

基于 PCA 和多目标优化的原材料订购与运输方案

摘要

某板材生产企业的原材料共分为 A、B、C 三种类型，共由 402 家供应商提供，订购的原材料由 8 家转运商进行转运。原材料的购买、运输和存储会产生成本，企业的订购量不一定能完全得到满足，并且转运过程中还会产生一定的损耗。本文以保障企业的正常生产、提高企业的经济效益为目标，对供应商的实力进行综合评估、制定合理的订购和转运方案，并对方案进行了合理评价。

针对问题一供应商保障企业生产重要性的量化评估模型的建立，首先提取年平均供货量、供货稳定性系数、订单满足率、交易频率和订单增长系数五项观测指标，构建了供应商综合评价体系。通过 PCA 方法，对评价指标进行重新整合并消除相关性，给出了供应商综合评价得分的计算模型，并根据具体得分筛选出包括 S140, S229, S361 等在内的 50 家最重要的供应商。

针对问题二保证企业生产的供应商数量下限的确定和最优订购、转运方案的制定，首先根据供应商、转运商的历史数据提取统计特征，作为未来 24 周的预测依据；以供应商数量为优化目标，以每周供应量能够满足当周产能为约束条件，建立单目标优化模型求解出最少 39 家供应商能够保障企业生产，并对结果进行了敏感性分析。针对订购方案的确定，以购买成本和存储成本最低为目标，建立了受到供货上限和库存下限约束的优化模型；针对转运方案的确定，以转运损耗最小为目标，建立了受到转运上限和集中转运约束的优化模型。通过贪心策略设计算法对两模型进行综合求解，求出在自拟参数的条件下最低成本为 181.4 万元，最小总损耗量为 6783.2m^3 ，对应的订购和转运方案分别保存在“附件 A”和“附件 B”中，并分析了方案的实施效果。

针对问题三在全体 402 家供应商中制定最经济订购方案和最少损失转运方案的问题，同样建立了以购买成本和存储成本最低为优化目标的订购优化模型和以转运损耗最小为优化目标的转运优化模型。在问题二的基础上进行了算法的升级和优化使之适应更多供应商的求解条件，求出在自拟参数的条件下最低成本为 179.8 万元，最小总损耗量为 6433.2m^3 ，与问题二的结果进行了对比分析，并将对应的订购和转运方案分别保存在“附件 A”和“附件 B”中，并分析了方案的实施效果。

针对问题四在当前供应商和转运商条件下企业每周产能提高上限的确定，以订货量矩阵和转运量矩阵为决策变量，以企业实际产能为优化目标，建立受到供货上限、库存下限、转运上限和集中转运约束的单目标优化模型。通过贪心策略设计算法求解出企业产能最高可以提高 26.14%，最大周产能为 35571.95m^3 ，并将对应的订购和转运方案分别保存在“附件 A”和“附件 B”中。

最后对模型的优、缺点进行了评价，并指出了模型可能的优化方向。

关键词：原材料订购与运输，多目标优化模型，主成分分析法，贪心算法

一、问题重述

某建筑和装饰板材的生产企业共需求 A、B、C 三种类型的原材料。原材料的采购流程为根据产能要求确定需要订购的原材料供应商(称为“供应商”)和相应每周的原材料订购数量(称为“订货量”),确定第三方物流公司(称为“转运商”)并委托其将供应商每周的原材料供货数量(称为“供货量”)转运到企业仓库。在实际转运过程中,原材料会有一定的损耗(损耗量占供货量的百分比称为“损耗率”),转运商实际运送到企业仓库的原材料数量称为“接收量”。

已知该企业按每年 48 周安排生产,由于原材料的特殊性需要提前制定 24 周的原材料订购和转运计划。

我们掌握的题目要求和数据条件有:

- ◆ 该企业每周的产能为 2.82 万立方米,每立方米产品需消耗 A 类原材料 0.6 立方米,或 B 类原材料 0.66 立方米,或 C 类原材料 0.72 立方米;
- ◆ 在成本方面,A 类和 B 类原材料的采购单价分别比 C 类原材料高 20% 和 10%,三类原材料运输和储存的单位费用相同;
- ◆ 该企业为保证正常生产的需要,需要保持不少于满足两周生产需求的原材料库存量,为此该企业对供应商提供的原材料总是全部收购;
- ◆ 每家转运商的运输能力为 6000 立方米/周,一家供应商每周供应的原材料尽量由一家转运商运输;
- ◆ 已知近 5 年 402 家供应商的订货量和供货量数据和 8 家转运商的运输损耗率数据。

我们需要解决的问题有:

1. 对 402 家供应商的供货特征进行量化分析,建立反映保障企业生产重要性的数学模型,在此基础上确定 50 家最重要的供应商;
2. 计算该企业应至少选择多少家供应商供应原材料才可能满足生产的需求,为企业制定未来 24 周每周最经济的原材料订购方案,据此确定损耗最少的转运方案,并对分析方案的实施效果;
3. 为了压缩生产成本,该企业现计划尽量多地采购 A 类和尽量少地采购 C 类原材料,同时希望转运商的转运损耗率尽量少。据此制定新的订购方案及转运方案,并分析方案的实施效果;
4. 根据现有原材料的供应商和转运商的实际情况,确定该企业提高每周产能的上限,并给出未来 24 周的订购和转运方案。

二、问题分析

2.1 对问题一的分析

问题一要求建立数学模型,量化评估 402 家供应商对保障企业生产的重要性,并据此选定 50 家最重要的供应商。

对此,应该首先选取评价供应商对保障企业生产重要性的若干项指标,据此建立供应商的量化评估体系。我们认为一家重要的供应商应该具备较高的供货能力和稳定性,对于订单的需求能最大程度的满足,能与企业保持较强的交易活跃度,在供货量上还应具备一定的增长能力,据此我们选择年平均供货水平、供货的稳定性指标、对订单的满足率、交易的频率和供应商供货能力的增长潜力作为我们对供应商的评价指标。

在利用评价指标对供应商进行综合评价的问题上,我们认为主成分分析法能够较好

地消除评价指标之间的相关影响,更好地反映供应商的综合能力。据此我们基于 PCA 方法可以建立起一套反映保障企业生产重要性的综合评价体系。通过计算每家供应商的量化评估值,可以筛选出最重要的 50 家供应商。

对问题一的建模求解流程如图 1 所示。

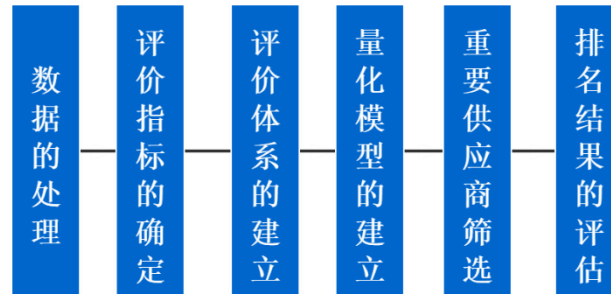


图 1 问题一建模求解思路

2.2 对问题二的分析

问题二要求确定最少由多少家供应商提供原材料可以满足企业的生产需求,针对这些供应商制定最经济、损耗最少的订购和转运方案,并分析方案的实施效果。

我们认为首先需要根据历史五年的供货商、转运商数据,对供货商、转运商特征进行提取,通过供货量矩阵、满足率矩阵和损失率矩阵预测未来 24 周的供货商、转运商特征,作为建模的条件;在求解最少需要多少家供应商时,我们认为对于所有实际供货的供应商都应以供货能力上限进行订货,运用贪心算法可以求解供应商数量的最小值;在制定订货方案时,出于风险防控的角度考虑我们认为应该按照重要性排序进行依次订货,以成本最少为目标,以供货上限和库存下限为约束建立优化模型;在制定转运方案时,我们认为应该以转运损耗最小为目标,以转运能力上限和集中转运原则为约束建立优化模型。最后设计算法对订货、转运模型进行统一求解。

对问题二的建模求解流程如图 2 所示。

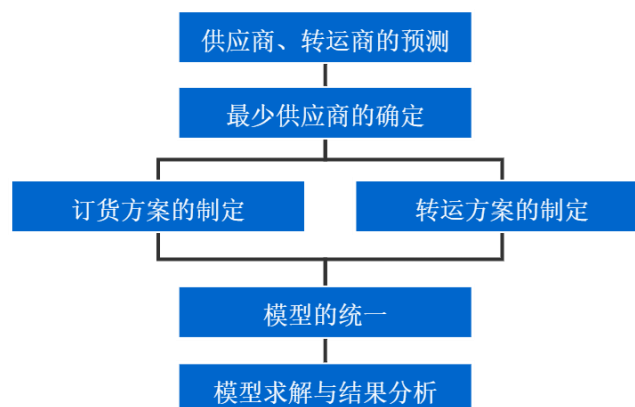


图 2 问题二建模求解思路

2.3 对问题三的分析

问题三要求在通过多采购 A 类少采购 C 类原材料以压缩成本的条件下，制定新的订购方案和使得转运商转运损耗率最小的转运方案，并评价方案的实施效果。

与问题二中订购和转运模型相类似，要求以经济性为主要指标建立模型计算订购量矩阵、订单满足率矩阵和转运量矩阵，进而制定最优的订购和转运方案。与问题二不同之处在于，可供选择的供应商由问题二中的不多于 50 家变为了 402 家，另外对于企业采购偏好也设置了要求。针对数据复杂度增加的问题，我们考虑设计效率更高的启发式算法，对模型结果进行求解，并通过施加扰动，模拟实际情况下模型的求解结果，并完成敏感性分析。

对问题三的建模求解流程如图 3 所示。

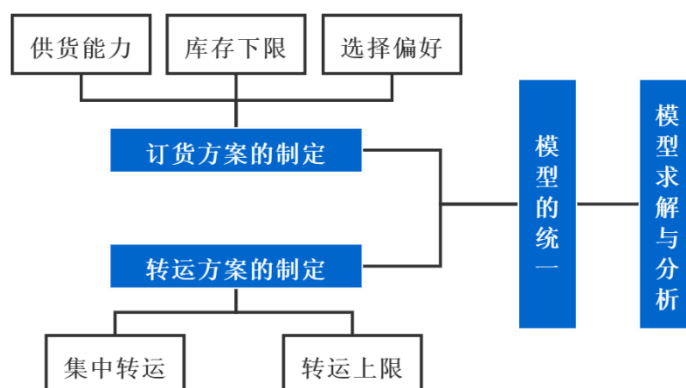


图 3 问题三建模求解思路

2.4 对问题四的分析

问题四要求根据现有原材料的供应商和转运商的实际情况，确定该企业提高每周产能的上限，并给出未来 24 周的订购和转运方案。

在前三个问题中，企业产能保持一个定值不变，是建立模型的一个重要约束条件。在本问中则无需考虑问题二、问题三所在意的经济性，只需在原材料订购基础上求产能能达到的最大稳定值。因此应以实际稳定产能为优化目标，以原材料订购和转运能力为约束条件建立优化模型。在模型求解时，可以采用贪心的策略，在原材料订购、转运时是各指标尽可能大，从而快速求解得到最大的稳定产能。之后还可以进行稳定性分析，在产能继续增大时观测转运量、库存量等的异常情况，以说明最大产能的合理性。

对问题四的建模求解流程如图 4 所示。

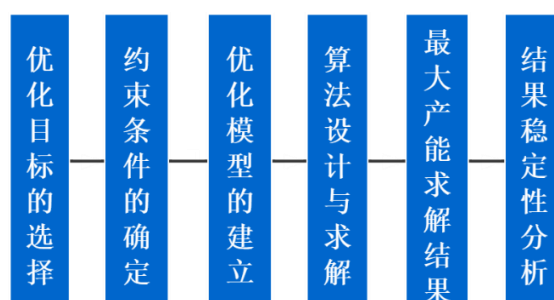


图 4 问题四建模求解思路

三、模型假设与变量说明

3.1 模型假设

- 1.各供应商提供的 A、B、C 类原材料质量相同，企业在原材料质量上没有偏好；
- 2.历史供、订货数据真实有效，能够反映供应商、转运商真实的运营特征；
- 3.在未来 24 周的运营之初，企业已有能够满足两周产能的库存量；
- 4.每周的原材料订购在每周开始前到达，能够用于当周生产，不计送货时间。

