非线性特征**。实验结果表明，所提出的混合预测模型在销量预测精度及库存变化动态捕捉能力方面均优于传统预测方法，为企业提供了更加精准的大促期间库存与销量预测数据。

针对问题二，本研究构建了一个**多目标优化模型**，旨在最小化搬仓成本与爆仓风险。该模型在定义仓储容量、日出库产能等关键约束条件的基础上，采用了**混合遗传算法与粒子群优化算法 (GA-PSO)**进行求解。遗传算法的全局搜索能力与粒子群优化的快速收敛特性相结合，使得该模型在处理大规模数据时，能够高效地逼近最优解。**研究的创新之处在于引入了自适应调整策略，通过动态平衡目标函数中各项权重，增强了算法的鲁棒性与适用性**。最终的分仓规划结果表明，优化方案在降低爆仓天数、减少搬仓成本以及提升仓储资源利用效率方面表现出显著的优势。

综上所述，本研究提出的预测与优化框架不仅能够有效应对电商大促期间复杂的库存管理与仓储规划问题，还能为企业的物流决策提供可靠的理论依据。展望未来，该模型具有较强的扩展性，可进一步应用于其他动态场景，如区域化分仓和实时物流调度等，从而提高供应链管理效率提供更加系统的理论支持。

**关键词：**电商物流、时间序列预测、ARIMAX、LightGBM、GA-PSO、供应链优化

# 目录

一、问题背景 .....	1
二、问题分析 .....	1
2.1 问题 1 重述与分析 .....	1
2.2 问题 2 重述与分析 .....	2
三、模型假设 .....	3
3.1 数据与需求假设 .....	3
3.2 仓库与物流假设 .....	3
四、数据探索与预处理 .....	4
4.1 数据描述与符号设定 .....	4
4.2 数据探索与标准化 .....	5
4.2.1 缺失值处理 .....	5
4.2.2 时间序列矩阵构建: .....	5
4.3 时间序列与相关性分析 .....	6
4.3.1 动态时间规整 (DTW) 距离 .....	8
4.3.2 基于 DTW 的 K 均值聚类 .....	8
4.3.3 聚类结果分析 .....	9
4.4 相关性分析 .....	10
4.4.1 品类-品类相关性分析 .....	11
4.4.2 高级品类-趋势-件型相关性分析 .....	12
五、问题一模型建立与求解 .....	12
5.1 库存量预测模型 .....	13
5.1.1 平稳变化趋势类库存: ARIMAX 模型 (ARIMA + ETS ) .....	13
5.1.2 剧烈波动趋势类库存: Prophet 模型 .....	15
5.2 销量预测模型 .....	17
5.2.1 平稳变化趋势类销量: ARIMAX+LightGBM .....	17
5.2.2 剧烈波动趋势类销量: Prophet+LightGBM .....	20
5.3 求解结果讨论 .....	22
5.3.1 库存量大促预测 .....	22
5.3.2 销量大促预测 .....	23
六、问题二模型建立与求解 .....	24
6.1 问题描述与建模 .....	24
6.1.1 符号与参数定义 .....	24
6.1.2 决策变量 .....	25
6.1.3 目标函数 .....	27
6.1.4 约束条件 .....	27
6.2 分仓规划: 混合遗传-粒子群算法 (GA-PSO) .....	28
6.2.1 遗传算法: 全局搜索 .....	28
6.2.2 粒子群优化: 局部精细化搜索 .....	30
6.2.3 自适应调整策略 .....	31
6.3 求解结果讨论 .....	32

七、模型的综合评价 .....	33
参考文献.....	33
附录 A 文件列表 .....	33
附录 B 代码.....	错误!未定义书签。

## 一、问题背景

### Top 5 Things People Look for on Black Friday 2020



图 1 2020 年“黑色星期五”消费者关注要素统计

随着全球电子商务的迅猛发展，大型促销活动，如“双十一”、“六一八”和“黑色星期五”等，已成为物流行业面临的重要挑战。这些活动不仅推动了在线销售额的显著增长，同时也对物流配送效率、库存管理和分仓策略提出了更为苛刻的要求。在面对销量激增、库存波动以及退货处理等复杂问题时，物流企业通过数字化转型、智能化升级及技术创新（例如自动化设备的应用），显著提升了运营效率和服务质量。为满足消费者对快速配送日益增长的需求，许多物流企业采取了多仓分布的战略，从而增强了供应链的灵活性与响应速度。这些策略和创新不仅保障了大促期间物流运作的顺畅进行，也为企业带来了降本增效的双重优势。

## 二、问题分析

### 2.1 问题 1 重述与分析

**问题重述：**基于附件中的数据，预测 2023 年 10 月 15 日至 11 月 15 日大促期间各品类的平均库存量和销量。

**问题分析：**在大促期间，准确预测各品类的销量是供应链管理的核心任务之一，其对库存控制、物流调度及资源配置具有重要影响。销量预测需要综合考虑多种因素，如历史销售数据、季节性波动、促销活动的力度、市场趋势以及竞争态势等。这些因素之间的复杂关系使得销量预测成为一项具有挑战性的任务。

为实现高精度的销量预测，可以采用时间序列分析与机器学习方法相结合的建模策略。时间序列分析可以有效捕捉销量数据的趋势性和季节性特征，帮助识别长期规律和周期性波动；而机器学习模型则能够从非线性关系和复杂的交互效应中挖掘潜在模式，提升预测精度。通过数据预处理、特征工程以及模型优化，能够进一步提高模型的鲁棒

性和预测性能。例如，通过特征工程提取历史销量的滞后效应、滚动统计特征、节假日效应等，可以更准确地反映销量波动的影响因素；同时，通过模型优化，调整超参数和选择适当的算法，可以确保预测结果在大规模数据和动态环境中的稳定性。

精准的销量预测不仅能够为库存安排提供科学依据，避免因过度备货或缺货导致的资源浪费与销售损失，还能为物流规划提供支持，确保运输资源的合理调配。这将直接提高供应链响应速度，提升客户满意度，并在大促期间最大化企业利润。因此，构建高效、精确的销量预测模型对于优化供应链管理、提升整体运营效率至关重要。

## 2.2 问题 2 重述与分析

**问题重述：**根据复赛问题 1 的库存量及销量预测结果，重新制定品类分仓规划方案。

**问题分析：**针对复赛问题一所提供的库存量及销量预测结果，问题二的核心任务是重新制定各品类的分仓规划方案。具体而言，新的分仓规划需在保证大促期间库存和销量需求平衡的基础上，优化仓储资源配置，以最大化物流效率并降低相关成本。

首先，基于预测的销量与库存量，必须重新评估现有仓储布局和库存分配策略。通过对不同品类的需求特征及其在各区域市场的分布情况进行分析，可以确定每个仓库所需承担的具体负荷。



图 2 全国物流集中点-广东省仓储分布图

其次，为降低整体物流成本，新的分仓方案需要考虑运输路径、仓储容量、日出库产能等约束条件，通过科学的库存调度和分仓策略实现资源的最优配置。

因此，基于问题一的销量与库存预测结果，问题二的分仓规划方案应着重解决如何根据动态需求优化仓储布局和库存分配（如图 2 中的多点分仓物流配送机制），以确保在大促期间实现供应链的高效运行、成本控制及客户满意度的提升。

## 三、模型假设

### 3.1 数据与需求假设

**需求预测准确性:** 在本研究中, 假设大促期间各品类的销量和库存预测数据准确。即, 设定预测模型输出的销量和库存量为  $\hat{d}_i(t)$  和  $\hat{I}_i(t)$ , 其中  $i$  表示品类,  $t$  表示时间点。在假设条件下, 实际销量  $d_i(t)$  和实际库存量  $I_i(t)$  与预测值之间的误差被认为为零, 故有:

$$d_i(t) = \hat{d}_i(t), \quad I_i(t) = \hat{I}_i(t)$$

这一假设的目的是确保预测数据的准确性, 从而能够有效支撑后续库存安排和仓储分配策略的制定。

**品类独立性:** 本假设假定不同品类的需求彼此独立, 即品类  $i$  和品类  $j$  的需求  $d_i(t)$  和  $d_j(t)$  不受彼此的影响。在数学上则是需求之间的协方差为零, 表现为:

$$\text{Cov}(d_i(t), d_j(t)) = 0, \quad \forall i \neq j$$

这一假设简化了需求预测模型, 使得各品类的需求预测可以单独进行, 无需考虑跨品类的依赖关系, 从而减少了模型的复杂性。

### 3.2 仓库与物流假设

**仓库容量与产能固定:** 假设大促期间每个仓库的存储容量  $C_k$  和出库产能  $O_k$  在时间段内保持不变, 即:

$$C_k = \text{constant}, \quad O_k = \text{constant}, \quad \forall k$$

**搬仓可行性与无损耗:** 假设所有品类可搬仓, 且搬仓过程中无损耗。

## 四、数据探索与预处理

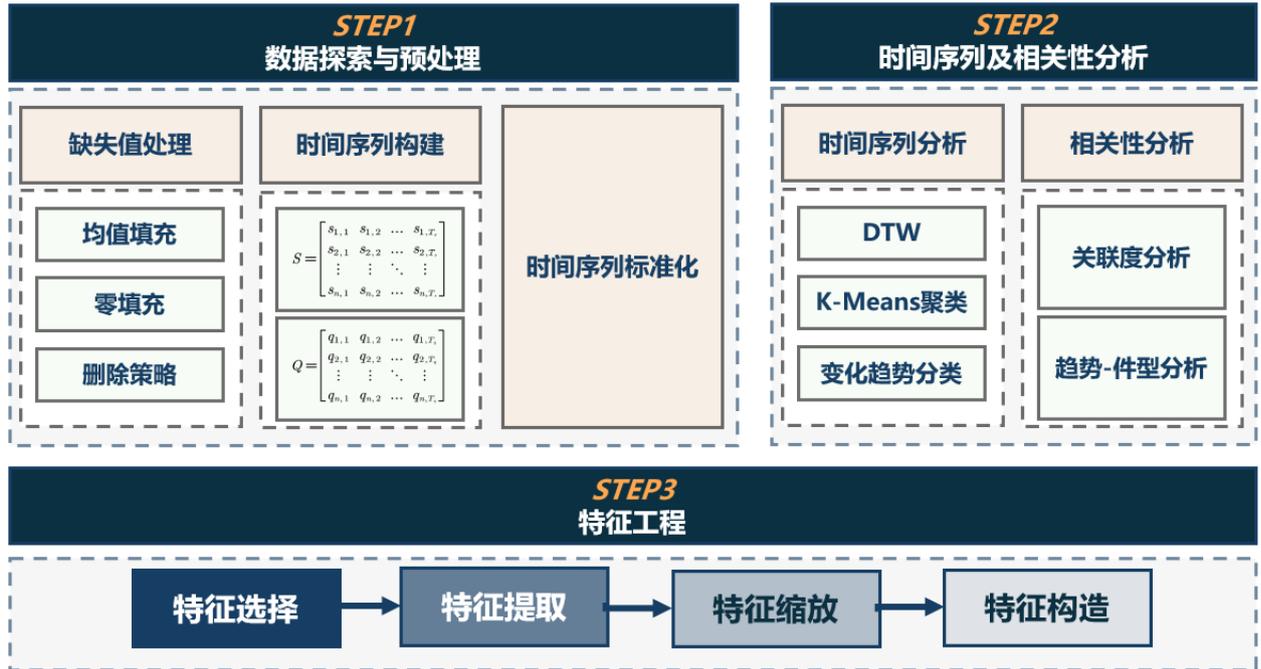


图 3 数据分析与预处理流程

### 4.1 数据描述与符号设定

由附件数据可得，商品共有  $C = 350$  种品类，记为  $\{c_i\}_{i=1}^C$ 。

**库存量数据集** 定义为  $D_{\text{库存}}$ ，包含每个品类的月库存量，基本格式为三元组  $(c, m, I)$ ，其中：

- $c \in C$  表示品类；
- $m$  为日期，即该月份第一天
- $I \in \mathbb{R}^+$  表示月初时该商品库存量。

**销量数据集**：销量为每日记录一次，定义为  $D_{\text{销量}}$ ，基本格式为三元组  $(c, d, Q)$ ，其中：

- $c \in C$ ：品类。
- $d \in D$ ：日期。
- $Q \in \mathbb{R}^+$ ：在日期  $d$  时品类  $c_i$  的销售量。

