基于 TOPSIS 和目标规划的企业订购与转运方案的量化分析

摘要

某建筑和装饰板材的生产企业需购入 A、B、C 三种类型的原料用于生产，且 A、B、C 原料的转化率和单价均不相同，为保证企业每周的生产需求，需要我们完成以下问题。问题一：对供货商的供货特征进行量化分析，建立数学模型为供货商排名。问题二：计算企业至少需要选择多少家供应商才能满足生产的需求，并针对这些供应商制定最经济的原料订购方案和损耗最小的转运方案。问题三：企业为压缩成本，希望多采购 A，少采购 C，根据此原则重新制定订购和转运方案。问题四：企业具备提高产能的潜力，根据实际情况制定提升产能的计划，并给出订购和转运方案。

针对问题一，我们建立了供应商评价模型。从供应商的信誉、能力以及供货稳定性三个方面构造了三个指标来衡量供应商的特征，并通过熵权法确定它们之间的权重，再使用优劣解距离法 (TOPSIS) 计算得出每个供应商的最终得分并据此排名。对排名结果分析发现，排名靠前的供应商总供应量都很大，违约占比较小且订单完成度较高，说明这些供应商在信誉、能力和稳定性三方面都表现较好，证明我们的模型具有合理性。

针对问题二，我们分别建立了以最小化供应商数量和最大化利润为目标函数的目标规划，利用模拟退火算法对其进行求解。结果显示企业至少需要选择 276 家供应商提供原料才能满足生产的需求，其中排名靠前的供应商都在列，说明求解结果具有一定的可信度。并得出了每周的最经济采购方案，对采购方案分析发现基本需要向所有的 276 家供应商下单才能满足生产需求。建立以最小化转运损耗产能为目标函数的目标规划，得出转运方案，观察发现转运原则是优先由损耗较小的转运商转运单价最贵且单位产能最大的 A 原料。

针对问题三，我们建立了以最大化 A 原料和 C 原料数量的差值为目标函数的目标规划，通过模拟退火算法对其进行求解，得到新的订购方案，观察结果发现企业会对所有生产 A 原料的供应商下单，若所购入的 A 原料不足以满足生产需求再向生产 B 原料的供应商下单，若还不够才选择向生产 C 原料的供应商下单。建立与问题二类似的转运方案求解目标规划解出相应转运方案。

针对问题四，我们建立了以最大化利润为目标函数的目标规划，在约束条件中引入表示产能提升的变量，通过模拟退火算法计算得出新的订购方案，并根据订购方案求出可行的产能提升规划。观察订购方案发现为保证产能提升不被原料影响，企业会向所有供应商下单以保证原料充足。这里我们假定每周产能的提升是相同的，最终计算得出每周产能提升 1108.94 立方米。

最后我们对建立的模型进行评价和分析，认为建立的模型具有分析角度较全面的优点，但也存在由于求解算法导致的结果不全面的缺点，因此模型的主要改进方向是各个目标规划的算法方面。

关键字：熵权法　优劣解距离法　目标规划　模拟退火

目录

一、 问题重述

1.1 问题提出

企业生产产品需要从供货商处购入原材料并委托第三方物流公司运输材料，对于企业而言，为尽可能减小成本，需要考虑从哪家供货商购入原材料、购入多少以及选择哪家物流公司进行运输，因此制定出最经济的选购以及运输方案对于企业的经营而言至关重要。

现有某建筑和装饰板材的生产企业需要选用 A，B，C 三种原材料进行生产，企业每周的产能确定，生产每单位的产品消耗原材料的数量均不相同。在考虑供应商不能严格按订单量供货的情况下，为保证生产需要，企业要尽可能保持不少于两周生产需求的原材料库存量。并且在实际转运中，原材料会有一定的损耗。在此背景下完成以下三个问题。

问题一，根据企业过去五年 402 家供应商的订货量和供货量数据，对供货商进行评价，建立反映保障企业生产重要性的数学模型，并在此基础上选出 50 家最重要的供货商。

问题二，在问题一的基础上确定企业至少需要选择多少家供货商才能满足生产需求。针对这些供货商制定未来 24 周最经济的订购方案，对此制定损耗最小的运输方案。并对其实施效果进行分析。

问题三，在尽可能多的采购 A 类材料和尽可能少的采购 C 类材料，同时转运的损耗量尽可能少的情况小制定订购和转运方案并分析实施效果。

问题四，若企业存在提升产能的潜力，根据当前供应商和转运商的实际情况，确定企业每周提升产能的多少，并制定未来 24 周的订购和转运方案。

1.2 问题分析

针对问题一：

为对供货商进行评价，需要构造合理的指标。由于提供的数据仅有企业过去对供货商的订货量和供货量，因此指标的构造需要从供应商的信誉、供应商的能力等方面进行考虑，并根据这些指标对各个供货商进行评分，进而得出供货商的排名，选出前 50 名的供货商。

针对问题二：

通过 $0-1$ 规划计算出企业应该从中至少选择多少家供应商才能满足生产的需求，在能够满足生产需求的供应商选择方案中以最小化成本作为目标函数，通过优化算法求解，得到原材料的订购方案。并在该订购方案基础上，以最小化转运损耗为目标函数优化得到转运方案。

针对问题三：

建立以最大化 A 类原料与 C 类原料的数量差为目标函数，以订购方案为变量的目标规划，通过优化算法求解，得到调节后的订购方案。并在该订购方案基础上，以最小化转运损耗为目标函数得到转运方案。

针对问题四:

建立以最大化利润为目标函数，以订购方案为变量构建目标规划问题，令订购方案通过约束条件与提高产能的大小关联，求解得到订购方案，再根据得到的订购方案计算每周提高的产能大小以及转运方案。

二、 模型假设

为了简化模型，使得模型合理，我们采取如下合理假设:

1. 只要为企业供应过原料都算作供应商。我们认为只要为企业供应过原料的供应商，即使在某些周供应量为 0，仍然算作该企业在本周的供应商。

2. 企业老板是精明的，知道尽量用 A 原料作为仓库存储，从而减少存储成本。由于 A 原料的转换率最大，因此相同产能情况下，A 原料的体积是最小的，即存储 A 原料的存储成本是最小的。

3. 产能不能降低。由于数据有限，为使得模型较符合实际，我们不考虑可能会使企业产能降低的因素的影响，即认为产能不会降低。、

4. 每一家供货商未来的供货量仅与过去的供货量存在关系。由于给予的数据只有供货商过去的供货数量，因此我们这里不考虑其他因素的影响，认为供货量仅与过去供货量有关。

三、 符号说明

符号	意义
$N_o(i,t)$	企业第 t 周对 i 供货商的订单量
$N_s(i,t)$	i 供货商第 t 周的实际供货量
$N_d(i)$	i 供货商的违约次数
$R_d(i)$	i 供货商的违约率
$R_s(i)$	i 供货商的总供货比例
$R_c(j)$	单位 j 原料的产品转化率
$C(j)$	j 原料的单价
$Rep(i)$	i 供货商的信誉值
$Abi(i)$	i 供货商的能力值
$D(i,t)$	i 供货商第 t 周的实际供货量与每周的订单量的差值
$\bar{D}(i)$	$D(i,t)$ 的均值
$C.V(i)$	$D(i,t)$ 的变异系数
$\hat{N}_s(t)$	预测的第 t 周的供应量矩阵
R	转换矩阵
$M(M(t))$	(第 t 周的) 供应关系矩阵
$Cap(Cap(t))$	企业 (第 t 周) 的产能
$G(t)$	第 t 周仓库获得的产能
$Q(t)$	第 t 周进货前仓库存有的产能
$\hat{N}_r(i,t)$	第 t 周企业收到的 i 原料的总量
$Cost_{in}(t)$	第 t 周企业的购入成本
$Cost_s(t)$	第 t 周企业的存储成本
$Price_i$	单位体积 i 原料的价格