3.2 变量说明

本文主要用到的变量名称和解释如表 1 所示，未列出或重复的符号以出现处的定义和解释为准。

表 1 主要变量说明表

变量名称	变量解释
S	年平均供货量
λ	供货稳定性系数
ρ	交易频率
k	订单增长系数
η	订单满足率
Z	PCA 综合评价得分
s_{ij}	第 i 家供应商第 j 周供货量
η_{ij}	第 i 家供应商第 j 周订单满足率
ϕ_{ij}	第 i 家转运商第 j 周损耗率
t_{ij}	第 i 家转运商第 j 周转运量
l_{ij}	第 i 家转运商第 j 周损耗量
w_j	第 j 周库存量能够满足的产能
δ	转运控制因子
e_i	价格转换因子
θ_i	生产因子
Q	企业周产能
$[d_{1j}, d_{2j}, d_{3j}]^T$	第 j 周三种原材料库存量向量

四、模型的准备

4.1 对历史数据的分析

4.1.1 企业原材料库存量分析

我们经过分析认为，企业的原材料库存量所能满足的产能能够从整体上体现企业在原材料管理方面的经营情况。一方面，处于风险管控的角度，原材料库存不应过少，所有库存应至少保障两周企业的正常生产；另一方面，处于成本控制的角度，原材料库存不应过多，否则会产生大量的存储成本，降低经济效益。好的原材料采购方案应该能够保证企业原材料库存量在两倍产能附近浮动，以取得最优的综合效益。

我们根据历史五年数据，绘制出了该企业在过去五年库存量随时间的变化曲线(我们假定企业在五年之初恰好保有能维持两周产能的库存量)，如图 5 所示。

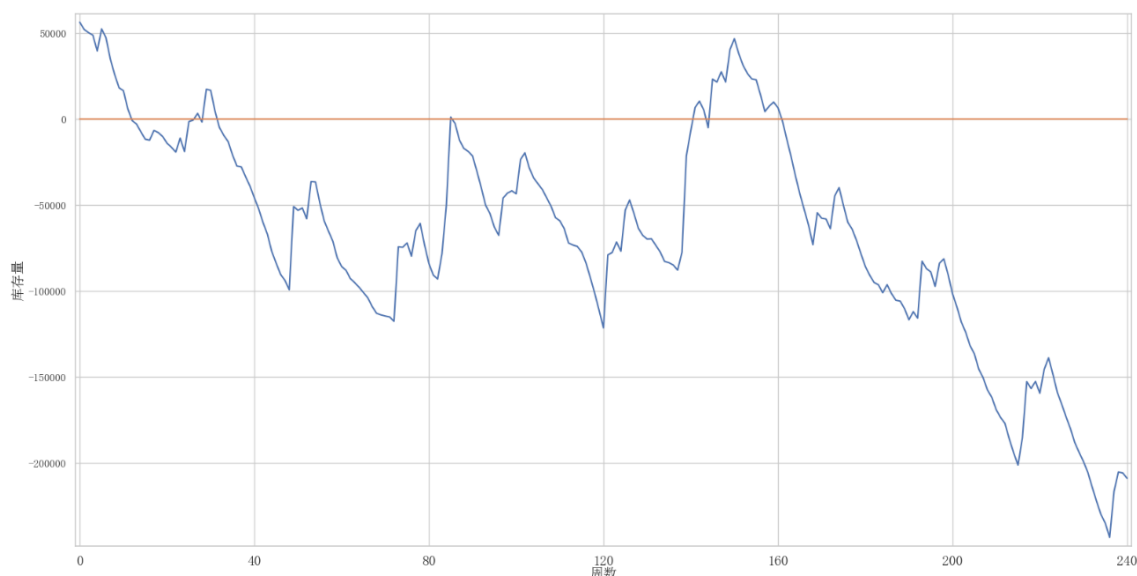


图 5 过去五年库存量随时间变化图

从图中可以看出，在开始运营后库存量整体趋势为逐渐下降，并迅速出现负值，表示企业不能保持正常的生产活动。因此对于 28200m^3 的周产能，过去的订货量远远不能达到保证生产的目的。企业原有的订货策略存在一定问题，需要进行调整。

4.1.2 企业原材料订购量分析

从图 5 中我们还可以发现，库存量存在陡升陡降的现象，无法保证稳定的变动，这说明企业在订货上存在的问题。我们根据题目提供数据绘制出过去五年订货量随时间的变化曲线。如图 6 所示。

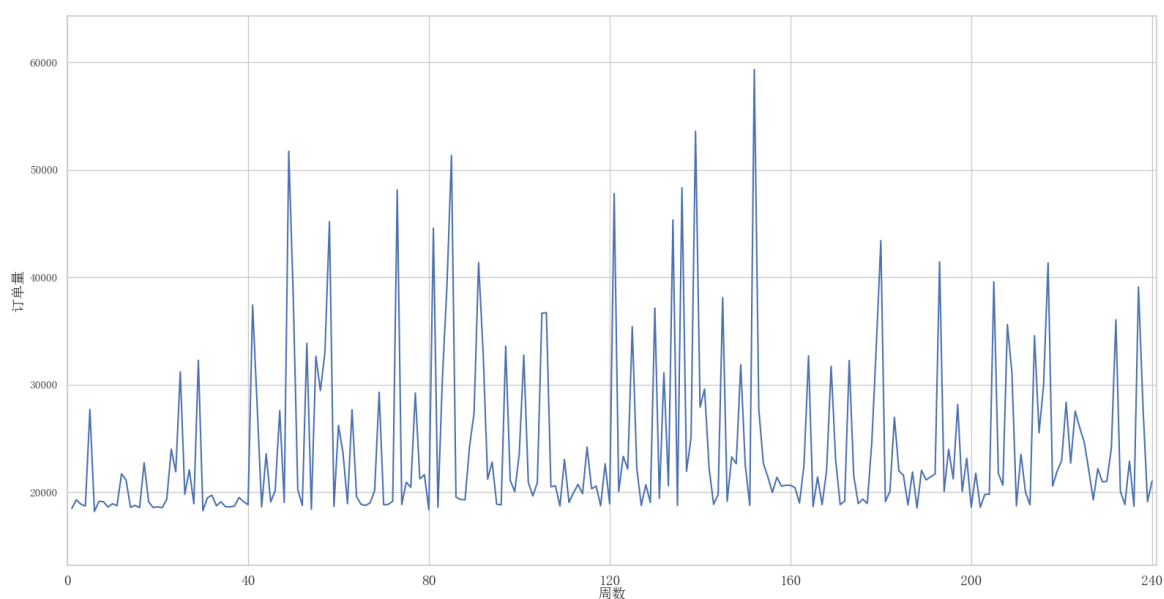


图 6 过去五年订货量随时间变化图

一个好的原材料订购和管理方案应该保证与供应商的长期稳定合作，这体现在订货量在一定范围内保持稳定，供应商对订单的响应较好，能够足量提供所需的原材料。图6所示的订货方案更像是“亡羊补牢”式的方案，在库存量不足时大量订货，如此反复不利于保障企业生产的稳定性，需要进行调整。

4.2 对历史数据的总结

通过对历史数据的分析我们认为：企业对经济性的考虑应建立在稳定运营的基础上，即需要选择更可靠的原材料供应商，每周订购足量的原材料保证生产的正常进行；企业在进行订购原材料时，应结合供应商的供应能力进行平稳订购，在短时间内的大量订购不利于维持企业运营的正常和有序性；历史数据中有很多周企业对一些供应商没有订货，但并不意味着此时企业没有能力供货，因此应当更加深入地评估企业的供货能力。

五、模型的建立与求解

5.1 问题一：供应商量化评价模型的建立和重要供应商的筛选

5.1.1 供应商量化评价指标的选取

原材料供应商作为企业生产经营活动的基础，其稳定性、活跃性、经济性对企业的经营活动和盈利情况具有重要作用。本节着重从管理学的角度，结合题目提供的数据，探讨供应商量化评估指标的选择问题。我们认为，一家良好的供应商应该具备较高的供货能力和稳定性、对于订单的需求能最大程度的满足、能与企业保持较强的交易活跃度、在供货量上还应具备一定的增长能力。^[5,6]据此，我们选取以下五个方面的参数作为对供应商重要性的量化评估指标。

(1)年平均供货量

考虑到企业每周对原材料需求数量较大，需要同时由多家供应商提供，因此供货商的供货能力是对其评价的重要指标。**年平均供货量 S** 是反映供货商供货能力的直观指标，以年为单位统计也能消除季节性影响，因此我们选择其作为我们的第一个评估指标。

记第 i 家供货商第 j 周的供货量为 $s_{(i,j)}$ ，则第 i 家供货商过去五年的年平均供货量为

$$S_i = \frac{1}{5} \sum_{j=1}^{5 \times 48} s_{i,j} \quad (1)$$

(2)供货稳定性系数

除了供货商每周能够提供的原材料数量，各周供货量的稳定性也是反映其供货能力的重要指标。由于供货商供货量受到企业提出的订购量相关，我们利用供货量与订购量的差值消除这一影响，并计算各周供货量与订购量差值的方差作为模型的一个评估指标，称为**供货稳定性系数 λ** 。

记企业第 j 周向第 i 家供货商的订货量为 $r_{(i,j)}$ ，第 i 家供货商第 j 周的供货量为 $s_{(i,j)}$ ，则第 i 家供货商的供货稳定性系数为

$$\lambda_i = \frac{1}{47} \sum_{j=1}^{48} \left[(s_{(i,j)} - r_{(i,j)}) - \overline{(s_{(i,j)} - r_{(i,j)})} \right]^2 \quad (2)$$

(3) 订单满足率

企业在对供应商评价时，能否根据企业订货量提供足量原材料是又一重要因素。尤其在大额订单中供应商的违约或者不能足量供货将可能导致企业生产原材料的缺口。定义**订单满足率** η 作为一个评价指标，表示供应商对企业订货量的完成情况，其计算方式为某一周供货商总供货量与企业总订货量的比值。

记企业第 j 周向第 i 家供货商的订货量为 $r_{(i,j)}$ ，第 i 家供货商第 j 周的供货量为 $s_{(i,j)}$ ，则第 i 家供货商第 j 周的订单满足率为

$$\eta_{ij} = \frac{s_{(i,j)}}{r_{(i,j)}} \quad (3)$$

若 $r_{(i,j)} = 0$ ，即该周交易没有发生，则不计算该周的订单满足率。

(4) 交易频率

一家供货商与企业的交易频次反映出供货商的商业活力，企业反复选择其进行合作也反映出企业对其的信任。定义**交易频率** ρ 作为一个评价指标，表示合作的活跃性，其计算方式为一段时间内企业与供应商发生成功交易的周数占总周数的比值。

记一年内第 i 家供货商共有 n_i 周成功向企业供货，则其与企业的交易频率为

$$\rho_i = \frac{1}{48} n_i \quad (4)$$

(5) 订单增长系数

供应商在过去五年也在不断发展，供货水平发生了一定的变化。我们引入**订单增长系数** k 刻画这一变化，并作为模型的一个评价指标，其计算方式为利用最小二乘法对供货商过去五年的年供货量进行线性回归，以回归函数的斜率作为订单的增长系数。^[1]

记第 i 家供应商第 t 年的年供货量为 $S_i^{(t)}$ ，则其订单增长系数为

$$k = \frac{\overline{tS_i^{(t)}} - \bar{t} \cdot \overline{S_i^{(t)}}}{\overline{t^2} - (\bar{t})^2} \quad (5)$$

5.1.2 基于 PCA 的供应商量化评价方案

5.1.2.1 供应商评价综合体系的确定

经过对评价供应商重要程度客观原因的分析，结合题目给出的数据，我们确定了一家良好的供应商应该具备较高的供货能力和稳定性、对于订单的需求能最大程度的满足、能与企业保持较强的交易活跃度、在供货量上还应具备一定的增长能力。据此建立了包含年平均供货量 S 、供货稳定性系数 λ 、订单满足率 η 、交易频率 ρ 和订单增长系数 k 五项观测指标了供应商综合评价体系，如图 7 所示。

