

双目标贪心优化下的企业原料订购运输方案

摘要

生产企业的原料订购与运输计划受采购成本、运输损耗、原料类型等各种因素影响，且部分特征与时间相关。本文基于某企业所需三种原料属性与多家原料供应商、原料转运商的历史交易信息，建立存贮模型下的多目标贪心优化等模型，对该企业未来 24 周的订购运输决策提供方案。

针对问题一，对给定的 402 家供应商进行评价分析。对所有供应商进行数据预处理，清洗掉历史参考价值较低的供应商。对筛选留下的 176 家供应商，建立反映**合作可靠性的评价指标体系**，遴选出 9 类代表性指标量化各方面供货特征，其中重点关注订单信誉度、货量波动等方面。运用**熵权法**确定各指标的初始体系权重。分析初始结果，运用**层次分析法**对模型进行修正：将 9 项指标分为三类，引入人工判断二次调节权重。该模型保留了数据本身区分属性，且符合人类经验判断，同时通过了一致性检验。应用此模型对 176 家供货商打分排序，选择前 50 家最优的供货商列在文中，并列举前 5 家的指标雷达图。

针对问题二，“最经济”方案的制定属于**双目标优化问题**：采购成本最少，运输损耗最少。将第一问中 50 家供应商依据供货时间特征分为**平稳型、周期型**，分别构造供应模式。建立企业的**经济进货存贮模型**，简化原料储存策略。在多个子模型基础上建立第二问的**双目标定量规划式**，分别定义采购、转运、存储过程的目标函数和约束关系。参照外层供应商、内层转运商的最优评分排序，针对不同商家供货量及供应-转运交接关系，构建**双层贪心算法**并论证其合理性。提出四种**关键式**对供货量作决策，实验求解对比各关键式结果，选取最优关键式嵌入模型。该模型求得的方案平均采购成本为 20488 单位/周，平均转运损耗为 135.33 单位/周，合作商家数量大约为 16 家。在模型中对各商家参数与关键环节引入供货参数等**随机变量**，使模型更加吻合真实情景，并提高其自动调整订购策略的能力，加强了模型的稳健性。

针对问题三，以压缩存储成本为目标，对第二问模型进行**定向权重重排筛选操作**。首先论证存储成本最小化等价于推论“尽量增加 A 类订单、减少 C 类订单”。基于此推论，对原 50 家供应商**二次评分**，指定重判权重：A 类 1.05，B 类 1，C 类 0.95，处理原评分得到新体系下供应商评分及排序。另外在原贪心算法中添加**定向筛选环节**，提高对 C 类原料的动态排斥度。应用新模型求解，得 A 类/C 类周供应量比值为 7.34，平均转运损耗为 72.73 单位/周，验证了推论方向的正确性，并给出了具体供货量与商家匹配决策方案。

针对问题四，产能提升的方法应当基于前文贪心优化模型。扩大供应商范围至全部 176 家有效商家。论证在提高产能前提下，企业的产品收益不会亏损，该优化方向是合理的。另外，此处存贮模型与订购运输方案满足独立不干扰性。对提高后的产能进行建模，提出**动态产能、静态产能**两种模式：产能的动态演变使该模型所得方案可根据当前状态累积提升产能消耗，具备进化性，适用于发展期企业；静态产能保证了企业各量的稳定性，相关比例可定制单周产能的个性化分配方案，适用于稳定期企业。本文选取动态产能求解实验，得出该方案下动态产能平均量为 $43299m^3$ /周，相比原固定产能提升了 53%，24 周前后自身提升了 27%。

关键词：双目标优化 存贮模型 贪心 评价指标体系

1 问题重述

1.1 问题背景

已知某建筑板材生产企业需要对自身产能原料的订购与运输进行未来 24 周的计划，使得任何时候都能够满足企业未来两周的产能需求，同时尽量使方案更加经济可行。原材料总体分为 A, B, C 三种类型。

原料从购入到生产大体经过以下三个环节：订货、转运与存储。这三个环节中涉及的对象除了该生产企业之外，还有原材料供应商（“供应商”）和运输的物流公司（“转运商”）；各供应商、转运商均有多家可供选择。在订货环节中，每家供应商只生产一种货物，企业根据自身的产能需求对某些供应商提出下一周原材料需求（“订货量”）；但供应商的实际生产受各种因素影响，实际供货量（“供货量”）与订货量不完全一致，并且根据合约企业会全部收购供货量的货物。而在转运途中，原材料由于路途损耗（以“损耗率”记录），最终抵达仓库的量将少于供货量（“接受量”）；一家供货商的原料在一周内只由一家转运商运输。

一年以 48 个生产周计算，企业的产能需求为一周 2.82 万立方米，每立方米生产需要特定的 A 或 B 或 C 原料量。三个环节中均会产生成本，其中订货成本各有差别（以 C 为标准记录相对价格），而转运与存储成本三者相同。A, B, C 的产能值与相对价格见表 1：

原料	每立方米产能所需 (m^3)	采购相对价格
A	0.6	120%
B	0.66	110%
C	0.72	100%

表 1: 原材料相关数据表

企业仓库存储的要求为，任何时候的库存应当保证未来两周的需求。

1.2 问题提出

现在给出两个附件，其中附件 1 包含了 402 家供应商在 5 年内 240 周自身原料的订货量及对应供应量。订货量为 0 时说明该周该供应商没有供货订单。附件 2 包含 8 家转运商 5 年内运输的损耗率，其中损耗率为 0 说明该转运商该周没有进行转运作业。

由问题背景及附件，尝试帮助该企业拟订合理的原料订购与运输方案：

- (1) 问题一：对于附件 1 中的 402 位供货商的供货特征进行分析评价，从企业需求保证考虑出发，选出最适于合作的 50 家供应商。
- (2) 问题二：参考问题一的分析评价，选择至少某几家供应商进行订货需求，在满足仓库要求的基础上，以“订购方案最经济”和“转运损耗最少”为目标，制定未来 24 周的订购转运方案。并且对该方案的实施效果进行分析。

- (3) 问题三：注意到 A, B, C 的生产成本与单位产值关系，考虑在压缩转运存储成本的目标下，可以尽量提高 A:C 产品量的比值等等。基于该“转运存储成本的压缩”，同时延续问题二“转运损耗最少”，制定新的订购转运方案。对方案效果进行分析。
- (4) 问题四：该企业的产能具有了提高的能力，故根据当前原材料情况出发，探讨每周产能可以提升多少，并且给出订购转运方案。

2 问题假设

一、供货商和转运商的特征属性不考虑季节变化，但考虑周期性变化。

由于多数供货商数据缺少足量季节样本分析，且观察数据可知特征随时间变化趋势不明显，故本题中不考虑季节影响。注意：此处季节指周的年坐标，与“周期”类循环量概念不同。

二、一家转运商在一周中可运多家供应商的货物。

由题可知一家供应商一周内的货物均由一家转运商负责，而转运商的最大运输量、转运商个数、供应商单次供货量反映可知：若假设一家转运商只运一家供应商货物，则实际接收量难以满足一周要求，故原假设不成立。

三、企业的储存容量是无限的，初始库存满足两周产能约束。储存成本为一次性，即持续储存不消耗成本。

为满足题给约束，规定企业的初始库存有 $28200 \times 2m^3$ 。在题目中未给出储存容量的约束，且未给出三种货物采购、运输、储存之间的相对成本比例，故无法对持续性储存消耗建模。考虑简化情况，即单次储存消耗。

四、商家的信誉、交货准确度等属性是确定的。

假设 5 年中每家商家的交货偏差、货量波动、周期等属性固定不变，或有确定的对应关系。忽略掉 5 年中偶然因素对其的影响，以便于提取商家准确内在属性。

五、企业对于库存中 A,B,C 货物的选取消耗规则是自动的。

该题解重点考虑原料的采购、运输方案，对于企业仓库中每周不同货物选取规则进行简化。企业总是会自动选取最优方案消耗，如，当 A 料较多，而 B,C 料紧缺时，优先利用 A 料转化产能。基于此点考虑，除开问题三我们不关心仓库中三种货物分别存储量，故该题的存贮模型可简化为转化后的“产值”的量。

3 符号说明

符号	含义说明	单位
k	周数 ($1 \leq k \leq 24$)	次序
i	50 家供应商排名 ($1 \leq i \leq 50$)	次序
j	转运商排名 ($1 \leq j \leq 8$)	次序
$d_{i,k}$	第 i 个供应商在第 k 周的订货量	m^3
$s_{i,k}$	第 i 个供应商在第 k 周的供货量	m^3
$t_{i,k}$	第 i 个供应商在第 k 周的接收量	m^3
$j_{i,k}$	第 i 个供应商在第 k 周对接的转运商	编号
w_k	企业第 k 周的库存产能	m^3
Δw_k	企业第 k 周新接收的产能	m^3
v_k	企业第 k 周消耗的动态产能	m^3
C_k	企业第 k 周的采购成本	C 单位价格 / m^3
L_k	企业第 k 周的运输损失产能	m^3
p_i	第 i 个供应商的偏差率	%
Dp_i	第 i 个供应商偏差率的标准差	%
l_j	第 j 个转运商的损耗率	%
Dl_j	第 j 个转运商损耗率的标准差	%
r_i	第 i 家供应商对应产品类别的相对成本	-
z_i	第 i 家供应商对应产品类别的单位产能转化率	-
$n_{\hat{i}}$	第 \hat{i} 个转运商在 240 周中的总订单数 ($1 \leq \hat{i} \leq 240$)	笔
$m_{\hat{i}}$	第 \hat{i} 个转运商在 240 周中的完成订单数 ($1 \leq \hat{i} \leq 240$)	笔

表 2: 符号说明表

注：表中未出现的符号在文中均有详细说明。

4 问题分析

4.1 问题一的分析

该问主要对 402 家供货商的情况进行整体评价分析，以便于后续方案的建模与求解。评价分析的思路流程如下：

首先对数据进行预处理。依据附件 1，对于 240 周内所得样本订单不足的供货商，由于其数据过少，难以在忽略偶然因素的影响下确定其供货特点，故对其进行排除。该合格指标可通过订单数量、总订货量等等进行判断。

构建供货特征评价指标，对预备供货商集进行量化分析。此处的指标评定参考文献并结合该数据集分析进行构建。根据历史订单情况给每个供货商打分，并且引入层次分析法、熵权法等打分式

模型，将各指标融合多角度判断评价。

根据该供应商评价模型，确定 50 家最具合作潜力的供应商，列表给出在文中。

4.2 问题二的分析

第二问基于前一问给出的 50 个供应商，结合 8 个转运商，确定未来 24 周的理想订购转运方案，以达到订购成本最少化、转运损耗最小化的目标。最后对该方案进行实施效果的分析。

该问内含多层模型，外层为双目标定量规划问题，内层为经济进货存贮模型，同时还包含了第一问的商家建模。建模时首先分析存贮模型的基本属性，再将其嵌套入目标规划中作为约束部分。另外从时间角度考虑细化供应商模型，将问一中 50 家按照供货量时间特征分为“平稳型”和“周期型”，分类讨论其供货模式。

在模型的求解部分，基于优化的目标与模型中排名因素，自然考虑贪心算法的应用。对采购、转运环节展开讨论，提出一种双层贪心算法，论证贪心在此处优化的合理性；并讨论关键式-商家订购量计算公式的选择，对不同公式进行程序求解、对比分析得出可行方案。在最后的效应分析中，画出最优方案中各类型的供货变化、总库存变化状态、成本变化等指标，并进行解释说明。

4.3 问题三的分析

题目对订货运输方案要求进行更改，提出了压缩成本的角度，并且得出推论应使 A 类原料尽可能多、C 类尽可能少。应当对这条结论进行证明解释，再加以应用。压缩成本与单位产能密度、原料性价比相关，分析三种原料这方面的性质。

利用推论构思新的模型算法思路。从供应商排序模型出发，设计新评价方式使 A 类供应商的评价提高、C 类评价降低，引入调整权重做此步处理。除此之外，对原先贪心算法的实现过程添加筛选机制，使 C 类商家被选中的频率进一步降低，可从优先度相近的其余供应商作替代考虑。

最后对实验结果的 A、C 类商家比例，新方案的运输损耗量进行展示及分析说明。

4.4 问题四的分析

问题四的要求与问题二、三相对，将前两问的约束条件转化成了最优化目标，而先前的最优化目标在此处不做限制，因此模型上可继承第二问模型框架思路。不同的工作为，此处产能的消耗不再是原先规定定值，考虑两种情况：动态产能变化与静态产能变化。分别对这两种情况进行建模、推理、讨论，依据更新后的模型进行引入随机参数的模拟，重复实验并分析其代表性效益量在产能提升前后的变化。

动态产能与静态产能的思路不一样，动态主要约束条件为存贮模型警戒点，而静态产能需要进行解析分析，推算得出表达式。

5 问题一的供应商评价选择

5.1 建立评价模型

5.1.1 数据预处理

首先, 根据历史 5 年供应商的供货总量, 分离开供货总量过小的供应商, 使得数据更有参考价值。作出图1可知, 选取总供货量 100 为分界线, 先行考虑大于 100 的供应商, 共筛选得到 176 家供应商。

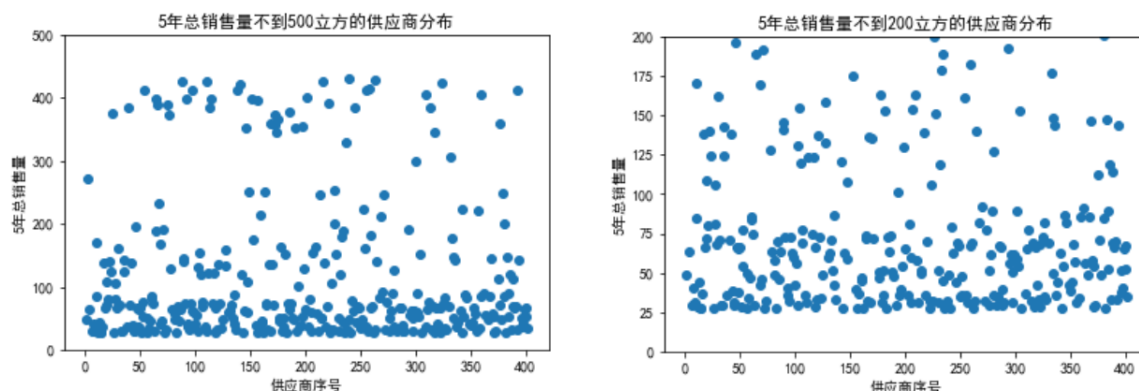


图 1: 历史总销售量供应商分布

同时, 可以检验本题中不存在“延迟交货”这一现象, 即并不存在某订单量为 0, 而相应的供货量不为 0 的情况。

5.1.2 指标选择

整个供应链系统是一种人工和自然相结合的多变量、多目标、多约束条件的复杂非线性开放系统, 供应商系统是这其中的一个子系统。对于这类系统的指标设计, 应遵循如下原则:

(一) 指标系统性。由于系统受到多重因素影响, 所以选择的参数要尽可能全面、系统地反映供应商目前地综合水平, 并对未来有一定指导。

(二) 指标科学性。将绝对数指标和相对数指标相结合, 通过绝对数指标反映出供应商总量上、规模上地情况, 通过相对数指标反映出速度和比率等。两类指标相辅相成, 结合分析, 可以更为准确地反映实际情况。

(三) 指标实用性。这主要包括指标的可计算性及计算所需数据的可行性。指标具有一定的灵活性, 使得供应商根据情况合理利用。

为了量化第 i 家供应商的供货特征, 从企业合作能力、绩效、灵活性等角度定义如下指标³:

指标	计算方法	数学表达式
供货及时性 $f_i^{(1)}$	$\frac{\text{准时供货次数}}{\text{总订货次数}} \cdot 100\%$	$\frac{m_i}{n_i} \cdot 100\%$
交货准确度 $f_i^{(2)}$	$(\frac{\text{供货量}-\text{订货量}}{\text{订货量}})^2$ 的平均值	$\frac{1}{n_i} \sum_{i=1}^{240} (\frac{s_{ik} - d_{ik}}{d_{ik}})^2$
供货波动性 $f_i^{(3)}$	$\frac{(\text{供货量}-\text{订货量}) \text{ 的标准偏差}}{(\text{供货量}-\text{订货量}) \text{ 的平均值}}$	$\frac{n_i \cdot SD(s_{ik} - d_{ik})}{\sum_{i=1}^{240} (s_{ik} - d_{ik})}$
供货量方差 $f_i^{(4)}$	供货量在题给 240 周下的方差	Ds_i
产品占有率 $f_i^{(5)}$	$\frac{\text{供货量}}{\text{此类材料总供货量}} \cdot 100\%$	$\frac{1}{n_i} \sum_{i=1}^{240} \frac{s_{ik}}{\sum_{\text{此类材料}} s_{ik}} \cdot 100\%$
信誉度 $f_i^{(6)}$	每周的交易准确度 $\cdot \ln(\text{订货量})$	$\frac{s_{ik} - d_{ik}}{d_{ik}} \cdot \ln d_{ik}$
平均忙期量 $f_i^{(7)}$	供货量 > 5 时供货量的平均值	$\frac{1}{n_i} \sum_{d_{ik} > 5} s_{ik}$
批量柔性 $f_i^{(8)}$	每周 $\frac{\text{供货量}}{\text{订货量}}$ 最大能增加的百分比	$\max_k \frac{s_{ik} - d_{ik}}{d_{ik}}$
供给弹性 $f_i^{(9)}$	$\frac{\text{供货量方差}}{240 \text{ 周供货量平均值}} \cdot 100\%$	$\frac{n_i \cdot Ds_i}{\sum_{i=1}^{240} s_{ik}} \cdot 100\%$

表 3: 指标确定

下面给出这 9 个指标的实际意义 [1]:

- $f_i^{(1)}$ 供货及时性反映供应商订单的完成率, 鉴于上文已经验证了不存在“延迟交货”的情况。
- $f_i^{(2)}$ 交货准确度衡量了供货量与订货量之间的偏差情况。
- $f_i^{(3)}$ 供货波动性反映供货与订货的差量这组数据的离散程度, 体现了供货是否稳定。
- $f_i^{(4)}$ 供货量方差体现供应商所能供货的波动性大小, 这是一个极小型指标。
- $f_i^{(5)}$ 产品占有率反映供应商在供应链中的重要性, 以此反映它对于保障生产的重要性。
- $f_i^{(6)}$ 信誉度考虑了在订货数量大小有差别时, 供货商的供货数据有效性并不相同。能够很好满足更大量的订货, 比满足较小的订货, 应当更有参考价值。这是一个极小型指标。
- $f_i^{(7)}$ 平均忙期量比单纯计算供货量的平均值更有意义, 它去除了供应商没有订单而导致的供货量为 0 的情况。这个指标在第二问也有不小的作用。
- $f_i^{(8)}$ 批量柔性用来衡量供应商的速度, 如果企业增加订单, 供应商在供给批量中最大能增加的百分比。
- $f_i^{(9)}$ 供给弹性形容的是供应商能够同时满足大订货量和小订货量的能力。