仓库 ( $w \in W$ ), 则定义以下属性:

- 仓租日成本:  $\text{cost}(w) \in \mathbb{R}^+$ , 表示仓库  $w$  的每日租赁成本。
- 仓容上限:  $\text{capacity}(w) \in \mathbb{R}^+$ , 表示仓库  $w$  的最大存储容量。
- 产能上限:  $\text{production}(w) \in \mathbb{R}^+$ , 表示仓库  $w$  的最大处理能力(每日最大出库量)。

对于**品类关联度**, 定义品类关联度函数  $\text{assoc}(c_i, c_j)$ , 其中  $c_i, c_j \in C$ 。

- 未出现的品类关联组合, 其关联度为:  $\text{assoc}(c_i, c_j) = 0$ 。

对于**品类分类信息**, 进行如下定义:

- 高级品类编码  $\text{code}(c)$ : 品类的标识编码。
- 件型  $\text{type}(c)$ : 品类的件型 (包装类型)。

## 4.2 数据探索与标准化

在构建预测模型之前, 我们首先对数据中的缺失值进行了填充处理, 并将不同品类的销量数据整理为**连续的时间序列格式**, 以便于后续的聚类分析和预测建模。为消除不同品类数据之间的数量级差异, 我们为每个品类构建了相应的时间序列矩阵, 并对其进行了标准化处理。这一步骤有助于将各品类的销量数据调整到同一量纲, 从而确保模型能够有效地比较和分析各时间序列的趋势性变化。

### 4.2.1 缺失值处理

在实际数据收集过程中, 由于信息不全或系统延迟, 导致部分数据存在缺失值。为确保预测和优化的准确性, 需要对缺失数据进行合理填充。对于不同类型数据的缺失值处理, 我们采取以下策略:

**均值填充:** 对于库存量和销量等数值型数据, 应用同一品类的历史均值填充缺失值, 以保持数据的稳定性。

**零填充:** 对于小概率出现的异常缺失值, 如部分仓库的产能数据或关联度矩阵的稀疏值, 应用零值填充, 表示不相关或无产能限制。

**删除策略:** 对于个别数据缺失过多或严重缺乏连续性的样本, 直接删除, 以避免对模型产生较大偏差。

### 4.2.2 时间序列矩阵构建:

对于每个品类, 我们提取其库存量历史数据, 按照时间顺序排列, 形成长度为  $T$  的一维数组, 代表该品类的时间序列。通过遍历所有品类, 构造出一个时间序列矩阵  $X = \{x_i\}_{i=1}^C$ , 其中每行  $x_i = [x_{i1}, x_{i2}, \dots, x_{iT}]$  对应一个品类的数据序列。

库存量时间序列矩阵:

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,T_s} \\ s_{2,1} & s_{2,2} & \dots & s_{2,T_s} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \dots & s_{n,T_s} \end{bmatrix}$$

- $s_{i,j}$  表示品类  $c_i$  在第  $j$  个月的库存量。

销量时间序列矩阵:

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \dots & q_{1,T_q} \\ q_{2,1} & q_{2,2} & \dots & q_{2,T_q} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,1} & q_{n,2} & \dots & q_{n,T_q} \end{bmatrix}$$

其中,  $q_{i,j}$  表示品类  $c_i$  在第  $j$  天的销量

**时间序列标准化:**

由于不同品类的库存量&销量数据在数量级上存在较大的差异(从几十到几千不等),如果直接应用原始数据可能会导致模型更倾向于较大数量级的时间序列的拟合。为了解决这一问题,我们对每个品类的库存量序列进行标准化处理,确保模型在后续分析时,不会受到不同品类间数量级差异的干扰。

设库存量序列为  $x_i = [x_{i1}, x_{i2}, \dots, x_{iT}]$ , 则标准化后得到的时间序列  $x_i'$  为:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}$$

其中:

- $\mu_i$  为品类  $i$  的时间序列均值
- $\sigma_i$  为其标准差。通过变换, 每个品类的标准化序列  $x_i'$  均值为 0、方差为 1, 适用于模型的统一分析。

### 4.3 时间序列与相关性分析

在电子商务库存管理中, 商品数据的时间序列通常表现出**显著的波动性和复杂性**, (如图 4 所示的 category01 销量序列为例)。这种波动性主要体现在**变化趋势的多样性**上: 某些品类的库存量呈现**平稳**的变化趋势, 而其他品类则显示出明显的**季节性波动**, 甚至在较短时间内出现剧烈的波动和波峰。这种复杂的变化模式使得销量预测和库存管理面临较大的挑战, 要求采用更加精细的建模方法以有效捕捉数据中的趋势性和波动性特征。

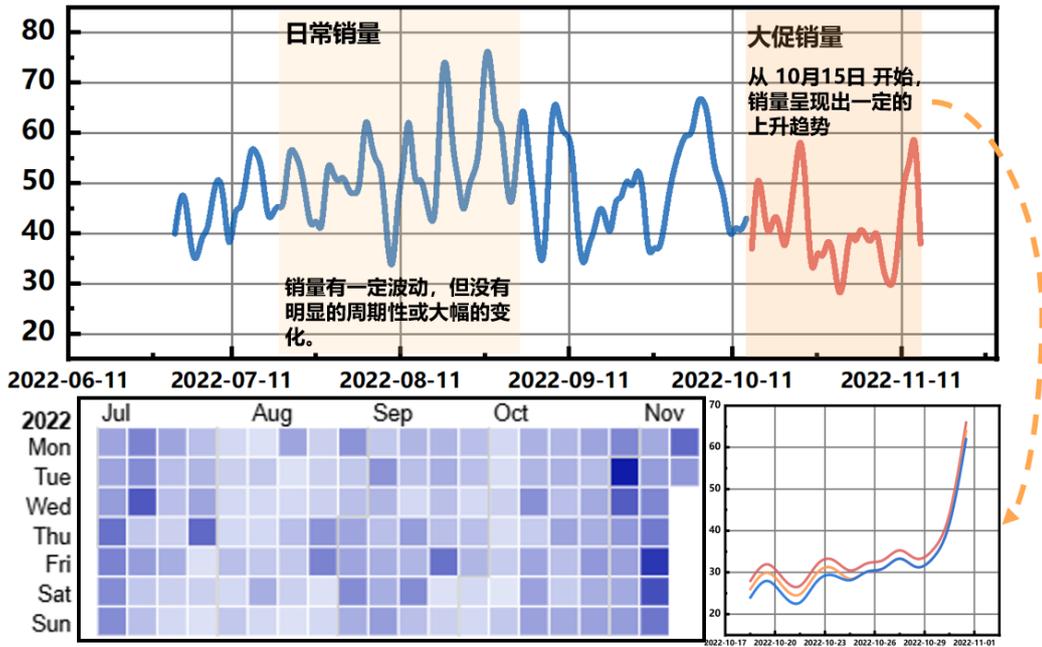


图 4 category01 在 2022 年销量时间序列统计

因此，在大促期间销量和库存量的预测中，采用聚类分析方法具有显著的优势。通过在预测模型中识别出具有相似趋势的品类，我们能够提取潜在的变化模式，从而使模型在预测未来时间序列时更加精准。为此，本研究采用了基于动态时间规整（Dynamic Time Warping, DTW）和 K-均值聚类（K-Means）的时间序列聚类方法，对各品类的时间序列数据进行分组。通过这一聚类分析，模型能够针对每个品类组的共同特征进行优化调整，进而提升大促期间销量和库存量预测的精度。

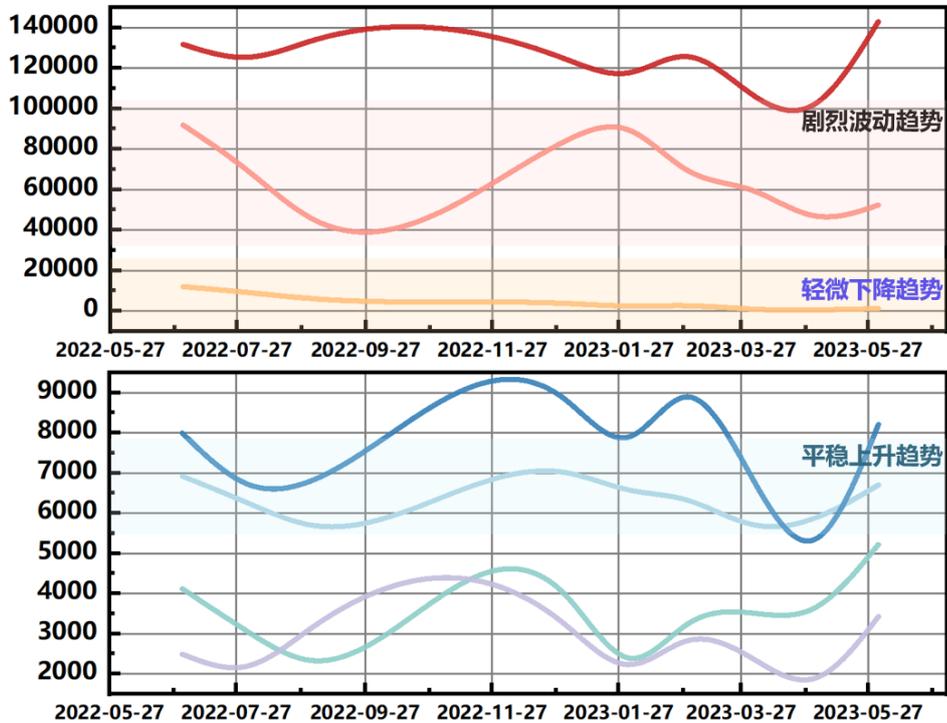


图 5 历史库存量时间序列趋势分类图

### 4.3.1 动态时间规整 (DTW) 距离

动态时间规整是一种时间序列相似性的度量方法，适合处理具有不同步长的时间序列。DTW 可以通过“弹性匹配”允许时间点的对齐发生偏移，从而更好地捕捉序列间的相似性。

对于两个时间序列  $x = [x_1, x_2, \dots, x_T]$  和  $y = [y_1, y_2, \dots, y_T]$ ，DTW 距离  $d_{\text{DTW}}(x, y)$  定义为：

$$d_{\text{DTW}}(x, y) = \min_{(i,j) \in \text{Path}} \sum |x_i - y_j|$$

其中，Path 表示对齐路径。而通过选择最优路径，DTW 可以最小化时间序列间的距离，从而适应时间上的不一致性。

### 4.3.2 基于 DTW 的 K 均值聚类

为了对各品类的时间序列进行聚类，我们在 DTW 的基础上使用 **K 均值聚类算法 (TimeSeries-KMeans)**，目标是最小化各时间序列到其聚类中心的 DTW 距离之和。