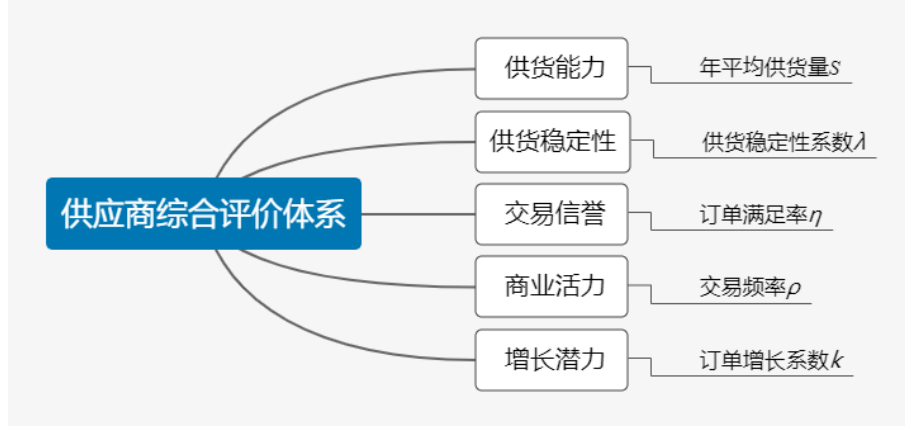


图 7 供应商综合评价体系

5.1.2.2 对数据的处理

通过对题目给出“近 5 年 402 家供应商的相关数据”的分析，发现题目数据完整有效，无需额外进行数据清理，因此根据 4.1.1 构造和计算出每家供应商的观测指标，其中第 i 家观测指标的数据结构为

$$(ID_i, S_i, \lambda_i, \eta_i, \rho_i, k_i) \quad (6)$$

对数据的处理包括数据规范化和指标正向化。

◆ 指标正向化

根据年平均供货量 S 、订单满足率 η 、交易频率 ρ 和订单增长系数 k 的构造方法，供应商的重要性与这些指标正相关，而供货稳定性系数 λ 通过方差的方式构造，供应商的重要性与该指标负相关，需要进行正向化处理。

我们考虑映射

$$\lambda' = \begin{cases} 2 \frac{\lambda - m}{M - m}, & m \leq \lambda \leq \frac{1}{2}(M + m) \\ 2 \frac{M - \lambda}{M - m}, & \frac{1}{2}(M + m) \leq \lambda \leq M \end{cases} \quad (7)$$

其中 M 为 λ 可能取得的最大值， m 为 λ 可能取得的最小值。基于此我们可以将所有指标对模型贡献的方向性统一。

◆ 数据规范化

依题意共 402 个待评价对象，每个对象有五维观测指标，构造原始数据矩阵

$$A = \begin{bmatrix} S_1 & \lambda_1' & \eta_1 & \rho_1 & k_1 \\ S_2 & \lambda_2' & \eta_2 & \rho_2 & k_2 \\ \dots & \dots & \dots & \dots & \dots \\ S_n & \lambda_n' & \eta_n & \rho_n & k_n \end{bmatrix} \quad (8)$$

该矩阵共 5 列 402 行 ($n = 402$)，记第 i 行 j 列的元素为 a_{ij} ，将属性进行标准化处理，

各指标值转化为标准化指标 a_{ij} ，有

$$a_{ij} = \frac{a_{ij} - \mu_j}{s_j} \quad (9)$$

式中 $\mu_j = \frac{1}{n} \sum_{i=1}^n a_{ij}$ ， $s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (a_{ij} - \mu_j)^2}$ ，由此得到规范化处理后的标准化矩阵 A 为

$$\tilde{A} = \begin{bmatrix} \tilde{S}_1 & \tilde{\lambda}_1' & \tilde{\eta}_1 & \tilde{\rho}_1 & \tilde{k}_1 \\ \tilde{S}_2 & \tilde{\lambda}_2' & \tilde{\eta}_2 & \tilde{\rho}_2 & \tilde{k}_2 \\ \dots & \dots & \dots & \dots & \dots \\ \tilde{S}_n & \tilde{\lambda}_n' & \tilde{\eta}_n & \tilde{\rho}_n & \tilde{k}_n \end{bmatrix} \quad (10)$$

对应地，称

$$x_{ij} = \frac{x_{ij} - \mu_j}{s_j} \quad (11)$$

为标准化指标变量，基于此我们可以基于 PCA 建立供应商量化评价方案。[2]

5.1.2.3 基于 PCA 的供应商量化评价方案的建立

本节主要解决基于已经建立的供应商综合评价体系，如何计算一个量化指标，来对所有 402 家供应商进行重要性排序的问题。考虑到评价体系共涉及五维的观测指标，我们选择主成分分析法(Principal Component Analysis, PCA)进行数据降维，同时概括供应商的最重要特征，据此建立供应商量化评价的解决方案。具体过程如下。

(1)相关性系数矩阵 R 的计算

由于供应商评价指标共有五维，因此令供应商评价指标相关性系数矩阵 $R = (r_{ij})_{5 \times 5}$ ，有

$$r_{ij} = \frac{\sum_{k=1}^n a_{ki} \cdot a_{kj}}{n-1}, i, j = 1, 2, \dots, 5 \quad (12)$$

式中 $r_{ii} = 1$ ， $r_{ij} = r_{ji}$ ， r_{ij} 为第 i 个指标与第 j 个指标的相关系数。

(2)特征值与特征向量的计算

计算供应商评价指标相关性系数矩阵 R 的特征值 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_5$ 及对应的标准化特征向量 u_1, u_2, \dots, u_5 ，其中 $u_j = [u_{1j}, u_{2j}, \dots, u_{5j}]^T$ ，由特征向量组成 5 个新的供应商评价指标变量

$$\begin{cases} y_1 = u_{11}x_1 + u_{21}x_2 + \dots + u_{51}x_5, \\ y_2 = u_{12}x_1 + u_{22}x_2 + \dots + u_{52}x_5, \\ \dots \\ y_5 = u_{15}x_1 + u_{25}x_2 + \dots + u_{55}x_5, \end{cases} \quad (13)$$

式中 y_1 为第一主成分, y_2 为第二主成分,..., y_5 为第五主成分, 从五个维度对供应商重要性进行评价。

(3)综合评价得分的计算

引入特征值 λ_j 的信息贡献率和累计贡献率。称

$$b_j = \frac{\lambda_j}{\sum_{k=1}^5 \lambda_k} \quad (14)$$

为供应商评价指标 y_j 的信息贡献率, 并且称

$$\alpha_p = \frac{\sum_{k=1}^p \lambda_k}{\sum_{k=1}^5 \lambda_k} \quad (15)$$

为前 p 个评价指标的累计贡献率。当 α 接近于 1 时, 选择前 p 个供应商评价指标变量 y_1, y_2, \dots, y_p 作为 p 个主成分, 则有最终的综合评价得分模型

$$Z = \sum_{j=1}^p b_j y_j \quad (16)$$

供应商的综合评价得分 Z 越高, 其综合评价越好, 说明其对企业的重要性越强。

至此, 通过 PCA 方法可以计算得到各供货商的综合评价得分 Z , 可以以此为据对供应商的重要性进行排序, 完成建立供应商量化评估方案的任务。^[2]

5.1.3 五十家最重要供应商的筛选

5.1.3.1 供应商量化评估模型的求解

利用 MATLAB 软件求得相关系数矩阵的五个特征根及其贡献率如表 2 所示。

表 2 主成分分析结果

序号	特征根	贡献率(%)	累计贡献率(%)
1	2.0545	41.0899	41.0899
2	1.3713	27.4261	68.5160
3	0.9902	19.8048	88.3208
4	0.3028	6.0558	94.3766
5	0.0812	5.6233	100.0000

可以看出，前四个特征根的累计贡献率达到 94%以上，主成分分析效果较好。因此我们选取前四个主成分进行综合评价，分别为

$$\begin{aligned} y_1 &= 0.6356x_1 - 0.4353x_2 + 0.1758x_3 + 0.6128x_4 + 0.0118x_5, \\ y_2 &= 0.0047x_1 + 0.5851x_2 + 0.7670x_3 + 0.1942x_4 - 0.1781x_5, \\ y_3 &= 0.0209x_1 + 0.0953x_2 + 0.1600x_3 - 0.0187x_4 + 0.9821x_5, \\ y_4 &= -0.3177x_1 + 0.3824x_2 - 0.4615x_3 + 0.7324x_4 + 0.0588x_5 \end{aligned} \quad (17)$$

分别以四个主成分的贡献率为权重，确定本问题的主成分综合评价模型为^[4,7]

$$Z = 0.4108y_1 + 0.2743y_2 + 0.1980y_3 + 0.0606y_4 \quad (18)$$

据此计算得到各供应商的综合评价得分，保存在附件 6 中。

5.1.3.2 五十家最重要供应商的筛选结果

对 4.1.3.1 计算结果进行排名，可以筛选出 50 家对企业经营最重要的原材料供应商。结果如表 3 所示，完整结果见附件 6。

表 3 五十家最重要供应商筛选结果

排名	1	2	3	4	5	6	7	8	9	10
供应商编号	S140	S229	S361	S108	S340	S282	S131	S374	S352	S275
综合评分	3.507	2.948	2.936	2.830	2.591	2.428	2.363	2.351	2.329	2.258
(a)										
排名	11	12	13	14	15	16	17	18	19	20
供应商编号	S329	S308	S268	S306	S139	S356	S194	S330	S247	S143
综合评分	2.163	2.142	2.059	1.996	1.921	1.860	1.855	1.707	1.642	1.624
(b)										
排名	21	22	23	24	25	26	27	28	29	30
供应商编号	S031	S365	S284	S266	S040	S364	S294	S218	S080	S367
综合评分	1.575	1.558	1.551	1.454	1.453	1.444	1.436	1.431	1.426	1.385
(c)										
排名	31	32	33	34	35	36	37	38	39	40
供应商编号	S123	S346	S007	S244	S055	S151	S307	S126	S037	S003
综合评分	1.379	1.367	1.358	1.356	1.247	1.235	0.925	0.923	0.916	0.858
(d)										
排名	41	42	43	44	45	46	47	48	49	50
供应商编号	S005	S189	S273	S114	S292	S338	S078	S291	S154	S221
综合评分	0.791	0.755	0.614	0.486	0.429	0.429	0.426	0.422	0.417	0.404
(e)										

5.1.3.3 筛选结果的合理化分析

图 8 展示了五十家最重要供应商中前五名供应商的供货量。从图中展示的供应商来

看，对供应商的评价考虑到了诸多指标，不同类型供应商在保证供货能力和满足率的条件下都有机会被标记为重要，具有一定的合理性。比如企业 S229、S361，是稳定性供应商，虽然供货数量不大，但是每周都能保证基本恒定的供货量，且保持着较高的订单满足率；再比如企业 S140、S108，是大额供应商，在大多数周中供应量少，但是在企业有较大订单量时也能以较高的满足率进行大额供货。

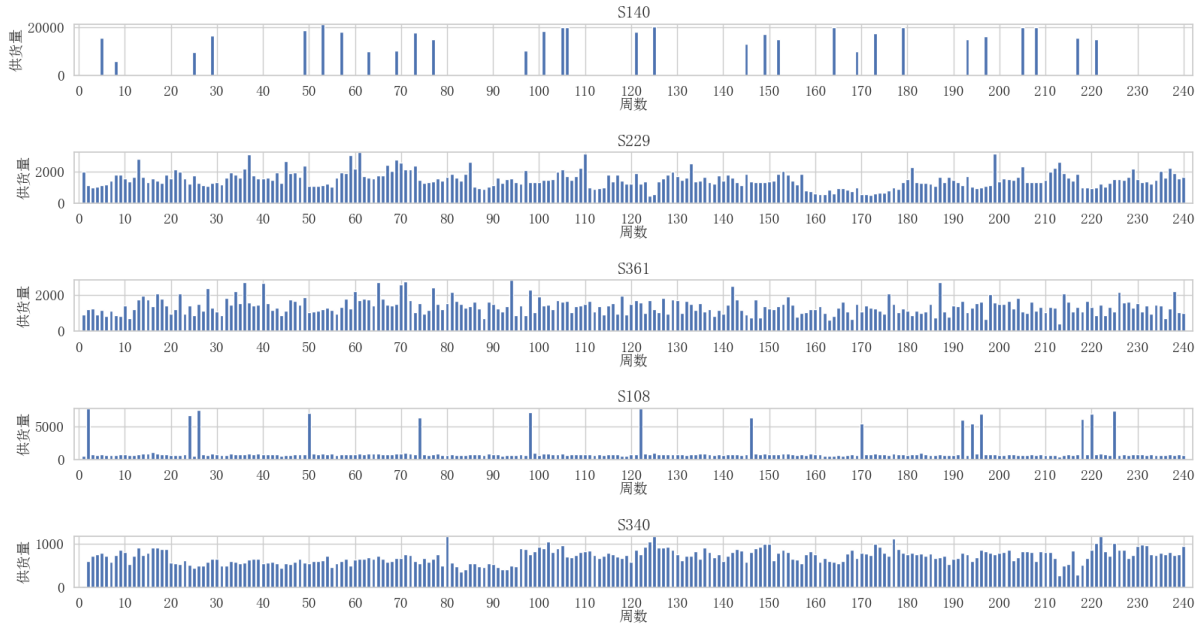


图 8 部分重要供应商的供货量展示图

5.2 问题二：最少供应商的确定与订购、转运模型的建立求解

5.2.1 供应商和转运商的预测建模

本节主要解决的问题是根据各供应商、转运商过去 5 年的供货量、转运损耗率等数据，结合问题一的评估结果对未来 24 周他们可能的供货能力、转运损耗等进行预测，从而为企业的订购方案和转运商的选取提供条件。