四、 模型建立

4.1 供货商的筛选

为了筛选出对企业最重要的 50 家供货商，需要先构造评价供货商特征的指标。由于提供的数据仅有企业过去五年对各个供货商的订单量以及各个供货商的供货量，因此我们从供货商的信誉、供货商的能力等方面进行指标的构造。

4.1.1 供货特征指标的构造

信誉指标

供货商的信誉与供货商违约次数呈负相关，与供货商完成的总供货比例呈正相关。我们定义 i 供货商的违约率 $R_d(i)$ 为

$$R_d(i) = \frac{N_d(i)}{\sum_{t=1}^{240} N_o(i,t)} \tag{1}$$

其中 $N_d(i)$ 为 i 供应商的违约次数，即出货量未达到订单量的次数，$N_o(i,t)$ 为企业第 t 周对 i 供应商的总下单次数。

定义总供货比例 $R_s(i)$ 为

$$R_s(i) = \frac{\sum_{t=1}^{240} N_s(i,t)}{\sum_{t=1}^{240} N_o(i,t)} \tag{2}$$

其中 $N_s(i,t)$ 为 i 供货商第 t 周的实际供货量。

因此，i 供货商的信誉 $Rep(i)$ 计算公式为 [1]

$$Rep(i) = \frac{\alpha(1 - R_d(i))}{1 - R_s(i) * R_d(i)} \tag{3}$$

其中 α 为参数。

能力指标

供货商的供货能力与其总实际出货量 $\sum_{t=1}^{240} N_s(i,t)$ 有关，总实际出货量 $\sum_{t=1}^{240} N_s(i,t)$ 越大则供货商的供货能力越强。考虑到不同的材料转化为产品的转化率不同，且不同供货商提供的材料种类不同，因此需要将供货商的出货量进行转化。

定义 i 材料的转化率 $R_c(i)$ 等于每单位 i 原料所能生产的产品数量，由于只存在三种材料，因此只存在三种转化率 $R_c(a) = \frac{1}{0.6} = \frac{5}{3}$、$R_c(b) = \frac{1}{0.66} = \frac{50}{33}$、$R_c(c) = \frac{1}{0.72} = \frac{25}{18}$，分别对应 A、B、C 原料。

根据题目，假定 C 原料单价 $C(c) = 1$，则 B 原料单价 $C(b) = 1.1$，A 原料单价 $C(a) = 1.2$。

观察数据发现供应商实际供应量之间差距较大，因此我们选用对数函数进行放缩。

定义 i 供应商供应能力 $Abi(i)$ 为

$$Abi(i) = ln(\frac{\sum_{t=1}^{240} N_s(i,t)}{R_c(j)^2 C_(j)} + 1)(j = a, b, c) \tag{4}$$

稳定性指标

供货商的供货稳定性与其完成订单量的程度以及完成订单量的次数有关，若供货商能够长期完成订单所需的供货量，说明该供货商的供货稳定性较好。

i 供货商第 t 周的实际供货量与每周的订单量差值 $D(i,t)$ 计算公式为

$$D(i,t) = |N_s(i,t) - N_o(i,t)| \tag{5}$$

我们选用值 $D(i,t)$ 的变异系数作为指标，i 供货商的变异系数 $C.V(i)$ 计算公式为

$$C.V(i) = \frac{\sum_{t=1}^{240}[D(i,t) - \bar{D}(i)]^2}{\sum_{t=1}^{240} D(i,t)} \tag{6}$$

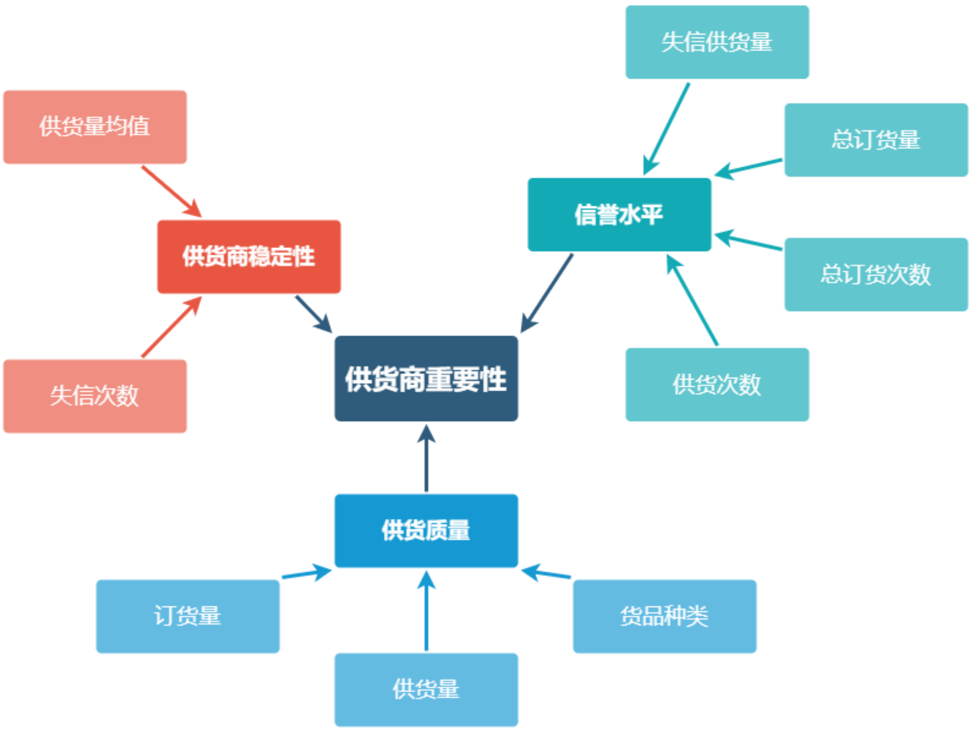

图 1: 指标示意图

4.1.2 指标权重的确定

确定指标后我们需要确定各个指标的权重，这里我们选用熵权法计算指标的权重[2]。

熵权法介绍

在信息论中，熵是对不确定性的一种度量。熵越大，说明不确定性越大，包含的信息量也越大；反之，则不确定性越小，包含的信息量也越小。

根据熵的特点，可以通过计算熵值来判断一个指标的离散程度，若指标离散程度越大，即指标的取值越多，说明该指标对最终评价的影响越大。依据熵权法计算所得的权重完全取决于数据本身，具有较好的客观性。

熵权法步骤

假设存在 n 个样本，m 个指标，x_{ij} 表示第 i 个样本的第 j 个指标的数值，其中 $i=1,2,...,n$；$j=1,2,...,m$。

由于各个指标的单位不同，取值方向也可能不同，因此需要进行标准化处理。对正向指标，即越大越好指标采取如下标准化处理

$$z_{ij}=\frac{x_{ij}-min(x_{1j},...,x_{nj})}{max(x_{1j},...,x_{nj})-min(x_{1j},...,x_{nj})} \tag{7}$$

对负向指标，即越小越好指标采取如下标准化处理

$$z_{ij}=\frac{max(x_{1j},...,x_{nj})-x_{ij}}{max(x_{1j},...,x_{nj})-min(x_{1j},...,x_{nj})} \tag{8}$$