5.1.3 评价体系的建立

(一) 熵权法初步确定权重

为了尽可能的精确客观，选择熵权法对于上述 9 个指标确定权重，打分评价，用各个供应商的得分来反映它们对于保障企业生产的重要性。

熵权法用于对有限个决策进行评价，通过对不同指标的信息熵进行对比，确定指标的权重，最终决定最优策略。通常来说，如果某个指标的信息熵越小，表明不同决策对应该指标的变化程度越大，提供的信息也就越多，起到的作用越大，它的权重也就越大。它的具体步骤如下：

Step1. 数据标准化

设 X_{ij} 表示评价对象 i 在第 j 个指标处的取值。对于极大型指标和极小型指标，应用不同的方法对其进行标准化。对极大型指标，定义标准化后的值 $Y_{ij} = \frac{X_{ij} - \min_i X_j}{\max_i X_j - \min_i X_j}$ ；对极小型指标，定义标准化后的值 $Y_{ij} = \frac{\max_i X_j - X_{ij}}{\max_i X_j - \min_i X_j}$ 。

Step2. 求出各指标的信息熵

根据信息熵的定义，一组数据的信息熵

$$E_j = -\frac{1}{\ln n} \sum_{j=1}^n p_{ij} \ln p_{ij} \quad (5.1.1)$$

其中 $p_{ij} = \frac{Y_{ij}}{\sum_{i=1}^n Y_{ij}}$ ，如果 $p_{ij} = 0$ ，定义 $\lim_{p_{ij} \rightarrow 0} p_{ij} \ln p_{ij} = 0$ 。

Step3. 确定各指标的权重

利用（2）中得到的信息熵，计算出权重

$$W_j = \frac{1 - E_j}{k - \sum_i E_i} \quad (5.1.2)$$

（二）层次分析法适度修正

由于熵权法仅是根据数据自身混乱程度，来比较得出各个指标的权重。但根据实际操作结果来看，这难免会出现与事实不吻合的情形，需要在适度范围内修正。于是加入人工干预，以熵权法得到的权重作为参考，应用层次分析法改进此评价体系。

层次分析法是一种系统性的分析方法。简洁使用，需要定量数据。

本问中将 9 个指标归类并修改为二级指标，新定义三个一级指标：绩效评价、战略意义、潜力估计，得到如下图2所示的评价指标体系。二级指标相关部分的权重依据熵权法确定；一级指标相互间的权重查阅文献，利用层次分析法得到，具体步骤如下：

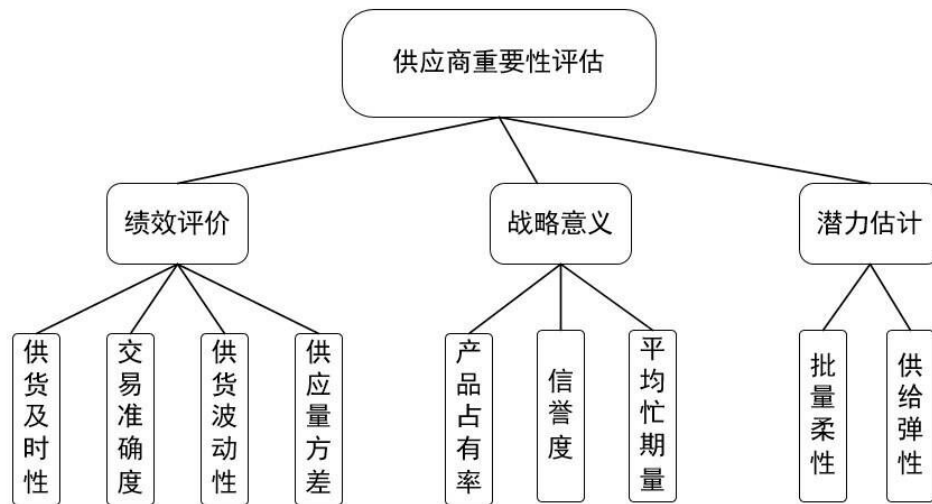


图 2: 评价指标体系

Step1. 计算比较矩阵的特征值。将最大特征值的特征向量归一化，作为权向量。确定各因素的权重。

Step2. 一致性检验，防止确定权重随意而导致矛盾。引入一致性指标 CI 和随机一致性指标 RI ，计算一致性比率 $\frac{CI}{RI}$ ，当小于 0.1 则满足条件。

5.2 模型求解

5.2.1 供货特征的量化分析

用交易型、战略型、大额型来描述供应商的供货特征。通常来说，交易型指为数众多，但交易金额较小的供应商；战略型指公司战略发展必需的几家供应商；大额型指交易数额巨大，但是战略意义一般的供应商。[2]

这三种特征在所给的供应商中都有所体现。对于交易型特征为主的供应商，它们供应量不大、灵活性较强，但是增长潜力有限，对于保障企业生产重要性有限，可以预见它们的排名较后，这在下文评价体系求解中得到验证。对于战略型或者大额型特征为主的供应商，它们生产规模较大，有较好的财务状况，往往对该企业生产保障起到重要作用。

除此以外，求出 9 个指标对应的值。注意到指标中供货量方差 $f_i^{(4)}$ 、供货波动性 $f_i^{(3)}$ 、交货准确度 $f_i^{(2)}$ 是极小型指标其余是极大型指标，分别用不同的方法将它们标准化，并且均化为极大型指标，便于观察。具体见附录 A。

5.2.2 供应商重要性的评价与分析

首先，运用 SPSSAU 求出各指标权重，如下图3蓝字所示。譬如“供给弹性”指标，它在评价保障生产的重要性时作用较为次要，但因为其离散程度较大，解得的权重相应较大，与具体情况矛

盾，需要进行合理范围内的修正。

然后，利用层次分析法，根据文献与经验，给出比较矩阵 $\begin{bmatrix} 1 & 0.2 & 0.167 \\ 5 & 1 & 0.769 \\ 6 & 1.3 & 1 \end{bmatrix}$ ，解得绩效评价、战略意义、潜力估计对应的权重为 40.467%,51.219%,8.314%，且通过一致性检验。结合先前二级指标的权重，得到修正后的指标权重，如下图3黄字所示。

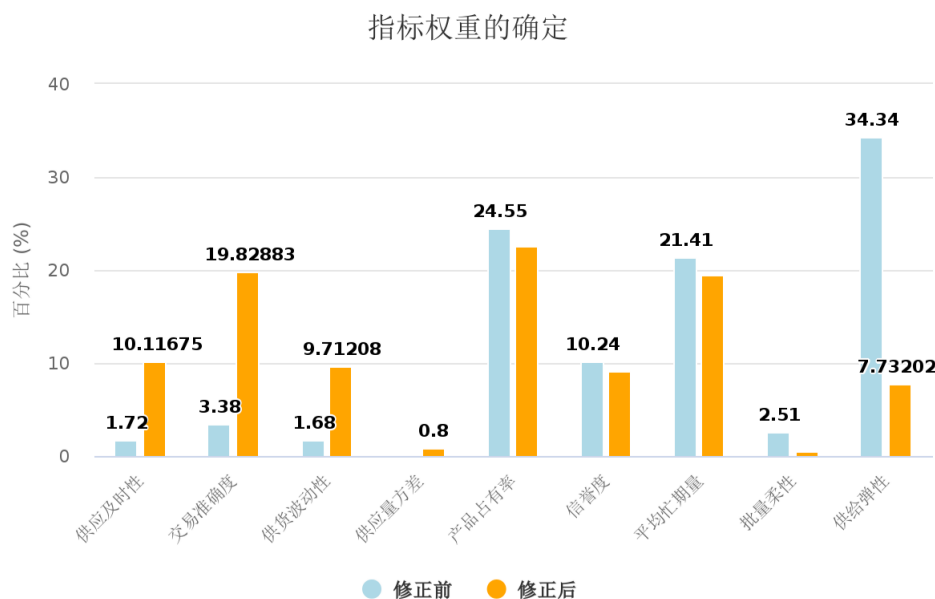


图 3: 层次分析法修正前后的权重的比较

基于修正后的权重，对供应商 i 进行打分。记分数为 w ，则

$$w_i = 0.101 \cdot f_i^{(1)} + 0.198 \cdot f_i^{(2)} + 0.097 \cdot f_i^{(3)} + 0.008 \cdot f_i^{(4)} + 0.225 \cdot f_i^{(5)} + 0.092 \cdot f_i^{(6)} + 0.195 \cdot f_i^{(7)} + 0.006 \cdot f_i^{(8)} + 0.077 \cdot f_i^{(9)} \quad (5.2.1)$$

打分得到 50 家最重要的供应商，如下表4所示：

排名	供应商 ID	评分	排名	供应商 ID	评分	排名	供应商 ID	评分
1	S201	0.75974645	18	S348	0.45921065	35	S005	0.388482065
2	S229	0.646160998	19	S374	0.452996397	36	S123	0.385770748
3	S361	0.626986785	20	S126	0.452707551	37	S367	0.38271972
4	S140	0.582720318	21	S352	0.451897697	38	S037	0.37993823
5	S108	0.562163907	22	S356	0.451782973	39	S189	0.376542632
6	S282	0.541186211	23	S284	0.443962706	40	S346	0.374376793
7	S340	0.536377517	24	S247	0.442722083	41	S244	0.372966686
8	S275	0.529990375	25	S031	0.434323795	42	S364	0.372366756
9	S329	0.526701861	26	S307	0.430694683	43	S055	0.371128447
10	S395	0.523379239	27	S365	0.423267153	44	S210	0.365853787
11	S131	0.507056588	28	S143	0.423261774	45	S074	0.362386554
12	S268	0.503219779	29	S294	0.410719677	46	S078	0.361104426
13	S306	0.495833855	30	S139	0.405768113	47	S086	0.354186795
14	S151	0.489199392	31	S218	0.404435297	48	S007	0.350107187
15	S308	0.483788422	32	S080	0.400158729	49	S273	0.349888122
16	S330	0.465845283	33	S040	0.394213106	50	S003	0.347399242
17	S194	0.465013512	34	S266	0.394086124			

表 4: 排名前 50 的供应商

如下图4, 作出了部分排名在前的供应商的特征分布。结合下图4和附录 B 中的数据, 可以发现排名在前的供应商大多有两个及以上的指标表现较为出色, 或者大部分指标均表现良好, 这与现实认知相符。这些排名在前的供应商与企业合作频繁, 供货量大, 能准时供货的情况数多; 那些交易量少, 食言次数多的企业均排名在后。除此以外, 排名靠前的企业有的供货弹性较大, 有的供货潜力较好……能够满足企业的各种策略要求, 对于保障企业生产发挥着至关重要的作用。由此, 评价体系的合理性可见一斑。

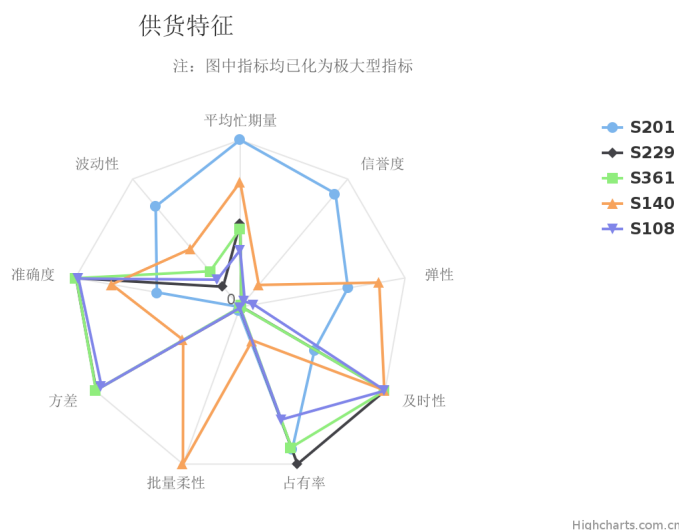


图 4: 前 5 名供货特征比较

6 问题二中贪心算法求解双目标优化问题

6.1 模型建立

在分析中我们指出，这一问的模型包含多个部分：内层的经济进货存贮模型，外层的双目标定量规划问题，以及问题一中求得的供应商模型。以下对这几个子模型分别进行建模与解说。

6.1.1 经济进货存贮模型

存贮模型是数学建模中一种常用模型，存贮论本身也涉及有最优存储策略的运筹求解。由于该题着重对存贮之前的采购-运输环节的策略进行讨论，故依据假设五，此处的存贮模型进行了适当简化，保证在完整表示目标约束条件的同时，不将模型过于复杂化。[3]

该情景模型中所涉及到的存贮要素有：

- 需求量：每周对于产能的需求量为 $28200m^3$ ，间接代表对 A, B, C 三种物品的需求量。
- 订货批量：一次订货中，包含 A, B, C 的数量。此为我们本问优化求解的对象。
- 订货间隔期：一周一次订货（进货）。
- 相关费用：分为订货费、运输费和存贮费。

该企业的存贮策略为 (t, s) 型策略，即，每隔一周补充进货一次，补充量参考存贮警戒点 s 进行设计。警戒点约束条件为满足未来两周的产能供应，即 $s = 28200 * 2 = 56400(m^3)$ 。

依据假设五，对 A, B, C 的存储进行简化处理，实际分开存储的三种货物在此模型中统一转换为产能进行存储，第 i 家供应商对应商品的单位产能转换系数 z_i 依题可知。故对第 i 个商家在第 k

周的接收量 $t_{i,k}$ 所得到的产能，有转换公式：

$$w_{i,k} = \frac{t_{i,k}}{z_i} \quad (6.1.1)$$

由此，第 k 周企业获得的产能值 Δw_k 为

$$\Delta w_k = \sum_{i=1}^{50} w_{i,k} = \sum_{i=1}^{50} \frac{t_{i,k}}{z_i} \quad (6.1.2)$$

第 k 周总库存产能值 w_k 基于上一周库存，还要进行消耗和补充操作，有

$$w_k = w_{k-1} + \Delta w_k - 28200 \quad (6.1.3)$$

由于模型警戒点的设置，我们假设初始状态时企业库存量 $w_0 = 56400$ ，刚好满足产能要求约束。该假设在保证模型自洽的同时，也没有给未来周的状态带来外来影响。

6.1.2 供应商模型的细化分类

分析数据发现，部分商家存在供应量非常小的时间段，而其余商家供货较为平稳、大致维持在同一水平波动。分析认为，周期性的存在表示了该商家具有短时间内提供大量货物的能力，但是之后需要一段时间进行供应量的积累恢复，从而造成了明显周期变化。

据此，定义两种供应商类型：**平稳型**、**周期型**。当该供应商某周供货量少于 5 时，该周算作供应“空期”，视为无法提供（中大额）订单。相应的，高于 $5m^3$ 的周记为“忙期”，即可以提供相对应数量级的大额订单。对于连续空期和连续忙期，算出其 240 周内平均时间长度，作为空期/忙期转换的周期长度。

在方案中对于周期型供应商的行为基于此周期建模：初期该供应商处于周期中某一随机位置，随着周数增加缓慢推进周期变化。当该周处于空期时，此商家不接受订单，产出量为 0；当该周处于忙期时，依据往期忙期平均量、偏差率等属性，确定其接受的订单量。如果某一商家的平均空期为 0，则说明其属于平稳型。人工粗略筛查 50 组数据确保，不存在位于 5 附近平稳波动的商家被误判为周期型的情况。

附录 C 为前 50 组商家的周期/平稳型分类情况。

6.1.3 双目标定量规划建模

企业一周进行订货流程包含如下三个环节：

1. **采购**。该环节交接对象为供货商与企业：企业每周向供应商预订订单上的订货量 $d_{i,k}$ ，供应商根据实际情况回返相应供货量 $s_{i,k}$ 。供货量与订货量之间有一定差异，通过偏差率 $p_{i,k}$ 计算。对于供应商生产提供的供货量，无论其与订货量差别多少，企业均全盘接受。故采购环节产生的成本按照供货量计算，记为 C_k 。（ i 表示供货商编号， k 表示周数）
2. **转运**。此过程涉及供货商与转运商的匹配问题。一家供货商将提供的供货量交给仅一家转运商转运，第 i 家供货商在第 k 周对接的转运商记为 $kj_{i,k}$ 。转运过程中可能出现损耗，即损耗率 l_j ，经过转运实际递交给企业的量为接收量 $t_{[i,k]}$ ，小于先前供货量。

3. **存贮**。企业将最终获得的接收总量存入仓库。仓库的容量假设是无限的，第 k 周仓库库存量记为 w_k 。转运与存贮环节成本统一记录。

(一) 基本量计算与随机变量假设

各量之间的关系有以下公式计算：

$$s_{i,k} = p_{i,k} d_{i,k} \quad (6.1.4)$$

$$t_{i,k} = (1 - l_{j,i,k,k}) s_{i,k} \quad (6.1.5)$$

假设偏差率 $p_{i,k}$ 、损耗率 $l_{j,k}$ 均为正态分布，由附件 1 处理得到每家供应商或转运商对应率的均值和标准差 p_i, Dp_i, l_j, Dl_j ，生成正态分布的随机量来模拟未来 24 周中损耗率、偏差率的波动。

$$p_{i,k} \sim N(p_i, Dp_i) \quad l_{i,k} \sim N(l_j, Dl_j) \quad (6.1.6)$$

此处对于偏差率和损耗率进行随机处理的考虑为，由假设一商家对应概率波动不存在季度变化，上下浮动看做真实世界中广泛存在的独立不相干干扰量的影响，在大部分情况下是基于正态分布的。同时，偏差率等本身具备了偏置性，正态分布的对称性并不会扭曲货量变化时的非对称性。另外，随机量的引入使得未来 24 周方案抛弃了一成不变的平稳型而具有波动性，在丰富方案的同时，也具备了自动上下调整库存的能力，使模型具有更强的稳健性。

(二) 目标函数

现有如下目标：

a. “最经济”的订购方案

该目标要求方案的订购环节“最经济”，即订购成本越少越好。企业第 i 家供应商某一周的订单最终交易量应当以实际供货量为准，故有目标函数 a：

$$\min \sum_{k=1}^{24} \sum_{i=1}^{50} r_i s_{i,k} \quad (6.1.7)$$

b. 转运损耗最少

基于目标函数 a 的最优解下，进一步确定转运方案。对于转运途中存在物品损耗的情况，也可看做成本的负方向指标。对于损耗量价值的量化估计，我们将其转换为产能，表示“在路途中损失的产能”。据此有以下目标函数 b：

$$\min \sum_{k=1}^{24} \sum_{i=1}^{50} \frac{s_{i,k} - t_{i,k}}{z_i} \quad (6.1.8)$$

(三) 约束条件

由题中各约束条件整理如下：

1. 转运商每周最多运送 $6000m^3$ ；

2. 贮存模型中存贮警戒点 s 约束, 要求满足未来至少两周库存;
3. 贮存模型中库存量 w_k 的时间变化关系;
4. 偏差率、损耗率基于正态分布的随机变量;
5. 订货量需满足一定合理性, 不可过大或过小;
6. 各参数的非负性。