对于供货商，我们认为其重要指标为供应量上限和订单满足率。我们取历史供货量最大值作为供应商周供应能力的上限，取历史满足率的期望作为对订单满足率的预测。为了提高预测的真实性，使之更符合真实的供应情况，我们依分布为订单满足率增加波动系数 ε ，据此构造供应量上限矩阵 $S_{n \times m}$ 和订单满足率矩阵 $H_{n \times m}$ 。

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{bmatrix}, H = \begin{bmatrix} \eta_{11} & \cdots & \eta_{1m} \\ \vdots & \ddots & \vdots \\ \eta_{n1} & \cdots & \eta_{nm} \end{bmatrix} \quad (19)$$

对于转运商，我们计算了 8 家转运商在过去五年损耗率的期望值并从小到大进行排名，结果如表 4 所示。

表 4 转运商历史损耗率排名

转运商编号	损耗率(%)	排名
T3	0.0907	1
T6	0.4894	2
T4	0.6675	3
T8	0.8545	4
T2	0.9214	5
T5	0.9994	6
T1	1.9048	7
T7	2.0788	8

用历史损耗率的期望作为对其转运特征进行刻画，同样地，依分布对其增加波动系数，使之更符合实际情况，据此构造转运损耗率矩阵 $\Phi_{p \times m}$

$$\Phi = \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1m} \\ \vdots & \ddots & \vdots \\ \varphi_{p1} & \cdots & \varphi_{pm} \end{bmatrix} \quad (20)$$

通过供应量上限矩阵 $S_{n \times m}$ 和订单满足率矩阵 $H_{n \times m}$ ，可以预测未来 24 周供应商行为；通过转运损耗率矩阵 $\Phi_{p \times m}$ ，可以预测未来 24 周 8 家转运商行为。

5.2.2 最少供应商数量的确定

根据第一问的重要性评估，我们认为排名更高的供货商供货能力、供货稳定性更高，选择排名更高的供货商能够更好地降低风险，保障企业的正常生产。结合题目的要求，我们规定企业在选择供应商时，优先选择在 4.1 的量化评估中排名更高的供货商。

5.2.2.1 模型的建立

设企业将选择 n 家供应商采购原材料，则确定最少的供应商数量只需以 n 为优化目标求解优化模型

$$\min n \quad (21)$$

模型的约束条件为保障企业的正常生产，即每周的原材料采购数量应能满足企业一周内的生产所需。根据 4.2.1 可知各供应商的供货量矩阵

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{bmatrix} \quad (22)$$

引入生产因子 θ_i ，根据第 i 家供应商供应原材料的类型将供货量向量 $s_j (j = 1, 2, \dots, m)$ 修正为能够满足企业产能的数量 $\theta_i s_{ij}$ ，其值为

$$\theta_i = \begin{cases} \theta_1 = 1/0.6 = 1.667, A \text{类原材料} \\ \theta_2 = 1/0.66 = 0.515, B \text{类原材料} \\ \theta_3 = 1/0.72 = 1.389, C \text{类原材料} \end{cases} \quad (23)$$

记企业的规定产能 $Q_0 = 28200\text{m}^3$ ，为保证在任意转运条件下都能保证企业实际接受量满足生产需求，因此令转运损耗率取历史最大值 $\eta_0 = 0.05$ 。则模型的约束条件可表示

为

$$\sum_{i=1}^n (1-\eta_0) \cdot \theta_i \cdot s_{ij} \geq Q_0, \forall j=1,2,\dots,m \quad (24)$$

综上，确定的求解最少供应商数量的优化模型为

$$\begin{aligned} & \min n \\ & s.t. \sum_{i=1}^n (1-\eta_0) \cdot \theta_i \cdot s_{ij} \geq Q_0, \forall j=1,2,\dots,m \end{aligned} \quad (25)$$

5.2.2.2 求解算法

由于我们约定了选择供应商的顺序应按照 4.1 中的排名进行，为使供应商总量最小只需在每一家供应商处都购进其供应量的最大值，因此求解算法设计如下。

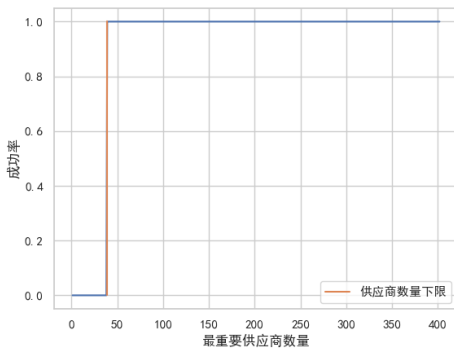
- Step1:** 将周数 w 初始化为 1、计数变量 $count$ 初始化为 0，最小供应商数量 n 初始化为 402，库存量初始化为满足两周产量的原材料；
- Step2:** 令 $count = count+1$ ，按照供应商重要性排序将第一家供应商最大供货量可转化为的库存量与当前库存量相加；
- Step3:** 判断当前库存量是否满足两周产能，若能满足则跳转到 **Step5**，若无法满足则跳转到 **Step4**；
- Step4:** 将下一家供应商最大供货量可转化为的库存量与当前库存量相加， $count = count+1$ ，跳转到 **Step3**；
- Step5:** 令 $n = \min\{count, n\}$ ， $count = 0$ ， $w = w+1$ ，判断 w 是否大于 24，若是则跳转 **Step6**，否则跳转 **Step2**；
- Step6:** 输出最小供应商数量 n 。

5.2.2.3 求解结果

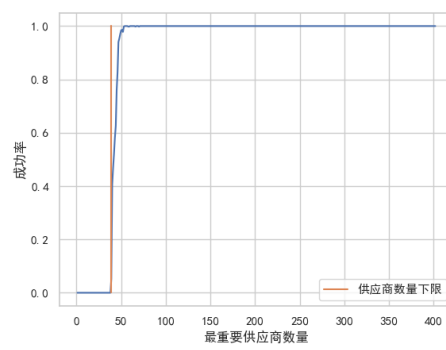
利用 5.2.2.2 中算法对最小供应商数量模型进行求解，可得该企业至少选择 39 家供应商供应原材料才可能满足生产的需求。

5.2.2.4 求解结果分析

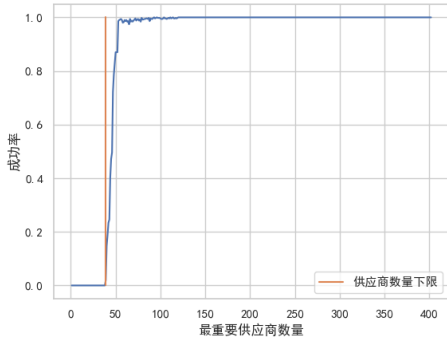
我们通过计算机模拟的方式对结果进行敏感性分析。我们在不同波动程度下，分别对选择 1-402 家供应商提供原材料进行 500 次模拟，计算在这 500 次模拟中有多少次能够保证企业的正常生产。图 9 展示了计算机模拟的图形结果。



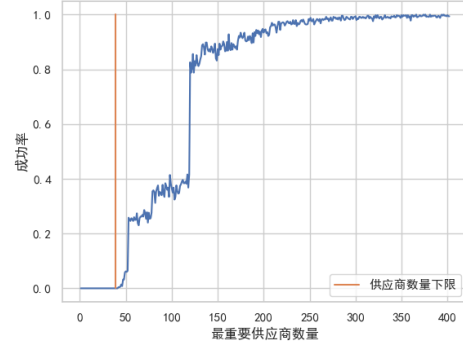
(a) 波动程度 0.05



(b) 波动程度 0.2



(c) 波动程度 0.25



(d) 波动程度 0.5

图 9 不同波动程度下仿真成功率与供应商数量关系图

从图中我们可以直观地看出，在波动程度较小时，当选择供应商数量达到 39 家时，成功率迅速升高达到 100%；而在波动程度较大时，需要的最小供应商数量也有所增加，之前的 39 已经不能达到要求；当波动程度达到 0.5 时，企业生产的保障能力明显下降。据此，根据模型求解的最小供应商数量 39 家在模拟的真实环境中得到了验证，客观上说明了模型的准确性。

5.2.3 订购、转运模型的建立

5.2.3.1 转运模型的建立

企业在选择转运商时，主要通过分配不同的转运商运送不同的供货商的原材料，使得转运过程出现的损耗最小。因此，我们以转运损耗为优化目标，建立优化模型来刻画企业的转运方案。

(1) 决策变量

记转运商总数 $p = 8$ ，总周数 $m = 24$ ，第 i 家转运商在第 j 周的实际转运量为 t_{ij} ($i = 1, 2, \dots, p, j = 1, 2, \dots, m$)。构造转运量矩阵

$$T = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{p1} & \cdots & t_{pm} \end{bmatrix} \quad (26)$$

则 $T_{p \times m}$ 能够反映企业在每周对待转运原材料的安排情况，不同的安排情况将决定不同的转运损耗，我们将其作为我们的决策变量。

(2) 优化目标

企业选择转运商、安排转运方案的原则是使得转运过程原材料损耗最小，在 4.2.1 中我们已经给出了转运损耗率矩阵 $\Phi_{p \times m}$ ，则损耗量矩阵 $L_{p \times m}$ 表示为

$$L_{p \times m} = T_{p \times m} \circ \Phi_{p \times m} = \begin{bmatrix} l_{11} & \cdots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{p1} & \cdots & l_{pm} \end{bmatrix} \quad (27)$$

其中 l_{ij} 为第 i 家转运商在第 j 周的转运过程的损耗量，则转运模型的优化目标为

$$\min \sum_{i=1}^p \sum_{j=1}^m l_{ij} \quad (28)$$

(3)约束条件

转运模型的第一方面约束是**转运上限约束**，即单个转运商每周转运量不应超过 6000 m³,即

$$t_{ij} \leq 6000\delta, \forall i = 1, 2, \dots, p, j = 1, 2, \dots, m \quad (29)$$

式中 δ 为转运控制因子, $0 < \delta < 1$ 。引入转运控制因子的目的是为每一家转运商留出余量,以防止在实际转运过程中出现由于供货量的突然增加而导致无法满足转运任务的情况。在实际计算过程中,我们取余量为 500,也即取 δ 为 0.9167。

转运模型的第二方面约束是**集中转运约束**,即一家供应商每周提供的原材料应由一家转运商运输,所有供应商共同完成所有转运任务,即

$$\bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j = 1, 2, \dots, m \quad (30)$$

其中 U_j 为第 j 周所有需要转运的供应商的集合, Tr_{ij} 为第 i 家转运商第 j 周负责转运的供应商的集合。

综合上述分析,最终转运模型建立为以**转运量矩阵**为决策变量,转运损耗量最少为优化目标,受到转运上限约束和集中转运约束的单目标优化模型,即

$$\begin{aligned} & \min \sum_{i=1}^p \sum_{j=1}^m l_{ij} \\ & s.t. \begin{cases} t_{ij} \leq 6000\delta, \forall i = 1, 2, \dots, p, j = 1, 2, \dots, m \\ \bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j = 1, 2, \dots, m \end{cases} \end{aligned} \quad (31)$$

5.2.3.2 订购模型的建立

由于不考虑原材料质量问题,企业在订购原材料时将合理选择供应商、分配在各供应商的订货量以达到成本最低的目标,同时一方面要保证原材料库存应能维持两周的生产需要,另一方面周、月、年采购量不应超过供应商周、月、年供货能力的上限。据此我们建立了企业的原材料订购模型。

(1)决策变量

记 4.2.2 中最少供应商总数 $n = 39$, 总周数 $m = 24$, 企业在第 j 周向第 i 家供应商的实际订购量为 $r_{ij} (i = 1, 2, \dots, n, j = 1, 2, \dots, m)$ 。构造订货量矩阵

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{bmatrix} \quad (32)$$

则 $R_{n \times m}$ 是由企业确定、将对企业生产成本以及能否保障生产产生直接影响。我们以

此作为决策变量，构建企业原材料订购模型

(2)目标函数

企业在原材料采购上产生的成本分为购买成本和存储成本。

假设 C 类原材料采购单价为 e ，A、B、C 类原材料运输和存储单位费用为 τe ，其中 $0 < \tau < 1$ 为运输存储系数。记第 i 家供应商原材料单价为 e_i ，则根据题意 e_i 仅取决于其提供的原材料种类，即

$$e_i = \begin{cases} 1.2e, A \text{类原材料} \\ 1.1e, B \text{类原材料} \\ e, C \text{类原材料} \end{cases} \quad (33)$$