对数据标准化处理后需要计算第 j 项指标下第 i 个样本值占该指标的比重 p_{ij}

$$p_{ij}=\frac{z_{ij}}{\sum_{i=1}^{n} z_{ij}} \tag{9}$$

其中 $i=1,...,n$；$j=1,...,m$。特别的，若 $p_{ij}=0$，则令 $p_{ij}*lnp_{ij}=0$。

得到各样本在对应指标所占全中后即可计算第 j 项指标的熵值 H_j

$$H_j=-K\sum_{i=1}^{n}[p_{ij}ln(p_{ij})] \tag{10}$$

其中 $j=1,...,m$，$k=\frac{1}{ln(n)}$。

通过计算信息熵的差异值 d_j 可以知道指标的变异程度，差异值越大说明指标的变异程度越小。

$$d_j=1-H_j \tag{11}$$

其中 $j=1,...,m$。

通过信息熵的差异值 d_j 计算各项指标的权重 ω_j

$$\omega_j=\frac{d_j}{\sum_{j=1}^{m} d_j} \tag{12}$$

最终得到各个指标的权重如下

指标名称	稳定性指标	能力指标	信誉指标
权重	0.009	0.763	0.228

图 2: 指标权重

将各个指标的数值乘以对应的权重作为供货商在该指标下的得分，记为 $z\prime_{ij}$

$$z\prime_{ij} = \omega_j * z_{ij} \tag{13}$$

为方便下文使用起见，更新后的 $z\prime_{ij}$ 仍记为 z_{ij}。

4.1.3 供应商排名的确定

得到各个指标的权重后需要根据各个指标的数据对供应商进行排名，这里我们选用优劣解距离法 (TOPSIS)[3]。

首先需要找出每个指标的最大值，记为 z_j^+，组成向量

$$Z^+ = [z_1^+, ..., z_m^+]$$

该向量表示最理想的供货商，即每个指标都是最大。同样的找出每个指标的最小值，记为 z_j^-，组成向量

$$Z^- = [z_1^-, ..., z_m^-]$$

该向量表示最不理想的供货商，即每个指标都是最小。

定义第 i 个供货商与最理想目标的距离为 d_i^+

$$d_i^+ = \sqrt{\sum_{j=1}^{m} (z_j^+ - z_{ij})^2} \tag{14}$$

同理，定义与最不理想目标的距离为 d_i^-

$$d_i^- = \sqrt{\sum_{j=1}^{m} (z_j^- - z_{ij})^2} \tag{15}$$

根据供货商与最理想、最不理想目标的距离计算供货商在该指标的得分 S_i

$$S_i = \frac{d_i^-}{d_i^+ + d_i^-} \tag{16}$$

显然 S_i 越接近 1，说明该供应商距离最理想目标越近，则该企业选择该供应商越可

靠。反之，说明该供应商距离最不理想的目标越近，则该企业不应该选择该供应商。

4.2 订购方案与转运方案的制定

要计算企业至少需要选择多少家供应商供应的原材料才能满足生产需求，需要先对供应商未来可能的供应量进行估计。考虑到一部分供货商为企业长期供货，存在一定的时间规律，适用 ARIMA 模型进行预测。

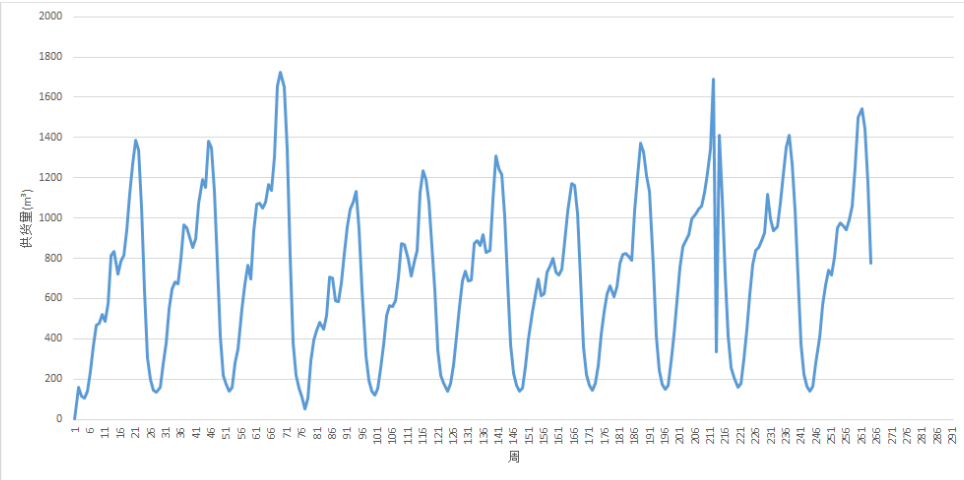

图 3: 存在周期性的数据展示

而部分供货商在相当长时间内的供货量为 0，仅在相当小一段时间为企业供货，这部分数据不适用 ARIMA 模型，因此针对这部分数据我们使用无规律供货量预测模型进行预测。

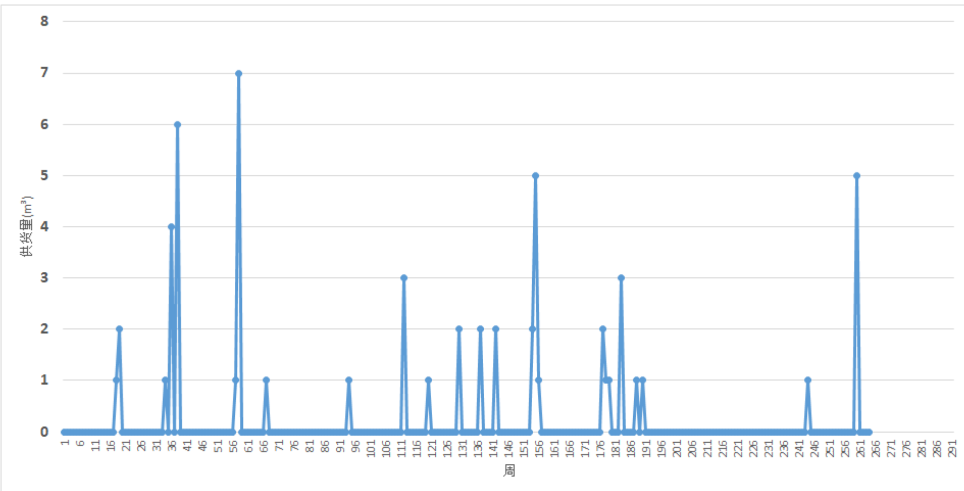

图 4: 不存在周期性的数据展示

4.2.1 ARIMA 模型的建立

时间序列的平稳性检验

常用的平稳性检验方法是单位根检验 (ADF)，原假设为时间序列具有单位根，通过检验过去供货商的供货量数据的平稳性，判断在一定置信水平上是否显著拒绝原假设。ADF 是通过以下三个模型实现的：

$$\begin{aligned}
\Delta N_s(i,t) &= \delta N_s(i,t-1) + \sum_{i=1}^{m} \beta_i \Delta N_s(i,t-i) + \epsilon_t \\
\Delta N_s(i,t) &= \alpha + \delta N_s(i,t-1) + \sum_{i=1}^{m} \beta_i \Delta N_s(i,t-i) + \epsilon_t \\
\Delta N_s(i,t) &= \alpha + \beta t + \delta N_s(i,t-1) + \sum_{i=1}^{m} \beta_i \Delta N_s(i,t-i) + \epsilon_t
\end{aligned} \tag{17}$$

原假设 H_0 与备择假设 H_1 如下

$$\begin{aligned}
H_0 &: \delta = 0 \\
H_1 &: \delta < 0
\end{aligned} \tag{18}$$

其中 t 是时间变量。进行假设检验时只需上述三个模型中存在一个拒绝了原假设，则认为时间序列是平稳的。

对时间序列进行差分处理：

由于供货商实际的供货量时间序列是不平稳的，且部分可能存在较明显的季节性和周期性，而不平稳的时间序列不适用与 ARIMA 模型，因此我们需要对这部分不平稳的时间序列进行差分处理，直到得到一个平稳的时间序列，同时需要记录差分的阶数 d 作为 ARIMA 模型的参数之一。差分过程如下

$$\triangledown_d N_s(i,t) = N_s(i,t) - N_s(i,t-d) \tag{19}$$

其中 d 为差分的阶数。

若在第一步对供货商的供货量序列进行 ADF 检验后发现该序列不是平稳的，则需要先对序列进行差分过程，再进行单位根检验，若发现序列仍不是平稳的，则继续差分过程直至序列变为平稳的。