(四) 双目标规划方程

$$\min \sum_{k=1}^{24} \sum_{i=1}^{50} r_i s_{i,k} \quad (6.1.9)$$

$$\min \sum_{k=1}^{24} \sum_{i=1}^{50} \frac{s_{i,k} - t_{i,k}}{z_i} \quad (6.1.10)$$

$$s.t. \begin{cases} \sum_{j=0}^{50} s_{i,k} \leq 6000, & (\forall j, k, j_{i,k} = j) \\ w_k \geq 28200 * 2 \\ w_k = w_{k-1} + \Delta w_k - 28200 \\ p_{i,k} \sim N(p_i, Dp_i), & l_{i,k} \sim N(l_j, Dl_j) \\ s_{i,k} = p_{i,k} d_{i,k} \\ t_{i,k} = (1 - l_{j_{i,k},k}) s_{i,k} \\ 0 \leq d_{i,k} \leq 6000 \\ s_{i,k}, t_{i,k}, w_k \geq 0 \end{cases} \quad (6.1.11)$$

6.2 算法模型求解

6.2.1 构建贪心算法

对于双目标优化类问题, 考虑贪心算法进行求解。此处构架双层贪心, 优先满足目标函数 a 即供应环节, 其次满足目标函数 b 即转运环节。算法伪代码见下:

Algorithm 1 双层贪心算法：求采购最经济方案及其对应运输方案

相关量说明 X - 供应商集, Y - 转运商集, $KeyExpression$ - 关键式

```

1: for week  $k = 1 \rightarrow 24$  do
2:    $w_k \leftarrow w_{k-1} - 28200$ 
3:   for  $x_i, i = 1 \rightarrow 50$  do
4:     if  $x_i \in X_{steady}$  then
5:        $d_{i,k} \leftarrow KeyExpression$ 
6:     else  $x_i \in X_{periodic}$ 
7:       if  $x_i$  处于忙期 then
8:          $d_{i,k} \leftarrow KeyExpression$ 
9:       else  $x_i$  处于空期
10:        continue
11:      $p_{i,k} \leftarrow p \sim N(p_i, Dp_i)$ 
12:      $s_{i,k} \leftarrow p_{i,k} * d_{i,k}$ 
13:
14:      $\forall j : clean \ y_i.storage$ 
15:     for  $y_j, j = 1 \rightarrow 8$  do
16:       if  $y_j.storage + s_{i,k} \leq 6000$  then
17:          $J_{i,k} \leftarrow y_j$ 
18:          $y_i.storage \leftarrow y_i.storage + s_{i,k}$ 
19:          $l_{i,k} \leftarrow l \sim N(l_{J_{i,k}}, Dl_{J_{i,k}})$ 
20:          $t_{i,k} \leftarrow (1 - l_{i,k} * s_{i,k})$ 
21:
22:      $w_k \leftarrow w_k + t_{i,k} / z_i$ 
23:     if  $w_k \geq 28200 * 2$  then
24:       break
25:
26:    $C_k \leftarrow \sum_{i=1}^{50} s_{i,k} * r_i$ 
27:    $L_k \leftarrow \sum_{i=1}^{50} (s_{i,k} - t_{i,k}) / z_i$ 

```

6.2.2 贪心的合理性论证

贪心策略首先需要有一个满足界限，定义为**贪心满意目标**。在每周进货时，企业只需要满足库存足够未来两周的产能，一旦大于该产能立即“见好就收”，停止增加订单。过多的产能不仅可能来

自较差的供应方和运输方、从而导致偏差率和损耗较大，还有可能提高存贮、运输的成本。故此处的贪心满意目标为：当前库存量大于未来两周所需产能。

接下来构建双层嵌套贪心，分别探讨订货量的选择和供货运输匹配。

(一) 外层贪心：最优供货商

由问题一得 50 家按照合作可靠度高低排序好的供货商，基于贪心，从最优合作商开始下订单，直到估计该周的最终接受量达到贪心满意目标。由供货商的供货周期类型可知，周期型供货商在“空期”时无法提供订单，则此时跳过处于空期的供货商，依次选择次优的合作商下单。

下单时的订货量根据关键式判断，将在下文中讨论。

(二) 内层贪心：最优运输商

参考供货商可靠度排序，对转运商同样构建排序操作。参考平均损耗量、损耗量偏差等，获得转运商的最优排序表：(见附录 C)

本周有订单的供货商，由最优到最次大致订单量呈递减趋势，因为可靠度越高的供货商的合作量也相对更大、更划算。故对最优到最次的供货商，依次分配当前仍有空间的最优转运商。越高合作量的供应商，具有优先选择高级转运商的权利。若当前最优转运商的剩余空间无法装下该供应商商品，则考虑次优转运商，直到与一名转运商匹配。

多次试验表明，不存在转运商不足的情况。

6.2.3 关键式-商家订购量计算公式

贪心策略确定之后，对 $d_{i,k}$ 的选取进行探讨，并由此提出以下四个较具代表性的公式。

首先 $d_{i,k}$ 需要与历史 5 年数据中该供应商的供货量具有相关性。往年该供应商与企业的供货量平均值反映了该供货商的货物供给能力、与企业合作量大小、合作频率密度等等，在考虑未来方案的时候也应当遵循该规律。而供货量的方差反映与其订单是否平稳，也可作为参考加入未来方案的订货量考虑中。运用第一问得出的供货量平均值、平均方差等数据。故四个公式的供货量基础均来自往年平均值 D_i 。

经过供应商评价模型后，贪心算法优先选择的商家均具有较高的合作潜力，故引入**进货参数** 1.5 乘以往年平均值，在不脱离历史数据的合理范围内提高与该商的合作力度，从而向更优方向探索。同时，类似偏差率的随机生成，订货量也应当体现一定的随机性，以保证方案在一定波动范围内的自我调节能力。公式 A、B 分别将随机性赋给加法或乘法，留待观察效果。

同时，分析历史数据发现，当订货量过大时，一方面供货量的随机波动将增强，另一方面其偏差率也将大幅下降。故定义公式 C、D 描述这两种现象。

公式 A

$$d_{i,k} = \overline{D}_i * (1.5 + \epsilon) \quad (6.2.1)$$

其中 $\epsilon \sim N(0.2, 0.5)$

公式 B

$$d_{i,k} = \overline{D}_i * 1.5 + \epsilon \quad (6.2.2)$$

其中 $\epsilon \sim N(0, 500)$

公式 C

$$d_{i,k} = \overline{D}_i * 1.5 + \epsilon \quad (6.2.3)$$

其中

$$\epsilon \sim \begin{cases} N(0, 500), & \text{if } \overline{D}_i * 1.5 > 2500 \\ N(0, 200), & \text{if } \overline{D}_i * 1.5 \leq 2500 \end{cases} \quad (6.2.4)$$

公式 D

$$d_{i,k} = \overline{D}_i * 1.5 + \epsilon \quad (6.2.5)$$

其中

$$\epsilon \sim \begin{cases} N(0, 500), & \text{if } \overline{D}_i * 1.5 > 2500 \\ N(0, 200), & \text{if } \overline{D}_i * 1.5 \leq 2500 \end{cases} \quad (6.2.6)$$

且

$$p_{i,k} \sim \begin{cases} 0.7 * N(p_i, Dp_i), & \text{if } \overline{D}_i * 1.5 > 2500 \\ N(p_i, Dp_i), & \text{if } \overline{D}_i * 1.5 \leq 2500 \end{cases} \quad (6.2.7)$$

6.3 方案的实施效果分析

(一) 结果分析

取定进货参数为 1.5，分别对四个关键式进行实验，求得结果分别如下⁵:

分析该图，公式 B 出现了未满足约束要求的情况，而公式 D 进货成本略高于 A、C，且在初始阶段刚刚满足约束条件，稳健性不强；公式 A 与 C 的结果均较为可观且相接近，但公式 C 的定义更加贴合实际，故最终选择公式 C 为关键式。

基于此方案，多次实验求得相关目标值。选取频率较高且目标较优的典型代表方案，将方案列入附件 A，且求得该方案下的某次单周采购成本 C_k 如下：

[21508.2 20326.5 19584.1 20590.4 20351.7 20808.9 20719.3 19879.6 20656.8 20378.3 20788.9 19947.4 20950.5 20085.2 20671.8 21109.2 19764.3 20432.9 20412.2 21566.2 20051.1 20061.8 20125.7 20940.8]

其平均订购成本为 20488/周。

同样，求得其转运损耗量 L_k 为：

[146.36363636 134.34343434 139.74747475 147.85353535 134.92424242 141.36363636 120.37878788 140. 123.05555556 141.99494949 123.40909091 131.59090909 139.09090909 137.85353535 136.21212121 144.87373737 129.04040404 128.88888889 137.82828283 131.91919192 132.1969697 133.08080808 131.36363636 140.60606061]

其平均转运损耗为 135.33/周。

该问还要求与尽量少的供应商合作，求得该值如下：

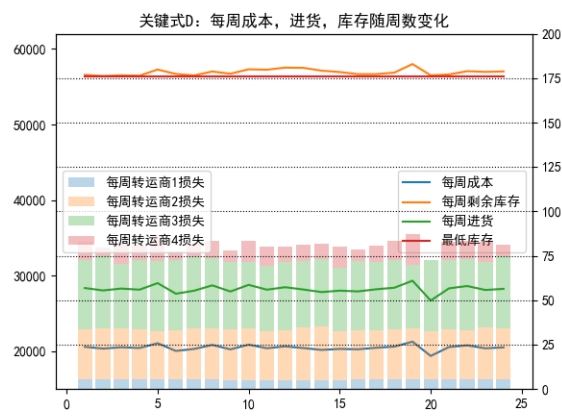
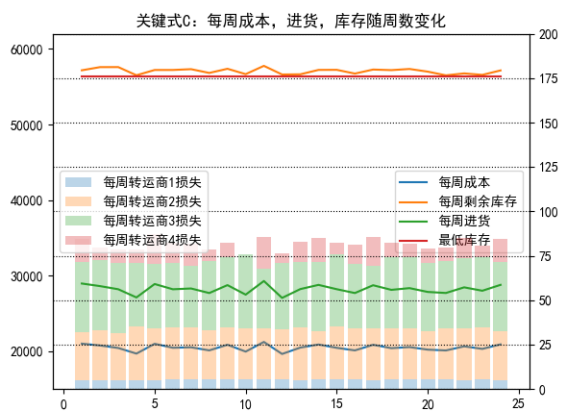
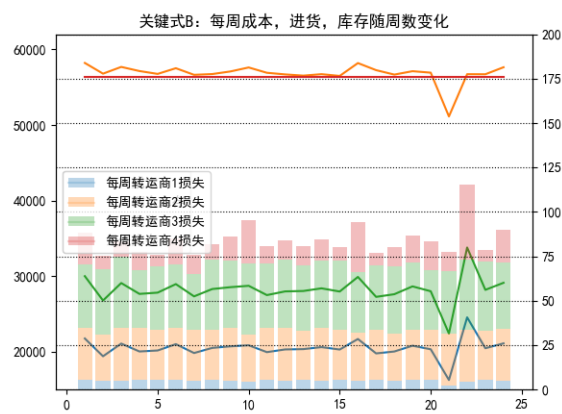
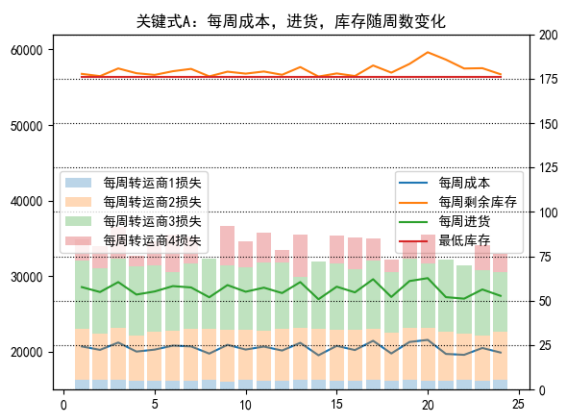


图 5: 四项公式比较

[16. 15. 16. 28. 17. 18. 18. 14. 10. 12. 14. 15. 11. 24. 16. 25. 15. 16. 16. 18. 17. 17. 17. 23.]

合作商家数量大致在 16 家左右，范围从 10 至 25 均有。

(二) 合理性说明

画出排名前 10 的供应商的准确性随订货量变化的重叠散点图6。通过比较下图6与实验所得数据，发现所得数据对应的供货准确度均较高，基本不存在偏差较大的情形，由此证明我们得到的订购方案是合理可行的。

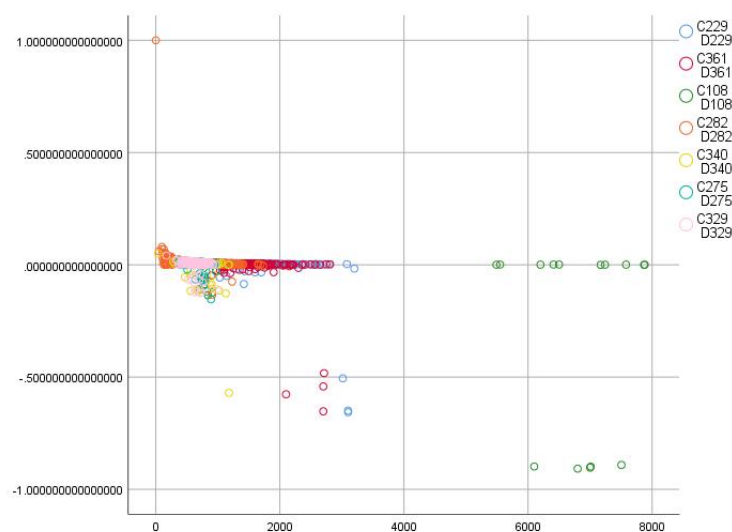


图 6: 供应商历史波动性分析

7 定向目标权重筛选在问题三的应用

7.1 目标模型的建立

题目引入压缩成本的考虑，也即尽量减小企业在存储环节所花的成本。题目给出未证明的推论如下：

推论：尽量采购 A 类原料、减少采购 C 类原料，可达成压缩成本效果。

首先对该推论进行证明，再基于此推论设计新订购模型。

7.1.1 原料属性分析及推论证明

由题表 1 已知各原料原始属性。其单位产能转化率 z 表示生产一立方米产能所需的原料量，而相对成本 r 反映原料价格。由此计算原料的性价比 $= r^{(x)} z^{(x)}$ ，表示以 x 为原料生产每单位产能所需的采购成本。结果如下表5：

种类	A	B	C
性价比	0.72	0.726	0.72

表 5: 原料性价比

A、C 类原料性价比相同，而由 $z^{(A)} < z^{(C)}$ 产生相同产能所需的 A 原料更少。因此为节约存贮成本，尽可能使单位存贮物产生产能更高，应提高 A 类原料采购量，减少 C 类原料采购量。推论证明完毕。

而对于 B 类原料，其性价比较低，且单位产能密度低于 A，但高于 C，故 B 作为中间量次于 A 类产品但优于 C 类产品。由于第三问成本仅从存贮方面考虑，故 B 类性价比低于 C 类的特点对重要性没有影响。

7.1.2 基于权重二次排序

基于推论对现有供应商排名进行更改，突出 A 类供应商的重要性，依次降低另两种。不同于第一问中各指标经过熵权法、层次分析法得出的分数评判，该处依照推论有明确的目标进行二次规划。在具备明确方向的背景下，人为重新指定类型权重简单有效。

现指定三类权重：A 类 1.05，B 类 1，C 类 0.95，乘以原评价分数得到新评价分数，排序获得新排名。权重的选取基于一般常识与多次实验判断，一方面权重不可过小，确保新方案中各类原料的显著突出度，具备与原方案排名的明显区别以保证模型三的思想贯彻始终；另一方面设定值不可过大，否则将会导致排名过于激进，导致解得方案过多关注存贮成本压缩，失去对采购成本的约束与供应商合作可靠性相关关系。

列出新排名中前十名的前后变化⁶:

代码	原评分	排名	新评分	新排名	材料分类
S201	0.759	1	0.797	1	A
S229	0.646	2	0.678	2	A
S361	0.626	3	0.595	3	C
S140	0.582	4	0.582	4	B
S282	0.541	6	0.568	5	A
S108	0.562	5	0.562	6	B
S275	0.529	8	0.556	7	A
S329	0.526	9	0.553	8	A
S395	0.523	10	0.549	9	A
S340	0.536	7	0.536	10	B

表 6: 供应商排名变化

7.1.3 定向筛选对算法进行更改

贪心算法的处理中加入定向筛选操作，进一步提高对 C 类原料的排斥度，以获得更好的存储成本。添加定向筛选算法至订购环节的优先供货商选取流程中。

该算法筛选条件为，若当前最优商家为 C 类，且下一非 C 类商家与其评价级别相近时，用下一商家代替该商家以减少 C 类供应订单。“相近”的筛选判断机制取 5%，该区间下前后供应商综合评价差别较小，生产原料上的优先级使得靠后的非 C 类供应商可以获得更高的被选中频率。

类似的，也可依据相同原理对 A 类商品进行向上提升的筛选，此处仅提出想法，不作赘述。

Algorithm 2 C 类定向筛选算法

- 1: 对厂家 i_1, i_2 :
 - 2: **if** $class(i_1) = 3$ **and** $class(i_2) = 1$ **or** 2 **and** $\frac{score(i_1) - score(i_2)}{score(i_1)} < 5\%$ **then**
 - 3: 跳过 i_1 ，选择 i_2 为当前首选供应商
-

7.2 求解所得方案分析

将基于新排名且引入定向筛选机制的模型称作新模型，进行实验求解与原模型对比。

求得其转运损耗量 L_k 为：

[79.7301424 72.32361493 80.16933504 72.70351575 75.00855177 74.96830567 74.95028003 69.99244121 79.80875954 70.31252042 70.49169535 74.72436144 72.65595341 71.41491977 74.98176174 83.20885243 71.10907587 69.18801292 79.57907641 71.95125798 80.34817925 29.04782128 74.34384891]

其平均转运损耗为 72.73/周，显见新模型下各转运商的转运损耗量整体下降了，而存储模型的警戒条件仍可被满足，说明新模型方法与先前相比，有效降低了方案的转运存储成本。

画出原排序方案与新排序方案的各项成本、损耗指标如下⁷

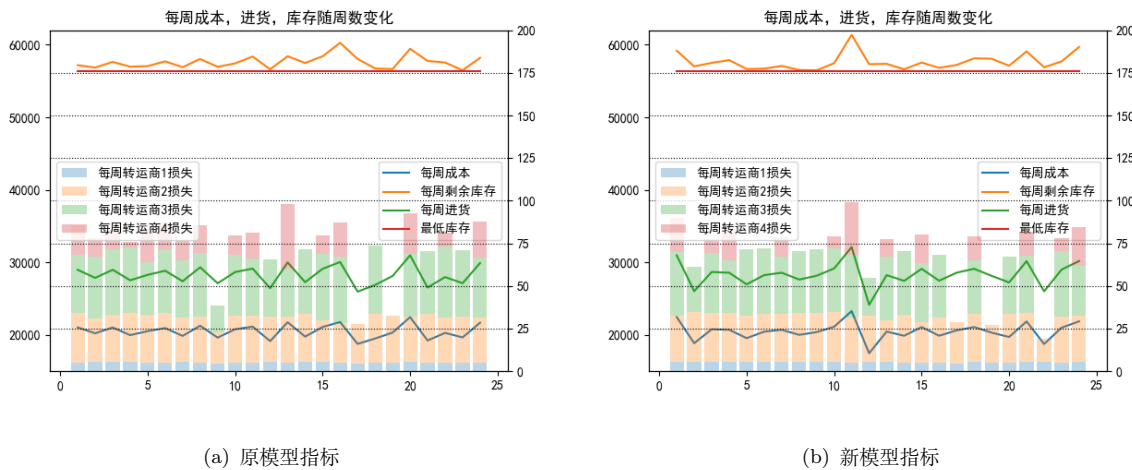


图 7: 模型各项指标变化

将转运损耗量单独画出，效果更为直观^{8(a)}:

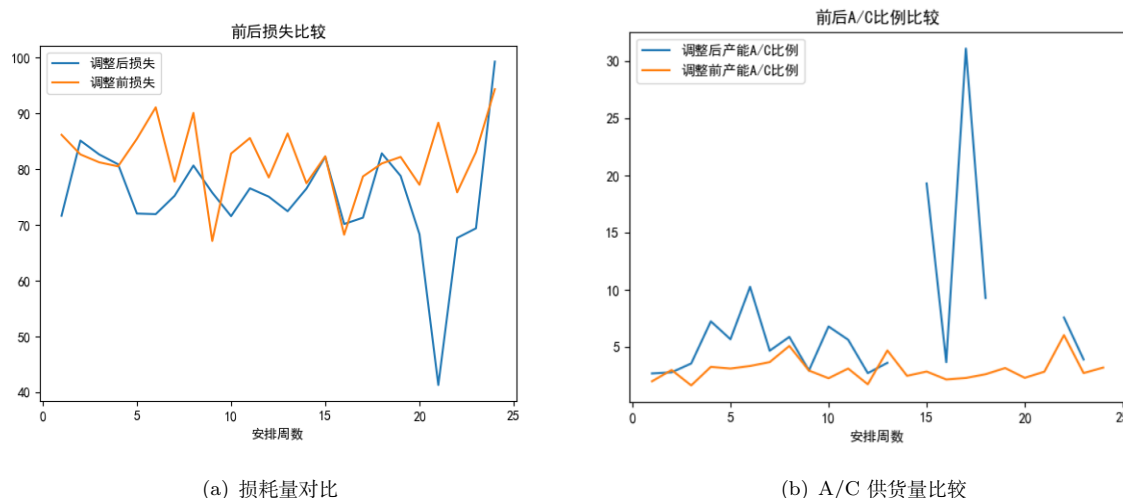


图 8: 前后对比参照

此外，对模型调整前后的 A 类、C 类合作量进行比较 8(b)，纵坐标为某周 A 类所有供货量与 C 类所有供货量的比值；调整后不存在比例的情况为，该周无 C 类商品供货。经过计算可知，在新方案下 A/C 类供货量之比平均在 7.34（除去比例不存在情况）。每周情况差别较大，最高可达无穷大，最低也将近有 3 倍。此处表明前述权重调整与定向筛选对 A、C 起到了实际吸引、排斥作用，且同样说明 A/C 值较大时，方案的损耗指标、存贮成本的确降低了，印证了题给推论的正确性。

8 问题四的产能拓展

8.1 提升产能的思路模型

问题四可以分为进货与投入生产这两个相互独立的部分，进货量的大幅提升无疑对产能提高有着显著作用，而原材料的投产分配方式在此问中并非主要影响因素，但在实际生活中是不容忽视的一部分。

现要求在被选商家的情况下，尽量提升每周产能，这要求每周进货量尽量增大，并且保证一定范围内成本和损耗值的合理性。显然，此处成本损耗不作为优化目标，进行了部分松弛操作。对于我们筛选出来的前 50 家供应商，在针对成本损耗优化的第二三问中，均取得了较好的效果。但是鉴于此问目标逆转，在较松的成本损耗约束情况下需要更多的进货量，故我们考虑全部 176 家数据具备参考性的供应商。这些供应商中评分较低者存在偏差率大、供应量小等情况，但对尽量提高每周产能具有积极作用。

8.1.1 产品收益亏损与否的证明

鉴于原材料在转运途中会有一定量的损失，但是最终售卖产品所得的收益能否抵消这部分亏损是一个疑问。经过证明，提出如下命题，所以此问无需考虑产品售价与提高产能之间的关系。

命题：产品不会亏损。

证明：假设 $1m^3$ 的 C 原料的定价为 1, $1m^3$ 产品的价格为 P , 即售价是原料 C 单价的 P 倍。

记 A 材料总供应量为 $A(m^3)$, B 为 $B(m^3)$, C 为 $C(m^3)$ 。则总材料成本为 $0.72A + 0.726B + 0.72C$, 到手产量为 $(\frac{A}{0.6} + \frac{B}{0.66} + \frac{C}{0.72}) \cdot (1 - s\%)$, 其中 s 为转运过程中的平均损耗率。当产品全部卖出时, 共能收入 $(\frac{A}{0.6} + \frac{B}{0.66} + \frac{C}{0.72}) \cdot (1 - s\%) \cdot P$ 。

得到总利润为总收入-总材料成本-转运费-库存费, 其中库存费与转运费题目并未给出具体数据与定价方式, 依据实际情况, 它们远小于材料的进货费用, 故此处略去不记。故总利润大致为

$$(\frac{A}{0.6} + \frac{B}{0.66} + \frac{C}{0.72}) \cdot (1 - s\%) \cdot P - (0.72A + 0.726B + 0.72C) \quad (8.1.1)$$

$$= [\frac{(1 - s\%) \cdot P}{0.6} - 0.72] \cdot A + [\frac{(1 - s\%) \cdot P}{0.66} - 0.726] \cdot B + [\frac{(1 - s\%) \cdot P}{0.72} - 0.72] \cdot C \quad (8.1.2)$$

由题给数据可知: $s < 5$, 所以上式在 $s = 5$ 时取到最小值, 最小值为 $(1.58P - 0.72)A + (1.44P - 0.726)B + (1.32P - 0.72)C \gg 0$ 。由此证明了转运过程中的损失不足以导致整体的亏损, 产品一般不会亏损。

8.1.2 原材料使用模式

如何合理使用原材料每周的库存量和进货量, 使得产能最大化? 我们提出两种方案。

(一) 动态产能情况

假设产能可随着当前周企业库存进行调整, 前提条件为满足存贮警戒点未来两周的产能供应。基于当前状态考虑暂时固定未来两周产能值, 则未来一周最大可达产能值为

$$v_k = \frac{w_{k-1}}{2} \quad (8.1.3)$$

其余量有计算公式:

$$w_k = w_{k-1} - v_k + \Delta w_k \quad (8.1.4)$$

$$\Delta w_k = \sum_{i=1}^{176} \frac{t_{i,k}}{z_i} \quad (8.1.5)$$

(二) 静态产能情况

为避免库存越积越多或者越用越少的情况, 且实际中我们要求企业每周的产能尽可能稳定, 为此我们提出更优的分配方案。每周投入的产量根据当前库存和本周、下周的进货数量情况, 计算出相应比例, 来确定单周产能的个性化分配方案。

假设第 k 周用于生产的产能为 a_k , 于是有 $a_k = x \cdot w_{k-1} + y \cdot t_k$, 其中 $x \cdot 100\%$ 、 $y \cdot 100\%$ 表示选择投入生产的百分比, w_k 、 t_k 分别表示第 k 周的库存产能与实际接受量。相应地有 $a_{k+1} = x \cdot w_k + y \cdot t_{k+1}$ 与 $w_k = (1 - x) \cdot w_{k-1} + (1 - y) \cdot t_k$ 。

(1) 实际应用中, 在技术改造后的一段时间内, 生产应尽快趋于稳定, 即要求第 k 周与第 $k+1$ 周投入生产的产能基本相同, 有如下等式:

$$a_k = a_{k+1} \quad (8.1.6)$$

$$\Rightarrow x \cdot w_{k-1} + y \cdot t_k = x \cdot w_k + y \cdot t_{k+1} \quad (8.1.7)$$

$$\Rightarrow x^2 \cdot w_{k-1} = (x - xy - y)t_k + y \cdot t_{k+1} \quad (8.1.8)$$

(2) 由题目要求, 企业库存量产能要尽可能满足两周生产需求。同时, 如果库存过多, 但并不投入生产, 无疑造成资源的浪费。因此, 库存应尽可能刚好满足接下来两周的生产需求, 有如下等式:

$$w_{k-1} = a_k + a_{k+1} \doteq 2a_k \quad (8.1.9)$$

$$\Rightarrow w_k \doteq \frac{2y \cdot t_k}{1 - 2x} \quad (8.1.10)$$

联立 (1)、(2) 中得到的两个等式, 解得

$$\frac{t_{k+1}}{t_k} \doteq -\frac{x}{y} + \frac{1-x}{1-2x} \quad (8.1.11)$$

由于 t_{k+1} 、 t_k 在订购方案的制定中已经给出, 是定值, 故可以求出 x 、 y 变量之间的比例, 确定第 k 周个性化的生产方案。

8.2 求解分析产能提升情况

8.2.1 最优供应分配方案

为使得每周进货量尽可能大, 在增加至 176 家供货商的同时, 除去模型二算法种贪心满意目标。从最优供应商开始选择, 与其达成制定订单量, 直到达成两个约束条件之一: 总共 176 家供货商全部选完, 或者所有 8 家转运商满载运输。

8.2.2 实验分析产能结果

该处实现仅考虑动态产能情况, 对于静态的解析论证见上文。求解动态产能模型, 获得以下数据⁹。

由图 (a), 对于以上两个约束条件, 我们发现通常情况下前者截止了订单的扩大, 而 8 家转运商少有满载情况。原因是即使纳入所有 176 家供应商, 由于评分较低的供应商多为周期型, 整体供货量贡献度较小, 仍然无法使所有转运商满负荷工作。对于这一结果存疑, 可能需要考虑更高的进货参数, 或者降低商家的偏差率。

图 (b) 中“每周产能”图例即为求得动态产能变化。随周期推进, 动态产能 v_k 呈现上下波动状态, 但总是优于最初 28200 的固定产能。并且随着周数的延长, 动态产能的平均趋势也逐渐提高, 24 周前后自身提升了约 27% 的产能值。这说明此方案中库存量及产能量将逐渐提高, 这也符合我们的常识认知。图上还能看出, 每周增加产能 Δw_k 对于动态产能变化的影响, 后者的波动趋势与前者几乎相同, 整体滞后约 3 周。

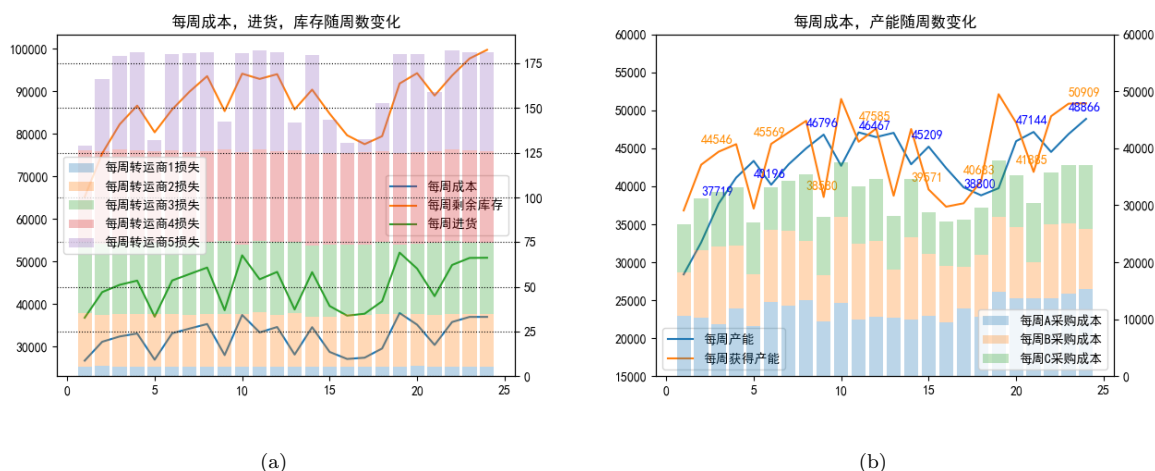


图 9: 动态模型实验结果

方案所涉及的供应商最低排名均延至 170+ 家, 而其中进行了订单交易的商家每周数量主要集中在 60 至 70 家之间, 中位数和众数约为 66 家。这说明其中将近一半的商家平均处于周期型的空期, 不参与贡献直接跳过。

对求得数据取平均值, 得出最终该方案的平均动态产能为 43299, 在原先 28200 的产能基础上提升了 53% 的收益。每周平均获得产能为 43667, 即在基本维持未来两周消耗需求的约束下, 库存量也将逐渐上升。

9 模型评价

9.1 优点分析

(一) 多种评价指标方式协同确定

模型一查阅相关文献, 提出九种具有实际意义的指标对供应商各方面能力评价。对各属性标准化、正向化后, 首先运用熵权法确定整体权重偏向, 根据求得反馈重新人为调整部分重点相关指标的权重设定, 套用层次分析法二次评价。多种评价指标体系协同确定最终评分等级, 保证了所得排名中既包含数据本身区分性质, 同时也包含人为一般经验知识引导, 最终结果对合作优先度具有更科学的体现。

(二) 对供货商供货周期特征的精确建模

在考虑信誉度等常规指标的基础上, 从时间维度分类不同供货商, 精确模拟周期变化规律。从历史数据中提取的周期规律包含了 5 年 240 周的信息, 作平均处理后对未来半年 24 周周期变化具有较为准确的预判模式。观察 50 家供应商数据表发现, 周期型供货商忙期内的供货量普遍高于平稳型, 表示了频率低、单次贡献高的合作订单; 而平稳型占据整体接收量比例较高, 说明整体产能获取量主要以相对量较低但分布稳定连续的订单为主。

（三）模拟随机变量波动影响

模型二中将供应商偏差量、每周订货量等视作随机变量，假设其为正态分布并从历史数据中提取相关参数，对未来 24 周模拟生成新的随机变量。该步随机操作符合真实世界的随机波动性，从损耗偏差、供货情况等方面考虑非确定性因素的干扰。同时，对于非对称的随机变量如偏差率，正态分布随机性的对称性不影响均值本身偏差方向的指引，不会引入坏的偏差影响。另外，更丰富的随机性可提高模型求解方案应对现实情况偶然波动因素的能力，使得该方案具备一定的自我调节能力，在真实设定下比平稳确定性模型表现更好。

（四）多层贪心对多目标优化问题的递进解法

求解多目标优化问题的方式较多，在该题模型二多目标的求解中，若选择线性规划则变量过多、约束条件不足求得少量较优解，若考虑动态规划则状态决策空间过大（ $6000 \times 50 \times 24$ ）、求解过程计算量过高且短时间内无法取得较优解。分析题目发现该题具备一种很强的贪心解法，且在多目标优先级已排序的情况下，依据优先级对最高目标首先进行贪心求解，再于求解过程中依次嵌套内层目标的贪心解法，即可求得最终较优解。虽然同样基于遍历思想，但由于贪心算法首先朝最好的方向进行搜索，故总能在少数时间内取得较好的解。

贪心算法成功的前提是贪心过程关键式的选取。同样模型二对于关键式进行了多种因素的讨论，并且通过实验对比确定最终最优关键式。模型三中也包含了多层贪心的运用。

（五）每周产能消耗的静态、动态情景讨论

在最终问的要求中我们给出了两种情景：静态产能与动态产能，并分别进行讨论。分类讨论的意义是，增加了方案的多样性，对于现实生产中不同的情形可选择更多方案，且动态和静态产能的计算生成原理均有其合理性。另外，进一步的扩展还可考虑动态、静态混合产能方案，此处列举思路供读者参考。

9.2 缺点分析

（一）贪心限定舍弃低排名供应商

在问题二、三的求解中，选取的供应商范围限定在模型一评价筛选出来的 50 家供应商之内。理论上可添加任意范围的供应商进入供货合作备选名单，但模型的贪心要求使得评分较低的供应商在方案中基本不存在贡献度，即低评分供应商在方案中几乎没有订单量。虽然该处理与第二问要求的“至少多少家供应商”相吻合，但在实际情况中并不一定是最优解。

（二）关键式选取影响方案构成

关键式是确定每周供货量，模拟得出后续转运方案与周库存产能变化的重要公式。对于该公式的选取构造，我们例举了四个典型式实验分析，但无法求得关键式的最优形式及其最优解析参数。应指出，模型贪心的关键式选取本身即为了降低计算上的复杂度；若强求得到关键式解析解，其复杂度并不小于使用线性规划或动态规划求解第二问的情况。故关键式的最优解改进无解。

不过，在当前关键式下得到的较优解相对也接近于理想最优解，此误差可以容忍。

(三) 随机变量带来不确定性影响

随机变量在具备优点（三）中好处的同时，同样也有少数劣势，最主要的一点即其使得最终得出的方案具有不确定性，每次求得的方案可能不一样。这是因为我们的较优解解集中方案不唯一，不同的随机变量会影响此次方案各决策量的变化。在以上几问的分析中，我们列出的结果均为多次实验出现频率较高的较好方案的典型代表。

在运用本模型实际求解的过程中，使用方宜多次运行求解过程，选取频率较高的较好方案作为最终解。

Bibliography

- [1] 徐呈祥. 糖果企业供应商选择研究 [D]. 东北农业大学,2017.
- [2] MBA 智库百科. 供应商 [EB]/[OL].<https://wiki.mbalib.com/wiki/供应商>, 2021-9-12.
- [3] 司守奎, 孙玺菁. 《数学建模算法与应用》[M]. 北京: 国防工业出版社,2011.

