基于动态规划的原材料采购策略

摘要

本文基于某建筑和装饰板材企业的生产需求和数据分析，构建数学模型帮助企业做出供应链的各环决策。首先评估各家供应商和转运商的实力和稳定性，结合各方生产和转运损失的周期性，构建最优订购量模型和转运商选择模型。

针对问题一，我们基于历史订购量和供货量数据提取特征，从供应商的生产实力、供货精准度、订单完成率和所占份额，筛选出**总供应量**、**订单完成率**、**订单误差率和大额订单占比**四项指标，并利用 **TOPSIS 综合评价法**为各家企业赋分，排名筛选出最重要的 50 家供应商。

针对问题二，我们预测各供应商未来 24 周的供货量，使用 **0-1 矩阵**表示各供应商和转运商的合作情况，以最少供应商为目标构建优化方程。同时提取转运商特征，为下题构建指标做铺垫。

针对问题三，为同时达到保障产能、压缩成本和减少囤积的目的，我们首先考虑仓储费用的节省，即为减少同一时间段仓储体积，以此为后续约束条件最优规划设置约束条件。对此我们使用了**动态规划** 算法，实现每周收货量正好略微多于能转化为一周产能都量（除第一周收货量需满足两周）。我们在划分出材料 A>C>B 的优先级顺序后，使用**贪心算法**将该优先次序运用与规划中：优先购买单位产能所需成本最小且对应所占空间也最小的 A，当 A 无法满足需求时则购买成本相同但所占空间略大的 C，最后考虑 B。我们把该题策略应用于附件 A 的历史数据进行交叉验证，发现按照每周最大产能的消耗速率，周均囤货量减少了 30%。

针对问题四，我们首先去掉了每周最大产能的约束，其余条件与问题三相同的优化方程，制定了最优的转运商选择计划。第二步，我们继续使用贪心算法和问题三的优先级，根据此计划，计算出了可能的最大收货量。根据所得最大收获量，我们利用**模拟退火**、**SLSQP** 等方法求解每周产能上限的期望值。由于目标函数总体呈现明显的单峰特征，因此上述算法对最优值的收敛速度和初始值敏感性相近。

关键字：TOPSIS 综合评价法，0-1 规划，贪心算法，动态规划，模拟退火

1 问题背景与重述

供应链的主要流程包括原材料的采购，运输，加工，储存，售卖。精心设计的供应链，能加快的产品生产，减少运输的时间，使货物合理周转，让企业利益最大化。然而，供应链本身十分复杂，需要考虑企业需求与供应商生产能力的可变性和外部环境（如季节、社会经济等）带来的不确定性。针对供应链，近年来学界做了许多调查，包括 EPQ 模型 [1]、目标成本管理 [2]、产量柔性生产批量决策模型 [3] 等。针对本题案例，本文致力于建立更具针对性的数学模型。

某生产企业所用原材料主要有 A、B、C 三种类型。该企业按每年 48 周安排生产，现在要根据产能提前制定未来 24 周的原材料订购及转运计划。该企业每周产能为 $28200m^3$, 每立方米产品需要 A 类 $0.6m^3$, 或 B 类 $0.66m^3$, 或 C 类 $0.72m^3$。其中，ABC 三类产品每立方米的单价分别是 1.2:1.1:1。为了保证生产，企业保持不少于两周生产需求的原材料库存量。由于配货的不确定性，供应商实际的供货量可能多于或少于订货量，但企业对实际供应商配给量总是全部接受。

在转运中，原材量会有损耗。每家转运商每周的最大运输能力为 $6000m^3$，损耗的历史数据由附件 2 给出。一家供应商一周尽可能只选择一家转运商。

(1) 已有 402 家供应商 240 周的数据，包括每周供货量和订单量，需制定评估标准选择其中最重要的 50 家。

(2) 参考 (1)，企业要确定能保障正常生产的供应商数量最小值，并针对这些供应商制定未来 24 周最经济的原材料订购计划，即为价格总和最便宜的订购方法。据此还需制定转运损耗最小的每周采购方案。