ARIMA 模型参数的确定

为确定 ARIMA 模型的 p 参数 (自回归项数) 和 q 参数 (滑动平均项数)，需要检查平稳时间序列的自相关图和偏自相关图。其中自相关函数为

$$r_k(i) = \frac{\sum_{t=1}^{n-k}(N_s(i,t) - \bar{N}_s(i))(N_s(i,t+k) - \bar{N}_s(i))}{\sum_{t=1}^{n}(N_s(i,t) - \bar{N}_s(i))^2} \tag{20}$$

其中 $k = 1, 2, ...$。

偏自相关函数为

$$p_{k,k}(i) = \begin{cases} r_1(i), & k=1 \\ \frac{r_k(i)-\sum_{j=1}^{k-1} p_{k-1,j}(i) r_{k-j}(i)}{1-\sum_{j=1}^{k-1} p_{k-1,j}(i) r_j(i)}, & k=2,3,... \end{cases} \tag{21}$$

其中

$$p_{k,j}(i) = p_{k-1,j}(i) - p_{k,k}(i) p_{k-1,k-j}(i) (j=1,2,...,k-1)$$

为精确的确定 p 和 q 的具体数值，我们采用准则函数定阶法。

⋆AIC 准则:

$$AIC = 2k - 2ln(L) \tag{22}$$

⋆BIC 准则:

$$BIC = kln(n) - 2ln(L) \tag{23}$$

⋆HQIC 准则:

$$HQIC = kln(ln(n)) - 2ln(L) \tag{24}$$

其中 k 为模型参数的个数，L 为似然函数，n 为供货商数量。

通过构造上述统计量使得拟合残差尽可能小, 依照上述准则计算所得统计量取最小值时即为最佳 p 值和 q 值。由此，可得与时间序列匹配的模型即为 ARIMA(p,d,q) 模型。

4.2.2 无规律供货量预测模型的建立

由于 0 观测值较多的这部分数据不存在周期性以及季节性规律，即随机性较大，因此我们将这部分供货商过去 240 周的供货量对应于一段均分为 240 份的 [0,1] 区间。通过产生 [0,1] 的随机数，决定下一周该供货商的供货数量。

将 [0,1] 区间分为 240 等分，即

$$[L_k, H_k](k=1,2,...,240)$$

其中 $L_{k+1} = H_k$，$L_1 = 0$，$H_240 = 1$，$H_k - L_k = H_{k+1} - L_{k+1}$。

随机产生 [0,1] 的数 λ，若 $L_k < \lambda < H_k$，则认为该供货商这周为企业提供过去第 k 周的供货量 $N_s(i,k)$。

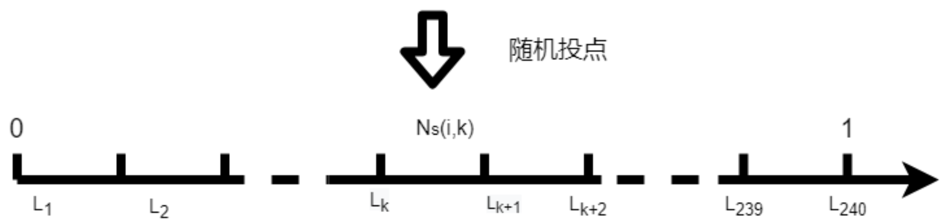

图 5: 无规律供货量预测模型示意图

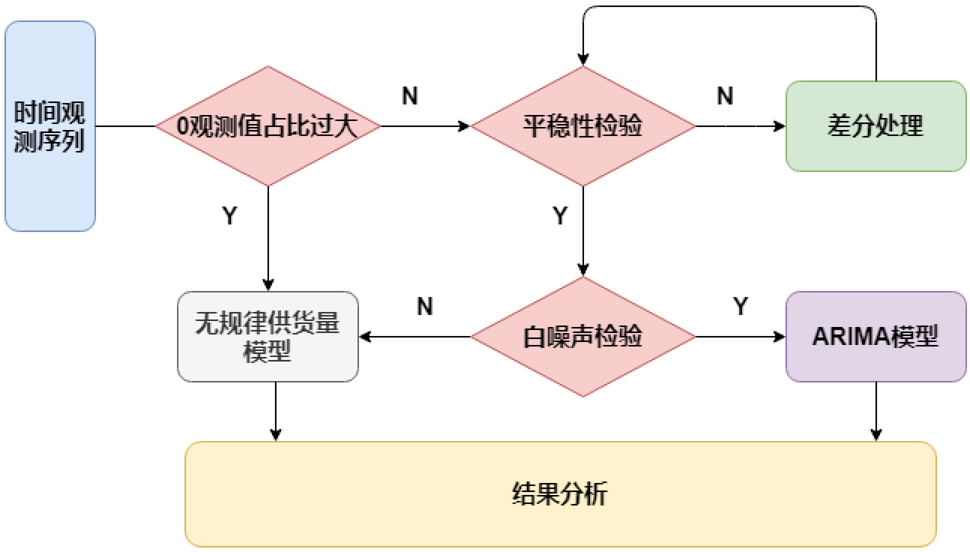

图 6: 预测流程图

4.2.3 最少供应商的确定

通过上述两个模型预测各个供货商未来 24 周的可能供货量，建立以最小化供货商数量为目标函数，以满足生产需求为条件的目标规划问题，求解该目标规划即可得出需要的最少供应商。

构造供应量矩阵 $\hat{N}_s(t)$

通过上述构造的两个模型预测各个供货商未来 24 周的供货量 $\hat{N}_s(i,t)$，其中 $t = 1, 2, ..., 24$，将其组成每周的供应量矩阵 $\hat{N}_s(t)$

$$\hat{N}_s(t) = \left[\hat{N}_s(1,t) \ldots \quad \hat{N}_s(402,t)\right]_{1\times 402}$$

构造转换矩阵 R

由于供货商生产的原料不同，不同原料与产品的转化率不同，因此我们需要构造各个供货商生产的原料对成品的转化率矩阵 R。

先挑选出生产 A 类原料的供货商的下标，将其置于集合 Set_a 中，同理，Set_b 中包含所有生产 B 类原料的供货商的下标，Set_c 中包含所有生产 C 类原料的供货商的下标。定义矩阵 R

$$r_i = \begin{cases} R_c(a), & \text{如果} i \in Set_a \\ R_c(b), & \text{如果} i \in Set_b \\ R_c(c), & \text{如果} i \in Set_c \end{cases} \tag{25}$$

转化率矩阵 R 是一个 1×402 的矩阵。

构造供应关系矩阵 M

每周为企业供应原料的供货商是一定的，即不是所有的供货商都为企业供货。将为企业供货的供货商的下标记录在供应关系矩阵 M 中

$$m_i = \begin{cases} 0, & \text{如果 } i \text{ 供货商没有为企业供过任何货物} \\ 1, & \text{如果 } i \text{ 供货商为企业供应过货物} \end{cases} \tag{26}$$

供应关系矩阵 M 是一个 402×1 的 0——1 矩阵。

目标规划函数的构建

根据上述构造的矩阵，我们可以写出企业要在未来 24 周满足生产需求所需的最少加供应商的目标规划如下

$$\begin{aligned} \min \quad & z = \sum_{i=1}^{402} m_i \\ \text{s.t.} \quad & G(t) = (R \bigotimes \hat{N}_s(t))M \\ & Q(t+1) = Q(t) + G(t) - Cap \\ & 5.64 \leq Q(t) + G(t) - Cap \end{aligned} \tag{27}$$

其中 m_i 是矩阵 M 的第 i 个元素，$G(t)$ 表示第 t 周仓库获得的产能，即本周购入的货物所能生产的产品数量 (下同)，$Q(t)$ 表示第 t 周进货前仓库内所存原料的产能，Cap 表示企业每周的产能，即 $Cap = 2.82$ 万立方米，运算 $\bigotimes$ 表示两个矩阵对应元素相乘。