附录

A 问题一指标初始数据

文件：176 家指标原始值.xlsx

表 7: 176 家指标原始值

排名	供应商 ID	材料分类	供应及时性	交货准确度	供货波动性	供货量方差	产品占有率	信誉度	平均忙期量	批量柔性	供给弹性
1	S201	1	0.571	0.474	1.92	3.43E7	0.247	3.34	2930	0.0333	11700
2	S229	1	1	0.00526	6.7	219000	0.272	0.0421	1470	0.018	148
3	S361	3	1	0.00616	5.78	162000	0.243	0.0482	1370	0.0115	119
4	S140	2	0.995	0.211	4.43	2.08E7	0.058	0.652	2190	1	15100
5	S108	2	1	0.0187	6.28	1500000	0.195	0.164	1000	0.0171	1500
6	S282	1	1	0.00506	1.92	150000	0.127	0.0023	707	1	212
7	S340	2	1	0.0021	4.18	25500	0.162	0.0146	712	0.05	35.7
8	S275	1	1	3.55E-4	1.98	13300	0.125	0.00235	659	0.0211	20.2
9	S329	1	1	5.93E-04	2.07	14200	0.123	0.00387	650	0.0412	21.8
10	S395	1	0.881	0.139	3.78	2450000	0.132	0.629	1050	0.333	2390
11	S131	2	1	0.0109	4.03	21300	0.13	0.043	574	1	37.1
12	S268	3	1	5.92E-4	1.76	5020	0.0989	0.00372	540	0.028	9.29
13	S306	3	1	6.52E-4	1.89	13500	0.0932	0.00411	524	0.0267	25.6
14	S151	3	1	0.0224	7.09	3330000	0.121	0.209	810	0.0429	4110
15	S308	2	1	0.0313	5.11	297000	0.121	0.273	571	0.1	521
16	S330	2	1	0.0213	6.43	454000	0.118	0.187	569	0.16	797
17	S194	3	1	0.00214	2.7	3400	0.0769	0.0134	421	0.0265	8.05
18	S348	1	0.975	0.109	6.48	8530000	0.039	0.478	516	1	17900
19	S374	3	1	0.268	0.85	3220000	0.0125	0.326	307	1	15700
20	S126	3	0.91	0.203	3.53	3220000	0.0561	0.965	609	1	6170
21	S352	1	1	0.00538	3.01	20900	0.0696	0.0318	370	0.075	56.3
22	S356	3	1	0.00664	5.58	125000	0.0937	0.0461	538	0.18	231
23	S284	3	1	0.00383	0.518	26100	0.034	0.0152	194	0.3	134
24	S247	3	1	0.00117	1.29	1100	0.043	0.00643	236	0.0529	4.67
25	S031	2	1	0.00395	1.44	1030	0.0393	0.0189	171	0.25	5.99
26	S307	1	0.994	0.356	5.57	3010000	0.0448	0.565	1260	1	6590
27	S365	3	1	0.00284	1.81	4160	0.0323	0.0148	173	0.1	24
28	S143	1	1	0.0422	4.98	75700	0.0633	0.249	346	0.9	219
29	S294	3	1	0.00893	1.1	177	0.0142	0.0383	78.7	0.18	2.25
30	S139	2	0.991	0.168	7.91	4820000	0.0464	0.471	744	1	7050
31	S218	3	1	0.038	1.38	1310	0.0116	0.123	64.5	0.9	20.3
32	S080	3	1	0.0509	1.83	3220	0.0142	0.157	80.2	0.9	40.2
33	S040	2	1	0.0141	3.55	5530	0.0277	0.0791	133	0.14	41.6
34	S266	1	1	0.0872	0.858	10.8	0.00513	0.265	27.1	0.5	0.398
35	S005	1	0.939	0.108	0.949	630	0.012	0.0838	66.3	1	9.75
36	S123	1	1	0.0963	1.57	103	0.0051	0.277	26.9	0.9	3.85

37	S367	2	1	0.0257	4.11	4430	0.0243	0.138	110	0.3	40.3
38	S037	3	0.975	0.219	5.48	596000	0.0438	0.359	473	1	1850
39	S189	1	0.95	0.093	2.17	516	0.0111	0.0934	62.5	1	8.75
40	S346	2	1	0.0187	4.52	299	0.0219	0.109	96.6	0.16	3.09
41	S244	3	1	0.058	3.72	1290	0.0119	0.203	68.1	0.9	18.9
42	S364	2	1	0.0118	4.97	10100	0.0235	0.0657	120	0.18	84.4
43	S055	2	1	0.046	4.59	36400	0.0176	0.283	100	0.3	363
44	S210	3	0.849	0.307	2.27	47100	0.0212	1.32	170	0.9	354
45	S074	3	0.934	0.258	2.68	32700	0.0132	0.973	111	1	428
46	S078	1	0.944	0.144	3.49	8010	0.0156	0.402	85.5	0.9	94.6
47	S086	3	0.986	0.241	3.87	70600	0.015	0.543	184	1	810
48	S007	1	1	0.255	2.57	879	0.00621	0.633	28.9	0.9	30.4
49	S273	1	0.909	0.149	4.27	8520	0.0196	0.345	114	0.9	80.8
50	S003	3	0.96	0.129	4.34	6350	0.0112	0.226	78.7	1	92.4
51	S291	1	0.986	0.176	4.27	11300	0.00708	0.398	51.3	1	310
52	S379	3	0.973	0.306	0.985	187	0.00113	0.101	22	1	27.1
53	S342	3	0.975	0.326	0.875	22	9.88E-4	0.159	9.74	1	3.84
54	S338	2	1	0.202	6.3	212000	0.0155	0.28	301	1	1430
55	S092	2	0.969	0.351	0.7	3.22	6.82E-4	0.121	5.93	1	1.02
56	S114	1	0.991	0.153	5.48	19800	0.00912	0.255	76.3	1	414
57	S221	1	0.976	0.401	0.338	3.15	6.23E-4	0.119	5.96	1	0.989
58	S392	2	0.935	0.357	0.773	3.17	7.09E-4	0.123	5.85	1	0.994
59	S292	1	0.763	0.279	3.05	4810	0.0199	0.537	104	0.9	47.3
60	S025	3	0.933	0.36	0.777	2.73	5.14E-4	0.121	5.36	1	0.916
61	S150	1	1	0.164	5.1	1320	0.00139	0.302	10.9	1	175
62	S178	1	1	0.425	0.451	7.1	6.19E-4	0.122	7.17	1	1.96
63	S314	3	1	0.176	4.97	38.4	0.00133	0.29	9.73	1	5.4
64	S169	2	0.959	0.418	0.469	2.47	6.48E-4	0.133	5.83	1	0.811
65	S174	2	0.992	0.459	0.286	2.49	6.11E-4	0.131	5.81	1	0.898
66	S256	2	0.946	0.362	1.7	6.47	7.38E-4	0.144	7.03	1	1.94
67	S141	2	0.923	0.408	0.749	4.39	7.47E-4	0.114	6.18	1	1.25
68	S324	2	0.926	0.412	0.731	3.43	7.39E-4	0.124	6.11	1	1.02
69	S113	3	0.955	0.432	0.676	4.23	5.2E-4	0.113	6.5	1	1.38
70	S208	1	0.714	0.377	2.79	22600	0.0191	0.899	101	0.8	232
71	S030	1	0.966	0.45	1.06	54.8	0.0011	0.233	15.1	1	9.47
72	S075	1	0.933	0.415	1.17	8.12	6.05E-4	0.128	6.67	1	2.6
73	S152	1	0.905	0.415	1.09	3.4	6.23E-4	0.135	6.26	1	1.06
74	S157	1	0.909	0.421	3.04	2.85	6.24E-4	1.13	5.68	1	0.865
75	S300	1	0.727	0.333	2.35	1050	0.00623	0.692	37.4	0.6	28.2
76	S154	1	0.74	0.337	3.53	9680	0.0168	0.542	116	1	97.7
77	S216	2	0.872	0.422	0.982	3.16	7.25E-4	0.118	5.74	1	0.96
78	S193	2	0.833	0.403	1.08	9.62	0.00104	0.136	7.86	1	2.38
79	S237	1	0.983	0.492	1.03	2.69	5.17E-4	0.159	6	1	0.978
80	S023	3	0.831	0.415	2.73	569	0.00223	0.972	26	1	46.4
81	S175	2	0.979	0.526	0.43	2.86	5.81E-4	0.126	6.18	1	1.08

82	S253	3	0.944	0.493	0.701	7.85	5.55E-4	0.113	8.07	1	2.39
83	S172	2	0.934	0.477	1.1	2.22	5.95E-4	0.149	5.47	1	0.798
84	S149	3	0.889	0.476	0.742	6.51	5.78E-4	0.113	7.67	1	1.87
85	S129	3	0.946	0.39	3.51	386	0.00151	0.449	29.2	1	48.6
86	S036	2	0.907	0.499	0.786	2.75	6.92E-4	0.137	5.83	1	1.09
87	S310	2	0.868	0.476	0.835	3.49	7.08E-4	0.122	6.16	1	1.08
88	S381	1	0.875	0.456	2.47	117	0.00129	0.417	20.6	1	16.3
89	S360	2	0.868	0.412	2.88	2.61	6.89E-4	0.187	5.67	1	0.843
90	S110	3	0.831	0.437	2.23	2.95	5.81E-4	0.178	5.61	1	0.885
91	S263	3	0.868	0.426	3.18	3.56	5.7E-4	0.244	5.87	1	1.1
92	S383	3	0.794	0.515	1.5	40.4	0.00102	0.209	12.6	1	7.42
93	S064	1	0.94	0.412	5.13	3.12	6.02E-4	0.523	5.88	1	0.991
94	S191	2	0.895	0.462	4.14	2.37	6.52E-4	0.63	5.6	1	0.804
95	S318	1	0.925	0.609	1.27	67.8	6.83E-4	0.146	33	1	19.4
96	S388	2	0.901	0.474	3.86	4.59	2.95E-4	0.665	7.43	0.5	2.57
97	S088	2	0.908	0.466	3.68	3.92	6.7E-4	0.304	6.33	1	1.28
98	S046	1	0.881	0.599	0.961	0.387	2.68E-4	0.149	0	1	0.275
99	S098	2	0.986	0.642	1.38	1.5	4.65E-4	0.204	6.29	1	0.747
100	S186	3	0.924	0.421	4.91	3.45	5.47E-4	0.442	6.19	1	1.12
101	S054	2	0.876	0.472	3.43	19.5	9.06E-4	0.261	9.64	1	5.33
102	S076	3	0.98	0.69	0.668	0.869	3.34E-4	0.153	5	1	0.455
103	S239	3	1	0.356	6.71	3.2	5.2E-4	0.182	5.56	1	1.08
104	S227	1	0.769	0.496	3.28	275	0.00131	0.566	21.3	1	32.5
105	S357	3	0.794	0.567	1.47	47.9	6.31E-4	0.126	10.6	1	11.8
106	S235	2	0.816	0.589	2.77	176	0.00139	0.78	19.3	1	28.8
107	S386	2	0.759	0.583	1.12	10.5	7.75E-4	0.152	10.8	1	3.63
108	S332	1	0.902	0.533	4.14	392	0.00132	0.411	26.3	1	59.1
109	S202	2	0.955	0.384	6.71	2.67	6.76E-4	0.264	6	1	0.847
110	S102	1	0.781	0.61	2.79	5.91	4.6E-4	0.9	7.57	1	2.26
111	S115	1	0.858	0.429	5.27	3.09	6.39E-4	0.284	6	1	0.94
112	S138	2	0.893	0.411	6.23	2.46	6.16E-4	0.337	5.43	1	0.851
113	S271	3	0.79	0.562	3.18	63.1	7.65E-4	0.438	12.5	1	12.5
114	S146	2	0.904	0.667	3.27	29.7	5.0E-4	0.685	26	1	14.3
115	S213	3	0.982	0.809	0.593	0.474	2.78E-4	0.165	0	1	0.314
116	S245	3	0.933	0.656	3.65	84	4.31E-4	0.491	58.3	1	33.1
117	S265	1	0.738	0.658	2.61	11.5	5.75E-4	1.12	9.2	0.5	4.82
118	S066	1	0.936	0.437	6.62	2.81	5.69E-4	0.268	5.67	1	0.954
119	S067	3	1	0.845	0.378	0.368	2.49E-4	0.164	0	1	0.267
120	S199	2	0.625	0.693	2.67	35.1	9.28E-4	1.73	13.1	1	8.09
121	S258	2	0.893	0.395	7.02	3.73	7.42E-4	0.198	6.07	1	1.13
122	S069	2	0.714	0.586	3.02	5.29	6.16E-4	0.525	7.88	1	2.19
123	S065	1	0.727	0.654	1.6	21.3	6.9E-4	0.304	12.4	0.8	7.2
124	S128	1	0.718	0.672	1.3	9.68	4.9E-4	0.192	8.86	1	3.71
125	S233	2	0.671	0.615	2.06	9.27	5.19E-4	0.293	8.18	1	2.95
126	S173	3	0.41	0.718	2.09	4770	0.00359	2.14	50	1	205

127	S039	3	0.714	0.626	3.53	646	0.00139	0.786	32.3	1	75.5
128	S376	2	0.877	0.49	6.15	1.96	6.32E-4	0.266	5.28	1	0.661
129	S336	1	0.66	0.658	1.5	48.6	8.58E-4	0.233	13.6	1	10.5
130	S232	2	0.65	0.666	1.96	31.8	9.94E-4	0.58	13.8	1	6.94
131	S197	2	0.952	0.449	7.56	3.23	6.41E-4	0.219	6.05	1	1.09
132	S313	3	0.551	0.677	2.45	148	0.00157	1.26	21.2	1	14.6
133	S117	2	0.622	0.711	3.03	6.18	3.08E-4	1.48	7.56	0.333	3.07
134	S153	2	0.232	0.917	1.88	1780	0.00308	3.77	154	1	132
135	S002	1	0.747	0.677	3.2	74.3	6.56E-4	0.438	17.8	1	19.3
136	S254	1	0.578	0.676	1.46	67.8	0.00118	0.216	18	1	11
137	S089	1	0.581	0.703	2.48	45.3	0.00105	0.965	12.3	1	8.04
138	S163	1	0.59	0.691	2.99	79.7	9.76E-4	1.03	18.1	1	14.6
139	S217	1	0.465	0.631	1.92	16.1	0.00187	0.464	8.4	1	2.31
140	S369	3	0.388	0.837	2.09	58.8	0.00105	2.6	16.1	1	10.5
141	S226	1	0.576	0.669	2.59	78.2	0.00108	0.417	17.6	1	13.3
142	S269	1	0.923	0.819	4.42	0.547	2.81E-4	1.07	5.5	1	0.371
143	S090	3	0.674	0.663	4.06	43.4	9.37E-4	0.619	14.7	1	8.69
144	S228	3	0.263	0.834	2.22	129	0.00197	2.96	16.9	1	12.8
145	S159	1	0.573	0.672	2.43	50.2	8.92E-4	0.242	12.3	1	11
146	S170	1	0.585	0.704	3.29	48	0.00106	0.828	14.4	1	8.53
147	S024	2	0.584	0.702	2.76	5.6	7.38E-4	0.504	7.14	1	2.03
148	S333	2	0.782	0.687	5.06	1.31	4.41E-4	0.626	5.2	1	0.712
149	S224	2	0.864	0.84	4.28	0.119	2.51E-4	1.11	0	1	0.105
150	S182	3	0.563	0.784	2.47	57.2	0.00104	0.849	16.1	1	10.1
151	S207	3	0.484	0.757	2.62	14.2	5.05E-4	1.09	9.3	1	4.05
152	S035	1	0.55	0.74	2.08	13.8	6.41E-4	0.382	10.6	0.5	5.26
153	S121	1	0.481	0.747	2.26	55.3	9.89E-4	0.637	17	1	10.5
154	S304	1	0.88	0.714	5.98	1.12	3.04E-4	0.392	5.33	1	0.632
155	S042	1	0.446	0.738	2.82	52.6	0.00104	0.789	14.7	1	9.53
156	S209	1	0.507	0.722	3.05	63.6	6.89E-4	0.46	12.3	1	14.4
157	S293	1	0.119	0.958	2.58	3680	0.00431	3.59	90.5	1	153
158	S280	2	0.647	0.748	4.58	3.4	3.6E-4	0.723	9.67	1	2.01
159	S105	2	0.455	0.694	3.02	86.1	0.00124	0.318	11.4	1	17.9
160	S334	3	0.139	0.904	2.2	10.6	0.00247	3.16	14.8	0.7	0.718
161	S028	2	0.388	0.731	2.02	18.8	3.06E-4	0.419	9.11	1	3.37
162	S104	1	0.347	0.793	2.51	92.9	0.00105	1.21	13	1	15
163	S017	1	0.373	0.767	2.07	50.3	0.00101	0.52	13.4	0.8	9.11
164	S260	2	0.883	0.708	7.31	0.484	2.99E-4	0.361	0	1	0.337
165	S167	1	0.4	0.776	2.45	43.6	9.29E-4	0.558	16	1	8.98
166	S147	2	0.4	0.84	2.82	572	0.00377	1.05	50	1	42.4
167	S010	2	0.364	0.792	3	48.5	0.00112	1.08	14.3	1	9.13
168	S112	2	0.313	0.788	2.88	785	0.00595	1.08	40.3	0.15	31.9
169	S142	2	0.326	0.848	2.48	133	0.0023	1.26	18.3	1	16.5
170	S394	1	0.152	0.886	2.52	26.3	0.00251	2.26	14.4	0.7	1.82
171	S375	2	0.213	0.911	3.13	659	0.00224	1.85	94	1	76.4

172	S022	1	0.242	0.883	2.83	142	0.00161	1.23	23.8	1	16.3
173	S019	2	0.207	0.89	2.8	246	0.00244	1.28	23.3	1	27.1
174	S071	1	0.167	0.871	2.15	78.1	0.00423	0.791	18.9	0.6	4.91
175	S077	2	0.207	0.933	3.11	329	0.00285	1.4	37.7	1	30.9
176	S127	1	0.375	0.784	6.26	60.3	0.00125	0.535	14	1	9.15