设所有时间序列组成的集合为  $X = x_1, x_2, \dots, x_N$ ，其中  $x_i$  表示第  $i$  个品类的时间序列，总共有  $N$  个品类。我们设定聚类数为  $K = 3$ ，旨在将时间序列集合划分为 3 个簇 ( $C_1, C_2, C_3$ )，使得簇内样本的趋势相似性最大化。目标函数为：

$$\min_{C_1, C_2, C_3} \sum_{k=1}^K \sum_{x_i \in C_k} \text{DTW}(x_i, \mu_k)$$

其中， $\mu_k$  为第  $k$  个簇的聚类中心（即簇内时间序列的“均值序列”）， $\text{DTW}(x_i, \mu_k)$  表示时间序列 ( $x_i$ ) 与聚类中心 ( $\mu_k$ ) 之间的 DTW 距离。

**初始阶段：** 随机选择 3 个时间序列作为初始聚类中心  $\mu_1, \mu_2, \mu_3$ 。

**迭代阶段**（直至收敛或达到最大迭代次数）：

(1) **分配步骤：**

对于每个时间序列  $x_i \in X$  和每个聚类中心  $\mu_k$ ，计算 DTW 距离：

$$d_{ik} = \text{DTW}(x_i, \mu_k)$$

将时间序列  $x_i$  分配到使得  $d_{ik}$  最小的聚类  $C_k$ ：

$$\text{令 } k^* = \arg \min_k d_{ik}, \text{ 将 } x_i \text{ 分配到 } C_{k^*}$$

(2) **更新步骤：**

对于每个聚类 $C_k$ ，重新计算聚类中心 $\mu_k$ ，即簇内所有时间序列的均值序列：

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

这里， $|C_k|$ 表示簇 $C_k$ 中的样本数量，时间序列的均值运算是在 DTW 度量下的取均值方法。

**终止条件：**当聚类中心 $\mu_1, \mu_2, \mu_3$ 稳定（即连续迭代中聚类中心不再变化）或达到预设的最大迭代次数时，停止迭代。

### 4.3.3 聚类结果分析

库存量时间序列聚类：

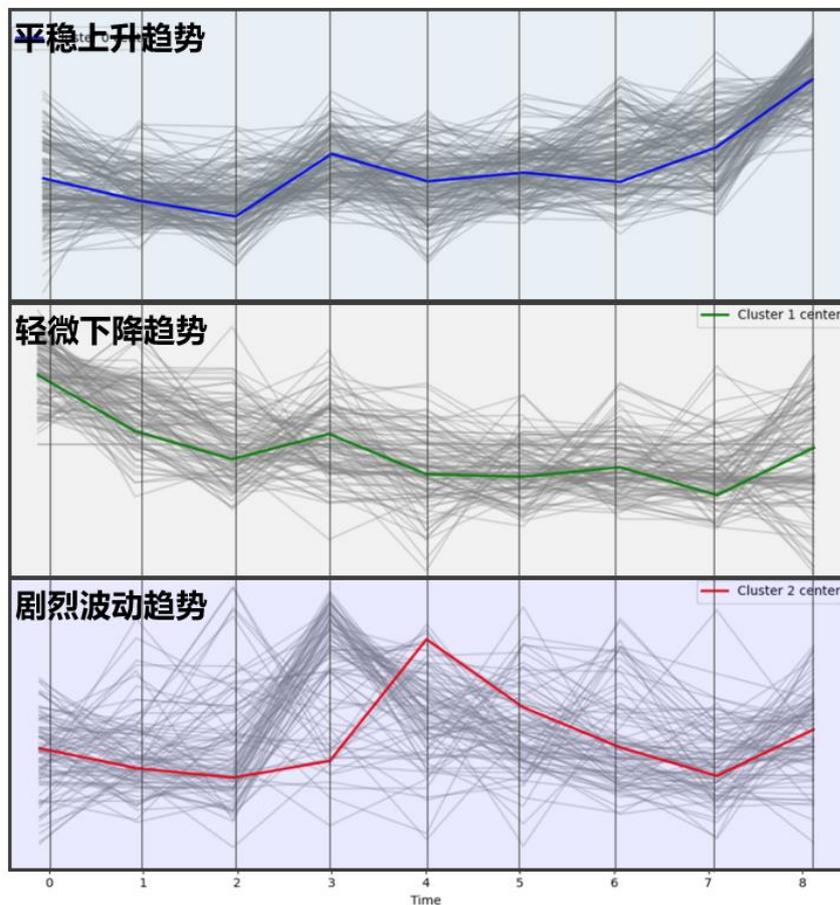


图 6 三类库存量-时间趋势

时间序列聚类结果如图 6 所示，每个聚类代表了一种库存量变化趋势：

**平稳上升趋势类（蓝色）：**这一类别的品类在大促期间显示出平稳的库存增长趋势。这可能与持续增长的市场需求或成功的促销活动有关。

**轻微下降趋势类（绿色）：**这一类别的品类在大促期间库存量呈现轻微下降。这可能是由于市场饱和、需求减少或竞争加剧导致的。

**剧烈波动趋势类（红色）：**这一类别的品类在大促期间库存量波动剧烈，可能受到季节性变化、市场波动或供应链问题的影响。

## 销量时间序列聚类：

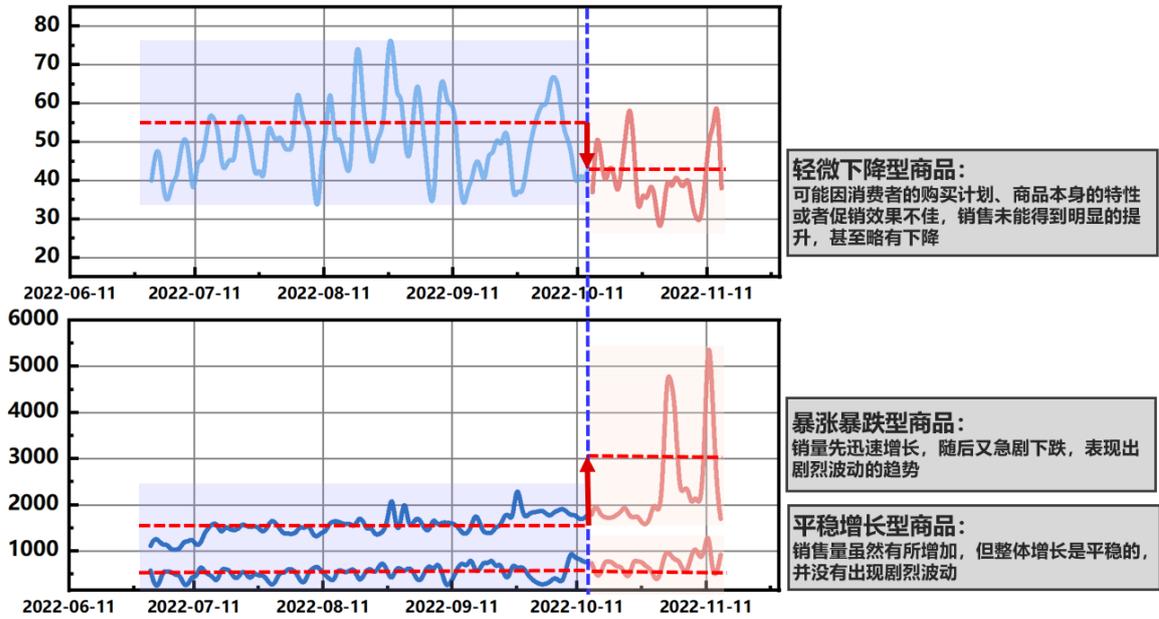


图 7 三类销量-时间趋势

我们对销量的聚类分析揭示了三种显著的销量趋势模式（如图 7）：

**轻微下降型商品：**这一类别的产品在大促期间内销量呈现轻微下降趋势。可能是由于市场饱和、消费者偏好转移或竞争产品的增加。

**暴涨暴跌型商品：**第二类别的产品在大促期间内表现出剧烈的销量波动，即暴涨暴跌型。这种模式可能与特定事件（如促销活动、季节性需求或供应链问题）有关。

**平稳增长型商品：**第三类别的产品在大促期间内显示出平稳的销量增长趋势。这表明这些产品在市场上的需求稳定增长，可能是由于产品本身的吸引力或有效的市场推广。

## 4.4 相关性分析

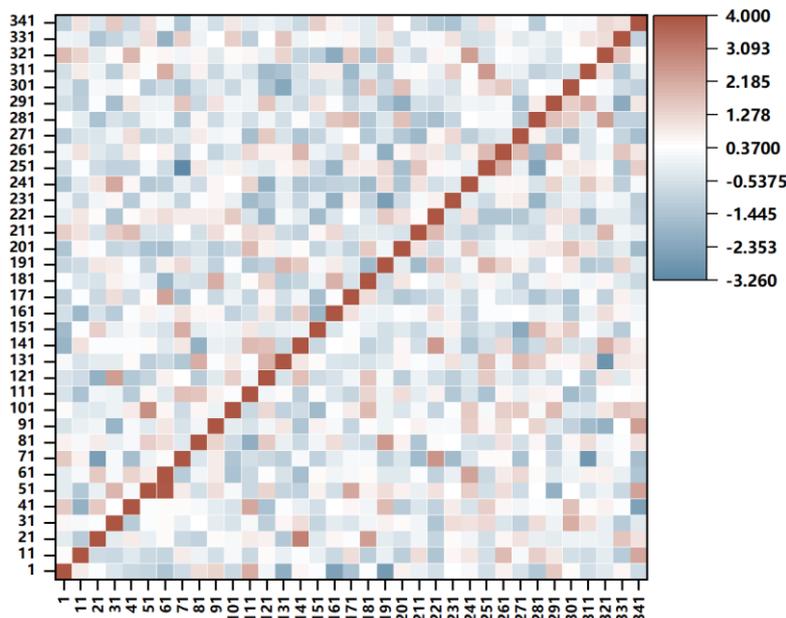


图 8 关联度热图

#### 4.4.1 品类-品类相关性分析

在分仓管理优化过程中，不同品类之间的关联度对于降低成本和提高分配效率具有重要影响。**高关联度的品类如果能够集中存放于同一仓库，将有助于减少仓库间的运输成本及协同成本。**因此，本研究基于附件中提供的商品品类关联度数据，进行了商品相关性分析（如图 8），以识别各品类之间的潜在关联，并为优化分仓策略提供数据支持：

以  $n$  为所有品类的数量，用  $c_i$  表示第  $i$  个品类； $r_{ij}$  表示品类  $c_i$  与品类  $c_j$  之间的关联度。由此，我们定义关联度矩阵  $R$  为：

$$R = [r_{ij}]_{n \times n}$$

根据关联度数据表，若  $r_{ij}$  未在原始数据中明确给出，则将其设定为 0，表示两者关联度较低，不产生明显的协同效益：

$$r_{ij} = \begin{cases} a_{ij} & \text{若 } (i, j) \text{ 在关联度数据表中给出} \\ 0 & \text{其他情况} \end{cases}$$