(3) 由于每立方米的仓储都需要钱，而单位产能对于 A 和 C 的成本相同且所要求的 A 材料体积更小，所以企业计划尽可能多采购 A，少采购 C。由于所有材料的单位运输费用相同，企业希望转运费用尽量小，那么目标即为减少转运商的数量。

(4) 现在企业有能力提高产能，请确定能提高到多少？并制定未来 24 周的方案

2 问题分析

2.1 问题一分析

问题一的目的是从 402 家供应商中选取最具有实力的 50 家，这就需要我们对 240 周的订货量和供应量进行综合评判，建立量化分析体系。思路如下：

(1) 利用附件一的数据，将供应和订单两张表进行对比，可以计算出很多评判企业的指标。比如完成率，平均误差率，方差和总供应量等等。由与总供应量十分重要且很有区分度，可以先以此来筛选，后续再根据所有指标进行打分。

(2) 利用层次分析法和一致矩阵，计算出所选指标的权重。

(3) 利用 Topsis 给各项指标打分，再分别乘上权重相加，得到最终分数。根据最终分数排名，筛选出得分的前 50 名（即为最重要的 50 家）。

2.2 问题二分析

问题二总体可以分为三个部分：1. 至少要多少供应商。2. 根据 1 的供应商，制定最经济的原材料制定方案。3. 制定最佳的转运方案。题目二只考虑题目一的 50 家供应商，步骤如下：

(1) 题干提到公司一年以 48 周作为周期，那么未来 24 周为一年的开始，即上半年。考虑到时间流动的影响和材料供应具有周期性，时而高时而低，在判断供应商的供应上限时着重关注。

(2) 由于供应商实际供货量和企业订货量存在差别，可以引入随机数来模拟。

(3) 预测出企业对应时间的供应量，然后用 0,1 矩阵分析，建立最优化数学模型，求解得到需要的最少供应商数量。

(4) 算出 A,B,C 三种材料在生产时的成本，得到各自的优先级。为了减少储存费，除了在订购是应尽量让得到的货堆积的少一些。

(5) 通过一些指标，算出 8 个转运商的优先级，优先让好的转运商运输，得到转运方法。

2.3 问题三分析

问题三分为两部分，第一部分是购买原材料，就要采用最经济的购买方法；第二部分是根据购买的材料，选择最佳的转运方法。

(1) 预估 402 家供运商未来 24 周的供应量上限，优先考虑最好的材料。除非最好的一种材料不能满足需求，才向下考虑。

(2) 减少转运费就要减少囤积，故尽量维持只有两周生产的量。

(3) 制定订货方案。

(4) 分析并预测 8 家转运商未来的运货损耗率。

(5) 制定转运方案。

2.4 问题四分析

题目四要设定产能上限，就可以关注所有供应商的产量之和。

(1) 在不超过转运上限的前提下，尽可能要更多。若超过，则同问题三（1)。

(2) 同问题 (三)，制定最佳转运方案。

(3) 购买量减去损耗量，得到原材料量，进而得到生产总量上限。

(3) 由于储存量不能小于两周产能，因此要根据这个限制条件设置梯度寻找最优解。

3 模型假设

(1) 研究对象在第一周的库存量为 0；

(2) 转运商实际运送到研究对象企业仓库的原材料被全部利用，即：接收量的利用率为 100%；

(3) 供应商的供应量以 48 周 (一年) 为周期；

(4) 未来 24 周为一年的开始，即上半年；

(5) 储存 $1m^3$ 的 A，B，C 材料的费用相同，均为单位 1 每周。在此前提下，$1m^3$ 的 A、B、C 价格分别为 120、110、100；A，B，C 材料的运费相同，都为 $5/m^3$。

4 符号说明

符号	意义
$STORE$	总储存费
$STORE(j)$	第 j 周未开始生产时的储存量
$P(A), P(B), P(C)$	A,B,C 在决策时的优先级
P	总购买费