B 问题一最终指标

文件：176 家标准化后的指标与最终得分.xlsx

表 8: 176 家标准化后的指标与最终得分

排名	供应商 ID	最终得分	及时性	准确度	供货波动性	供货量方差	产品占有率	信誉度	平均忙期量	批量柔性	供给弹性
1	S201	0.7597	0.5131	0.5054	0.7857	0	0.908	0.8859	1	0.0221	0.6536
2	S229	0.6462	1	0.9949	0.1587	0.9936	1	0.0106	0.5017	0.0066	0.0083
3	S361	0.627	1	0.9939	0.2794	0.9953	0.8933	0.0122	0.4676	0	0.0066
4	S140	0.5827	0.9943	0.78	0.4565	0.3936	0.2125	0.1724	0.7474	1	0.8436
5	S108	0.5622	1	0.9808	0.2138	0.9563	0.7167	0.0429	0.3413	0.0057	0.0838
6	S282	0.5412	1	0.9951	0.7857	0.9956	0.4664	0	0.2413	1	0.0118
7	S340	0.5364	1	0.9982	0.4892	0.9993	0.5952	0.0033	0.243	0.0389	0.002
8	S275	0.53	1	1	0.7778	0.9996	0.4591	0	0.2249	0.0097	0.0011
9	S329	0.5267	1	0.9998	0.766	0.9996	0.4517	0.0004	0.2218	0.03	0.0012
10	S395	0.5234	0.8649	0.8552	0.5417	0.9286	0.4848	0.1663	0.3584	0.3252	0.1335
11	S131	0.5071	1	0.989	0.5089	0.9994	0.4775	0.0108	0.1959	1	0.0021
12	S268	0.5032	1	0.9998	0.8067	0.9999	0.363	0.0004	0.1843	0.0167	0.0005
13	S306	0.4958	1	0.9997	0.7896	0.9996	0.342	0.0005	0.1788	0.0154	0.0014
14	S151	0.4892	1	0.977	0.1076	0.9029	0.4443	0.0549	0.2765	0.0318	0.2296
15	S308	0.4838	1	0.9677	0.3673	0.9913	0.4443	0.0718	0.1949	0.0895	0.0291
16	S330	0.4658	1	0.9781	0.1941	0.9868	0.4333	0.049	0.1942	0.1502	0.0445
17	S194	0.465	1	0.9981	0.6834	0.9999	0.2821	0.0029	0.1437	0.0152	0.0004
18	S348	0.4592	0.9716	0.8865	0.1876	0.7513	0.1426	0.1263	0.1761	1	1
19	S374	0.453	1	0.7205	0.926	0.9061	0.0451	0.0859	0.1048	1	0.8771
20	S126	0.4527	0.8978	0.7884	0.5745	0.9061	0.2055	0.2555	0.2078	1	0.3447
21	S352	0.4519	1	0.9948	0.6427	0.9994	0.2552	0.0078	0.1263	0.0642	0.0031
22	S356	0.4518	1	0.9934	0.3056	0.9964	0.3439	0.0116	0.1836	0.1705	0.0129
23	S284	0.444	1	0.9964	0.9696	0.9992	0.1242	0.0034	0.0662	0.2919	0.0075
24	S247	0.4427	1	0.9991	0.8683	1	0.1573	0.0011	0.0805	0.0419	0.0003
25	S031	0.4343	1	0.9962	0.8486	1	0.1437	0.0044	0.0584	0.2413	0.0003
26	S307	0.4307	0.9932	0.6286	0.3069	0.9122	0.1639	0.1493	0.43	1	0.3682
27	S365	0.4233	1	0.9974	0.8001	0.9999	0.1179	0.0033	0.059	0.0895	0.0013
28	S143	0.4233	1	0.9563	0.3843	0.9978	0.232	0.0655	0.1181	0.8988	0.0122
29	S294	0.4107	1	0.991	0.8932	1	0.0513	0.0096	0.0269	0.1705	0.0001
30	S139	0.4058	0.9898	0.8249	0	0.8595	0.1698	0.1244	0.2539	1	0.3939
31	S218	0.4044	1	0.9607	0.8565	1	0.0418	0.032	0.022	0.8988	0.0011
32	S080	0.4002	1	0.9472	0.7975	0.9999	0.0513	0.0411	0.0274	0.8988	0.0022

33	S040	0.3942	1	0.9856	0.5719	0.9998	0.101	0.0204	0.0454	0.13	0.0023
34	S266	0.3941	1	0.9093	0.925	1	0.018	0.0697	0.0092	0.4942	0
35	S005	0.3885	0.9308	0.8876	0.913	1	0.0432	0.0216	0.0226	1	0.0005
36	S123	0.3858	1	0.8998	0.8316	1	0.0179	0.0729	0.0092	0.8988	0.0002
37	S367	0.3827	1	0.9735	0.4984	0.9999	0.0885	0.036	0.0375	0.2919	0.0022
38	S037	0.3799	0.9716	0.7717	0.3187	0.9826	0.1603	0.0947	0.1614	1	0.1033
39	S189	0.3765	0.9432	0.9033	0.7529	1	0.0399	0.0242	0.0213	1	0.0005
40	S346	0.3744	1	0.9808	0.4446	1	0.0797	0.0283	0.033	0.1502	0.0002
41	S244	0.373	1	0.9398	0.5496	1	0.0429	0.0533	0.0232	0.8988	0.0011
42	S364	0.3724	1	0.988	0.3856	0.9997	0.0856	0.0168	0.041	0.1705	0.0047
43	S055	0.3711	1	0.9523	0.4355	0.9989	0.0638	0.0745	0.0341	0.2919	0.0203
44	S210	0.3659	0.8286	0.6798	0.7398	0.9986	0.0771	0.3497	0.058	0.8988	0.0198
45	S074	0.3624	0.9251	0.731	0.686	0.999	0.0477	0.2576	0.0379	1	0.0239
46	S078	0.3611	0.9364	0.85	0.5797	0.9998	0.0565	0.1061	0.0292	0.8988	0.0053
47	S086	0.3542	0.9841	0.7487	0.5299	0.9979	0.0543	0.1435	0.0628	1	0.0452
48	S007	0.3501	1	0.7341	0.7004	1	0.0219	0.1674	0.0099	0.8988	0.0017
49	S273	0.3499	0.8967	0.8448	0.4774	0.9998	0.0712	0.091	0.0389	0.8988	0.0045
50	S003	0.3474	0.9546	0.8657	0.4683	0.9998	0.0403	0.0594	0.0269	1	0.0052
51	S291	0.3415	0.9841	0.8166	0.4774	0.9997	0.0251	0.105	0.0175	1	0.0173
52	S379	0.3395	0.9694	0.6808	0.9083	1	0.0032	0.0262	0.0075	1	0.0015
53	S342	0.3374	0.9716	0.66	0.9227	1	0.0027	0.0416	0.0033	1	0.0002
54	S338	0.3373	1	0.7894	0.2112	0.9938	0.0561	0.0737	0.1027	1	0.0799
55	S092	0.3323	0.9648	0.6338	0.9457	1	0.0016	0.0315	0.002	1	0.0001
56	S114	0.3317	0.9898	0.8406	0.3187	0.9994	0.0326	0.0671	0.026	1	0.0231
57	S221	0.3273	0.9728	0.5816	0.9932	1	0.0014	0.031	0.002	1	0
58	S392	0.3263	0.9262	0.6276	0.9361	1	0.0017	0.032	0.002	1	0
59	S292	0.3259	0.731	0.709	0.6375	0.9999	0.0723	0.1419	0.0355	0.8988	0.0026
60	S025	0.3252	0.924	0.6244	0.9356	1	0.001	0.0315	0.0018	1	0
61	S150	0.3247	1	0.8291	0.3686	1	0.0042	0.0795	0.0037	1	0.0098
62	S178	0.3238	1	0.5566	0.9784	1	0.0014	0.0318	0.0024	1	0.0001
63	S314	0.3227	1	0.8166	0.3856	1	0.004	0.0764	0.0033	1	0.0003
64	S169	0.3205	0.9535	0.5639	0.976	1	0.0015	0.0347	0.002	1	0
65	S174	0.3181	0.9909	0.5211	1	1	0.0013	0.0342	0.002	1	0
66	S256	0.3154	0.9387	0.6224	0.8145	1	0.0018	0.0376	0.0024	1	0.0001
67	S141	0.3145	0.9126	0.5743	0.9393	1	0.0018	0.0296	0.0021	1	0.0001
68	S324	0.3145	0.916	0.5701	0.9416	1	0.0018	0.0323	0.0021	1	0.0001
69	S113	0.314	0.9489	0.5493	0.9488	1	0.001	0.0294	0.0022	1	0.0001
70	S208	0.3115	0.6754	0.6067	0.6716	0.9993	0.0694	0.238	0.0345	0.7977	0.013
71	S030	0.3106	0.9614	0.5305	0.8985	1	0.0031	0.0612	0.0052	1	0.0005
72	S075	0.3091	0.924	0.567	0.8841	1	0.0013	0.0334	0.0023	1	0.0001
73	S152	0.3071	0.8922	0.567	0.8945	1	0.0014	0.0352	0.0021	1	0.0001
74	S157	0.3058	0.8967	0.5608	0.6388	1	0.0014	0.2993	0.0019	1	0
75	S300	0.3057	0.6901	0.6526	0.7293	1	0.022	0.1831	0.0128	0.5953	0.0016
76	S154	0.3043	0.7049	0.6485	0.5745	0.9997	0.0609	0.1432	0.0396	1	0.0055
77	S216	0.3029	0.8547	0.5597	0.9087	1	0.0018	0.0307	0.002	1	0

78	S193	0.3019	0.8104	0.5795	0.8959	1	0.0029	0.0355	0.0027	1	0.0001
79	S237	0.3014	0.9807	0.4866	0.9024	1	0.001	0.0416	0.002	1	0
80	S023	0.3011	0.8082	0.567	0.6794	1	0.0073	0.2574	0.0089	1	0.0026
81	S175	0.3008	0.9762	0.4511	0.9811	1	0.0012	0.0328	0.0021	1	0.0001
82	S253	0.2999	0.9364	0.4856	0.9456	1	0.0011	0.0294	0.0028	1	0.0001
83	S172	0.2977	0.9251	0.5023	0.8932	1	0.0013	0.0389	0.0019	1	0
84	S149	0.2966	0.874	0.5033	0.9402	1	0.0012	0.0294	0.0026	1	0.0001
85	S129	0.2963	0.9387	0.5931	0.5771	1	0.0046	0.1186	0.01	1	0.0027
86	S036	0.2939	0.8944	0.4793	0.9344	1	0.0016	0.0358	0.002	1	0.0001
87	S310	0.2932	0.8502	0.5033	0.928	1	0.0017	0.0318	0.0021	1	0.0001
88	S381	0.2861	0.8581	0.5242	0.7135	1	0.0038	0.1101	0.007	1	0.0009
89	S360	0.282	0.8502	0.5701	0.6598	1	0.0016	0.049	0.0019	1	0
90	S110	0.2805	0.8082	0.544	0.745	1	0.0012	0.0466	0.0019	1	0
91	S263	0.2766	0.8502	0.5555	0.6204	1	0.0012	0.0642	0.002	1	0.0001
92	S383	0.2711	0.7662	0.4626	0.8408	1	0.0028	0.0549	0.0043	1	0.0004
93	S064	0.2698	0.9319	0.5701	0.3646	1	0.0013	0.1382	0.002	1	0
94	S191	0.2695	0.8808	0.5179	0.4945	1	0.0015	0.1666	0.0019	1	0
95	S318	0.2692	0.9149	0.3644	0.8709	1	0.0016	0.0381	0.0113	1	0.0011
96	S388	0.269	0.8876	0.5054	0.5312	1	0.0002	0.1759	0.0025	0.4942	0.0001
97	S088	0.2681	0.8956	0.5138	0.5548	1	0.0015	0.0801	0.0022	1	0.0001
98	S046	0.2676	0.8649	0.3749	0.9115	1	0.0001	0.0389	0	1	0
99	S098	0.2673	0.9841	0.33	0.8565	1	0.0008	0.0535	0.0021	1	0
100	S186	0.2669	0.9137	0.5608	0.3935	1	0.0011	0.1167	0.0021	1	0.0001
101	S054	0.2658	0.8593	0.5075	0.5876	1	0.0024	0.0687	0.0033	1	0.0003
102	S076	0.2643	0.9773	0.2799	0.9499	1	0.0003	0.04	0.0017	1	0
103	S239	0.2597	1	0.6286	0.1574	1	0.001	0.0477	0.0019	1	0.0001
104	S227	0.2591	0.7378	0.4824	0.6073	1	0.0039	0.1496	0.0073	1	0.0018
105	S357	0.2582	0.7662	0.4083	0.8447	1	0.0014	0.0328	0.0036	1	0.0007
106	S235	0.2569	0.7911	0.3853	0.6742	1	0.0042	0.2064	0.0066	1	0.0016
107	S386	0.2561	0.7264	0.3916	0.8906	1	0.0019	0.0397	0.0037	1	0.0002
108	S332	0.2525	0.8888	0.4438	0.4945	1	0.0039	0.1085	0.009	1	0.0033
109	S202	0.2509	0.9489	0.5994	0.1574	1	0.0016	0.0695	0.002	1	0
110	S102	0.2496	0.7514	0.3634	0.6716	1	0.0008	0.2383	0.0026	1	0.0001
111	S115	0.2493	0.8388	0.5524	0.3463	1	0.0014	0.0748	0.002	1	0
112	S138	0.246	0.8785	0.5712	0.2204	1	0.0014	0.0888	0.0019	1	0
113	S271	0.2449	0.7616	0.4135	0.6204	1	0.0019	0.1156	0.0043	1	0.0007
114	S146	0.2419	0.891	0.3039	0.6086	1	0.0009	0.1812	0.0089	1	0.0008
115	S213	0.2408	0.9796	0.1556	0.9597	1	0.0001	0.0432	0	1	0
116	S245	0.24	0.924	0.3154	0.5588	1	0.0007	0.1297	0.0199	1	0.0018
117	S265	0.2397	0.7026	0.3133	0.6952	1	0.0012	0.2967	0.0031	0.4942	0.0003
118	S066	0.2389	0.9274	0.544	0.1692	1	0.0012	0.0705	0.0019	1	0
119	S067	0.2381	1	0.118	0.9879	1	0	0.0429	0	1	0
120	S199	0.2372	0.5743	0.2767	0.6873	1	0.0025	0.4586	0.0045	1	0.0004
121	S258	0.236	0.8785	0.5879	0.1167	1	0.0018	0.0519	0.0021	1	0.0001
122	S069	0.2349	0.6754	0.3885	0.6414	1	0.0014	0.1387	0.0027	1	0.0001

123	S065	0.2342	0.6901	0.3174	0.8276	1	0.0016	0.0801	0.0042	0.7977	0.0004
124	S128	0.2313	0.6799	0.2986	0.867	1	0.0009	0.0503	0.003	1	0.0002
125	S233	0.2305	0.6266	0.3582	0.7673	1	0.001	0.0772	0.0028	1	0.0002
126	S173	0.2302	0.3303	0.2506	0.7634	0.9999	0.0123	0.5674	0.0171	1	0.0114
127	S039	0.2291	0.6754	0.3467	0.5745	1	0.0042	0.208	0.011	1	0.0042
128	S376	0.2271	0.8604	0.4887	0.2308	1	0.0014	0.07	0.0018	1	0
129	S336	0.2266	0.6141	0.3133	0.8408	1	0.0022	0.0612	0.0046	1	0.0006
130	S232	0.2266	0.6027	0.3049	0.7804	1	0.0027	0.1533	0.0047	1	0.0004
131	S197	0.2252	0.9455	0.5315	0.0459	1	0.0014	0.0575	0.0021	1	0.0001
132	S313	0.2244	0.4904	0.2934	0.7162	1	0.0049	0.3338	0.0072	1	0.0008
133	S117	0.2176	0.5709	0.2579	0.6401	1	0.0002	0.3922	0.0026	0.3252	0.0002
134	S153	0.2174	0.1283	0.0428	0.7909	0.9999	0.0104	1	0.0526	1	0.0074
135	S002	0.2162	0.7128	0.2934	0.6178	1	0.0015	0.1156	0.0061	1	0.0011
136	S254	0.2142	0.521	0.2945	0.846	1	0.0034	0.0567	0.0061	1	0.0006
137	S089	0.2138	0.5244	0.2663	0.7122	1	0.0029	0.2555	0.0042	1	0.0004
138	S163	0.2128	0.5346	0.2788	0.6453	1	0.0027	0.2728	0.0062	1	0.0008
139	S217	0.2106	0.3927	0.3415	0.7857	1	0.006	0.1225	0.0029	1	0.0001
140	S369	0.2092	0.3053	0.1264	0.7634	1	0.0029	0.6895	0.0055	1	0.0006
141	S226	0.2058	0.5187	0.3018	0.6978	1	0.0031	0.1101	0.006	1	0.0007
142	S269	0.2058	0.9126	0.1451	0.4578	1	0.0001	0.2834	0.0019	1	0
143	S090	0.2042	0.63	0.308	0.505	1	0.0025	0.1637	0.005	1	0.0005
144	S228	0.2034	0.1635	0.1295	0.7463	1	0.0063	0.785	0.0058	1	0.0007
145	S159	0.2021	0.5153	0.2986	0.7188	1	0.0024	0.0636	0.0042	1	0.0006
146	S170	0.2005	0.5289	0.2652	0.606	1	0.003	0.2192	0.0049	1	0.0005
147	S024	0.1989	0.5278	0.2673	0.6755	1	0.0018	0.1332	0.0024	1	0.0001
148	S333	0.198	0.7526	0.283	0.3738	1	0.0007	0.1655	0.0018	1	0
149	S224	0.197	0.8456	0.1232	0.4761	1	0	0.294	0	1	0
150	S182	0.1925	0.504	0.1817	0.7135	1	0.0029	0.2247	0.0055	1	0.0006
151	S207	0.1921	0.4143	0.2099	0.6939	1	0.0009	0.2887	0.0032	1	0.0002
152	S035	0.19	0.4892	0.2276	0.7647	1	0.0014	0.1008	0.0036	0.4942	0.0003
153	S121	0.1883	0.4109	0.2203	0.7411	1	0.0027	0.1685	0.0058	1	0.0006
154	S304	0.1861	0.8638	0.2548	0.2531	1	0.0002	0.1034	0.0018	1	0
155	S042	0.1826	0.3712	0.2297	0.6676	1	0.0029	0.2088	0.005	1	0.0005
156	S209	0.1815	0.4404	0.2464	0.6375	1	0.0016	0.1215	0.0042	1	0.0008
157	S293	0.1795	0	0	0.6991	0.9999	0.0149	0.9522	0.0309	1	0.0085
158	S280	0.1786	0.5993	0.2193	0.4368	1	0.0004	0.1913	0.0033	1	0.0001
159	S105	0.1786	0.3814	0.2757	0.6414	1	0.0036	0.0838	0.0039	1	0.001
160	S334	0.1783	0.0227	0.0564	0.749	1	0.0082	0.8381	0.0051	0.6965	0
161	S028	0.1775	0.3053	0.237	0.7726	1	0.0002	0.1106	0.0031	1	0.0002
162	S104	0.174	0.2588	0.1723	0.7083	1	0.0029	0.3205	0.0044	1	0.0008
163	S017	0.1699	0.2883	0.1994	0.766	1	0.0028	0.1374	0.0046	0.7977	0.0005
164	S260	0.1697	0.8672	0.2611	0.0787	1	0.0002	0.0952	0	1	0
165	S167	0.1685	0.319	0.19	0.7162	1	0.0025	0.1475	0.0055	1	0.0005
166	S147	0.1674	0.319	0.1232	0.6676	1	0.013	0.2781	0.0171	1	0.0024
167	S010	0.1669	0.2781	0.1733	0.644	1	0.0032	0.286	0.0049	1	0.0005

168	S112	0.1642	0.2202	0.1775	0.6598	1	0.021	0.286	0.0138	0.1401	0.0018
169	S142	0.1632	0.235	0.1149	0.7122	1	0.0075	0.3338	0.0062	1	0.0009
170	S394	0.1575	0.0375	0.0752	0.707	1	0.0083	0.5992	0.0049	0.6965	0.0001
171	S375	0.1487	0.1067	0.0491	0.627	1	0.0073	0.4904	0.0321	1	0.0043
172	S022	0.141	0.1396	0.0783	0.6663	1	0.005	0.3258	0.0081	1	0.0009
173	S019	0.1378	0.0999	0.071	0.6703	1	0.0081	0.3391	0.008	1	0.0015
174	S071	0.1322	0.0545	0.0908	0.7555	1	0.0146	0.2093	0.0065	0.5953	0.0003
175	S077	0.1292	0.0999	0.0261	0.6296	1	0.0096	0.371	0.0129	1	0.0017
176	S127	0.1151	0.2906	0.1817	0.2164	1	0.0037	0.1414	0.0048	1	0.0005