其中， $a_{ij}$  为关联度数据表中明确给出的关联度值。

不同品类间的关联度值分布可能差异较大，需要对关联度进行标准化，以确保在优化过程中各品类关联度影响的均衡性。为此，我们采用 **Z-score**（对关联度矩阵的离散值进行处理，使其均值为 0，方差为 1）：

$$z_{ij} = \frac{r_{ij} - \mu_r}{\sigma_r}$$

其中， $\mu_r$  和  $\sigma_r$  分别表示关联度  $R$  的均值和标准差， $z_{ij}$  为 *Z-score* 标准化后的值。

#### 4.4.2 高级品类-趋势-件型相关性分析

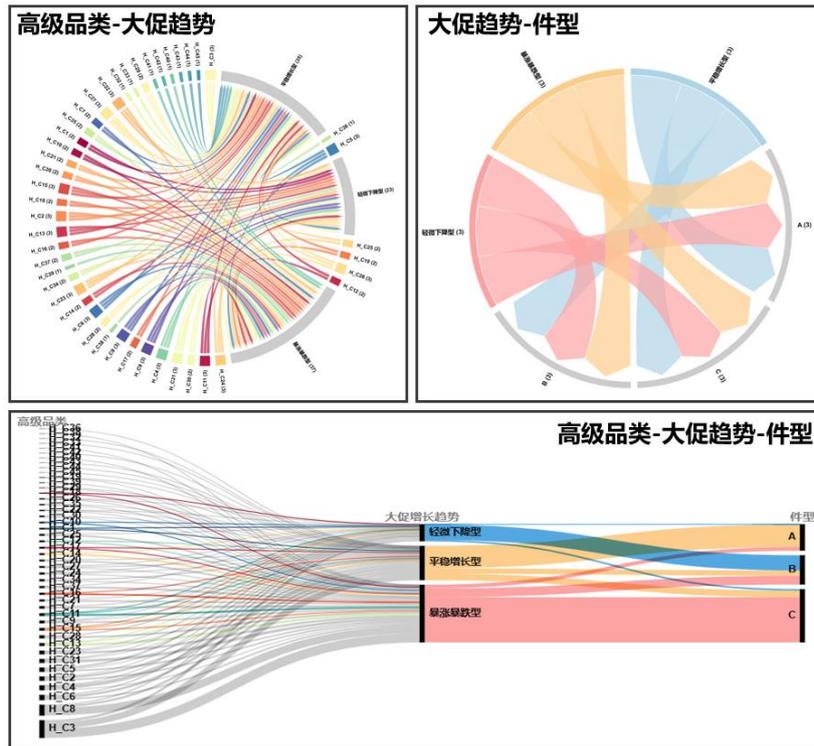


图 8 品类大促趋势与件型关联分析图

### 五、问题一模型建立与求解

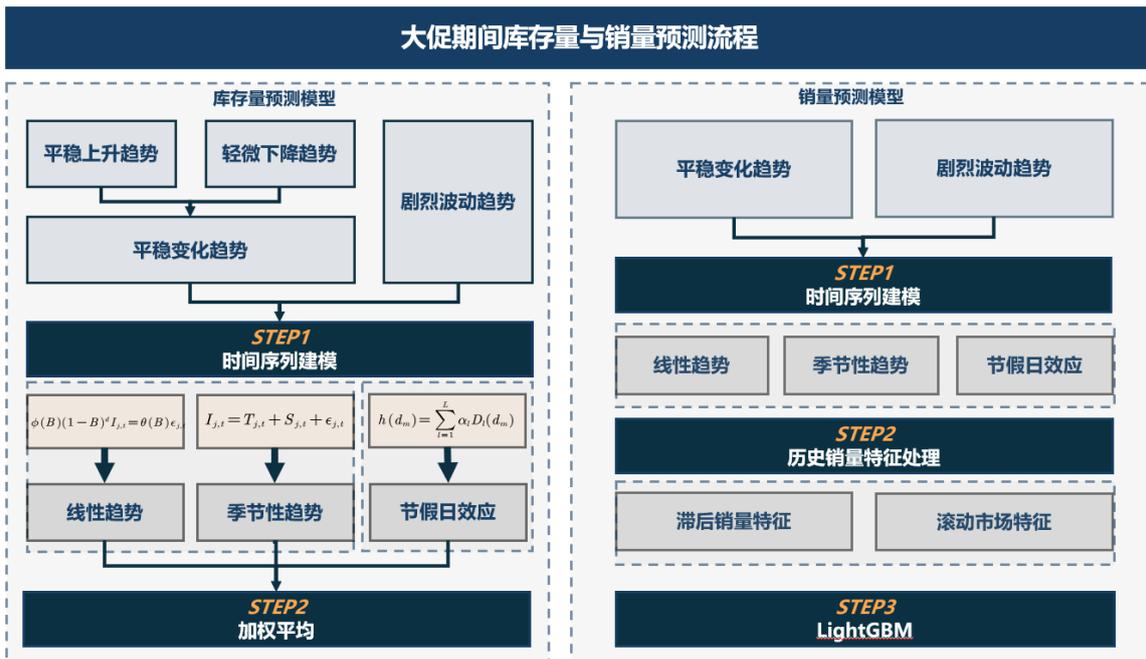


图 9 问题一求解流程图

## 5.1 库存量预测模型

针对大促期间的库存量预测，本研究根据库存变化的不同特征，选择了多种模型以提高预测准确性并增强对各类数据的适应性。对于呈现平稳变化的库存量，我们采用了 **ARIMAX 模型 (ARIMA + ETS)**，该模型能够有效捕捉时间序列中的趋势和季节性成分。对于库存量波动剧烈的品类，则采用了 **Prophet 模型**，该模型能够更好地处理具有复杂非线性变化的时间序列数据。通过这些模型的应用，我们能够全面把握库存数据的趋势性、季节性以及非线性特征，从而提高预测精度，为电商企业在日常运营及大促期间的库存管理和决策提供可靠支持，确保在搬仓阶段和大促运营期间的高效履约。

### 5.1.1 平稳变化趋势类库存：ARIMAX 模型 (ARIMA + ETS)

针对平稳变化趋势类库存（包括平稳上升和轻微下降趋势）的预测，我们采用了结合自回归积分滑动平均 (ARIMA) 与指数平滑法 (ETS) 的混合模型。**ARIMA 模型能够有效捕捉库存数据中的长期线性趋势**，利用自回归和移动平均部分对平稳变化的趋势进行建模，从而处理那些缺乏显著波动但呈现稳步增长或下降的库存数据。**ETS 模型则通过平滑方法进一步提炼库存数据中的长期趋势和季节性波动**，减少噪声的干扰，进而提供更加平稳的库存预测。通过将这两种模型进行融合，我们能够同时考虑库存数据的线性趋势与季节性变化，从而提高预测的准确性和稳定性。该混合模型在处理平稳上升趋势时，能够保持趋势的稳定性，并有效适应品类间可能存在的复杂关系，从而显著提升大促期间库存量预测的精度和可靠性。

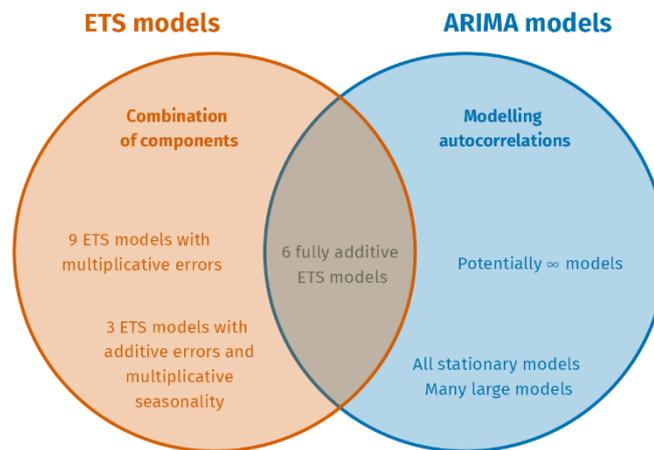


图 10 ARIMA/ETS 模型对比韦恩图

## I 时间序列建模

平稳变化趋势类库存量数据具有明显的**趋势性**和**季节性成分**，我们分别使用 ARIMA 和 ETS 模型来建模库存量的趋势和季节性特征。

### (1) 趋势特征提取

我们使用 ARIMA 模型来捕捉库存量的**线性趋势**。设库存量  $I_{j,t}$  为品类  $C_j$  在时间点  $t$

的实际库存量，则 ARIMA 的表达式为：

$$\phi(B)(1-B)^d I_{j,t} = \theta(B)\epsilon_{j,t}$$

其中：

- $\phi(B)$  是自回归（AR）部分的多项式， $\theta(B)$  是滑动平均（MA）部分的多项式， $B$  为滞后算子。
- $d$  是差分阶数，用于消除趋势。
- $\epsilon_{j,t}$  为白噪声残差。

我们通过拟合 ARIMA 模型，得到品类的预测值  $\hat{I}_{j,t}^{ARIMA}$ ，并用于后续模型融合。

## （2）季节性特征提取

我们通过平滑技术捕捉时间序列的季节性变化。假设库存量序列可以由以下三部分构成：

$$I_{j,t} = T_{j,t} + S_{j,t} + \epsilon_{j,t}$$

其中：

- $T_{j,t}$  为品类  $C_j$  在时间点  $t$  的趋势项；
- $S_{j,t}$  为品类  $C_j$  在时间点  $t$  的季节性项；
- $\epsilon_{j,t}$  为误差项。

通过平滑方法，我们可以得到每个品类在大促期间的预测值  $\hat{I}_{j,t}^{ETS}$ 。

## II 综合预测模型构建

我们采用加权平均的方法来融合两个模型的预测结果。这种方法考虑了不同模型在特定情况下的预测性能，通过赋予更高的权重给在特定时间段内表现更好的模型，以提高整体预测的准确性。

### （1）权重确定

首先，我们通过交叉验证来确定两个模型的权重，即在历史数据上比较两个模型的预测误差（如表 1），选择在大促期间表现更优的模型赋予更高的权重。

表 1 时间序列预测模型误差比较表

时间点	ARIMA模型 MSE	ARIMA模型 RMSE	ARIMA模型 MAPE	ETS模型 MSE	ETS模型 RMSE	ETS模型 MAPE
2022-10	0.15	0.39	3.5%	0.12	0.35	3.0%
2022-11	0.18	0.42	4.0%	0.11	0.33	2.8%
2022-12	0.14	0.37	3.2%	0.13	0.36	3.1%
2023-04	0.16	0.40	3.6%	0.14	0.37	3.3%
2023-05	0.17	0.41	3.9%	0.15	0.38	3.4%
2023-06	0.19	0.43	4.2%	0.16	0.40	3.6%