P	总损耗
s_i	第 i 家供应商
D_{ij}	第 j 周对第 i 家供应商的订货量
S_{Aj}	第 j 周供应商能供应 A 的量
S_{Bj}	第 j 周供应商能供应 B 的量
S_{Cj}	第 j 周供应商能供应 C 的量
$Limit(j)$	第 j 周将原材料全部购买能提供的产量
S	总供应量
s_{ij}	第 i 家供应商在第 j 周的供应量

5 问题一的模型建立与求解

本小问要求构建量化模型，对 402 家供应商在 240 周内的供货表现进行评估，选出对本企业最重要的 50 家公司，作为后续模型中优先选择的供应商。首先，我们提取出若干指标评估供应商的产能、订单完成能力、供应稳定性等表现。第二步，我们利用 TOPSIS 综合评价法（又称优劣解距离法）结合一定的初步筛选，完成对供应商的评分并选出得分最高的 50 家供应商。

5.1 附件一数据的总体分析与筛选

我们对于各供应商的历史总供应量和总订单量进行加和与排序，发现了若干特点。首先，大部分供应商的供货量存在较多空值，且在一定的周期内（通常以 12 周或 24 周为周期）具有高峰期和低谷期，说明我们难以要求一家供应商持续无间断地供货，应当为周期内不同的时间点制定灵活的供应商选择策略。第二，按照总供应量降序排序后，数据出现了若干断层。为减轻数据分析量，可从部分明显的断层处截取样本，舍弃供应量明显过低的商家。

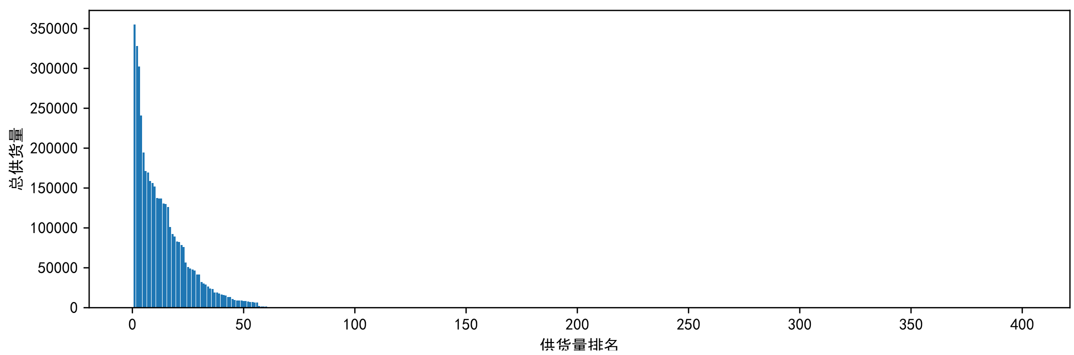


图 1: 排序后的总供应量

参照本问题的最终目标——确定 50 家供应商，考虑到一半以上的商家总供应量处于明显过低的水平，因此把供货量少于 $250m^3$（排名 109 及以后）的商家划到考虑范围之外，在前 109 家内按照指标进行排名。

5.2 评估指标

除了题目所给的各家供应商 s_i $(0 < i < 402)$ 在第 j 周的供应量 $S_j(m^3,\ 0 < j < 240)$ 和本企业对各供应商在第 j 周的订货量 $D_j(m^3)$，我们还提取出了下列指标以反映企业的供应能力：

1. **总供应量**

   总体供应量是一家供应商的总体产能的直观指标，体现出该供应商过往对企业的总体供应，也可表明它的生产潜能。该指标的计算方式是

$$TotalSupply = \sum_{j} S_{ij}$$

2. **订单完成率**

   原材料的稀缺性和企业对保障后续产能稳定的需求，导致公司需保证收货的足量。鉴于实际供货量会在一定范围内上下波动，超额相比缺额供货更易于接受。我们把供应量大于订单量的情况称为完成订单，在有订单的时间内完成订单的周数占比即为订单完成率。

$$\xi = \frac{Count(S_j - D_j > 0)}{240 - Count(D_j > 0)}$$

3. **订单误差率**

   由于原材料的特殊性，供应商难以保证精确供货。我们希望供应商尽可能提供与订单一致的数额，以免与订单相差过大造成物资紧缺或货物囤积。因此，我们可以通过实际供货量与订单的误差率，衡量一家供应商精准履约的能力，优先选出精准供货的商家，便于问题二中商家数量选择方案的制定。