C 问题三新评价

文件：供给商分析 - 第三题新评分.xlsx

表 9: 供给商分析 - 第三题新评分

代码	评分	排名	新评分	新排名	材料分类	供应偏差	平均忙期量	类型 (0-平稳, 1-周期)
S201	0.75974645	1	0.797733773	1	1	0.938597099	2930	1
S229	0.646160998	2	0.678469048	2	1	0.994387609	1470	0
S361	0.626986785	3	0.595637446	3	3	0.992012141	1370	0
S140	0.582720318	4	0.582720318	4	2	1.079308646	2190	1
S282	0.541186211	6	0.568245521	5	1	1.009170688	707	0
S108	0.562163907	5	0.562163907	6	2	0.983967758	1000	0
S275	0.529990375	8	0.556489894	7	1	1.003358179	659	0
S329	0.526701861	9	0.553036954	8	1	1.001927732	650	0
S395	0.523379239	10	0.549548201	9	1	1.034072873	1050	1
S340	0.536377517	7	0.536377517	10	2	0.999286394	712	0
S131	0.507056588	11	0.507056588	11	2	0.993171931	574	0
S308	0.483788422	15	0.483788422	12	2	0.974273932	571	0
S348	0.45921065	18	0.482171182	13	1	1.085455292	516	1
S268	0.503219779	12	0.47805879	14	3	1.003486449	540	0
S352	0.451897697	21	0.474492582	15	1	1.001532219	370	0
S306	0.495833855	13	0.471042162	16	3	1.004553487	524	0
S330	0.465845283	16	0.465845283	17	2	0.985486559	569	0
S151	0.489199392	14	0.464739422	18	3	0.983204211	810	0
S307	0.430694683	26	0.452229417	19	1	1.130532651	1260	1
S143	0.423261774	28	0.444424863	20	1	1.002846257	346	0
S194	0.465013512	17	0.441762836	21	3	1.001890698	421	0
S031	0.434323795	25	0.434323795	22	2	1.019790955	171	0
S374	0.452996397	19	0.430346578	23	3	1.284341248	307	1
S126	0.452707551	20	0.430072173	24	3	1.066199786	609	1
S356	0.451782973	22	0.429193825	25	3	1.002601807	538	0
S284	0.443962706	23	0.421764571	26	3	1.037560804	194	0
S247	0.442722083	24	0.420585979	27	3	1.013863546	236	0
S266	0.394086124	34	0.41379043	28	1	1.204428173	27.1	0

S005	0.388482065	35	0.407906168	29	1	1.098158422	66.3	1
S139	0.405768113	30	0.405768113	30	2	1.24830339	744	1
S123	0.385770748	36	0.405059285	31	1	1.13866868	26.9	0
S365	0.423267153	27	0.402103796	32	3	1.021675868	173	0
S189	0.376542632	39	0.395369764	33	1	1.055992216	62.5	1
S040	0.394213106	33	0.394213106	34	2	1.018639413	133	0
S294	0.410719677	29	0.390183693	35	3	1.042336086	78.7	0
S218	0.404435297	31	0.384213532	36	3	1.099639896	64.5	0
S367	0.38271972	37	0.38271972	37	2	1.030739196	110	0
S080	0.400158729	32	0.380150792	38	3	1.085448421	80.2	0
S078	0.361104426	46	0.379159648	39	1	1.046795208	85.5	1
S346	0.374376793	40	0.374376793	40	2	1.01692339	96.6	0
S364	0.372366756	42	0.372366756	41	2	1.038313593	120	0
S055	0.371128447	43	0.371128447	42	2	1.039237966	100	0
S007	0.350107187	48	0.367612547	43	1	1.288458314	28.9	0
S273	0.349888122	49	0.367382528	44	1	1.097859688	114	0
S037	0.37993823	38	0.360941318	45	3	1.074224244	473	1
S244	0.372966686	41	0.354318352	46	3	1.071043207	68.1	0
S210	0.365853787	44	0.347561098	47	3	0.991005503	170	1
S074	0.362386554	45	0.344267227	48	3	1.055334615	111	1
S086	0.354186795	47	0.336477455	49	3	1.08576467	184	1
S003	0.347399242	50	0.33002928	50	3	1.069185997	78.7	1

D 问题一代码

文件: Problem1.py

```

1 #第一个模块——对供应及时性的考量,可以考虑称之为“履约率”
2 #目标: 给出shape=(402,)的数列, 对应各个供应商的供货及时性
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import numpy as np
6 from collections import Counter
7 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
8 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←
    供货量 (m³) ')
9
10 temp = df[df.columns[2:]]
11 temp1 = df1[df1.columns[2:]]
12 '''
13 temp.replace(0.,np.nan,inplace=True)
14 temp1.replace(0.,np.nan,inplace=True)
15 '''

```

```

16 temp = np.array(temp)
17 temp1 = np.array(temp1)
18
19 bool_m=temp!=0
20 bool_m1=temp1!=0
21
22 bool_m_count=np.sum(bool_m,axis=1)
23 bool_m1_count=np.sum(bool_m1,axis=1)
24
25 result=(bool_m1_count)/(bool_m_count)
26 print(result)
27
28
29 #第二个模块——产品占有率
30 #目标：给出三个数列，对应三种原材料各个供应商的产品占有率
31 import pandas as pd
32 import matplotlib.pyplot as plt
33 import numpy as np
34 from collections import Counter
35 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的↵
    供货量 (m³) ')
36
37 dfA_bool=df1["材料分类"]=="A"
38 dfA=df1[dfA_bool]
39
40 dfB_bool=df1["材料分类"]=="B"
41 dfB=df1[dfB_bool]
42
43 dfC_bool=df1["材料分类"]=="C"
44 dfC=df1[dfC_bool]
45
46 def every_week_total_offer(df):
47     temp = df[df.columns[2:]]
48     return temp.values,np.sum(temp.values,axis=0)
49
50 A_offer,A_sum=every_week_total_offer(dfA)
51 B_offer,B_sum=every_week_total_offer(dfB)
52 C_offer,C_sum=every_week_total_offer(dfC)
53
54 def rate(offer,sum,n):
55     position=offer[n]!=0
56
57     return np.mean(offer[n][position]/sum[position])
58
59 A=[]

```



```

60 B=[]
61 C=[]
62 for i in range(A_offer.shape[0]):
63     A.append(rate(A_offer,A_sum,i))
64 for i in range(B_offer.shape[0]):
65     B.append(rate(B_offer,B_sum,i))
66 for i in range(C_offer.shape[0]):
67     C.append(rate(C_offer,C_sum,i))
68
69 print(A)
70 print(B)
71 print(C)
72
73
74 #第三个模块——批量柔性，也即每一批次最大增加的百分比
75 #目标：给出一个数列，对应每个供应商的批量柔性指标
76 import pandas as pd
77 import matplotlib.pyplot as plt
78 import numpy as np
79 from collections import Counter
80 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
81 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←
    供货量 (m³) ')
82
83 temp = df[df.columns[2:]]
84 temp1 = df1[df1.columns[2:]]
85
86 delta=temp1-temp
87 data=delta/temp
88 data=data.fillna(0)
89
90 result=np.max(data.values,axis=1)
91 print(result)
92
93
94 #第四个模块——交货准确度
95 #目标：给出一个数列，对应每个供应商的交货准确度指标
96 import pandas as pd
97 import matplotlib.pyplot as plt
98 import numpy as np
99 from collections import Counter
100 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
101 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←

```

```

    供货量 (m³) ')
102
103 temp = df[df.columns[2:]]
104 temp1 = df1[df1.columns[2:]]
105
106 delta=temp1-temp
107 data=delta/temp
108 data=data*data
109 data=data.fillna(0)
110 data=data.values
111
112 result=[]
113
114 for i in range(402):
115     position=data[i]!=0
116     result.append(np.mean(data[i][position]))
117
118 print(result)
119
120
121 #第五个模块——供货波动性
122 #目标：给出一个数列，对应每个供应商供货波动性的指标
123 import pandas as pd
124 import matplotlib.pyplot as plt
125 import numpy as np
126 from collections import Counter
127 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
128 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←
    供货量 (m³) ')
129
130 temp = df[df.columns[2:]]
131 temp1 = df1[df1.columns[2:]]
132
133 delta=temp1-temp
134 data=delta/temp
135 data=data.fillna(0)#这几步是为了排除未产生的订单
136 data*=temp
137 data=data.values
138
139 result=[]
140
141 for i in range(402):
142     position=data[i]!=0
143     result.append(np.std(data[i][position],ddof = 1)/np.mean(data[i][position]))

```

```

144
145 print(result)
146
147
148 #第六个模块——信誉度
149 #目标：给出一个数列，对应每个供应商信誉度的指标
150 import pandas as pd
151 import matplotlib.pyplot as plt
152 import numpy as np
153 from collections import Counter
154 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
155 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←
    供货量 (m³) ')
156
157 temp = df[df.columns[2:]]
158 temp1 = df1[df1.columns[2:]]
159
160 delta=temp1-temp
161 data=delta/temp
162 data=data*data*log(temp)
163 data=data.fillna(0)
164 data=data.values
165
166 result=[]
167
168 for i in range(402):
169     position=data[i]!=0
170     result.append(np.mean(data[i][position]))
171
172 print(result)
173
174
175 #第七个模块——供给弹性
176 #目标：给出一个数列，对应每个供应商供给弹性的指标
177 import pandas as pd
178 import matplotlib.pyplot as plt
179 import numpy as np
180 from collections import Counter
181 df = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='企业的订货←
    量 (m³) ')
182 df1 = pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的←
    供货量 (m³) ')
183
184 temp = df[df.columns[2:]]

```

```

185 temp1 = df1[df1.columns[2:]]
186
187 data=temp1/temp
188 data=data*temp
189 data=data.fillna(0)
190 data=data.values
191
192 result=[]
193
194 for i in range(402):
195     position=data[i]!=0
196     result.append(np.var(data[i][position],ddof = 1)/np.mean(data[i][position]))
197
198 print(result)

```

E 问题二代码

文件: Problem2.py

```

1 # 贪心思路
2 #A属性的分数乘以1.05, B不变, C变为0.95
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 plt.rcParams['font.sans-serif'] = ['SimHei']
7 # 1. 对象定义与解释
8 # 供应商1-50对应的代号, 排序顺序为第一问中评分高低
9 GongyingName = [] # 字符串, 从1开始, 下同
10 # 转运商1-8对应的代号, 排序顺序为评分高低
11 ZhuanyunName = []
12 # 供应商类别, 0为平稳, 1为周期
13 Gclass = []
14 Gabc1 = [] # A, B, C, 此为成本
15 Gabc2 = [] # 此为产能转化
16 # 平均值
17 Gmean = []
18 # 供应商的偏差率、方差
19 p = []
20 p_fc = []
21 # 周期
22 Gweek = np.random.rand(51)
23 Gweekmark = []
24 Gworkweekmark = []

```

```

25 # 转运商的损耗率
26 l = []
27 l_fc = []#fc为方差
28 # 第k周的剩余库存
29 w = np.zeros([25]) # 0-24, 未减去当前一周消耗
30 w[0] = 28200 * 2
31 # 当前周的供货量、接收量
32 s = np.zeros([51], dtype=int)
33 t = np.zeros([51], dtype=int)
34 # 当前车装载情况
35 car = np.zeros([9], dtype=int)
36
37 # -----策略
38 # 第i个供应商的第k周订货量
39 d = np.zeros([51, 25], dtype=int)
40 # 第i个供应商的第k周选择的转运商
41 J = np.zeros([51, 25], dtype=int)
42
43 # -----目标
44 C = np.zeros([25])
45 L = np.zeros([25])
46 Furest=np.zeros([25])
47 True_use=np.zeros([25])
48
49
50 # 2. 数据载入
51 from xlrd import open_workbook
52
53 book=open_workbook(r'转运量分析.xlsx')
54 sheet=book.sheets()[0]
55 ZhuanyunName = np.array([x.value for x in sheet.col(0,start_rowx=1, end_rowx=9)])
56 ZhuanyunName = np.append([''], ZhuanyunName)
57 l = np.array([float(x.value) for x in sheet.col(1,start_rowx=1, end_rowx=9)])
58 l = np.append([0], l)
59 l_fc = np.array([float(x.value) for x in sheet.col(2,start_rowx=1, end_rowx=9)])
60 l_fc = np.append([0], l_fc)
61
62
63 book=open_workbook(r'供给商分析.xlsx')
64 sheet=book.sheets()[0]
65 GongyingName = np.array([x.value for x in sheet.col(0, start_rowx=1, end_rowx=51)]←
    ])
66 GongyingName = np.append([''], GongyingName)
67
68 p = np.array([float(x.value) for x in sheet.col(5, start_rowx=1, end_rowx=51)])

```

```

69 p = np.append([0], p)
70 p_fc = np.array([float(x.value) for x in sheet.col(6,start_rowx=1, end_rowx=51)])
71 p_fc = np.append([0], p_fc)
72
73 Gclass = np.array([int(x.value) for x in sheet.col(11, start_rowx=1, end_rowx=51)↵
    ])
74 Gclass = np.append([''], Gclass)
75 Gabc1 = np.array([float(x.value) for x in sheet.col(3, start_rowx=1, end_rowx=51)↵
    ])
76 Gabc1 = np.append([0], Gabc1)
77 Gabc2 = np.array([float(x.value) for x in sheet.col(4,start_rowx=1, end_rowx=51)↵
    ])
78 Gabc2 = np.append([1], Gabc2)
79
80 Gweekmark = np.array([float(x.value) for x in sheet.col(7, start_rowx=1, end_rowx↵
    =51)])
81 Gweekmark = np.append([0], Gweekmark)
82
83 Gworkweekmark = np.array([float(x.value) for x in sheet.col(9, start_rowx=1, ↵
    end_rowx=51)])
84 Gworkweekmark = np.append([0], Gworkweekmark)
85
86 # 周期随机初始化
87 Gweek = Gweek * Gworkweekmark
88 Gweek = Gweek.astype(int)
89
90 Gmean = np.array([float(x.value) for x in sheet.col(10, start_rowx=1, end_rowx↵
    =51)])
91 Gmean = np.append([0], Gmean)
92
93 Everyweek_get=np.zeros([25])
94 LOST=np.zeros([9,25])
95
96
97 # 3. 贪心过程, 对每一周来记
98 for k in range(1, 25):
99     w[k] = w[k-1] - 28200
100
101     for i in range(1, 51):
102         if Gclass[i] == 0:
103             if Gmean[i]>2500:
104                 d[i, k] = Gmean[i]*1.3 + np.random.normal(0, 500)
105             else:
106                 d[i, k] = Gmean[i] * 1.3 + np.random.normal(0, 200)
107         else:

```

```

108         if Gweek[i] >= Gworkweekmark[i]:
109             if Gmean[i] > 2500:
110                 d[i, k] = Gmean[i] * 1.3 + np.random.normal(0, 500)
111             else:
112                 d[i, k] = Gmean[i] * 1.3 + np.random.normal(0, 200)
113             Gweek[i] = 0
114         elif Gweek[i] >= Gweekmark[i]:
115             if Gmean[i] > 2500:
116                 d[i, k] = Gmean[i] * 1.3 + np.random.normal(0, 500)
117             else:
118                 d[i, k] = Gmean[i] * 1.3 + np.random.normal(0, 200)
119         else:
120             continue # next i
121     True_use[k] += 1
122     Furest[k] = i
123
124     pp = np.random.normal(p[i], p_fc[i], 1)
125     s[i] = pp * d[i, k]
126     for j in range(1, 9):
127         if car[j] + s[i] <= 6000:
128             J[i, k] = j
129             car[j] += s[i]
130             LOST[j,k] += s[i]*np.random.normal(1[J[i, k]], 1_fc[J[i, k]], 1)
131             break
132     ll = np.random.normal(1[J[i, k]], 1_fc[J[i, k]], 1)
133
134     t[i] = (1-ll) * s[i]
135
136     w[k] += t[i] / Gabc2[i]
137     if w[k] >= 28200 * 2:
138         break
139     Gweek += 1
140     Everyweek_get[k]=sum(t/ Gabc2)
141     C[k] = sum(s * Gabc1)
142     L[k] = sum((s - t) / Gabc2)
143
144     # clean
145     # 当前周的供货量、接收量
146     s = np.zeros([51], dtype=int)
147     t = np.zeros([51], dtype=int)
148     # 当前车装载情况
149     car = np.zeros([9], dtype=int)
150
151
152     # 4. 绘制图表

```

```

153 ax1 = plt.gca()
154 ax1.set_ylim(15000,62000)
155 ax2 = ax1.twinx()
156 ax1.plot(range(1,25),C[1:],label='每周成本')
157 ax1.plot(range(1,25),w[1:],label='每周剩余库存')
158 ax1.plot(range(1,25),Everyweek_get[1:],label='每周进货')
159 ax1.plot(range(1, 25), [28200 * 2]*24, label='最低库存')
160 ax2.bar(range(1,25),LOST[1,1:],label='每周转运商1损失',alpha=0.3,bottom=0)
161 ax2.bar(range(1,25),LOST[2,1:],label='每周转运商2损失',alpha=0.3,bottom=LOST[1,1:])
162 ax2.bar(range(1,25),LOST[3,1:],label='每周转运商3损失',alpha=0.3,bottom=LOST[1,1:]+LOST[2,1:])
163 ax2.bar(range(1,25),LOST[4,1:],label='每周转运商4损失',alpha=0.3,bottom=LOST[1,1:]+LOST[2,1:]+LOST[3,1:])
164 ax2.set_ylim(0,200)
165 ax2.legend(loc=6,ncol=1)
166 plt.xlabel('安排周数')
167 ax1.legend(loc=0,ncol=1)
168 plt.grid(linestyle=":", color="k")
169 plt.title('每周成本, 进货, 库存随周数变化')
170
171 plt.show()

```

F 问题三代码

文件: Problem3.py

```

1  # 贪心思路
2  import random
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6  plt.rcParams['font.sans-serif'] = ['SimHei']
7
8  random.seed(0)
9  def test_func(sup_pram,name):
10     # 1. 对象定义与解释
11     Cost_accumulated = np.zeros([3, 25])
12
13     # 供应商1-50对应的代号, 排序顺序为第一问中评分高低
14     GongyingName = [] # 字符串, 从1开始, 下同理
15     # 转运商1-8对应的代号, 排序顺序为评分高低
16     ZhuanyunName = []

```



```

17
18     # 供应商类别, 0为平稳, 1为周期
19     Gclass = []
20     Gabc1 = [] # A, B, C, 此为成本
21     Gabc2 = [] # 此为产能转化
22     # 平均值
23     Gmean = []
24     # 供应商的偏差率、方差
25     p = []
26     p_fc = []
27     # 周期
28     Gweek = np.random.rand(51)
29     Gweekmark = []
30     Gworkweekmark = []
31
32     # 转运商的损耗率
33     l = []
34     l_fc = [] # fc即是方差
35
36     # 第k周的剩余库存
37     w = np.zeros([25]) # 0-24, 未减去当前一周消耗
38     w[0] = 28200 * 2
39
40     # 当前周的供货量、接收量
41     s = np.zeros([51], dtype=int)
42     t = np.zeros([51], dtype=int)
43     # 当前车装载情况
44     car = np.zeros([9], dtype=int)
45
46     # -----策略
47     # 第i个供应商的第k周订货量
48     d = np.zeros([51, 25], dtype=int)
49     # 第i个供应商的第k周选择的转运商
50     J = np.zeros([51, 25], dtype=int)
51
52     # -----目标
53     C = np.zeros([25])
54     L = np.zeros([25])
55
56     # 2. 数据载入
57     from xlrd import open_workbook
58
59     book = open_workbook(r'转运量分析.xlsx')
60     sheet = book.sheets()[0]
61     ZhuanyunName = np.array([x.value for x in sheet.col(0, start_rowx=1, end_rowx=