我们计算每个模型在测试集时间点上的平均误差：

对于 ARIMA 模型，平均误差  $\bar{E}_{ARIMA}$  为：

$$\bar{E}_{ARIMA} = \frac{1}{n} \sum_{i=1}^n E_{ARIMA,i}$$

对于 ETS 模型，平均误差  $\bar{E}_{ETS}$  为：

$$\bar{E}_{ETS} = \frac{1}{n} \sum_{i=1}^n E_{ETS,i}$$

我们使用误差的倒数作为权重，权重  $w$  表示为：

$$w_{ARIMA} = \frac{1/\bar{E}_{ARIMA}}{1/\bar{E}_{ARIMA} + 1/\bar{E}_{ETS}}$$

$$w_{ETS} = \frac{1/\bar{E}_{ETS}}{1/\bar{E}_{ARIMA} + 1/\bar{E}_{ETS}}$$

## (2) 融合公式

我们综合预测模型的预测值,得到**最终的预测结果**  $\hat{I}_{j,t}^{综合}$ ：

$$\hat{I}_{j,t}^{综合} = w_{ARIMA} \cdot \hat{I}_{j,t}^{ARIMA} + w_{ETS} \cdot \hat{I}_{j,t}^{ETS}$$

### 5.1.2 剧烈波动趋势类库存：Prophet 模型

在电商企业中，某些商品的库存量波动可能受到多种因素的影响，如季节性变化、促销活动以及外部冲击等。针对这类**库存量表现出剧烈波动的**时间序列数据，本研究采用了 **Prophet 模型**进行预测。该模型能够有效捕捉复杂的季节性波动和外部事件的影响，尤其适用于具有显著波动性的库存量预测，进而为库存管理和运营决策提供有力支持。

## I 时间序列建模

在对大促时期的剧烈波动库存量进行预测时，节假日效应是一个不容忽视的特征。我们通过 Prophet 模型引入节假日效应作为模型的一个关键特征，该模型以其在处理时间序列数据中的**强季节性特征和节假日效应**而闻名。通过将节假日效应纳入模型的关键特征集，可以有效捕捉和模拟特定节假日（大促时期）对数据的影响。

我们使用 Prophet 模型将时间序列分解为：

$$s(c_i, d_m) = g(d_m) + s(d_m) + h(d_m) + \epsilon_m$$

其中：

- $g(d_m)$ ：趋势函数，用于捕捉长期趋势变化；
- $s(d_m)$ ：季节性函数，用于模拟周期性变化；
- $h(d_m)$ ：节假日效应。
- $\epsilon_m$ ：误差项，表示未能解释的随机波动。

### (3) 趋势特征提取

我们对趋势函数采用分段线性模型，表示为：

$$g(d_m) = (k + a_m^T \delta)(d_m - t_{m-1}) + (m + a_m^T \gamma)$$

其中：

- $k$ ：初始斜率；
- $m$ ：初始截距；
- $a_m$ ：指示函数，用于标记变化点；
- $\delta$ ：斜率变化量；
- $\gamma$ ：截距变化量。

### (4) 季节性特征提取

对于季节性函数，我们采用傅里叶级数形式，表示为：

$$s(d_m) = \sum_{n=1}^N (A_n \cos(2\pi n d_m / P) + B_n \sin(2\pi n d_m / P))$$

其中：

- $N$ ：傅里叶项数，决定了季节性变化的复杂度；
- $P$ ：季节周期；
- $A_n$  和  $B_n$ ：傅里叶系数，决定了每个周期性成分的振幅。

### (5) 节假日效应提取

对于节假日效应，我们使用指示函数表示，表示为：

$$h(d_m) = \sum_{l=1}^L \alpha_l D_l(d_m)$$

其中：

- $L$ ：节假日数量。
- $\alpha_l$ ：第  $l$  个节假日的影响参数。
- $D_l(d_m)$ ：指示函数（若日期  $d_m$  为第  $l$  个节假日，则为 1，否则为 0）

### (6) 模型参数优化

我们通过最小化以下目标函数来优化模型参数：

$$\min_{k, m, \delta, \gamma, A_n, B_n, \alpha_i, \sigma^2} \sum_{m=1}^M (s(c_i, d_m) - (g(d_m) + s(d_m) + h(d_m)))^2 + \lambda \sum_{l=1}^L \alpha_l^2$$

通过参数的优化之后，我们得到了大促期间的预测值  $\hat{I}_{j,t}^{Prophet}$

## 5.2 销量预测模型

### 5.2.1 平稳变化趋势类销量：ARIMAX+LightGBM

针对平稳变化的销量预测，本研究提出了一种创新的方法，**将传统的时间序列分析模型（如 ARIMA 和 ETS）与先进的机器学习算法（LightGBM）相结合**。首先，利用 ARIMA 和 ETS 模型对销量数据进行初步预测，从中提取关键的趋势性和季节性特征。这些初步预测结果被整合为特征向量，并结合滞后值和滚动统计量，形成一个多维度的特征集。

随后，将该特征集输入到 LightGBM 模型中进行**进一步建模**。作为一种高效的梯度提升框架，LightGBM 能够对销量数据进行非线性建模，捕捉特征之间的复杂交互作用，而这一点是传统时间序列分析方法难以实现的。通过交叉验证和网格搜索技术优化 LightGBM 的超参数设置，确保模型在预测性能上达到最佳水平。

最终，所构建的混合模型不仅能够准确预测销量的平稳变化趋势，还能有效适应销量数据的非线性特征和复杂性，从而为电商企业在大促期间的销量预测提供了一种新的解决方案，提升了预测的精度和可靠性。

## I 时间序列建模

对于每个商品品类  $c_i$  在时间点  $t$ ，我们首先利用 ARIMA 和 ETS 模型得到对应的预测值：

**ARIMA 模型预测值：**

$$\hat{y}_{ARIMA,t}^{c_i} = \text{ARIMA}(q_{t-1}^{c_i}, q_{t-2}^{c_i}, \dots)$$

**ETS 模型预测值：**

$$\hat{y}_{\text{ETS},t}^{c_i} = \text{ETS}(q_{t-1}^{c_i}, q_{t-2}^{c_i}, \dots)$$

## II 历史销量特征处理

滞后销量特征和滚动统计特征是两种重要的特征工程，它们能够为模型提供关于时间序列动态变化的重要信息：

### 滞后销量特征 (Lag Features):

滞后特征通过考虑之前  $l$  个时间步的销量数据来构建。这些特征有助于捕捉时间序列中的短期依赖性。对于每个商品品类  $c_i$ ，在时间点  $t$  的滞后特征表示为：

$$q_{t-l}^{c_i}, \quad l = 1, 2, \dots, L$$

其中， $L$  是选定的最大滞后期数， $q_{t-l}^{c_i}$  表示品类  $c_i$  在时间  $t-l$  的销量。

### 滚动统计特征 (Rolling Statistics Features):

滚动统计特征通过在固定大小的滚动窗口上计算统计量来捕捉时间序列的局部特性。这些特征对于理解销量数据的波动性至关重要。对于每个商品品类  $c_i$ ，在时间点  $t$  的滚动统计特征表示为：

$$\text{MA}_t^{c_i} = \frac{1}{w} \sum_{k=1}^w q_{t-k}^{c_i}$$

其中， $w$  是滚动窗口的大小， $\text{MA}_t^{c_i}$  表示品类  $c_i$  在时间  $t$  的滚动平均值。

以及

$$\text{STD}_t^{c_i} = \sqrt{\frac{1}{w} \sum_{k=1}^w (q_{t-k}^{c_i} - \text{MA}_t^{c_i})^2}$$

其中， $\text{STD}_t^{c_i}$  表示品类  $c_i$  在时间  $t$  的滚动标准差，它衡量了在窗口  $w$  内销量的波动性。

通过将这些特征整合到模型中，我们可以增强模型对销量时间序列动态变化的理解，从而提高预测的准确性。这些特征不仅捕捉了销量的线性趋势，还考虑了销量的波动性和季节性变化，为 LightGBM 模型提供了丰富的信息，以进行更精确的销量预测。

## III 历史销量特征处理

综合了以上所有特征，我们构建了用于模型训练的特征向量  $\mathbf{x}_t^{c_i}$ ：

$$\mathbf{x}t^{c_i} = \begin{bmatrix} \hat{y}_{\text{ARIMA}, t^{c_i}}, \hat{y}_{\text{ETS}, t^{c_i}}, \\ qt - 1^{c_i}, q_{t-2}^{c_i}, \dots, q_{t-L}^{c_i}, \\ \text{MA}t^{c_i}, \text{STD}t^{c_i}, \\ \text{Month}_t, \text{DayOfWeek}_t, \text{IsHoliday}_t, \\ \text{Code}^{c_i}, \text{Type}^{c_i} \end{bmatrix}^T$$

## IV LightGBM 模型构建

### (1) 目标变量

我们将模型的目标变量定义为品类 $c_i$ 在时间 $t$ 的实际销量：

$$y_t^{c_i} = q_t^{c_i}$$

### (2) 模型定义

LightGBM 是一个基于梯度提升树 (Gradient Boosting Trees) 的模型，我们的目的是让其学习特征向量 $\mathbf{x}_t^{c_i}$ 与目标变量 $y_t^{c_i}$ 之间的映射关系：

$$\hat{y}_t^{c_i} = f(\mathbf{x}_t^{c_i} | \Theta)$$

其中， $f$  表示 LightGBM 模型的预测函数， $\Theta$  表示模型参数集合。

### (3) 损失函数

我们仍然采用均方误差 (Mean Squared Error, MSE) 作为损失函数，以衡量模型预测值与实际值之间的差异：

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^N (y_t^{c_i} - \hat{y}_t^{c_i})^2$$

其中， $N$  为训练样本的总数。

### (4) 模型训练

训练的目标是通过最小化损失函数来优化模型参数：

$$\Theta^* = \arg \min_{\Theta} L(\Theta)$$

### (5) 模型预测

利用训练完成的模型，我们对新的数据进行销量预测：

$$\hat{y}_t^{c_i} = f(\mathbf{x}_t^{c_i} | \Theta^*)$$

## V 模型优化与超参数调优

LightGBM 模型包含多个超参数，这些超参数的调优对于提高模型性能至关重要。超参数包括但不限于：

- 学习率  $\eta$

- 最大深度  $\text{max\_depth}$
- 子采样率  $\text{subsample}$
- 特征采样率  $\text{colsample\_by\_tree}$
- 正则化参数  $\lambda, \gamma$

通过定义超参数的搜索空间，我们遍历可能的参数组合，以找到使验证误差最小的参数配置。

### 5.2.2 剧烈波动趋势类销量：Prophet+LightGBM

针对销量在大促期间剧烈波动的品类，我们依然采用时间序列分析（Prophet）与机器学习（LightGBM）混合的方法。我们首先利用 Prophet 模型对销量数据进行初步预测，捕捉销量中的长期趋势、季节性变化和节假日效应，然后将 Prophet 的预测结果作为重要特征之一，整合到特征向量中。这个特征向量还包括滞后销量、滚动统计量以及其他可能影响销量的外部因素，如节假日、促销活动等。

最后，将构建的特征向量输入 LightGBM 模型，学习特征与销量之间的复杂非线性关系。通过优化 LightGBM 模型的参数，准确地预测销量的剧烈变化。

#### I 时间序列模型：Prophet

我们通过加性模型来分解时间序列的不同成分，包括趋势、季节性和节假日效应。表达式为：

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

通过最大化对数似然函数来进行参数估计，得到  $g(t)$ ,  $s(t)$ ,  $h(t)$  等成分的最佳估计。

#### II 特征向量构建

在使用 Prophet 进行时间序列预测后，得到的预测结果（即  $\hat{y}(t)$ ）将作为一个重要特征被加入到后续的机器学习模型中。这个特征表示了模型对时间序列的长期趋势、季节性变化和节假日效应的捕捉。