   当订货量 D 不等于 0 时，第 i 家供应商在第 j 周订单误差率 ϵ 的计算公式如下

$$\epsilon_{ij} = \frac{|S_{ij} - D_{ij}|}{D_{ij}}, where\ D_{ij} \neq 0$$

   在此之上，一家供应商的总体订单误差率等于

$$\sigma = \sum_{j} \epsilon_{ij}$$

4. **大额订单数量占比**

   大部分供应商的供货量振幅较大，然而有部分商家能够相对持续地提供大额货物，这对企业保持稳定且高额的货源起到至关重要的作用。根据我们对供货量数额的基本观察，我们把供货量大于 $500m^3$ 的订单视为大额订单，然后计算大额订单数量比：

$$\gamma = \frac{Count(Supply > 500)}{Count(Supply \neq 0)}$$

5.3 TOPSIS 综合评价系统

TOPSIS 综合评价法的特点是充分利用原始数据的信息，避免人为赋权导致的指标偏差，精确反映各方案之间的差距。在算出所有数据的上述指标得分后，经过一定的正向化、标准化和归一化处理，得出各样本的综合评分与排名。

5.3.1 正向化与归一化

正向化用于极小型指标的转换。对于部分指标我们希望它的数值越小越好，比如误差越小说明稳定性越强，此为极小型指标。然而在综合评分时，我们以分值大者为优，因此需要把极小型指标转为极大型，转换式为

$$max(x) - x$$

归一化用于消除纲量影响，一般的计算式是

$$\frac{max(x)-x}{max(x)-min(x)}$$

1. **总供应量** 不同企业的供应能力千差万别，周供货量由个位数至数万不等。采取一般的归一化方法极其容易受到极端值的影响，使标化能力变弱。对此我们采取的归一化公式是

$$\frac{X-\mu}{S}$$

，其中 μ 是所以供应商总供货量的平均值。

2. **订单完成率，订单误差率，大额订单数量占比** 由于该指标的取值范围是 [0,1]，经过正常的归一化即可使用。

5.3.2 标准化处理与综合评分

进行完正向化和归一化处理后，假设构成的矩阵是 $\boldsymbol{X}=\{x_{ij}\}$，那么对于标准化后的矩阵 Z，元素 $z_{ij}=x_{ij}/\sqrt{\sum_i x_{ij}^2}$。接着，我们计算各个样本的优劣解距离。定义最优值 $Z^+=[max(\boldsymbol{z_1}),max(\boldsymbol{z_2}),\ldots,max(\boldsymbol{z_n})]$，最劣值 $Z^-=[min(\boldsymbol{z_1}),min(\boldsymbol{z_2},\ldots,min(\boldsymbol{z_n})]$，第 i 个评价对象标准化处理后的各项得分为向量 $\boldsymbol{x}_i$，那么第 i 个评价对象与最优值的距离是 $D^+=\parallel\boldsymbol{x}_i-Z^+\parallel_2$，与最劣值的距离是 $D^-=\parallel\boldsymbol{x}_i-Z^-\parallel_2$。接着，计算第 i 个评价对象的综合得分 $Score_i=D^+/(D^+D^-)$，对此排名即可获得对象 i 的综合排名。我们所选出最重要的 50 家商家排名如下。

表 2: 最重要的 50 家供应商排名

排名	商家	排名	商家	排名	商家	排名	商家	排名	商家
1	S229	11	S268	21	S395	31	S210	41	S037
2	S140	12	S131	22	S126	32	S374	42	S005
3	S108	13	S330	23	S284	33	S346	43	S218
4	S151	14	S308	24	S365	34	S247	44	S114
5	S282	15	S329	25	S338	35	S086	45	S074
6	S275	16	S352	26	S031	36	S040	46	S078
7	S340	17	S348	27	S055	37	S292	47	S266
8	S356	18	S306	28	S364	38	S007	48	S294
9	S139	19	S194	29	S367	39	S143	49	S123
10	S361	20	S307	30	S080	40	S201	50	S208