求解该目标规划问题得出 M 矩阵，计算 M 矩阵所有元素之和即可得可保证生产需求的最少供货商数量。

4.2.4 最经济订购方案的制定

在上述模型中求出的供应关系矩阵 M 的基础上，我们需要为企业制定未来 24 周每周最经济的原料订购方案。通过构造新的目标规划并求解得到最经济的原料订购方案。值得注意的是，由于企业存在仓库，因此我们认为企业从供货商收到的货物将直接运至仓库，再由仓库送给企业生产基地进行生产。

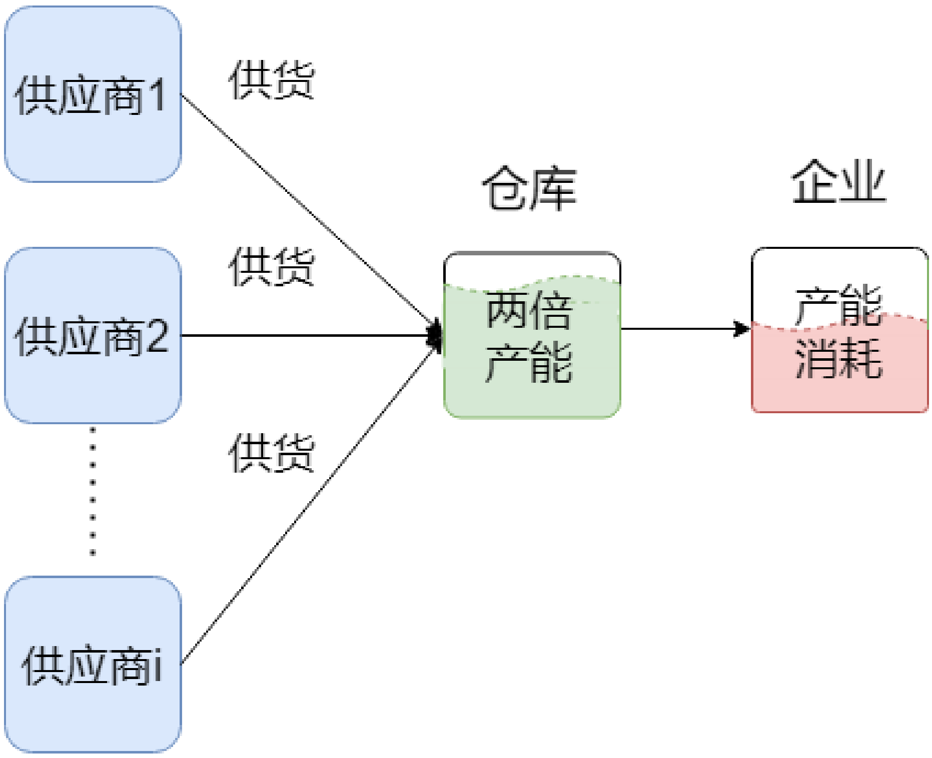

图 7: 情景示意图

进货成本

企业每周向供货商购入的原料需要成本，我们把这部分成本称为进货成本 $Cost_{in}(t)$

$$Cost_{in}(t) = Price_a N_r(a,t) + Price_b N_r(b,t) + Price_c N_r(c,t) \tag{28}$$

其中 $N_r(i,t)$ 表示第 t 周企业受到的 i 原料的总量，$Price_i$ 表示单位体积 i 原料的价格。

$$\begin{aligned} N_r(a,t) &= \hat{N}_s(t)A \\ N_r(b,t) &= \hat{N}_s(t)B \\ N_r(c,t) &= \hat{N}_s(t)C \end{aligned} \tag{29}$$

其中 A 是记录生产 A 原料的供货商的 402×1 的矩阵，

$$a_i = \begin{cases} 0, & i \notin Set_a \\ 1, & i \in Set_a \end{cases} \tag{30}$$

矩阵 B、C 同理。

存储成本

当企业购入过多原料，当周无法消耗完则会产生存储成本 $Cost_s(t)$。我们假设老板

是精明的，会尽量用 A 原料作为储备产能，在对企业过去五年的购入数据计算发现，确实可以只用 A 原料作为储备产能，因此我们认为仓库内的储备产能全为 A 原料。

因此存储成本 $Cost_s(t)$ 为

$$Cost_s(t) = Q(t)R_c(a) \tag{31}$$

其中 $Q(t)$ 表示第 t 周进货前仓库的存储产能，$R_c(a)$ 表示单位体积的 A 原料可生产的产品数量。

$Q(t)$ 满足以下关系

$$\begin{gathered} Q(t) + G(t) - Cap = Q(t+1) \\ Q(t) \geq 0 \end{gathered} \tag{32}$$

其中 $G(t)$ 表示第 t 周仓库获得的产能。

储备产能要求

由于需要尽量保证仓库的储备产能不少于两周的产能，因此我们构建了仓库储备产能 $Q(t)$ 的函数 $f(Q(t))$ 以保证仓库的弹性。

$$f(Q(t)) = \alpha|2Cap - Q(t)| \tag{33}$$

其中 α 为参数。

若 $Q(t)$ 距离企业两周产能越大，则 $f(Q(t))$ 越大，因此通过最小化函数 $f(Q(t))$ 可以有效调节仓库储备产能 $Q(t)$ 在企业两周产能附近浮动。

多目标规划的建立

第一目标为最小化成本函数 $Cost(t)$，成本函数包含两个部分，即进货成本 $Cost_{in}(t)$ 和存储成本 $Cost_s(t)$

$$Cost(t) = Cost_{in}(t) + Cost_s(t) \tag{34}$$

第二目标为保证仓库储备产能 $Q(t)$ 为两周企业的产能，通过最小化 $f(Q(t))$ 函数可以实现。

因此目标规划如下

$$\min \quad z=\sum_{t=1}^{24}(p_1Cost(t)+p_2f(Q(t)))$$

$$\begin{aligned}
\text{s.t.}\quad G(t)&=[R\bigotimes\hat{N}_s(t)]M(t)\\
Q(t+1)&=Q(t)+G(t)-Cap\\
m_i(t)&\le m_i\\
Q(t)&\ge 0\\
G(t)&\ge 0
\end{aligned}\tag{35}$$

其中 $M(t)$ 表示第 t 周的供应关系矩阵，$m_i(t)$ 为 $M(t)$ 的元素，m_i 为 M 的元素。

求解该线性规划得出的一组供应关系矩阵 $M(t)$ 即为最佳订购方案。

4.2.5 最小损耗转运方案的制定

损耗矩阵的构造

由于每家转运商每周的转运损耗率均不相同，因此我们首先需要对每家转运商未来 24 周每周的转运损耗率进行预测。观察过去五年转运商的转运损耗率数据发现，存在部分转运商较多观测值为 0，部分转运商的转运损耗率呈周期性变化。因此可以效仿前面对供应商供货量预测的方法，对呈周期性变化的转运商使用时间序列模型进行预测，对观测值较多为 0 的转运商使用无规律供货量预测模型进行预测。模型构建过程不再赘述。

将通过预测模型得到的未来 24 周所有转运商的转运损耗率组成一个 24×8 的损耗矩阵 L，则 $L_{i,j}$ 表示第 j 家转运商在第 i 周的转运损耗率，用 L_i 表示矩阵 L 的第 i 行。

转运数量矩阵的构造

将第 t 周各转运商为各供货商转运商品的数量作为元素，组成一个 8×402 的转运数量矩阵 $Tr(t)$，则 $Tr_{i,j}(t)$ 表示第 t 周，第 i 家转运商为第 j 家供货商转运商品的数量。