• **滞后销量**：滞后销量指前一时间步的销量，它提供了销量变化的历史信息，是预测销量波动的一个重要特征。表达式为：

$$\mathbf{X}_{\text{lag}} = [y(t-1), y(t-2), \dots, y(t-k)]$$

其中， $y(t-k)$  表示时间  $t-k$  时刻的销量， $k$  为滞后阶数。

- **滚动统计量**：滚动统计量是对过去一段时间内的销量进行平滑处理的统计量，用

于捕捉销量的局部趋势和波动性。对于滚动均值：

$$\mathbf{X}_{\text{rolling mean}} = \frac{1}{k} \sum_{t=1}^k y(t)$$

其中， $k$  是滚动窗口的大小。

• **外部因素**：除了 Prophet 预测结果、滞后销量和滚动统计量，其他可能影响销量的外部因素，如节假日、促销活动等，也是构建特征向量的重要组成部分。这些外部因素通过一个二元变量或定量特征进行表示，例如：

$$\mathbf{X}_{\text{external}} = [I(t, \text{Holiday1}), I(t, \text{Promo})]$$

### III LightGBM 模型的应用

在构建了特征向量之后，接下来将其输入到模型中。这里我们依然使用 LightGBM 模型，能够处理大规模数据集并学习特征之间的复杂非线性关系。

LightGBM 的目标是通过一系列弱分类器（即决策树）的集成来优化预测性能，目标函数表示为：

$$\mathcal{L}(\theta) = \sum_{i=1}^N (L(y_i, \hat{y}_i) + \Omega(f))$$

通过优化目标函数  $\mathcal{L}(\theta)$ ，LightGBM 可以得到最优的树结构和分裂点，从而准确地预测销量的剧烈波动。

## 5.3 求解结果讨论

### 5.3.1 库存量大促预测

#### I 预测结果

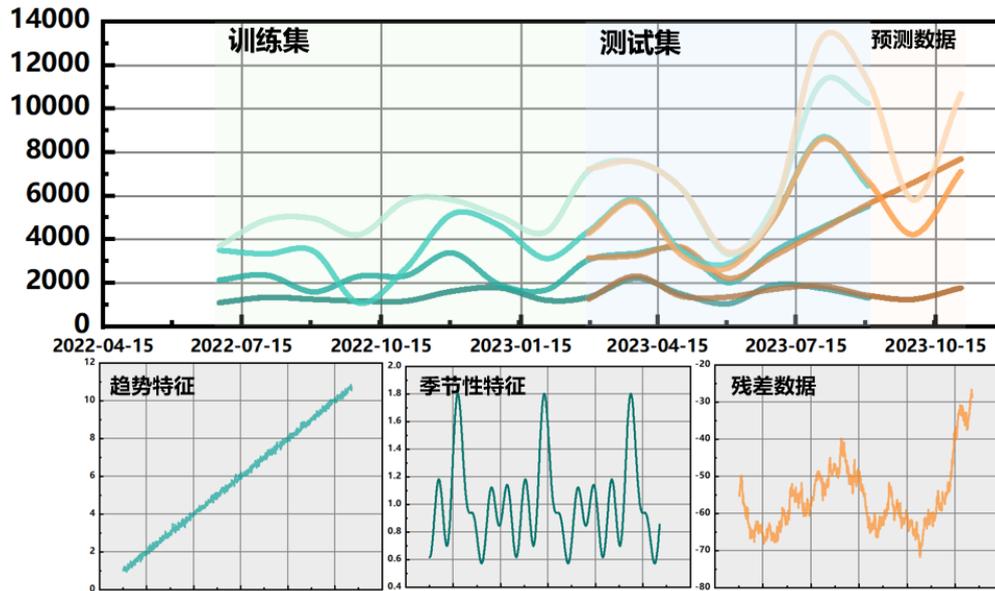


图 11 库存量时间序列-预测结果

总体而言，构建的预测模型在处理常规时期的库存量变化时展现出较为稳定的预测性能，能够有效地拟合数据中的长期趋势和季节性波动。然而，在面对大促期间的特殊情况时，模型的预测精度显著受到挑战，表现为预测残差的增大以及预测曲线与实际观测数据之间的偏差扩大。大促期间库存量的剧烈波动超出了模型在常规数据集上训练过程中所学习到的规律，这导致模型在应对突发性变化时存在一定的局限性，未能有效捕捉这些异常波动。因此，模型在大促期间的预测结果呈现出较大的波动性，未能充分反映库存量的剧烈变化。

表 2 2023 年大促期间库存量预测数据

month	category25	category47	category84	category141
May, 2023	1,496	3,545	3,484	6,350
Jun, 2023	1,052	2,010	2,891	3,319
Jul, 2023	1,910	3,432	4,963	5,545
Aug, 2023	1,757	4,519	8,686	11,206
Sep, 2023	1,319	5,520	6,456	10,240
<b>Oct, 2023</b>	<b>1,244</b>	<b>6,607</b>	<b>4,227</b>	<b>5,806</b>
Nov, 2023	1,775	7,695	7,109	10,689

## II 误差分析

表 3 库存量预测数据分步误差

	步1_绝对误差	步2_绝对误差	步3_绝对误差	均方误差(MSE)
category1	1	1	3	0
category31	0	0	2	0
category61	100	76	0	1
category91	9	6	100	0
category121	50	72	0	0
category151	64	100	0	1
category181	0	1	0	0
category211	1	0	25	0
category241	10	26	54	0
category271	12	12	0	0
category301	2	4	0	0
category331	46	48	0	0

总体而言，误差分析结果表明，该多步骤预测模型在不同类别上表现出明显的性能差异。模型在大多数类别上展现了较高的预测准确性，特别是在第三阶段，多个类别的绝对误差降至零，表明模型能够有效地拟合这些类别的数据。低均方误差（MSE）值进一步验证了模型在整体预测性能上的优越性，尽管在少数类别中，仍存在一定的偏差。

### 5.3.2 销量大促预测

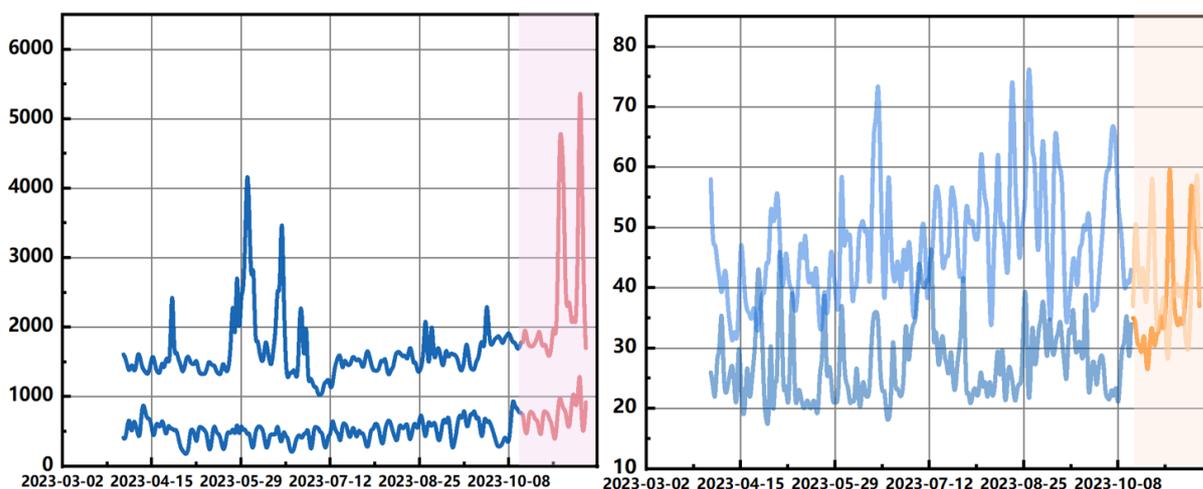


图 12 大促期间销量预测结果

从图表分析来看，销量预测结果整体上呈现出令人满意的效果：

**趋势捕捉：**预测模型在捕捉销量的整体趋势方面表现出色。在大多数时间段内，预测销量与实际销量的波动趋势高度一致，特别是在销量的高峰期，模型能够精确预测销量的增长趋势。

**误差控制：**图表中的波动范围展示了预测的误差和不确定性。尽管存在一定程度

的预测误差，但整体上误差控制在一个合理范围内，表明模型能够在大部分情况下保持较为稳定的预测结果。

总体而言，该销量预测模型在捕捉销量趋势、控制误差并在不同市场环境下提供稳定预测方面表现出色。其结果能够为企业在库存管理、生产规划和市场策略制定等方面提供有效的支持。

## 六、问题二模型建立与求解

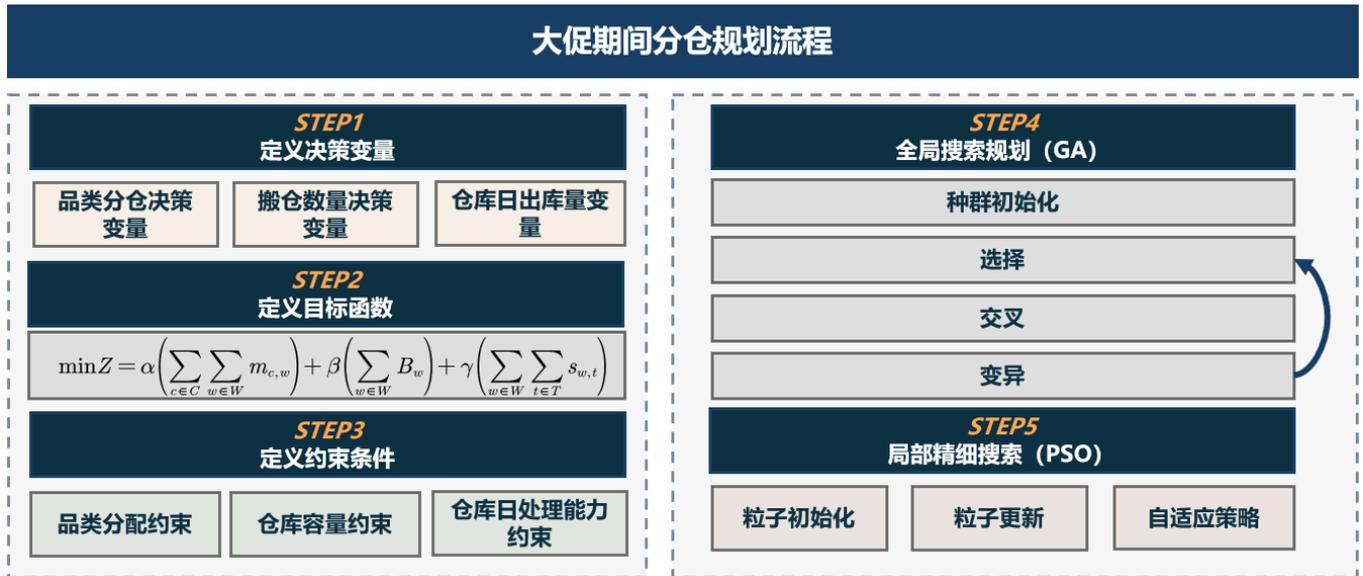


图 13 问题二求解流程图

### 6.1 问题描述与建模

#### 6.1.1 符号与参数定义

- 品类集合： $C = \{c_1, c_2, \dots, c_N\}$
- 仓库集合： $W = \{w_1, w_2, \dots, w_M\}$
- 大促期间的日期集合： $T = \{t_1, t_2, \dots, t_L\}$