6 问题二的模型建立与求解

现假设我们分别使用 A、B、C 三种材料达到 1 单位的产能，计算成本分别对应如下：通过对各类产品数据的归类与加和，我们发现产品 B 和产品 C 有较强的周期性特征（图 2）。按照企业 24 周的周期，我们不妨假设每个周期内的同一时间段的总体最大供货量是一致的，以此作为企业能在该周期内时间点可能获得该材料的上限。转运商的损耗率也有周期性（图 3），各家峰值时间点不一。注意部分转运商（如 T2，T3）的损耗率存在噪音，去除噪音即可估计周期峰值的平均值。

材料种类	A	B	C
空间	0.6	0.66	0.72
单位空间的价格	120	110	100
单位产量的价格	72	72.6	72

表 3: 三类原材料的成本对比

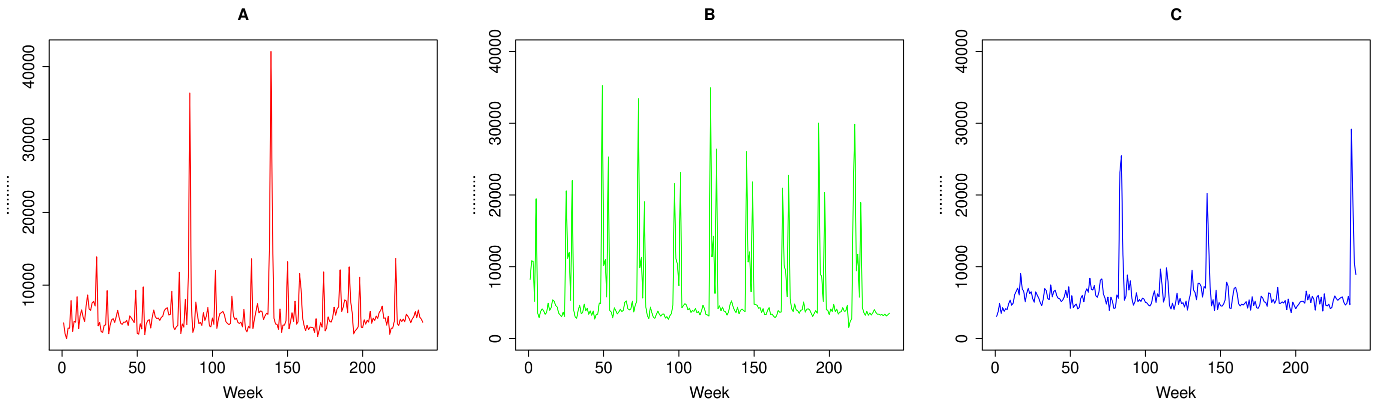


图 2: 三种产品的总供应量

6.1 问题二模型的建立

为预留接下来两周的生产需求，企业需要规划供应商选择计划。本题是一个双阶段的最优规划问题，分为第一阶段的供应商选取和第二阶段的经济成本优化。第一步，我们仅仅聚焦于供应商，利用 0-1 规划模型求出合理的各类供应商数量，优先考虑第一问里排名高的供应商；第二步，针对所选供应商，规划成本最低的方案；第三步，利用 0-1 规划模型，目标函数为损耗最小。

6.1.1 预测模型

由于我们需要规划未来 24 周的量，所以必须对未来的 402 家供应商的供货上限做预测。供应商供货呈现周期性，所以我们对未来某一周的预测时，只参考过去 5 年来的相同时期。比如第一周只参考附件的第 1+48k 周，k 是整数。如果该供应商未来在这一时期都没有出现过缺货，则供应商在过去 5 年的" 潜力 "未被完全激发。那么我们认为供应商生产极限在最大供应量 1-1.5 倍之间呈均匀分布波。若供应商在过去五年内发生过缺货，则供应上限在过去五年的均值和最大值之间波动。若无缺货，我们在订购时，选择订购最大值的 1.3 倍。若有缺货，我们在订购时，选择过往五个数据的最大值。