对于购买成本，根据 4.2.1 的分析企业第 j 周向第 i 家供应商购买的原材料总量为 $r_{ij}\eta_{ij}$ ，则在 24 周内所有的购买成本为

$$\sum_{i=1}^n \sum_{j=1}^m e_i r_{ij} \eta_{ij} \quad (34)$$

对于存储成本，引入企业库存量矩阵 $D_{3 \times m}$ ，

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ d_{31} & d_{32} & \dots & d_{3m} \end{bmatrix} \quad (35)$$

其中 d_{1j} 、 d_{2j} 、 d_{3j} 分别表示第 j 周周末仓库剩余的 A、B、C 类原材料总量。则在 24 周内总的存储成本为

$$\sum_{i=1}^3 \sum_{j=1}^m \tau e d_{ij} \quad (36)$$

综合购买成本和存储成本，原材料订购模型的目标函数为

$$\min \sum_{j=1}^m \left(\sum_{i=1}^n e_i r_{ij} \eta_{ij} + \sum_{i=1}^3 \tau e d_{ij} \right) \quad (37)$$

(3)约束条件

订购模型的第一方面约束是**订购上限约束**。从每一家供应商处实际购得的原材料数量不应多于供应商的最大供应能力，即

$$r_{ij} \eta_{ij} \leq s_{ij}, \forall i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (38)$$

订购模型的第二方面约束是**库存下限约束**。对于第 j 周的库存向量 $\mathbf{d}_j = (d_{1j}, d_{2j}, d_{3j})^T$ ，能够满足的产能为

$$w_j = \theta_1 d_{1j} + \theta_2 d_{2j} + \theta_3 d_{3j} \quad (39)$$

库存量对应的产能在时间序列上具有前后的连续性，满足递推关系式

$$w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{r_{ij}} - Q_0 \quad (40)$$

其中 tr_i 表示转运第 i 家供应商所提供的原材料的转运商编号。根据企业保障生产的要求，库存量对应的产能不应小于企业两周的实际产能，即需满足

$$w_j \geq 2Q_0, \forall j = 1, 2, \dots, m \quad (41)$$

综合上述分析，最终订购模型建立为以订购量矩阵为决策变量，总成本最少为优化目标，受到订购上限约束和库存下现约束的单目标优化模型，即

$$\begin{aligned} & \min \sum_{j=1}^m \left(\sum_{i=1}^n e_i r_{ij} \eta_{ij} + \sum_{i=1}^3 \tau e d_{ij} \right) \\ & s.t. \begin{cases} r_{ij} \eta_{ij} \leq s_{ij}, \forall i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{tr_j} - Q_0 \\ w_j \geq 2Q_0, \forall j = 1, 2, \dots, m \end{cases} \end{aligned} \quad (42)$$

5.2.3.3 模型的统一

虽然订货策略和转运策略的制定分别独立，但是由于实际的原材料订购和转运在时间上相互联系，因此求解时需要同时考虑订购模型和转运模型的目标和限制。因此，我们将两个模型合成为一个双目标优化的订购、转运模型，作为我们最终求解所利用的模型

$$\begin{aligned} & \min \sum_{j=1}^m \left(\sum_{i=1}^n e_i r_{ij} \eta_{ij} + \sum_{i=1}^3 \tau e d_{ij} \right) \\ & \min \sum_{i=1}^p \sum_{j=1}^m l_{ij} \\ & s.t. \begin{cases} r_{ij} \eta_{ij} \leq s_{ij}, \forall i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{tr_j} - Q_0 \\ w_j \geq 2Q_0, \forall j = 1, 2, \dots, m \\ t_{ij} \leq 6000\delta, \forall i = 1, 2, \dots, p, j = 1, 2, \dots, m \\ \bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j = 1, 2, \dots, m \end{cases} \end{aligned} \quad (43)$$

5.2.4 模型的求解与分析

5.2.4.1 模型的求解算法

针对订购转运模型的特点，转运费用我们采用贪心策略设计算法，算法具体流程如下。

Step1: 将周数初始化为 $w = 1$ ，成本初始化为 0，库存量初始化为满足企业两周产量的原材料；

- Step2:** 将所选择的 39 家供应商首先按照 A-B-C 顺序排列，其次按评价指标排序；转运商按照损耗率升序排列；
- Step3:** 选择排名最高的供应商，将其所有供应量分配给排名最高的转运商运输；
- Step4:** 选择下一名的供应商，将其所有供应量分配给排名最高的、有足够余量的转运商运输；若所有转运商均无余量，返回转运错误；若无下一名供应商，返回供应错误；
- Step5:** 计算企业总的接受量，判断其是否满足一周产能，若是则跳转 **Step6**，否则跳转 **Step4**；
- Step6:** 记录该周订货量向量和转运量向量，计算总成本， $w = w + 1$ ，判断 w 是否大于 24，若是则跳转 **Step7**，否则跳转 **Step2**；
- Step7:** 输出订货量矩阵、转运量矩阵和总成本。

5.2.4.2 求解结果

运用 5.2.4.1 中算法对模型进行求解，得到了对于 39 家企业而言的订货量矩阵和转运量矩阵，分别保存在“附件 A”、“附件 B”中。以此作为最经济订购方案和最小损耗转运方案，在自拟的参数条件下企业 24 周总的原材料采购和存储成本为 181.4 万元，转运损耗量为 6783.2 m³。根据订购和转运方案，可以绘制在理想情况下库存量与时间的关系图如图 10 所示。

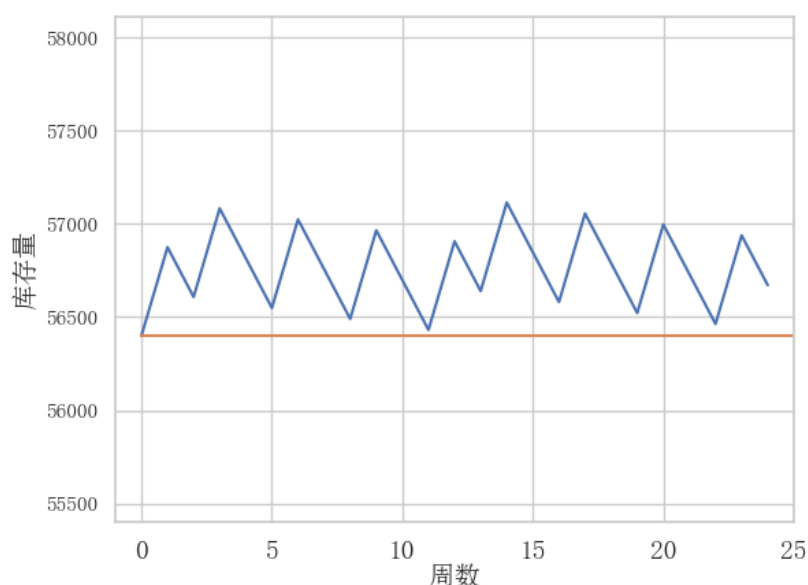


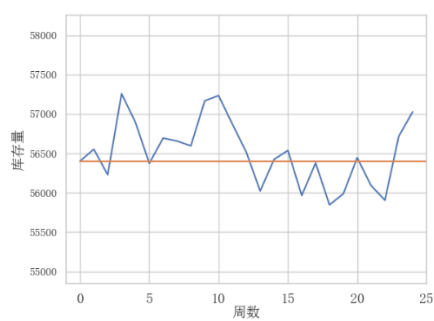
图 10 理想情况下库存量与时间关系图

图 10 体现了在我们给出的订购和转运方案下，理想情况时，企业原材料库存量始终在满足两周产能的库存量上方波动，能够较好的保证企业的正常生产，同时兼顾经济性。

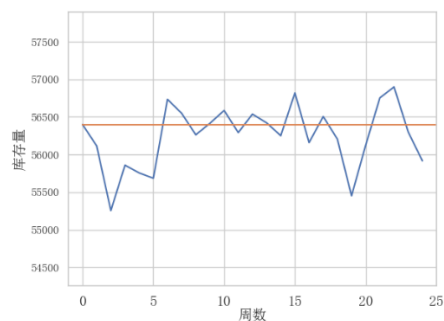
5.2.4.3 实施效果及稳定性分析

在方案实际实施的过程中，最容易出现的异常情况是由于企业供货能力的波动性，导致企业难以获得足够的原材料或者原材料的转运安排不合理导致无法正常转运。如果方案稳定性较差则会出现失衡的情形。我们通过对供应商的供货能力施加扰动，观察企业原材料库存量、各转运商转运量变化情况，来验证方案的可行性和稳定性。

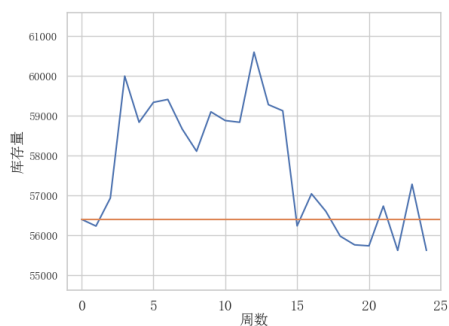
图 11 展示了在不同等级的随机扰动下，模拟出的企业库存量随时间变化曲线。根据图象可以判断，在各等级扰动的随机模拟中企业的原材料库存虽然不能完全保证在两周产能之上，但是总能快速调节，并未出现明显失稳的趋势。



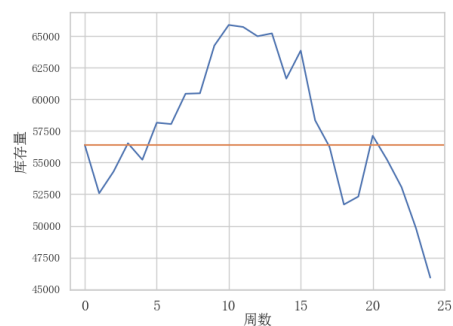
(a) 波动程度 0.05



(b) 波动程度 0.1



(c) 波动程度 0.2



(d) 波动程度 0.5

图 11 不同波动程度下库存量与时间关系图

而图 12 则展示了不同转运商每周的转运量（图中 T1、T2、T5、T7 的转运量为 0，转运量曲线重合），可以看到评价排名靠前的转运商将承担更多的转运任务，这样有助于降低整个过程的损耗；且各转运商的每周转运量较为稳定，仅有少数转运商有小幅波动，这样将为企业与转运商的合作提供方便，也说明了我们的转运方案的合理性、优越性。

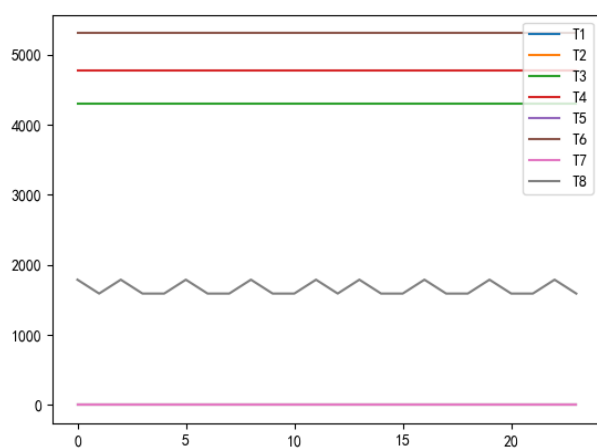


图 12 不同转运商每周的转运量

5.3 问题三：对 402 家供应商最经济订购方案和最少损耗转运方案的制定

本问题主要在问题二的基础上，研究当供应商选择的范围由 39 家增加到 402 家时，订购方案和转运方案的变化。与第二问相比，在订购模型和转运模型的建立上基本一致，我们着重设计在更大数据规模下的优化算法和比较求解结果与第二问结果之间的差异性，并分析其原因。

5.3.1 模型的建立

对于订购模型，是以最低成本为优化目标，以订购量矩阵和转运量矩阵为决策变量，受到订购上限约束和库存下限约束的单目标优化模型；对于转运模型，是以最低损耗为优化目标，以订购量矩阵和转运量矩阵为决策变量，受到转运上限和集中转运约束的单目标优化模型。根据 5.2.3 的分析和计算，最经济订购方案和最少损耗方案的统一优化模型为

$$\begin{aligned}
 & \min \sum_{j=1}^m \left(\sum_{i=1}^n e_i r_{ij} \eta_{ij} + \sum_{i=1}^3 \tau e d_{ij} \right) \\
 & \min \sum_{i=1}^p \sum_{j=1}^m l_{ij} \\
 & s.t. \begin{cases} r_{ij} \eta_{ij} \leq s_{ij}, \forall i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{r_{ij}} - Q_0 \\ w_j \geq 2Q_0, \forall j = 1, 2, \dots, m \\ t_{ij} \leq 6000\delta, \forall i = 1, 2, \dots, p, j = 1, 2, \dots, m \\ \bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j = 1, 2, \dots, m \end{cases} \quad (44)
 \end{aligned}$$