参数：

- 大促前现状平均库存（10月1日-10月14日）： $[S_{\text{pre}}(c), \quad \forall c \in C]$
- 预测的大促期间库存量： $[I_{c,t}, \quad \forall c \in C, \forall t \in T]$
- 预测的大促期间每日销量： $[D_{c,t}, \quad \forall c \in C, \forall t \in T]$

- 仓库容量：  $[SC_w, \forall w \in W]$
- 仓库日最大处理能力（产能上限）：  $[PC_w, \forall w \in W]$
- 允许的爆仓比例：
  - 爆仓天数比例上限：  $\theta_d = 10\%$
  - 单次爆仓程度上限：  $\theta_p = 20\%$

## 6.1.2 决策变量

### I 品类分仓决策变量

定义二元变量  $x_{c,w}$  用于表示品类  $c$  是否分配到仓库  $w$ ：

$$x_{c,w} = \begin{cases} 1, & \text{如果品类 } c \text{ 分配到仓库 } w \\ 0, & \text{否则} \end{cases}, \quad \forall c \in C, \forall w \in W$$

取值为 1 时，表示品类  $c$  被分配到仓库  $w$ ；取值为 0 时，则表示未分配。这一定义为我们进一步构建优化模型提供了基础，可以用于确保每个品类根据业务需求和资源限制被合理分配到相应的仓库。

### II 搬仓数量决策变量

定义连续非负变量  $m_{c,w}$  用于表示品类  $c$  从当前仓库搬运到仓库  $w$  的搬仓件数，以确保搬仓操作的物理可行性和非负性：

$$m_{c,w} \geq 0, \quad \forall c \in C, \forall w \in W$$

**搬仓数量**的计算则基于品类  $c$  的当前存储位置和目标仓库  $w$ 。设品类  $c$  在大促前的现有仓库为  $w_{old,c}$ ，则搬仓数量  $m_{c,w}$  可以通过以下方式确定：

$$m_{c,w} = \begin{cases} S_{pre}(c) \cdot x_{c,w}, & \text{如果 } w \neq w_{old,c} \\ 0, & \text{如果 } w = w_{old,c} \end{cases}, \quad \forall c \in C, \forall w \in W$$

这种搬仓数量的计算方法确保了在大促期间，品类的库存能够根据新的分仓规划进行有效转移，同时避免了不必要的搬仓操作，从而优化了搬仓成本和效率。

### III 仓库日出库量变量

在电商物流管理中，准确预测每个仓库在特定日期的出库量是优化仓储和物流资源

分配的关键。为此，我们定义了仓库日出库量变量 $O_{w,t}$ ，它表示仓库 $w$ 在大促期间的每一天 $t$ 的总出库量：

$$O_{w,t} = \sum_{c \in C} D_{c,t} \cdot x_{c,w}, \quad \forall w \in W, \forall t \in T$$

#### IV 爆仓指示变量

"爆仓"指仓库的出库量超过了其产能上限 $PC_w$ ，需要尽可能避免，为了在数学模型中表示和处理爆仓情况，我们引入了爆仓指示变量 $z_{w,t}$ ：

$$z_{w,t} = \begin{cases} 1, & \text{如果 } O_{w,t} > PC_w \\ 0, & \text{否则} \end{cases}, \quad \forall w \in W, \forall t \in T$$

线性化约束：

由于 $z_{w,t}$ 是一个二元变量，直接在模型中使用可能会造成非线性问题。为了线性化这个问题，我们采用大 $M$ 法，引入以下约束：

##### 1. 爆仓上限约束：

$$O_{w,t} - PC_w \leq M \cdot z_{w,t}, \quad \forall w \in W, \forall t \in T$$

##### 2. 爆仓下限约束：

$$O_{w,t} - PC_w \geq \epsilon - M(1 - z_{w,t}), \quad \forall w \in W, \forall t \in T$$

#### V 爆仓程度变量

监控和控制仓库的爆仓程度是确保运营效率和客户满意度的关键。为了量化仓库在特定日期的爆仓程度，我们引入了连续非负变量 $s_{w,t}$ ，表示仓库 $w$ 在日期 $t$ 的爆仓程度，即超出产能上限的比例：

$$s_{w,t} = \frac{O_{w,t} - PC_w}{PC_w} \cdot z_{w,t}, \quad \forall w \in W, \forall t \in T$$

#### VI 总爆仓天数变量

定义总爆仓天数变量 $B_w$ ，它表示仓库 $w$ 在整个大促期间发生爆仓的总天数：

$$B_w = \sum_{t \in T} z_{w,t}, \quad \forall w \in W$$

### 6.1.3 目标函数

目标函数直接反映了业务目标和约束的平衡。针对大促期间的库存管理和分仓规划，我们提出了一个综合目标函数，旨在最小化总搬仓件数、爆仓天数以及爆仓程度：

$$\min Z = \alpha \left( \sum_{c \in C} \sum_{w \in W} m_{c,w} \right) + \beta \left( \sum_{w \in W} B_w \right) + \gamma \left( \sum_{w \in W} \sum_{t \in T} s_{w,t} \right)$$

其中：

- $\alpha, \beta, \gamma$  是权重系数，用于平衡不同目标的重要性。

### 6.1.4 约束条件

#### I 品类分配约束

品类分配约束确保每个品类  $c$  被且仅被分配到一个仓库  $w$ ：

$$\sum_{w \in W} x_{c,w} = 1, \quad x_{c,w} \in \{0, 1\}, \quad \forall c \in C, \forall w \in W$$

#### II 仓库容量约束

仓库容量约束确保每个仓库  $w$  的总库存量不超过其容量上限  $SC_w$ ：

$$\sum_{c \in C} S_{\max}(c) \cdot x_{c,w} \leq SC_w, \quad \forall w \in W$$

#### III 仓库日处理能力约束

仓库日处理能力约束考虑了正常运营和爆仓情况下的处理能力：

$$O_{w,t} \leq PC_w \cdot (1 + \theta_p \cdot z_{w,t}), \quad \forall w \in W, \forall t \in T$$

其中：

- $PC_w$  是仓库  $w$  的正常处理能力（产能上限）。
- $\theta_p$  是一个参数，表示爆仓时允许超出正常处理能力的比例。

## IV 搬仓数量约束

搬仓数量约束确保每个品类 $c$ 从其当前仓库 $w_{old,c}$ 搬到新仓库 $w$ 的数量满足以下条件:

$$m_{c,w} \geq S_{pre}(c) \cdot (x_{c,w} - \delta_{w,w_{old,c}}), \quad \forall c \in C, \forall w \in W$$

### 6.2 分仓规划：混合遗传-粒子群算法（GA-PSO）

在大促期间多仓分配问题中，问题的复杂性使单一优化算法可能无法高效地找到全局最优解。混合遗传-粒子群算法（GA-PSO）结合了遗传算法（GA）和粒子群优化（PSO）的优点，利用遗传算法的全局搜索能力和粒子群优化的局部收敛优势，在解决多目标优化问题上表现出色。

#### 6.2.1 遗传算法：全局搜索

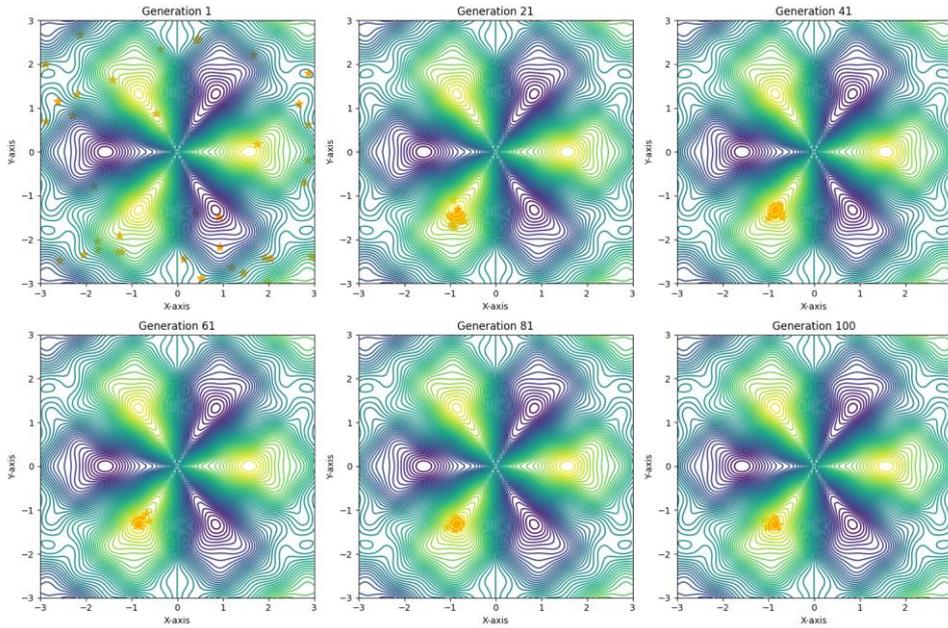


图 14 GA 全局搜索收敛过程

全局搜索阶段旨在初始化具有高多样性的解集，以确保覆盖广泛的搜索空间，提升全局探索能力。

#### I 种群初始化

生成一个初始种群，其中每个个体表示一个可能的仓库分配方案，以确保种群的多样性。

我们采用整数编码，将每个个体表示为一个长度为 $n$ 的染色体向量：

$\mathbf{X} = [x_1, x_2, \dots, x_n]$  其中， $x_i \in \{1, 2, \dots, m\}$  表示将第 $i$ 个品类分配到编号为 $x_i$ 的仓库。

对于种群中的每个个体  $\mathbf{X}^{(k)}$  ( $k = 1, 2, \dots, N$ )，在允许的仓库编号范围内随机分配品类，使得初始种群多样性最大化：

$$x_i^{(k)} = \text{RandInt}(1, m), \quad \forall i \in \{1, 2, \dots, n\}$$

## II 选择操作

根据适应度值选出适应度较高的个体，为下一代的繁殖做准备。

我们选择了**轮盘赌选择策略**，确保具有较高适应度的个体有更大的概率被选中，但同时保留了一定的随机性，防止过早收敛。选择过程可以视为对适应度值的加权随机采样。

## III 交叉操作

我们使用单点交叉的方法，通过交换父代个体的部分基因，产生新的子代个体，探索新的解空间：

**1. 交叉概率：**首先，我们设定一个交叉率  $P_c = 0.8$ ，意味着在种群中每对父代个体之间，有 80% 的概率发生交叉操作。

**2. 选择交叉点：**如果决定进行交叉，我们随机选择一个交叉点  $k$  ( $1 \leq k < n$ )， $n$  是个体的基因长度)。这个交叉点将决定哪些基因被交换。

**3. 生成子代：**根据交叉点  $k$ ，我们交换父代个体的部分基因来生成两个新的子代个体：

$$\begin{aligned} \text{子代1: } & [x_1^{(1)}, x_2^{(1)}, \dots, x_k^{(1)}, x_{k+1}^{(2)}, \dots, x_n^{(2)}] \\ \text{子代2: } & [x_1^{(2)}, x_2^{(2)}, \dots, x_k^{(2)}, x_{k+1}^{(1)}, \dots, x_n^{(1)}] \end{aligned}$$