目标规划的构建

由于不同原料的单价以及效用不相同，因此需要统一单位再进行运算。这里我们将

其转化为对应产品的产量进行计算，则目标规划如下

$$\min \quad z = \frac{L_A}{0.6} + \frac{L_B}{0.66} + \frac{L_C}{0.72}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j=1}^{402} Tr_{i,j}(t) \le 6000(i = 1, 2, ..., 8) \\ & L_A = \sum_{t=1}^{24} L_t Tr(t)(A \cap M(t)) \\ & L_B = \sum_{t=1}^{24} L_t Tr(t)(B \cap M(t)) \\ & L_C = \sum_{t=1}^{24} L_t Tr(t)(C \cap M(t)) \end{aligned} \tag{36}$$

其中 $\cap$ 表示矩阵对应位置元素进行且的逻辑运算，A 是记录生产 A 原料的供货商的 402×1 的矩阵30，B、C 同理。

求解该目标规划问题得到每周的转运数量矩阵 $Tr(t)$ 即可知最佳转运方案。

4.3 原料购入原则调整后的订购及转运方案的确定

企业为了压缩成本，计划多采购 A 类原料，少采购 C 类原料从而减少转运以及仓储的成本，因此需要重新制定订购方案和转运方案。

新订购方案的制定

新的目标规划与上述构造过程类似，具体构造如下

$$\min \quad z = \sum_{t=1}^{24} (\hat{N}_r(a,t) - \hat{N}_r(c,t))$$

$$\begin{aligned} \text{s.t.} \quad & Q(t+1) = Q(t) + G_A(t) + G_B(t) + G_c(t) - Cap \\ & Q(t) \ge 2Cap \\ & G_A(t) = (R \bigotimes \hat{N}_s(t)) M_A(t) \\ & G_B(t) = (R \bigotimes \hat{N}_s(t)) M_B(t) \\ & G_C(t) = (R \bigotimes \hat{N}_s(t)) M_C(t) \\ & M(t) = M_A(t) + M_B(t) + M_C(t) \end{aligned} \tag{37}$$

其中 $\hat{N}_r(i,t)$ 表示第 t 周企业受到的 i 原料的数量，计算公式为 $\hat{N}_r(i,t) =$，$G_i(t)$ 表示第

t 周购入的 i 原料对应的产能，$M_i(t)$ 表示第 t 周 i 原料的供应关系矩阵。

求解该目标规划问题得到 $M_A(t)$、$M_B(t)$、$M_C(t)$ 矩阵即可知最佳订购方案。

求解转运方案的目标规划与上述运输方案的目标规划类似，此处不再赘述。

4.4 产能提高的规划

由于该企业通过技术改造已具备提高产能的潜力，为提高经济效益，需要根据现有原材料的供应商和转运商的实际情况制定合理的提高产能的策略。这里我们假设产能是不能降低的，即要么提高要么维持现状。

我们从经济效益的角度出发设计目标函数，即以最大化利润作为目标函数，调整企业的订购方案，进而得出企业每周产能提高多少的规划。目标规划如下

$$\max \quad z = \lambda Cap(t) - Cost_{in}(t) + Cost_s(t)$$

$$\begin{aligned} \text{s.t.} \quad Cost_{in}(t) &= \sum_{i=a,b,c} Price_i \hat{N}_r(i,t) \\ Cost_s(t) &= \frac{c_1 Q(t)}{R_c(a)} \\ h &= Cap(t+1) - Cap(t) \\ Q(t+1) &= Q(t) + \sum_{i=a,b,c} R_c(i)\hat{N}_r(i,t) - Cap(t) \\ Q(t+1) &\geq 2Cap(t) \end{aligned} \tag{38}$$

其中 $Cap(t)$ 表示第 t 周企业的产能，$h(t)$ 表示第 t 周到第 t+1 周产能提升的大小，$\hat{N}_r(i,t)$ 表示第 t 周企业受到的 i 类原料的数量。

求解该目标规划得出一组 $M(t)$，即为未来 24 周的最佳订购方案，结合约束条件的可得出企业在每一周产能可提升的范围。

五、 模型求解

5.1 供货商评分

根据上述构造的指标评分以及计算所得权重，最终得到 402 家供应商的得分，由于篇幅有限，此处仅展示排名前十的供应商的得分，排名前 50 家的供应商得分置于附录。

供应商 ID	材料分类	综合得分
S229	A	0.9913
S361	C	0.9743
S108	B	0.9462
S282	A	0.9333
S275	A	0.9274
S329	A	0.9262
S340	B	0.9249
S140	B	0.9191
S139	B	0.9079
S131	B	0.9048

图 8: 前十排名的供应商及其评分

供应商 ID	总供应量	总订货量	总订货次数	失信次数
S229	354887	359885	240	13
S361	328080	333452	240	29
S108	240950	271445	240	13
S282	169340	168531	240	3
S275	158553	158150	240	13
S329	156518	156350	240	14
S340	171426	172000	240	14
S140	302047	481103	220	25
S139	151862	177748	224	7
S131	137512	139381	240	22

图 9: 排名前十供应商的部分特征

结合过去五年供应商的供应量数据，我们发现排名靠前的供应商过去五年的总供货量都在 100000 以上，订单完成度都比较好，并且失信次数占总订货次数的比例较小，说明我们的评分准则是合理的。

5.2 供货预测的求解

时间序列模型我们使用 SPSS 软件进行操作。无规律供货预测模型我们使用 MATLAB 编写代码，在附录进行展示。最终预测数据详见附件。

5.3 最少供应商的确定以及最经济方案的制定的求解

为了求解上述构造的目标规划问题，我们选用模拟退火算法进行优化求解，优化算法需要初始解，因此我们需要寻找目标规划问题的一个初始解。在前面我们根据供应商的供货特征构造了指标并对其进行了评分排名，这里我们选取评分排名前 50 名的供货

商作为初始解，即初始供应关系矩阵 M 中，下标为这些商家的元素取值为 1，其余均为 0。

$$m_i = \begin{cases} 1, \text{如果 i 供货商排名为前 50} \\ 0, \text{如果 i 供货商排名不为前 50} \end{cases} \tag{39}$$

将其代入算法进行计算，最终得到至少需要 276 家供货商才能满足企业的生产需求。

对最经济方案的目标规划求解，根据方案制作出如下每周购入原料对应的产能图

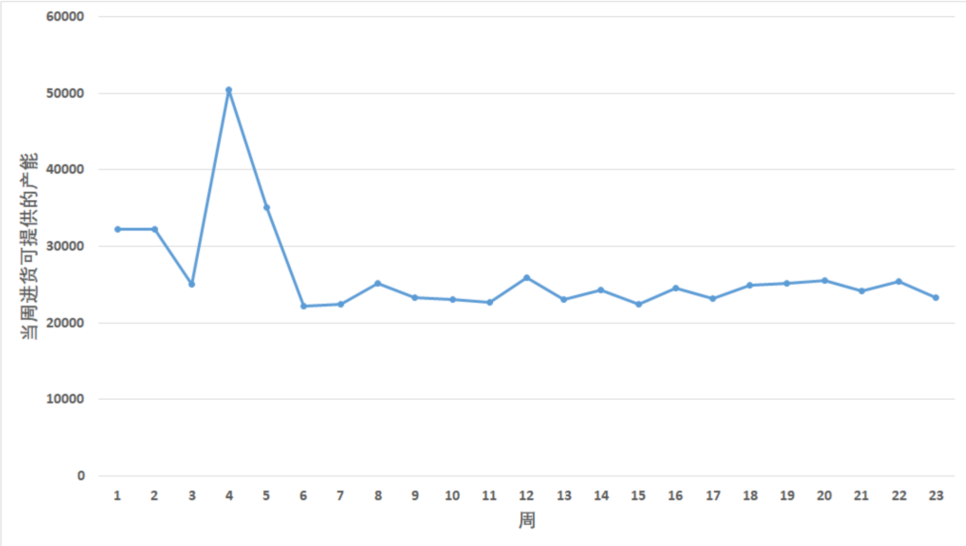

图 10: 每周购入原料对应的产能

观察发现，前期购入较多原料，存储于仓库，以保证后期每周消耗后，仓库中的储备产能大于企业两周的的产能，因为后期供应商的供应总量不足以满足每周的生产需求。由于前期购入较多，所以后期购入的原料对应的产能较稳定，略低于企业每周产能，以此消耗仓库内多余的产能，减小存储成本。