5.3.2 模型的求解

效法 5.2.4.1，针对订购转运模型的特点，我们依旧可以采用贪心算法，尽可能优先选择更好的供货商和转运商，保证成本、损耗均最小。算法具体流程如下。

- Step1:** 将周数初始化为 $w = 1$ ，成本初始化为 0，库存量初始化为满足企业两周产量的原材料；
- Step2:** 将所有供应商(402 家)首先按照 A-B-C 顺序排列，其次按评价指标排序；转运商按照损耗率升序排列；
- Step3:** 选择排名最高的供应商，将其所有供应量分配给排名最高的转运商运输；
- Step4:** 选择下一名的供应商，将其所有供应量分配给排名最高的、有足够余量的转运商运输；若所有转运商均无余量，返回转运错误；若无下一名供应商，返回供应错误；
- Step5:** 计算企业总的接受量，判断其是否满足一周产能，若是则跳转 **Step6**，否则跳转 **Step4**；
- Step6:** 记录该周订货量向量和转运量向量，计算总成本， $w = w + 1$ ，判断 w 是否大于 24，若是则跳转 **Step7**，否则跳转 **Step2**；
- Step7:** 输出订货量矩阵、转运量矩阵和总成本。

5.3.3 求解结果和分析

5.3.3.1 求解结果

运用 5.3.2 中的算法对模型进行求解，得到了对于全部 402 家企业而言的订货量矩阵和转运量矩阵，分别保存在“附件 A”、“附件 B”中。以此作为最经济订购方案 and 最小损耗转运方案，其中 283 家供应商参与了订购方案、7 家转运商参与了转运方案，在自拟的参数条件下企业 24 周总的原材料采购和存储成本为 179.8 万元，转运损耗量为 6433.2m³。根据订购和转运方案，可以绘制在理想情况下库存量与时间的关系图如图 13 所示。

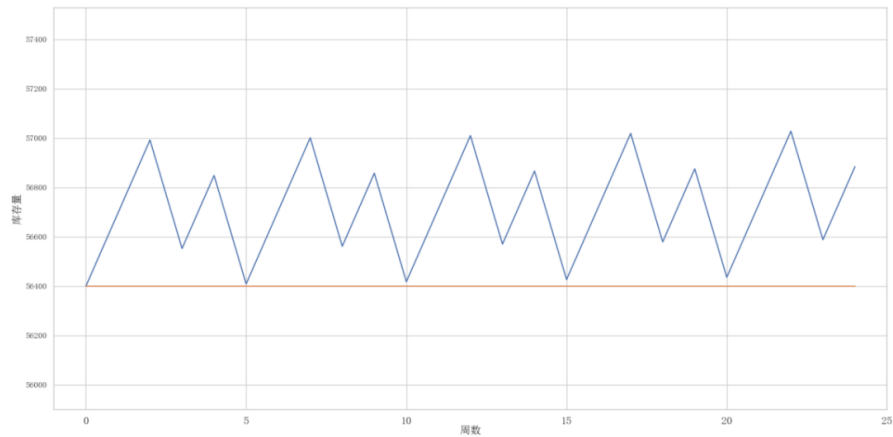


图 13 理想情况下库存量与时间关系图

5.3.3.2 实施效果及稳定性分析

我们再次引入对供货商供货能力的扰动，观察企业原材料库存量、各转运商转运量的变化情况，来验证方案的可行性和稳定性。

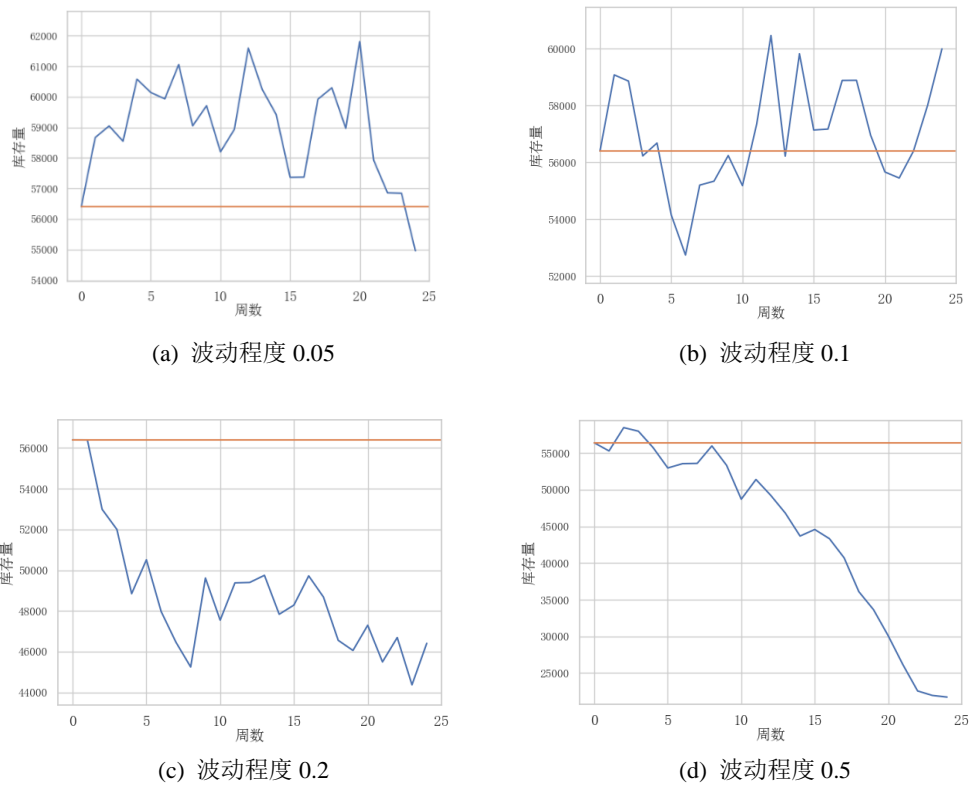


图 14 不同波动程度下库存量与时间关系图

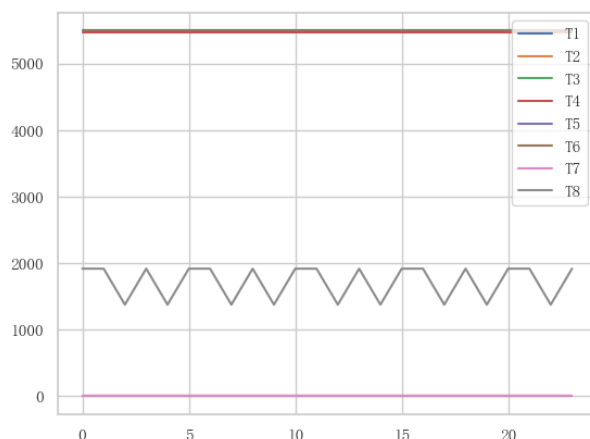


图 15 不同转运商每周的转运量

图 14 展示了不同波动程度下库存量随时间的变化图，图 15 展示了不同转运商每周的转运量（图中 T1—T6 的转运量均满额转运，转运量曲线重合）。根据图形我们发现，在扰动较小时，库存量保持在较为稳定的水平，说明方案的整体执行效果较好；而随着扰动程度的增加，库存量逐渐显露出衰减的态势，逐渐由稳态走向失稳，但在 24 周以内仍能保持企业的正常生产。这相比于问题二的供应商选择有所差距。

5.3.3.3 与问题二结果的比较

问题二求解的在 39 家筛选出的企业中求解的订购和转运方案最小成本为 181.4 万元，最少损耗量为 6783.2 m³；问题三求解的在全部 402 家企业中求解的订购和转运方案最小成本为 179.8 万元，最少损耗量为 6433.2 m³。

显然，问题三方案的成本小于问题二，这是因为问题二中供应 A 类原材料的供应商数目有限，企业为保证生产不得不去采购 B 和 C 类原材料，整体成本较高。但是问题三中的低成本也不是凭空而来，相比于问题二，问题三引入了大量的在问题一中评价并不良好的企业，选择他们作为原材料供应商将承担更大的风险，这在 5.3.3.2 中问题三方案抗干扰能力不如问题二的方案中也有直观的体现。

5.4 问题四：上限产能的建模求解和订购、转运方案的制定

本问题主要解决在企业具备提高产能的潜力时，产能提高的上限值。在对前面问题的求解中，产能被作为一个固定的约束条件而对供应商和转运商进行选择和分配。在本问中，产能由定值 $Q_0 = 28200\text{m}^3$ 转变为变量 Q ，在不考虑其他因素的情况下产能将只受到原材料采购和转运的约束，需要重新建立模型来对本问题进行求解。

5.4.1 产能上限模型的建立

5.4.1.1 目标函数、决策变量、约束条件的确定

(1) 目标函数

由于问题要求在条件允许下尽可能提高产能，而不考虑包括成本、转运损耗相关的因素，因此模型的优化目标为

$$\max Q \quad (45)$$

(2)决策变量

企业在制定计划、做出决策时，能够改变的量只有订购量矩阵和转运量矩阵 $T_{p \times m}$ 和订货量矩阵 $R_{n \times m}$ ($n = 402$)

$$T = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{p1} & \cdots & t_{pm} \end{bmatrix}, R = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{bmatrix} \quad (46)$$

在本问中，也正是对于订货量、转运量的安排和选择能够让企业获得更多的原材料，同时减小运输过程的损失，直接决定了产能的上限，因此我们选择转运量矩阵 $T_{p \times m}$ 和订货量矩阵 $R_{n \times m}$ 作为模型的决策变量。

(3)约束条件

模型的第一个约束条件是**订购上限约束**。从每一家供应商处实际购得的原材料数量不应多于供应商的最大供应能力，即

$$r_{ij}\eta_{ij} \leq s_{ij}, \forall i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (47)$$

模型的第二个约束条件是**库存下限约束**。库存量对应的产能在时间序列上具有前后的连续性，根据式(40)有

$$w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{tr_i j} - Q \quad (48)$$

其中 tr_i 表示转运第 i 家供应商所提供的原材料的转运商编号。根据企业保障生产的要求，库存量对应的产能不应小于企业两周的实际产能，即需满足

$$w_j \geq 2Q, \forall j = 1, 2, \dots, m \quad (49)$$

模型的第三个约束条件是**转运上限约束**。即单个转运商每周转运量不应超过 6000 m^3 ，即

$$t_{ij} \leq 6000\delta, \forall i = 1, 2, \dots, p, j = 1, 2, \dots, m \quad (50)$$

模型的第四个约束条件是**集中转运约束**，根据式(30)有

$$\bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j = 1, 2, \dots, m \quad (51)$$

其中 U_j 为第 j 周所有需要转运的供应商的集合， Tr_{ij} 为第 i 家转运商第 j 周负责转运的供应商的集合。

5.4.1.2 产能上限模型的确定

综上，我们建立了以转运量矩阵和订货量矩阵为决策变量，以最大化产能为目标，受到订购、库存和转运约束的单目标优化模型

$$\begin{aligned}
& \max Q \\
& \left\{ \begin{aligned}
& t_{ij} \leq 6000\delta, \forall i=1,2,\dots,p, j=1,2,\dots,m \\
& \bigcap_{i=1}^n Tr_{ij} = \emptyset, \bigcup_{i=1}^n Tr_{ij} = U_j, \forall j=1,2,\dots,m \\
& r_{ij}\eta_{ij} \leq s_{ij}, \forall i=1,2,\dots,n, j=1,2,\dots,m \\
& w_{j+1} = w_j + \sum_{i=1}^n \theta_i r_{ij} \eta_{ij} \varphi_{r_{ij}} - Q \\
& w_j \geq 2Q, \forall j=1,2,\dots,m
\end{aligned} \right. \quad (52)
\end{aligned}$$

5.4.2 模型的求解

模型求解的核心在于根据供货商供货特性，考虑在转运商容量限制下的最大产能，依旧考虑贪心算法，尽可能优先考虑更好的供应商和转运商，可以遇见求解算法或在达到转运容量限制时停止、或在遍历完所有供应商时停止，两种情况下我们均能得到产能上限及对应的订货、转运方案。详细算法流程如下。