```

```

        =9)])
62     ZhuanyunName = np.append([''], ZhuanyunName)
63     l = np.array([float(x.value) for x in sheet.col(1, start_rowx=1, end_rowx=9)↵
        ])
64     l = np.append([0], l)
65     l_fc = np.array([float(x.value) for x in sheet.col(2, start_rowx=1, end_rowx=↵
        =9)])
66     l_fc = np.append([0], l_fc)
67
68     book = open_workbook(name)
69     sheet = book.sheets()[0]
70     GongyingName = np.array([x.value for x in sheet.col(0, start_rowx=1, end_rowx=↵
        =51)])
71     GongyingName = np.append([''], GongyingName)
72
73     p = np.array([float(x.value) for x in sheet.col(5, start_rowx=1, end_rowx=51)↵
        ])
74     p = np.append([0], p)
75     p_fc = np.array([float(x.value) for x in sheet.col(6, start_rowx=1, end_rowx=↵
        =51)])
76     p_fc = np.append([0], p_fc)
77
78     Gclass = np.array([int(x.value) for x in sheet.col(11, start_rowx=1, end_rowx=↵
        =51)])
79     Gclass = np.append([''], Gclass)
80
81     Gscore = np.array([float(x.value) for x in sheet.col(12, start_rowx=1, ↵
        end_rowx=51)])
82     Gscore = np.append([0], Gscore)
83
84     Gabc1 = np.array([float(x.value) for x in sheet.col(3, start_rowx=1, end_rowx=↵
        =51)])
85     Gabc1 = np.append([0], Gabc1)
86     Gabc2 = np.array([float(x.value) for x in sheet.col(4, start_rowx=1, end_rowx=↵
        =51)])
87     Gabc2 = np.append([1], Gabc2)
88
89     Gweekmark = np.array([float(x.value) for x in sheet.col(7, start_rowx=1, ↵
        end_rowx=51)])
90     Gweekmark = np.append([0], Gweekmark)
91
92     Gworkweekmark = np.array([float(x.value) for x in sheet.col(9, start_rowx=1, ↵
        end_rowx=51)])
93     Gworkweekmark = np.append([0], Gworkweekmark)
94

```

```

95     Gweek = Gweek * Gworkweekmark
96     Gweek = Gweek.astype(int)
97
98     Gmean = np.array([float(x.value) for x in sheet.col(10, start_rowx=1, ↵
        end_rowx=51)])
99     Gmean = np.append([0], Gmean)
100
101     Everyweek_get = np.zeros([25])
102     LOST = np.zeros([9, 25])
103     Everyweek_chooseA=np.zeros([25])
104     Everyweek_chooseC = np.zeros([25])
105
106
107     # 3. 贪心过程, 对每一周来记
108     #从总体上说,
109     for k in range(1, 25):
110         w[k] = w[k - 1] - 28200
111
112         chooseA = 0
113         chooseC = 0
114
115         for i in range(1, 51):
116             if i<50 and 0.98*Gscore[i]<Gscore[i+1] and Gabcl[i]<Gabcl[i+1]:
117                 continue
118             if Gclass[i] == 0:
119                 d[i, k] = Gmean[i]*(sup_pram+np.random.normal(0.2, 0.5))
120             else:
121                 if Gweek[i] >= Gworkweekmark[i]:
122                     d[i, k] = Gmean[i]*(sup_pram+np.random.normal(0.2, 0.5))
123                 elif Gweek[i] >= Gweekmark[i]:
124                     d[i, k] = Gmean[i]*(sup_pram+np.random.normal(0.2, 0.5))
125                 else:
126                     continue # next i
127
128             if Gabcl[i]==1.2:
129                 chooseA+=1
130             elif Gabcl[i]==1:
131                 chooseC+=1
132
133             pp = np.random.normal(p[i], p_fc[i], 1)
134             s[i] = pp * d[i, k]
135
136             for j in range(1, 9):
137                 if car[j] + s[i] <= 6000:
138                     J[i, k] = j

```

```

139         car[j] += s[i]
140         LOST[j, k] += s[i] * np.random.normal(1[J[i, k]], l_fc[J[i, k]←
            ], 1)
141         break
142     ll = np.random.normal(1[J[i, k]], l_fc[J[i, k]], 1)
143
144
145     t[i] = (1 - ll) * s[i]
146
147     w[k] += t[i] / Gabc2[i]
148     if w[k] >= 28200 * 2:
149         break
150     Gweek += 1
151     Everyweek_get[k] = sum(t / Gabc2)
152     C[k] = sum(s * Gabc1)
153     L[k] = sum((s - t) / Gabc2)
154     Cost_accumulated[0][k] = sum(s[np.where(Gabc1 == 1.2)] / 0.6)
155     Cost_accumulated[1][k] = sum(s[np.where(Gabc1 == 1.1)] / 0.66)
156     Cost_accumulated[2][k] = sum(s[np.where(Gabc1 == 1)] / 0.72)
157
158     # clean
159     # 当前周的供货量、接收量
160     s = np.zeros([51], dtype=int)
161     t = np.zeros([51], dtype=int)
162     # 当前车装载情况
163     car = np.zeros([9], dtype=int)
164     Everyweek_chooseA[k]=chooseA
165     Everyweek_chooseC[k]=chooseC
166
167
168     ax1 = plt.gca()
169     ax1.set_ylim(15000, 62000)
170     ax2 = ax1.twinx()
171     ax1.plot(range(1, 25), C[1:], label='每周成本')
172     ax1.plot(range(1, 25), w[1:], label='每周剩余库存')
173     ax1.plot(range(1, 25), Everyweek_get[1:], label='每周进货')
174     ax1.plot(range(1, 25), [28200 * 2]*24, label='最低库存')
175     ax2.bar(range(1, 25), LOST[1, 1:], label='每周转运商1损失', alpha=0.3, bottom←
        =0)
176     ax2.bar(range(1, 25), LOST[2, 1:], label='每周转运商2损失', alpha=0.3, bottom←
        =LOST[1, 1:])
177     ax2.bar(range(1, 25), LOST[3, 1:], label='每周转运商3损失', alpha=0.3, bottom←
        =LOST[1, 1:] + LOST[2, 1:])
178     ax2.bar(range(1, 25), LOST[4, 1:], label='每周转运商4损失', alpha=0.3, bottom←
        =LOST[1, 1:] + LOST[2, 1:] + LOST[3, 1:])

```

```

179     ax2.set_ylim(0, 200)
180     ax2.legend(loc=6, ncol=1)
181     plt.xlabel('安排周数')
182     ax1.legend(loc=0, ncol=1)
183     plt.grid(linestyle=":", color="k")
184     plt.title('每周成本, 进货, 库存随周数变化')
185
186     plt.show()
187     # 每周实际收货量的变化图
188     return LOST[1, 1:] + LOST[2, 1:] + LOST[3, 1:] + LOST[4, 1:],↵
        Everyweek_chooseA,Everyweek_chooseC,Cost_accumulated[0],Cost_accumulated↵
        [2]
189
190
191 LOST1,Everyweek_chooseA1,Everyweek_chooseC1,Pa1,Pc1=test_func(1.5,r'供给商分析.↵
    xlsx')
192 LOST2,Everyweek_chooseA2,Everyweek_chooseC2,Pa2,Pc2=test_func(1.5,r'供给商分析 - ↵
    第二题.xlsx')
193
194 plt.xlabel('安排周数')
195 plt.plot(range(1, 25),LOST1,label="调整后损失")
196 plt.plot(range(1, 25),LOST2,label="调整前损失")
197 plt.legend(loc=0, ncol=1)
198 plt.title('前后损失比较')
199 plt.show()
200
201 plt.xlabel('安排周数')
202 plt.plot(range(1, 25),(Everyweek_chooseA1/Everyweek_chooseC1)[1:],label="调整后A/↵
    C比例")
203 plt.plot(range(1, 25),(Everyweek_chooseA2/Everyweek_chooseC2)[1:],label="调整前A/↵
    C比例")
204 plt.legend(loc=0, ncol=1)
205 plt.title('前后A/C比例比较')
206 plt.show()
207
208 plt.xlabel('安排周数')
209 plt.plot(range(1, 25),(Everyweek_chooseA1-Everyweek_chooseC1)[1:],label="调整后A,↵
    C之差")
210 plt.plot(range(1, 25),(Everyweek_chooseA2-Everyweek_chooseC2)[1:],label="调整前A,↵
    C之差")
211 plt.legend(loc=0, ncol=1)
212 plt.title('前后A/C比例比较')
213 plt.show()
214
215 plt.xlabel('安排周数')

```

```

216 plt.plot(range(1, 25), (Pa1/Pc1)[1:], label="调整后产能A/C比例")
217 plt.plot(range(1, 25), (Pa2/Pc2)[1:], label="调整前产能A/C比例")
218 plt.legend(loc=0, ncol=1)
219 plt.title('前后A/C比例比较')
220 plt.show()

```

G 问题四代码

文件: Problem4.py

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  plt.rcParams['font.sans-serif'] = ['SimHei']
5  # 1. 对象定义与解释
6
7  # 供应商1-50对应的代号, 排序顺序为第一问中评分高低
8  GongyingName = [] # 字符串, 从1开始, 下同
9  # 转运商1-8对应的代号, 排序顺序为评分高低
10 ZhuanyunName = []
11
12 # 供应商类别, 0为平稳, 1为周期
13 Gclass = []
14 Gabc1 = [] # A, B, C, 此为成本
15 Gabc2 = [] # 此为产能转化
16 # 平均值 ✓
17 Gmean = []
18 # 供应商的偏差率、方差
19 p = []
20 p_fc = []
21 #%% 周期 ✓
22 Gweek = np.random.rand(177)
23
24 #%%
25 Gweekmark = []
26 Gworkweekmark = []
27
28 # 转运商的损耗率
29 l = []
30 l_fc = []
31
32 # 第k周的剩余库存
33 w = np.zeros([25]) # 0-24, 未减去当前一周消耗

```

```

34 w[0] = 56800
35 # 当前周的供货量、接收量
36 s = np.zeros([177], dtype=int)
37 t = np.zeros([177], dtype=int)
38 # 当前车装载情况
39 car = np.zeros([9], dtype=int)
40
41 # -----策略
42 # 第i个供应商的第k周订货量
43 d = np.zeros([177, 25], dtype=int)
44 # 第i个供应商的第k周选择的转运商
45 J = np.zeros([177, 25], dtype=int)
46
47 # -----目标
48 C = np.zeros([25])
49 L = np.zeros([25])
50
51 Everyweek_get = np.zeros([25])
52 LOST = np.zeros([9, 25])
53
54 ### 2. 数据载入
55 from xlrd import open_workbook
56
57 book=open_workbook(r'转运量分析.xlsx')
58 sheet=book.sheets()[0]
59 ZhuanyunName = np.array([x.value for x in sheet.col(0,start_rowx=1, end_rowx=9)])
60 ZhuanyunName = np.append([''], ZhuanyunName)
61 l = np.array([float(x.value) for x in sheet.col(1,start_rowx=1, end_rowx=9)])
62 l = np.append([0], l)
63 l_fc = np.array([float(x.value) for x in sheet.col(2,start_rowx=1, end_rowx=9)])
64 l_fc = np.append([0], l_fc)
65
66 book=open_workbook(r'供给商分析.xlsx')
67 sheet=book.sheets()[0]
68 GongyingName = np.array([x.value for x in sheet.col(0, start_rowx=1, end_rowx=↵
    =177)])
69 GongyingName = np.append([''], GongyingName)
70 p = np.array([float(x.value) for x in sheet.col(5, start_rowx=1, end_rowx=177)])
71 p = np.append([0], p)
72 p_fc = np.array([float(x.value) for x in sheet.col(6,start_rowx=1, end_rowx=177)↵
    ])
73 p_fc = np.append([0], p_fc)
74
75 Gclass = np.array([int(x.value) for x in sheet.col(11, start_rowx=1, end_rowx=↵
    =177)])

```

```

76 Gclass = np.append([''], Gclass)
77 Gabc1 = np.array([float(x.value) for x in sheet.col(3, start_rowx=1, end_rowx=177)])
78 Gabc1 = np.append([0], Gabc1)
79 Gabc2 = np.array([float(x.value) for x in sheet.col(4, start_rowx=1, end_rowx=177)])
80 Gabc2 = np.append([1], Gabc2)
81
82 Gweekmark = np.array([float(x.value) for x in sheet.col(7, start_rowx=1, end_rowx=177)])
83 Gweekmark = np.append([0], Gweekmark)
84 Gworkweekmark = np.array([float(x.value) for x in sheet.col(9, start_rowx=1, end_rowx=177)])
85 Gworkweekmark = np.append([0], Gworkweekmark)
86
87 Gweek = Gweek * Gworkweekmark
88 Gweek = Gweek.astype(int)
89
90 Gmean = np.array([float(x.value) for x in sheet.col(10, start_rowx=1, end_rowx=177)])
91 Gmean = np.append([0], Gmean)
92
93 Productivity=np.zeros([25])
94 Cost_accumulated=np.zeros([3,25])
95
96 Furest=np.zeros([25])
97 True_use=np.zeros([25])
98
99 ### 3. 贪心过程，对每一周来记
100 for k in range(1, 25):
101     w[k] = w[k-1]/2 #每周消耗该周库存产能的一半
102     Productivity[k]=w[k-1]/2
103     for i in range(1, 177):
104         if Gclass[i] == 0:
105             d[i, k] = Gmean[i]*(1.3+np.random.normal(0.2, 0.5))
106         else:
107             if Gweek[i] >= Gworkweekmark[i]:
108                 d[i, k] = Gmean[i]*(1.3+np.random.normal(0.2, 0.5))
109                 Gweek[i] = 0
110             elif Gweek[i] >= Gweekmark[i]:
111                 d[i, k] = Gmean[i]*(1.3+np.random.normal(0.2, 0.5))
112             else:
113                 continue # next i.
114         True_use[k]+=1
115         Furest[k]=i

```



```

116     pp = np.random.normal(p[i], p_fc[i], 1)
117     s[i] = pp * d[i, k]
118     mk = 0
119     for j in range(1, 9):
120         if car[j] + s[i] <= 6000:
121             J[i, k] = j
122             car[j] += s[i]
123             LOST[j, k] += s[i] * np.random.normal(1[J[i, k]], l_fc[J[i, k]], 1)
124             break
125         if j == 8:
126             mk = 1
127     if mk == 1:
128         break
129     l1 = np.random.normal(1[J[i, k]], l_fc[J[i, k]], 1)
130     t[i] = (1-l1) * s[i]
131
132     w[k] += t[i] / Gabc2[i]
133     Gweek += 1
134     Everyweek_get[k] = sum(t / Gabc2)
135     C[k] = sum(s * Gabc1)
136     L[k] = sum((s - t) / Gabc2)
137     Cost_accumulated[0][k] = sum(s[np.where(Gabc1==1.2)]*1.2)
138     Cost_accumulated[1][k] = sum(s[np.where(Gabc1==1.1)]*1.1)
139     Cost_accumulated[2][k] = sum(s[np.where(Gabc1 == 1)] * 1)
140     # clean
141     # 当前周的供货量、接收量
142     s = np.zeros([177], dtype=int)
143     t = np.zeros([177], dtype=int)
144     # 当前车装载情况
145     car = np.zeros([9], dtype=int)
146
147     ax1 = plt.gca()
148     #ax1.set_ylim(15000, 62000)
149     ax2 = ax1.twinx()
150     ax1.plot(range(1, 25), C[1:], label='每周成本')
151     ax1.plot(range(1, 25), w[1:], label='每周剩余库存')
152     ax1.plot(range(1, 25), Everyweek_get[1:], label='每周进货')
153     ax2.bar(range(1, 25), LOST[1, 1:], label='每周转运商1损失', alpha=0.3, bottom=0)
154     ax2.bar(range(1, 25), LOST[2, 1:], label='每周转运商2损失', alpha=0.3, bottom=↵
        LOST[1, 1:])
155     ax2.bar(range(1, 25), LOST[3, 1:], label='每周转运商3损失', alpha=0.3, bottom=↵
        LOST[1, 1:] + LOST[2, 1:])
156     ax2.bar(range(1, 25), LOST[4, 1:], label='每周转运商4损失', alpha=0.3, bottom=↵
        LOST[1, 1:] + LOST[2, 1:] + LOST[3, 1:])

```

```

157 ax2.bar(range(1, 25), LOST[5, 1:], label='每周转运商5损失', alpha=0.3, bottom=↵
        LOST[1, 1:] + LOST[2, 1:] + LOST[3, 1:]+LOST[4, 1:])
158 #ax2.set_ylim(0, 200)
159 ax2.legend(loc=6, ncol=1)
160 plt.xlabel('安排周数')
161 ax1.legend(loc=0, ncol=1)
162 plt.grid(linestyle=":", color="k")
163 plt.title('每周成本, 进货, 库存随周数变化')
164
165 plt.show()
166
167 ax1 = plt.gca()
168 ax1.set_ylim(15000, 60000)
169 ax2 = ax1.twinx()
170 ax1.plot(range(1, 25), Productivity[1:], label='每周产能')
171 ax1.plot(range(1, 25), Everyweek_get[1:], label='每周获得产能')
172 ax2.bar(range(1, 25), Cost_accumulated[0, 1:], label='每周A采购成本', alpha=0.3, ↵
        bottom=0)
173 ax2.bar(range(1, 25), Cost_accumulated[1, 1:], label='每周B采购成本', alpha=0.3, ↵
        bottom=Cost_accumulated[0, 1:])
174 ax2.bar(range(1, 25), Cost_accumulated[2, 1:], label='每周C采购成本', alpha=0.3, ↵
        bottom=Cost_accumulated[0, 1:]+Cost_accumulated[1, 1:])
175
176 lis=np.linspace(3,24,8,dtype=int)
177
178
179 for i,j in zip(lis,Productivity[lis]):
180     ax1.text(i-1,j+1000,int(j),color="b")
181
182 for i,j in zip(lis,Everyweek_get[lis]):
183     ax1.text(i-1,j+1000,int(j),color="darkorange")
184
185 ax2.legend(loc=4, ncol=1)
186 plt.xlabel('安排周数')
187 ax1.legend(loc=3, ncol=1)
188 ax2.set_ylim(0, 60000)
189 #plt.grid(linestyle=":", color="k")
190 plt.title('每周成本, 产能随周数变化')
191 plt.show()

```