5.4 转运方案的求解

考虑到 A 类原料的采购单价较高，且单位转化率较大，因此应当优先选择委托最小损耗的商家进行运输，B 类原料次之，这样总可以保证转运损耗的产能是最小的。因此，我们针对所有的转运方案的求解都采用上述原则，即让预测当周损耗率从小到大排名靠前的转运商优先转运 A 原料，B 原料次之，C 原料最后。

5.5 原料购入原则调整后的订购方案的求解

为求解这个目标规划问题，我们选择模拟退火算法进行求解 [4]。由于企业计划尽可能多的采购 A 类原料，尽可能少的采购 C 类原料，因此我们选择优先订购所有生产

A 原料的供应商，若所订购的 A 原料不足以支持当周的生产需求，再向生产 B 类原料的供应商下单，直到满足当周的生产需求的订购方案作为初始解。

即初始供应关系矩阵 $M(t)$ 中，下标为生产 A 类原料的商家的元素取值为 1，其余均为 0，若 $\hat{N}_s(t)M(t) < Cap$ 则从下标为生产 B 类原料的商家的元素取值为 1，直到 $\hat{N}_s(t)M(t) \geq Cap$。

将其带入算法进行计算，最终发现初始解即为最优解，详情请见附件 A。

5.6 产能提高规划的求解

为求解这个目标规划问题，我们仍然选择模拟退火算法进行求解。初始供应关系矩阵 M 选取排名前 50 名的供应商对应的元素为 1，其余为 0，经过计算得到结果，详情请见附件。

通过对目标规划中的约束条件进行变换，可以得到如下产能提高量 $h(t)$ 与供应关系矩阵 $M(t)$ 的关系

$$Q(t-1) - Cap(t-1) + \sum_{i=a,b,c} R_c(i)\hat{N}_r(i,t)M_i(t) \geq h(t) \tag{40}$$

通过上述关系进行迭代可得每周的产能提升值。由于算法限制，这里我们认为每周的产能提升是相同的，即 $h(t) = h$，通过上述迭代关系最终求得 $h = 1108.94$。

六、 模型评价

6.1 模型优缺点

优点

1. 构造指标时综合考虑了供货商的信誉、能力以及稳定性等方面，构造的指标可以较全面的反映供货商的供货特征。

2. 对时间观测序列进行分批预测，针对存在季节性、周期性变化的时间观测序列，我们采用 ARIMA 模型进行预测，对于无显著周期性且 0 值占比较大的时间观测数据，我们采用概率投点进行模拟，均取得较为合理的结果。

缺点

1. 本文的大多数目标规划涉及维度较大的 0—1 规划，在实际求解过程中使用智能优化算法——模拟退火算法，无法每次都求得全局最优解，因此不能取得具有一般性的结果。

2. 由于算法限制，我们没有考虑每周的产能提升可能存在差异，而是认为产能的提升是稳定的，与实际情况可能存在出入。

6.2 模型改进

本次建模中，我们设计的目标规划较为合理，但是由于编程能力的不足，不能很好的求解这些目标规划，得出的结果可能陷入局部最优解，从而影响最终的效果，因此寻求更好的算法求解本文的目标规划是我们认为最主要的改进方向。

参考文献

[1] 李向朋. 城市交通拥堵对策—封闭型小区交通开放研究 [D]. 长沙理工大学,2014.

[2] 杨力, 刘程程, 宋利, 盛武. 基于熵权法的煤矿应急救援能力评价 [J]. 中国软科学,2013(11):185-192.

[3] 雷勋平, 邱广华. 基于熵权 TOPSIS 模型的区域资源环境承载力评价实证研究 [J]. 环境科学学报,2016,36(01):314-323.

[4] 谢云. 用模拟退火算法并行求解整数规划问题 [J]. 高技术通讯,1991,1(10):21-26.

[5] 司守奎. 数学建模算法与程序 [M]. 海军航空工程学院，2007.

表 1: 排名前 50 的供应商及其评分

供应商 ID	材料分类	综合得分
S229	A	0.9913
S361	C	0.9743
S108	B	0.9462
S282	A	0.9333
S275	A	0.9274
S329	A	0.9262
S340	B	0.9249
S140	B	0.9191
S139	B	0.9079
S131	B	0.9048
S151	C	0.9026
S308	B	0.8960
S330	B	0.8952
S268	C	0.8935
S356	C	0.8932
S306	C	0.8909
S352	A	0.8760
S194	C	0.8716
S143	A	0.8643
S307	A	0.8611
S348	A	0.8570
S395	A	0.8476
S247	C	0.8214

供应商 ID	材料分类	综合得分
S374	C	0.8095
S284	C	0.8050
S037	C	0.8046
S031	B	0.8007
S365	C	0.7951
S126	C	0.7828
S040	B	0.7771
S338	B	0.7716
S364	B	0.7695
S367	B	0.7593
S055	B	0.7486
S346	B	0.7479
S201	A	0.7386
S080	C	0.7291
S294	C	0.7281
S086	C	0.7127
S218	C	0.7125
S244	C	0.7125
S003	C	0.6924
S114	A	0.6882
S189	A	0.6787
S273	A	0.6742
S210	C	0.6679
S074	C	0.6644
S005	A	0.6630
S078	A	0.6619
S291	A	0.6596

```
clc
clear
load indict2 .mat
x = S2;
ind = [1 2 1];
[n,m]=size(x); % n个样本, m个指标
%% 数据的归一化处理
for i=1:m
if ind(i)==1 %正向指标归一化
X(:,i)=guiyi(x(:,i),1,0.002,0.996);    %若归一化到[0,1], 0会出问题
else %负向指标归一化
X(:,i)=guiyi(x(:,i),2,0.002,0.996);
end
end
%% 计算第j个指标下，第i个样本占该指标的比重p(i,j)
for i=1:n
for j=1:m
p(i,j)=X(i,j)/sum(X(:,j));
end
end
%% 计算第j个指标的熵值e(j)
k=1/log(n);
for j=1:m
e(j)=-k*sum(p(:,j).*log(p(:,j)));
end
d=ones(1,m)-e; %计算信息熵冗余度
w=d./sum(d); %求权值w

S = topsis(X,w); %topsis计算最终得分

```

entropy.m

```
function y=guiyi(x,type,ymin,ymax)
%实现正向或负向指标归一化，返回归一化后的数据矩阵
%x为原始数据矩阵, 一行代表一个样本, 每列对应一个指标
%type设定正向指标1,负向指标2
%ymin,ymax为归一化的区间端点
[n,m]=size(x);
y=zeros(n,m);
xmin=min(x);
xmax=max(x);
switch type
case 1
for j=1:m
y(:,j)=(ymax-ymin)*(x(:,j)-xmin(j))/(xmax(j)-xmin(j))+ymin;
end
case 2
for j=1:m
y(:,j)=(ymax-ymin)*(xmax(j)-x(:,j))/(xmax(j)-xmin(j))+ymin;
end
end