- Step1:** 将所有供应商(402 家)首先按照 A-B-C 顺序排列，其次按评价指标排序；转运商按照损耗率升序排列；
- Step2:** 选择排名最高的供应商，将其所有供应量分配给排名最高的转运商运输；
- Step3:** 选择下一名的供应商，将其所有供应量分配给排名最高的、有足够余量的转运商运输；
- Step4:** 若所有转运商均无余量，或无下一名供应商，运行结束输出订货量矩阵、转运量矩阵和总成本，跳转 **Step5**；否则跳转 **Step3**；
- Step5:** 根据订货量矩阵、转运量矩阵及已知信息，计算最大产能，输出。

5.4.3 求解结果

运用 5.4.2 中的算法对模型进行求解，计算出**最高产能为 35571.95m³**，较原始产能**提高 26.14%**，同时得到了对于全部 402 家企业而言的订货量矩阵和转运量矩阵，以此作为未来 24 周的最经济订购方案和最小损耗转运方案，分别保存在“附件 A”、“附件 B”中。在自拟参数条件下，企业 24 周总的原材料采购和存储成本为 227.3 万元，转运损耗量为 9318.7m³。根据订购和转运方案，可以绘制在理想情况下库存量与时间的关系图如图 16 所示。

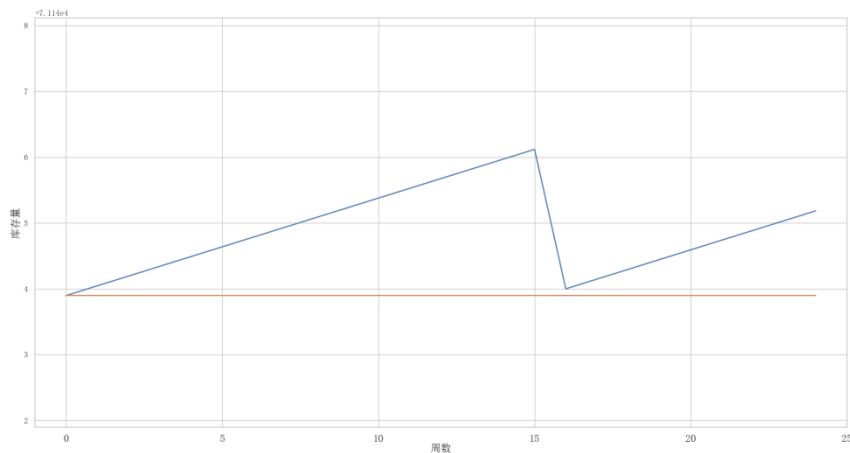


图 16 最大产能下库存量与时间关系图

六、模型的评价与改进方向

6.1 模型的优点

- ◆ 评价体系科学，评价结果准确。在对供应商保障企业生产重要性的评价问题中，我们建立的 PCA 评价模型取得较好的效果，且模型能够对供应商进行综合性的评价，对不同类型的供应商都能取得比较客观的结果；
- ◆ 模型实际执行的稳定性好，不易受到供应商变动的影响。在通过增加扰动对模型进行稳定性分析的过程中，模型的计算结果始终能够保证企业的正常经营和转运的正常进行，抗干扰能力较强；
- ◆ 泛化能力强，适用性广。订购和转运方案基于对供应商、转运商行为的预测开展，仅依靠供应商供应量矩阵、满足率矩阵和转运商损失率矩阵就可以进行方案的制定，能够应用在更一般的场合，模型泛化能力较强；

6.2 模型的缺点

- ◆ 对供应商、转运商行为的预测模型可以更加精确。在对供应量矩阵、满足率矩阵和损失率矩阵进行建模预测时，采取了用统计特征进行替代的做法，虽然在较长时间内实际效果与真实情况相近，但是不可避免的损失了一部分数据信息；
- ◆ 对供应商重要性的评价体系可以更加客观。目前我们对供应商重要性的评价体系主要从数据的角度出发，因此在一定程度上不能反映供应商全貌，可能在评价的结果上具有一定的偏颇。

6.3 模型的改进方向

- ◆ 在对供应商保障企业生产重要性的评价问题上，可以跳出数据限制，通过查阅相关专业文献建立更科学、更全面、更准确的供应商评价体系；
- ◆ 在建立供应商、转运商行为的预测模型时，利用实际分布特征对各供应商的供货量进行更精确的拟合，使得模型的约束条件更加准确、贴合实际；
- ◆ 在对优化模型进行求解时，可以考虑设计更具效率和准确性的启发式算法，提高模型求解的速度和精确度。

七、参考文献

- [1] 盛骤. 概率论与数理统计:第三版[M]. 高等教育出版社, 2001.
- [2] 司守奎, 孙玺菁. 数学建模算法与应用[M]. 国防工业出版社, 2011.
- [3] 杨琳. 多供应商选择及物料采购量分配问题的研究[D]. 东华大学.
- [4] 梁珊珊, 吕芳德, 蒋瑶,等. 美国山核桃坚果主成分分析及综合评价[J]. 中国南方果树, 2015, 44(003):123-128.
- [5] 辛霞. 供应商分类模型及指标体系研究[D]. 浙江大学.
- [6] Jakubowska-Gawlik Katarzyna and Kolanowski Wojciech and Trafialek Joanna. Evaluating suppliers of spices, casings and packaging to a meat processing plant using food safety audits data gathered during a 13-year period[J]. Food Control, 2021, 127
- [7] Karami Shirin and Ghasemy Yaghin R. and Mousazadegan Fatemeh. Supplier selection and evaluation in the garment supply chain: an integrated DEA-PCA-VIKOR approach[J]. The Journal of The Textile Institute, 2021, 112(4) : 578-595.

八、附录

8.1 附件清单

编号	文件名	类型	文件内容
A	附件 A 订购方案数据结果	xlsx	题目要求提交的订购方案数据结果
B	附件 B 转运方案数据结果	xlsx	题目要求提交的转运方案数据结果
3	quantization	csv	供应商 ID 对应的排名、周最大供货量、满足率
4	quantization(rank p3)	csv	较 quantization 多了材料分类
5	quantization(rank only)	csv	仅含各供应商的综合指标排名
6	evaluateSupply	csv	各供应商对应的综合指标评分及排名
7	evaluateTransfer	csv	各转运商对应的综合指标评分及排名
8	afterProcess	txt	各供应商的 5 个综合评价指标
9	afterProcess	csv	各供应商的 5 个综合评价指标及订货量数据
10	matrix	文件夹	内含损失率、满足率、供应量三个预测矩阵(csv 格式)
11	simulation&check(p2)	文件夹	内含问题 2 求解所需的 3 个数据表格
12	simulation&check(p3)	文件夹	内含问题 3 求解所需的 3 个数据表格
13	simulation&check(p4)	文件夹	内含问题 4 求解所需的 3 个数据表格
14	PCA	m	主成分分析及综合评价使用的代码
15	dataPreProcess	py	数据预处理、指标选取使用的代码
16	dataPreProcess2	py	计算周最大供货量及满足率使用的代码
17	satisfyAna	py	满足率分析相关代码
18	matrixGen	py	损失率、满足率、供应量 3 个预测矩阵的生成代码
19	excelProcess	py	生成各供应商对应的综合指标评分及排名的代码
20	drawGraph	py	生成不同供货商的供货量分布的代码
21	simulation5	py	问题 2 的贪心算法代码
22	check5	py	问题 2 的结果检验及敏感性分析代码
23	simulation(p3)	py	问题 3 的贪心算法代码
24	check(p3)	py	问题 3 的结果检验及敏感性分析代码
25	simulation(p4)	py	问题 4 的贪心算法代码
26	check(p4)	py	问题 4 的结果检验及敏感性分析代码

*所有附件将打包为压缩文件“支撑材料.zip”另行上交，其中附件 14、附件 15、附件 21 还将在 8.2 中展示。

8.2 代码示例

附件 14	PCA.m	主成分分析及综合评价使用的代码
		<pre> clc,clear gj=load('afterProcess.txt'); gj=zscore(gj); r=corrcoef(gj); [x,y,z]=pcacov(r) f=repmat(sign(sum(x)),size(x,1),1); x=x.*f; num=4; df=gj*x(:,[1:num]); </pre>

```
tf=df*z(1:num)/100;
[stf,ind]=sort(tf,'descend')
stf=stf';
ind=ind';
```

附件 15	dataPreProcess.py	数据预处理、指标选取使用的代码
-------	-------------------	-----------------

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
df_order=pd.read_excel('.\\data\\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)')
df_supply=pd.read_excel('.\\data\\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)')
for i in range(5):
    #plt.subplot(3,3,i+1)
    #print([df_order['W'+(3-len(str(j)))*str(0)+str(j)] for j in range(1,241)])
    #plt.plot(list(range(1,241)),np.array([df_order['W'+(3-len(str(j)))*str(0)+str(j)][i] for j in
range(1,241)])) #weekg
    yearsum=[sum([df_order['W' + (3 - len(str((j-1)*48+k))) * str(0) + str((j-1)*48+k)][i] for k
in range(1,49)]) for j in range(1, 6)]
    #plt.plot(list(range(1, 6)),yearsum) #year
    print(i)
matrix=[[0,0,0,0,0,0] for i in range(len(df_supply.index))]
for i in df_supply.index:
    x=list(range(1,6))
    yearsum_o = [sum(
        [df_order['W' + (3 - len(str((j - 1) * 48 + k))) * str(0) + str((j - 1) * 48 + k)][i] for k
in range(1, 49)])
        for j in range(1, 6)]
    yearsum_s = [sum(
        [df_supply['W' + (3 - len(str((j - 1) * 48 + k))) * str(0) + str((j - 1) * 48 + k)][i] for k
in range(1, 49)])
        for j in range(1, 6)]
    i_o_data = [df_order['W' + (3 - len(str(j))) * str(0) + str(j)][i] for j in range(1, 241)]
    i_s_data = [df_supply['W' + (3 - len(str(j))) * str(0) + str(j)][i] for j in range(1, 241)]
    matrix[i][0]=np.average(yearsum_o) #年平均订货量
    matrix[i][1] = np.average(yearsum_s) # 年平均供货量
    matrix[i][2] = np.var(np.array([i_s_data[j]-i_o_data[j] for j in range(0,240)])) #供货量-订货
量的方差
    matrix[i][3]=sum(yearsum_s)/sum(yearsum_o) #满足率
    matrix[i][4]=sum([1 for j in range(0,240) if (i_s_data[j]>5 and i_o_data!=0)]/240 #交易频
率
    matrix[i][5]=(5*sum([x[k]*yearsum_o[k] for k in range(0,5)])-
sum(x)*sum(yearsum_o))/(5*sum([k*k for k in x])-sum(x)**2)#订货量增长率 回归
dfw=pd.DataFrame(matrix,index=df_supply['供应商 ID'],columns=['年平均订货量','年平均供货量','供货
稳定性系数','订单满足率','交易频率','订单增长系数'])
dfw.to_csv('.\\data\\afterProcess.csv')

#plt.plot(list(range(1,403)),[i[0] for i in matrix])
sns.displot([i[1] for i in matrix],bins=50,kde=True)

#plt.plot(list(range(1,403)),[i[1] for i in matrix])
```

```

#sns.displot([i[1] for i in matrix])
plt.show()
s=0
output1=[]
output2=[]
for j in range(1,241):
    column='W'+(3-len(str(j)))*str(0)+str(j)
    o1=0
    o2=0
    for i in df_supply.index:
        if df_supply['材料分类'][i]=='A':
            o1+=df_supply[column][i]/0.6
        if df_supply['材料分类'][i]=='B':
            o1+=df_supply[column][i]/0.66
        if df_supply['材料分类'][i]=='C':
            o1+=df_supply[column][i]/0.72
        if df_order['材料分类'][i]=='A':
            o2+=df_order[column][i]/0.6
        if df_order['材料分类'][i]=='B':
            o2+=df_order[column][i]/0.66
        if df_order['材料分类'][i]=='C':
            o2+=df_order[column][i]/0.72
    output1.append(o1)
    s+=output1[-1]
    output2.append(o2)
print('ave',s/240)
print(len(list(range(1,241))))
print(len(output1))
plt.plot(list(range(1,241)),list(output1),list(range(1,241)),list(output2))
plt.show()

```

附件 21	simulation5.py	问题 2 的贪心算法代码
-------	----------------	--------------

```

# 模拟, 库存量曲线
# 订货量由历史 5 年数据确定

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from random import *
sns.set(style='whitegrid')
import matplotlib
import numpy as np
matplotlib.rcParams['font.sans-serif']=['simsum'] #显示中文标签
#matplotlib.rcParams['font.sans-serif']=['Times New Roman'] #显示中文标签
matplotlib.rcParams['axes.unicode_minus']=False
cost=0
damage=0
def simulation(k):
    global cost
    global damage
    #print('我们选择了%d 家供应商,他们分别是\n'%k)
    sas=0
    for i in range(k):
        r=rank[i] #供应商

```