6.1.2 最少供应商模型

由于本小题的重点是企业所需供应商的最小数量，因此我们的目标函数是使得供应商数量最小。我们用 x_i 标记第 i 家企业是否被选择，即

$$x_i = \begin{cases} 0, \text{不选择第}i\text{家供应商} \\ 1, otherwise \end{cases}$$

由于本小题着眼于供货阶段，对损耗率、成本等考虑较少，为便于计算，我们将损耗率固定为 1.5%（接近平均损耗率），并将各供应商的原始产能全部转化为目标产物进行计算。考虑到货物的产

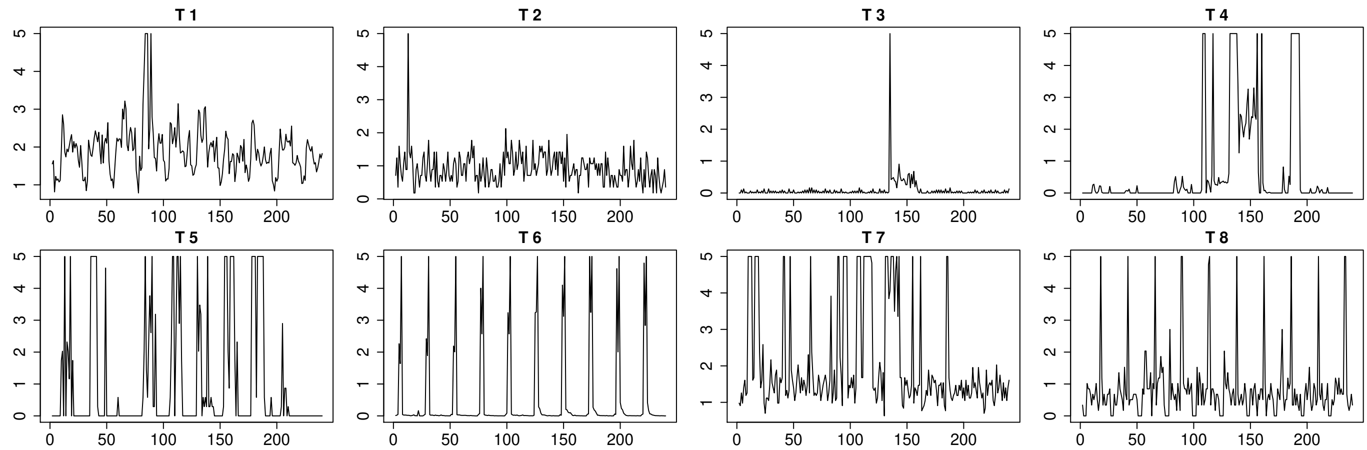


图 3: 8 家转运商的损耗率

能受季节周期性波动较为明显，我们可以假设每家供货商在本年第 x 周的产能上限为该商在过去五年第 x 周产能的最大值，由此得出 n 家供货商本年的产能上限矩阵 MAX

考虑到商家的选取是整形问题，我们采取 0-1 规划设置 24 行 n 列的矩阵 S, 当第 x 行 k 列为 1 时，代表第 x 周第 k 家供应商被选取，为 0 时，代表未被选取。将此矩阵作为 cvx 规划的变量。

6.1.3 最经济原材料订购模型

为使得所得原材料成本最低，我们应在满足最小需求供应量的情况下，优先选择成本较为低廉的原料所对应的供应商，且在上一问所选择的供应商的前提下。

将上一问所选择的供应 A、B、C 的供货商在未来 24 周的计划订购量生产的材料所转化为的产品量分别设为矩阵 P_A、P_B、P_C, 以此作为变量计算总成本以及优化约束条件

6.1.4 损耗最少转运模型

由于材料自身的物理，化学性质等会使得材料的运输损耗率随着季节而波动，因此我们取历史五年各公司对应周的平均损耗率作为该公司在当年该周损耗率的估计。各公司当年每周的损耗率计为矩阵 Q