```

guiyi.m

```
function S = topsis (Z,weigh)

%% 第三步：对正向化后的矩阵进行标准化
[n,m] = size(Z);
% Z = X ./ repmat(sum(X.*X) .^ 0.5, n, 1);

%% 第四步：计算与最大值的距离和最小值的距离，并算出得分
D_P = sum([(Z − repmat(max(Z),n,1)) .^ 2 ] .* repmat(weigh,n,1) ,2) .^ 0.5;   % D+ 与最大值的距离向量
D_N = sum([(Z − repmat(min(Z),n,1)) .^ 2 ] .* repmat(weigh,n,1) ,2) .^ 0.5;   % D− 与最小值的距离向量
S = D_N ./ (D_P+D_N);   % 未归一化的得分
% disp('最后的得分为：')
% stand_S = S / sum(S)
% [sorted_S,index] = sort(stand_S ,'descend')
end
```

topsis.m

```
%% 无规律供货量预测
clc;
clear;
load pre_data.mat
data = aaa; %348个需要预测的供货商数据240行，348列;
data2 = bbb; %4个需要预测的转运商数据240行，4列;
predict_X = zeros(24, size(data,2));
X = data;

for i = 1:size(data,2)
std1 = std(X(:,i)',1);
for j =1:24
rd = ceil(240*rand);
if X(rd,i)~=0
predict_X(j,i) = X(rd,i)+fix(abs(normrnd(0,std1/10)));
else
predict_X(j,i) = X(rd,i);
end
end
end
```

pos_predict.m

以下为问题二模拟退火算法的代码

```
clc
clear
load youhua1.mat;
%% 计算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

temp = ori_data';
for i = 1:size(temp,1)
```

```
temp(i ,:) =temp(i ,:) ./ class ';
end

temp = sum(temp,2);
temp2 = zeros (240,1) ; %存放5年每一周的的存货量

for i = 2:240
if temp(i)+temp2(i-1)<28200
temp2(i)=0;
else
temp2(i) = temp(i)+temp2(i-1)-28200;
end
end

cunhuo = temp2(end);

time = 1;
result = zeros(time,402+1);
for i=1:time
[ steal_best , best_value ] = SA_Backpack(data,cunhuo,class);      % 调用模拟退火算法函数
r=[ steal_best , best_value ];
result ( i ,:) =r;
end

fprintf ('最少供货商数量: ')
disp(min( result (:, end)));

data = data ';
for i = 1:24
data( i ,:) =data( i ,:) ./ class ';
end

```

main.m

```
function [ steal_best , best_value ] = SA_Backpack(data,cunhuo,class)

% value=-value;
Initial_temp = 1000;          % 模拟退火算法的初始温度
res = 1;                      % 模拟退火算法温度的最低限制
ratio = 0.98;                 % 降温参数 T(i+1)=T(i)*ration
temperature = Initial_temp ;
Markov_length = 100;          % 马尔科夫长度，控制自变量改变次数
g = [229 361 108 282 275 329 340 140 139 131 151 308 330 268 356 306 352 194 143 307 348 395 247 374 284 37 31 365 126 40
338 364 367 55 346 201 80 294 86 218 244 3 114 189 273 210 74 5 78 291 7 266 123 292 154 208 150 314 129 23 221 239 324 98
 75 92 175];
steal_new = zeros (1,402) ;
steal_new(g)=1;               %初始解为50家最重要的商家
Energy_current = inf ;                  % 初始化局部最优解
Energy_best = inf ;                     % 初始化全局最优解
steal_current = steal_new;
steal_best = steal_new;

while temperature > res
for i=1:Markov_length
```

```
% 单一位置的取反操作
 if  rand>0.5
 position  = randperm(size(data,1),2);
steal_new(position) = ~steal_new(position);

% 交换两位置的编码值
 else
 position=randperm(size(data,1),2);
 position=sort(position);
a=position(1);
b=position(2);
temp=steal_new(a);
steal_new(a) = steal_new(b);
steal_new(b)=temp;
end

% 限制条件
% 对于每一个编码结果, 考虑其是否满足条件
while 1
[q, iter]=  tiaojian(steal_new, data, cunhuo, class);
 if ~q
index=find(steal_new==0);
 if rand() >0.5
 position  = ceil(rand()*length(index));
steal_new(index(position)) = 1;
 else
steal_new(index(end)) = 1;
end
 else
break
end

end
Energy_new=sum(sum(steal_new));    % 计算满足重量限制的物品价值

% 利用Metropolis准则对解进行更新
% 更新局部最优解
 if Energy_new<Energy_current
Energy_current=Energy_new;
 steal_current =steal_new;

% 更新全局最优解
 if Energy_new<Energy_best
Energy_best=Energy_new;
 steal_best =steal_new;
end

 else
 if rand<(-(Energy_new-Energy_current)/temperature)
Energy_current=Energy_new;
 steal_current =steal_new;
 else
steal_new= steal_current ;% 否则, 保持解的编码不变, 到下一次循环中改变解的编码
end
end
```

```
    end

    temperature=temperature* ratio ;     % 降温操作
    end

    best_value=Energy_best;

```

SA_Backpack.m

```
function  [loap, iter ] =  tiaojian (x, data ,cunhuo, class )
data  =  data ';
for  i  =  1:24
data( i ,:) =data( i ,:)  ./ class ';
end

gt  =  data*x';

temp2 =zeros(25,1) ;
temp2(1)  =  cunhuo;
loap=1;
for  i  =  1:24
if  gt( i )+temp2(i)−28200<2*28200
loap  =  0;
 iter  =  i ;
break
else
temp2(i+1)  =  temp2(i)+gt( i )−28200;
end
end
 iter  =  −1;
end

```

tiaojian.m

```
clc ;
clear ;
load  best_decision .mat

class  =  class '.* choice ;
%3种材料及其索引
index_A = find( class==0.6);
index_B = find( class==0.66);
index_C = find( class==0.72);

loap3  =[];
for  i  =  1: size (new_data,2)
temp = choice ;
sum1 = zeros (8,1) ;
loap  =  zeros (402,8) ;
go = A(:, i ) ;
go(go==0)=6; %0的表示不转运，参与排序给一个极大值
[B,index]=  sort (go);
for  j  =  1:8
```

```
for k = 1:size(index_A,2)
if sum1(j) >=6000
break;
end
if (sum1(j) + new_data(index_A(k),i)<=6000)&&(temp(index_A(k))==1)
sum1(j) =sum1(j) + new_data(index_A(k),i);
temp(index_A(k))=0;
loap(index_A(k),j)=new_data(index_A(k),i);
new_data(index_A(k),i) = 0;
elseif (sum1(j) + new_data(index_A(k),i)>6000)&&(temp(index_A(k))==1)
loap(index_A(k),j)=6000-sum1(j);
new_data(index_A(k),i) = new_data(index_A(k),i)-(6000-sum1(j));
sum1(j) = 6000;
end
end

end
for j = 1:8
for k = 1:size(index_B,2)
if sum1(j) >=6000
break;
end
if (sum1(j) + new_data(index_B(k),i)<=6000)&&(temp(index_B(k))==1)
sum1(j) =sum1(j) + new_data(index_B(k),i);
temp(index_B(k))=0;
loap(index_B(k),j)=new_data(index_B(k),i);
new_data(index_B(k),i) = 0;
elseif (sum1(j) + new_data(index_B(k),i)>6000)&&(temp(index_B(k))==1)

loap(index_B(k),j)=6000-sum1(j);
new_data(index_B(k),i) = new_data(index_B(k),i)-(6000-sum1(j));
sum1(j) = 6000;
end
end

end
for j = 1:8
for k = 1:size(index_C,2)
if sum1(j) >=6000
break;
end
if (sum1(j) + new_data(index_C(k),i)<=6000)&&(temp(index_C(k))==1)
sum1(j) =sum1(j) + new_data(index_C(k),i);
temp(index_C(k))=0;
loap(index_C(k),j)=new_data(index_C(k),i);
new_data(index_C(k),i) = 0;
elseif (sum1(j) + new_data(index_C(k),i)>6000)&&(temp(index_C(k))==1)
loap(index_C(k),j)=6000-sum1(j);
new_data(index_C(k),i) = new_data(index_C(k),i)-(6000-sum1(j));
sum1(j) = 6000;
end
end
end

loap2 = loap;
for j = 1:8
```

```
        loap2 (:, index(j))=loap (:, j);  %该周各个转运商的转运量
    end
    loap3 = [loap3  loap2];  %24周各个转运商的转运量
end


```

zhuanyun_project.m

由于问题三和问题四的求解算法均为模拟退火，与问题二类似，因此问题三和问题四的代码此处不再展示。