```

# print(i+1, r, aveSupply[r], satisfy[r])
'''
if kind[r]=='A':
    sas+=aveSupply[rank[i]]*satisfy[rank[i]]/0.6
if kind[r]=='B':
    sas+=aveSupply[rank[i]]*satisfy[rank[i]]/0.66
if kind[r]=='C':
    sas+=aveSupply[rank[i]]*satisfy[rank[i]]/0.72
'''
total=28200
s_index=0
week=0
while week<24:
    cost+=3*28200-total
    Tleft = [6000 for i in range(8)] # 转运商余量
    while(total>0):
        sindex = int(rank[s_index][1:]) - 1
        if s_index>=k:
            return 'False, wrong supply'
        r = rank[s_index] # 供应商
        # print(i+1, r, aveSupply[r], satisfy[r])
        beta=1 # 测试增益
        t_index=0
        # while supply[sindex][week]>Tleft[t_index]-500:
        while aveOrder[sindex][week] * satisfy[sindex][week] > Tleft[t_index] - 500:
            t_index+=1
            if t_index>=8:
                return 'False, wrong transfer'
            Tleft[t_index]-=aveOrder[sindex][week] * satisfy[sindex][week]
            cost+=aveOrder[sindex][week] * satisfy[sindex][week]
            # print(Tleft)
            lindex=int(df_et['转运商 ID'][t_index][1:])-1 # 转运商序号
            transfer[sindex][week * 8 + lindex]=aveOrder[sindex][week] * satisfy[sindex][week]
            Ordertransfer[sindex][week]=lindex
            satisfaction[sindex][week]=str(aveOrder[sindex][week] *
satisfy[sindex][week])+','+str(satisfy[sindex][week])+','+str(loss[lindex][week])
            aveOrdernew[sindex][week]=aveOrder[sindex][week] * satisfy[sindex][week]
            '''
            if satisfy[s_index][week] != 0:
                aveOrder=supply[s_index][week]/satisfy[s_index][week]
            else:
                aveOrder=0
            if kind[r]=='A':
                total-=supply[s_index][week]/0.6*beta*loss[lindex][week]
                aveOrder[rank[index]] * satisfy[rank[index]]
            if kind[r]=='B':
                total-=supply[s_index][week]/0.66*beta*loss[lindex][week]
            if kind[r]=='C':
                total-=supply[s_index][week]/0.72*beta*loss[lindex][week]
            '''
            if kind[r]=='A':
                # total-=aveOrder[sindex][week] *
satisfy[sindex][week]/0.6*beta*loss[lindex][week]
                # total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.6 * beta *
0.95
                total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.6 * beta *
loss[lindex][week]

```



```

        damage+=aveOrder[sindex][week] * satisfy[sindex][week] / 0.6 * beta *(1-
loss[lindex][week])
        if kind[r]=='B':
            #total-=aveOrder[sindex][week] *
satisfy[sindex][week]/0.66*beta*loss[lindex][week]
            #total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.66 * beta *
0.95
            total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.66 * beta *
loss[lindex][week]
            damage += aveOrder[sindex][week] * satisfy[sindex][week] / 0.66 * beta *
(1 - loss[lindex][week])
        if kind[r]=='C':
            #total-=aveOrder[sindex][week] *
satisfy[sindex][week]/0.72*beta*loss[lindex][week]
            #total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.72 * beta *
0.95
            total -= aveOrder[sindex][week] * satisfy[sindex][week] / 0.72 * beta *
loss[lindex][week]
            damage += aveOrder[sindex][week] * satisfy[sindex][week] / 0.72 * beta *
(1 - loss[lindex][week])
        s_index+=1
        #print(total,s_index,rank[s_index-1],aveOrder[sindex][week],
satisfy[sindex][week],df_saf.values[sindex][week+1],(1-df_lf.values[lindex][week+1]/100))
        #print(Tleft)
        for si in range(s_index,402):
            aveOrdernew[int(rank[si][1:])-1][week]=0
            aveOrder[int(rank[si][1:])-1][week]=None
        s_index=0
        total+=28200
        week+=1
        print(total)
        storage.append(3*28200-total)
    return 'True'

```

```

df_rank=pd.read_csv('..\\data\\quantization(rank only).csv')
df_ap=pd.read_csv('..\\data\\afterProcess.csv')
df_order=pd.read_excel('..\\data\\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)')
df_supply=pd.read_excel('..\\data\\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)')
df_ls=pd.read_excel('..\\data\\附件 2 近 5 年 8 家转运商的相关数据.xlsx')
df_st=pd.read_csv('..\\data\\satisfy.csv')
df_et=pd.read_csv('..\\data\\evaluateTransfer.csv')
df_es=pd.read_csv('..\\data\\evaluateSupply.csv')
df_saf=pd.read_csv('..\\data\\matrix\\satisfyForecast.csv')
df_lf=pd.read_csv('..\\data\\matrix\\lossForecast.csv')
rank=list(df_rank['rank'])
for i in range(len(rank)):
    rank[i]='S'+(3-len(str(rank[i]))) * str(0)+str(rank[i])
aveOrder=[[None for i in range(24)] for j in range(402)] # 周平均订货量
aveOrdernew=[[0 for i in range(24)] for j in range(402)] # 周平均供货量
satisfy=[[0 for i in range(24)] for j in range(402)] # 订单满足率
kind=dict() # 材料分类
supply=[[0 for i in range(24)] for j in range(402)] # 供应限制
loss=[[0 for i in range(24)] for j in range(8)] # 供应限制

```



```

transfer=[[None for i in range(8*24)] for j in range(402)]
#供应限制
Ordertransfer=[[0 for i in range(24)] for j in range(402)] #订单对应的转运商
satisfaction=[[0 for i in range(24)] for j in range(402)] #订单对应的转运商
'''
for i in df_ap.index:
    aveOrder[df_ap['供应商 ID'][i]]=df_ap['年平均订货量'][i]/48
    satisfy[df_st['供应商 ID'][i]] = df_st['周平均满足率'][i]'''
for i in df_order.index:
    kind[df_order['供应商 ID'][i]]=df_order['材料分类'][i]
for i in range(402):

    sai=df_st['周平均满足率'][i]
    satisfy[i]=[sai for j in range(24)]
    sui=df_ap['年平均供货量'][i]/48
    supply[i]=[sui for j in range(24)]
    avo=df_ap['年平均订货量'][i]/48
    aveOrder[i] = [avo for j in range(24)]
    #aveOrder[i]=[np.average([df_order['W' + (3 - len(str(k*24+j+1))) * str(0) +
    str(k*24+j+1)])[i] for k in range(0,10)]) for j in range(24)]
print(len(aveOrder),len(aveOrder[0]))
print(aveOrder[0])

    #print(ls)

    #print(loss[i])
for i in range(8):
    loss[i] = [df_et['评分'][i]/100 for j in range(24)]

#print('supply',supply)
#print('loss',loss)
#print('satisfy',satisfy)
cost=0
for k in range(39,40):
    success=0
    storage=[28200*2]
    notice=simulation(k)
    print(notice)
    #plt.plot(list(range(len(storage))), storage)
    #percentage.append(success/count)
    #plt.show()
print('cost',cost)
dfwO=pd.DataFrame(aveOrder,columns=['W' + (3 - len(str(j))) * str(0) + str(j) for j in range(1,
25)],index=df_order['供应商 ID'])
dfwO.to_csv('.\\data\\simulation&check\\simulationOrder.csv')
dfwT=pd.DataFrame(transfer,columns=['W' + (3 - len(str(j))) * str(0) + str(j)+'T'+str(k+1) for j
in range(1, 25) for k in range(8)],index=df_order['供应商 ID'])
dfwT.to_csv('.\\data\\simulation&check\\simulationTransfer.csv')
dfwOT=pd.DataFrame(Ordertransfer,columns=['W' + (3 - len(str(j))) * str(0) + str(j) for j in
range(1, 25)],index=df_order['供应商 ID'])
dfwOT.to_csv('.\\data\\simulation&check\\simulationOrderTransfer.csv')
dfwsa=pd.DataFrame(satisfaction,columns=['W' + (3 - len(str(j))) * str(0) + str(j) for j in
range(1, 25)],index=df_order['供应商 ID'])
dfwsa.to_csv('.\\data\\simulation&check\\satisfaction.csv')
print(storage)
plt.plot(list(range(0,25)),storage)
plt.plot([0,25],[0,0])

```

```

plt.axis([-1,25,min(storage)-1000,max(storage)+1000])
plt.title('库存量变化图', fontsize=16)
plt.xticks(np.arange(0, 26, 5),fontsize=14)
plt.xlabel('周数', fontsize=14, labelpad=0)
plt.ylabel('库存量', fontsize=14)
plt.show()
print('cost',cost)
print('damage',damage)
#print(satisfy[0])
#print(aveOrder[0])
#print(loss[0])

#s=sum([aveOrder[rank[index]]*satisfy[rank[index]] for index in df_ap.index])
#print(28200/s)
'''
ave=[] for i in range(8)
for i in range(8):
    plt.subplot(3,3,i+1)
    #print([df_order['W'+(3-len(str(j)))*str(0)+str(j)] for j in range(1,241)])
    plt.plot(list(range(1,241)),np.array([df_lossrate['W'+(3-len(str(j)))*str(0)+str(j)])[i] for j in
range(1,241)])) #weekg
    #yearsum=[sum([df_lossrate['W' + (3 - len(str((j-1)*48+k))) * str(0) + str((j-1)*48+k)])[i] for j in
range(1,49)]) for j in range(1, 6)]
    #plt.plot(list(range(1, 6)),yearsum) #year
    a=[df_lossrate['W' + (3 - len(str(j)) * str(0) + str(j))][i] for j in range(1, 241)]
    for j in a:
        if j!=0:
            ave[i].append(j)
    ave[i]=sum(ave[i])/len(ave[i])

plt.show()
print(ave)
matrix=[[0,0,0,0,0] for i in range(len(df_supply.index))]
for i in df_supply.index:
    x=list(range(1,6))
    yearsum_o = [sum(
        [df_order['W' + (3 - len(str((j - 1) * 48 + k))) * str(0) + str((j - 1) * 48 + k)])[i] for k
in range(1, 49)])
        for j in range(1, 6)]
    yearsum_s = [sum(
        [df_supply['W' + (3 - len(str((j - 1) * 48 + k))) * str(0) + str((j - 1) * 48 + k)])[i] for k
in range(1, 49)])
        for j in range(1, 6)]
    i_o_data = [df_order['W' + (3 - len(str(j)) * str(0) + str(j))][i] for j in range(1, 241)]
    i_s_data = [df_supply['W' + (3 - len(str(j)) * str(0) + str(j))][i] for j in range(1, 241)]
    matrix[i][0]=np.average(yearsum_o) #年平均订货量
    matrix[i][1] = np.var(np.array([i_s_data[j]-i_o_data[j] for j in range(0,240)])) #供货量-订
    货量的方差
    matrix[i][2]=sum(yearsum_s)/sum(yearsum_o) #满足率
    matrix[i][3]=sum([1 for j in range(0,240) if (i_s_data[j]>5 and i_o_data!=0)])/240 #交易频
    率
    matrix[i][4]=(5*sum([x[k]*yearsum_o[k] for k in range(0,5)])-
sum(x)*sum(yearsum_o))/(5*sum([k*k for k in x])-sum(x)**2)#订货量增长率 回归

dfw=pd.DataFrame(matrix,index=df_supply['供应商 ID'],columns=['年平均订货量','供货量减订货量的方
差','满足率','交易频率','订单量增长趋势'])
dfw.to_csv('.\\data\\afterProcess.csv')

```

```

#plt.plot(list(range(1,403)),[i[0] for i in matrix])
sns.displot([i[1] for i in matrix],bins=50,kde=True)

#plt.plot(list(range(1,403)),[i[1] for i in matrix])
#sns.displot([i[1] for i in matrix])
plt.show()

for j in range(1,241):
    column='W'+(3-len(str(j)))*str(0)+str(j)
    output=0
    df=df_supply
    for i in df.index:
        if df['材料分类'][i]=='A':
            output+=df[column][i]/0.6
        if df['材料分类'][i]=='B':
            output+=df[column][i]/0.66
        if df['材料分类'][i]=='C':
            output+=df[column][i]/0.72
    print(output)
'''

```