## IV 变异操作

变异操作可以表示为：

$$x_i' = \begin{cases} \text{RandInt}(1, m) & \text{with probability } P_m \\ x_i & \text{otherwise} \end{cases}$$

通过这样的变异操作，我们可以进一步探索解空间，增加种群的多样性，从而有助于找到更优的解。

## 6.2.2 粒子群优化：局部精细化搜索

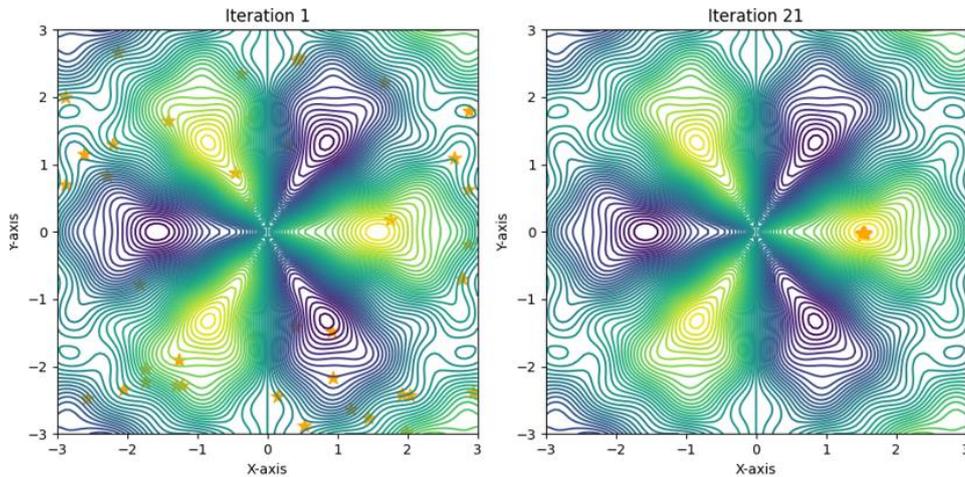


图 15 PSO 局部精细化搜索过程

粒子群优化算法（PSO）是一种基于群体的优化算法，模拟鸟群觅食的行为，通过个体和全局信息的交互来更新解的状态。我们将 PSO 用于在遗传算法生成的较优解基础上进行局部精细化搜索，以加速收敛到全局最优解。

### I 初始化粒子位置和速度

#### (1) 粒子位置初始化:

我们根据遗传算法（GA）生成的较优解初始化粒子的位置。由于 GA 已经产生了一个较优解，编码为  $\mathbf{X}_{GA}$ ，我们将该解作为 PSO 的初始位置。

$$\mathbf{X}_i(0) = \mathbf{X}_{GA}, \quad \text{for all particles } i$$

#### (2) 粒子速度初始化:

我们将粒子的速度进行随机初始化，以避免粒子在初期阶段出现过大的跳跃。假设粒子速度  $\mathbf{V}_i(0)$  在  $t = 0$  时刻的初始值为随机数。

$$\mathbf{V}_i(0) = \text{Rand}(\mathbf{V}_{\min}, \mathbf{V}_{\max}), \quad \text{for all particles } i$$

### II 速度和位置更新

#### (1) 速度更新公式:

粒子的速度通过以下公式更新:

$$\mathbf{V}_i(t+1) = w \mathbf{V}_i(t) + c_1 r_1 (\mathbf{P}_i - \mathbf{X}_i(t)) + c_2 r_2 (\mathbf{G} - \mathbf{X}_i(t))$$

该公式由三个部分组成:

- 惯性部分  $w \mathbf{V}_i(t)$ : 保持粒子的当前速度，增强搜索稳定性;

- 个体经验部分  $c_1 r_1 (\mathbf{P}_i - \mathbf{X}_i(t))$ ：根据粒子自身的历史最优解进行调整；
- 社会经验部分  $c_2 r_2 (\mathbf{G} - \mathbf{X}_i(t))$ ：根据群体的最优解进行调整。

(2) 位置更新公式：

粒子位置的更新根据粒子的速度和当前位置进行计算：

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1)$$

即粒子的下一位置由当前的速度和当前位置的加和得到。

### 6.2.3 自适应调整策略

粒子的适应度值用于衡量粒子的优劣，适应度函数包括多个目标。对于仓库规划问题，表示为：

$$f(\mathbf{X}_i) = \alpha \cdot C(\mathbf{X}_i) + \beta \cdot D(\mathbf{X}_i)$$

其中， $\alpha$ 和 $\beta$ 为加权系数，用于平衡总仓储成本和品类关联之间的权重。粒子群中的每个粒子都需要计算其适应度值，并根据适应度值来更新个体最优解 $P_i$ 和全局最优解 $G$ 。

**个体最优解更新：**

若粒子的适应度值优于其历史最优适应度，则更新个体最优解：

$$f(\mathbf{X}_i) < f(\mathbf{P}_i) \Rightarrow \mathbf{P}_i = \mathbf{X}_i$$

**全局最优解更新：**

若当前粒子的适应度值优于群体中所有粒子的适应度，则更新全局最优解：

$$f(\mathbf{X}_i) < f(\mathbf{G}) \Rightarrow \mathbf{G} = \mathbf{X}_i$$

### 6.3 求解结果讨论

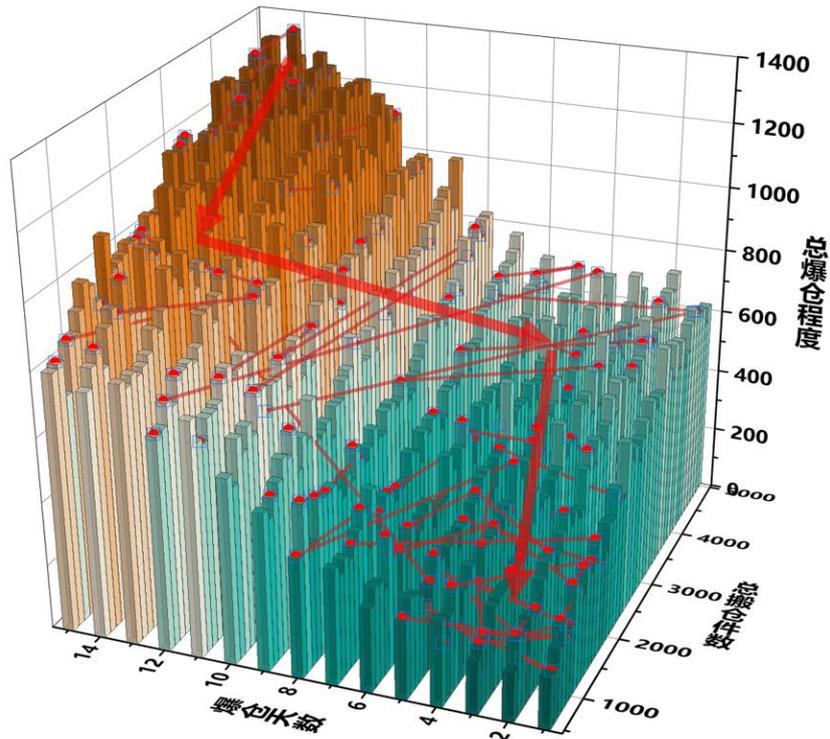


图 16 GA-PSO 收敛过程

图 16 展示了分仓算法在优化过程中的收敛过程，目标函数值随着迭代次数增加呈现下降趋势，表明算法在逐步接近最优解。后期迭代中，目标函数值的改善逐渐减小，表明收敛速度放缓并趋于稳定，且在最后几次迭代中波动较小，进一步确认算法已收敛。算法成功避免了局部最优解，快速接近全局最优解，且具有较高的收敛效率。参数选择对收敛性亦有影响，整体表明算法设计合理，能够有效解决分仓问题。

表 4 仓储规划最优分配表

	warehouse	warehouse1	warehouse2
category1	54	93	111
category31	85	19	62
category61	31	46	105
category91	96	108	88
category121	101	32	110
category151	17	104	45
category181	13	106	1
category211	122	106	55
category241	115	15	7
category271	18	33	46
category301	14	40	45
category331	58	83	119

## 七、模型的综合评价

本研究构建的时间序列预测模型与优化模型在解决电商大促期间的库存管理与仓储分配问题上展现了显著的优势。时间序列预测模型通过融合 ARIMAX、Prophet 和 LightGBM 等技术,不仅能有效捕捉数据中的趋势性、季节性和非线性特征,还能灵活适应不同波动类型的数据,显著提高了预测的准确性和稳健性。而在仓储分配优化方面,混合遗传-粒子群优化(GA-PSO)算法充分结合了遗传算法的全局搜索能力和粒子群优化的局部收敛优势,在复杂约束条件下高效逼近最优解。模型设计中的自适应调整策略进一步提升了算法的鲁棒性与适用性。

从实验结果来看,这些模型在应对大规模电商促销场景中的实际需求时均表现出色,不仅提供了精准的销量与库存预测数据,还生成了优化的仓储分配方案,大幅降低了物流成本和爆仓风险,同时提升了资源利用效率和客户满意度。整体上,研究成果具有较强的实用价值与推广潜力,可为电商企业的供应链管理决策提供可靠的理论支持和实践指导。

## 参考文献

- [1] 杨毅,卢益清.基于电商用户行为预测的电商物流研究[J].物流科技,2024,47(23):77-79.DOI:10.13714/j.cnki.1002-3100.2024.23.021.
- [2] 郑泓洋,刘云飞.跨境电商背景下的物流创新创业路径研究[J].中国航务周刊,2024,(49):54-56.
- [3] 章雪飞.企业成本管理存在的问题及对策探析——以京东物流为例[J].现代营销(下旬刊),2024,(11):107-109.DOI:10.19932/j.cnki.22-1256/F.2024.11.107.
- [4] 童思永,侯海荣,苏振宇.我国西南地区电商物流网络空间结构与形成机制[J/OL].西安理工大学学报,1-15[2024-12-13].<http://kns.cnki.net/kcms/detail/61.1294.N.20241125.1634.002.html>.
- [5] 黄琳琳.低碳背景下生鲜农产品电商高质量发展路径研究[J].中国商论,2024,33(21):45-48.DOI:10.19699/j.cnki.issn2096-0298.2024.21.045.
- [6] 杨伯靖,窦学诚.农村电商物流助推产业融合发展实践路径研究——基于 W 市新型农业经营主体的田野调查[J].物流科技,2024,47(21):76-78+92.DOI:10.13714/j.cnki.1002-3100.2024.21.018.
- [7] 杨芸.跨境电商退货逆向物流选择和策略[J].对外经贸实务,2024,42(08):55-63.
- [8] 高磊.基于精益生产的电商物流运营路径与效果[J].中国经贸导刊,2024,(14):112-114.
- [9] 常荣华.物流管理的跨组织创新:以电商与新零售为例[J].物流科技,2024,47(20):22-25.DOI:10.13714/j.cnki.1002-3100.2024.20.006.
- [10] 邢力元,赵霞.“一带一路”倡议背景下山西省跨境电商物流的发展路径研究[J].物流科技,2024,47(20):72-74+78.DOI:10.13714/j.cnki.1002-3100.2024.20.019.

## 附录 代码