由于各供应商仅能选择一家转运商进行合作，因此本题也是 0-1 规划模型，设转运商与供应商的合作关系为 I 矩阵，I 的第 K 行 m 列为 1，即代表此转运商与供应商进行合作

6.2 问题二模型的求解

6.2.1 最少供应商模型求解

目标为选取的供应商数量最小，我们只需最小化 S 矩阵中列和不为 0 的列数即可（列和不为 0 时，该供应商至少有一周被选取）。用向量 X 表示，X = sumofcolumn(S)

$$min \quad num = sum(X)$$

对于生产产能，应尽可能保持至少两周的供应量，即每周的产能应大于 $2.82 * 10^4$ 立方米，特别地，对于第一周的产能，至少要满足两周的库存量，即 $5.64 * 10^4$ 立方米

6.2.2 最经济原材料订购模型求解

目标函数:

使得订购总成本为最小值即

$$min \quad z = sum(0.72 * P_A + 0.726 * P_B + 0.72 * P_C)$$

约束条件:

1. 所有公司每周的产能值均不高于该公司在过去五年对应周产能的最大值，即

$$P_A <= MAX_A$$

$$P_B <= MAX_B$$

$$P_c <= MAX_C$$

2. 为了保证正常生产的需要, 该企业要尽可能保持不少于两周生产需求的原材料库存量, 即每周的产能应大于 $2.82*10^4$ 立方米, 特别地, 对于第一周的产能, 至少要满足两周的库存量, 即 $5.64*10^4$ 立方米

$$0.985*(sumofcolumn(P_A)+sumofcolumn(P_B)+sumofcolumn(P_C)) >= [5.64*10^4 \quad 2.82*10^4 ...]$$

6.2.3 损耗最少转运模型求解

目标函数:

使得材料损耗率最低，即

$$min \quad z_0 = \begin{pmatrix} P_A \\ P_B \\ P_C \end{pmatrix} * I * Q$$

约束条件:

1. 每家转运商的转运能力为 6000

$$\begin{pmatrix} P_A \\ P_B \\ P_C \end{pmatrix} * I <= \begin{pmatrix} 6000 & \cdots \\ \cdots & \end{pmatrix}$$

2. 某家供应商有且仅有一家转运商为其运输货物

$$sumofcolumn(Q) = [11...]$$

7 问题三的模型建立与求解

7.1 问题三模型的建立

为了压缩成本，企业希望尽可能少购买 C 并减少仓储成本。考虑到每立方米的仓储都有成本，不能进货太多而使货物囤积或影响材料质量。

由于企业需要保持 2 周生产的原材料量，且按照假设（1），第一周的仓储量为 0，所以第一周需要购足两周生产需求的原材料，之后每周购买一周的量即可，储存量必定能提供至少的一周生产。因此在进行决策时，我们希望在保证接下来两周的生产的前提下，尽可能减少多余的仓储，尽可能购得刚好 1 周的原料。

现在假设购买 A 材料 $P_a m^3$，B 材料 $P_b m^3$，C 材料 $P_c m^3$。在第一部分的求解中，假设损耗率为 1.5%

购买费 (PURCHASE)：

$$PURCHASE = 120 \times P_a + 110 \times P_b m^3 + 100 \times P_c$$

储存费 (STORE)：

$$STORE = (1 - 0.015) \times P_a + (1 - 0.015) \times P_b + (1 - 0.015) \times P_c$$

总费用（TOTAL）；

$$TOTAL = STORE + PURCHASE$$

限制条件是原材料能提供的生产量 (PRODUCT) 大于且接近生产需求:

$$PRODUCT = \frac{0.985 \times P_a}{0.6} + \frac{0.985 \times P_b}{0.66} + \frac{0.985 \times P_c}{0.72} > 28200$$

在限制条件下，求解 TOTAL 最小。

7.2 问题三模型求解

7.2.1 动态规划求解储存费最低策略

理想情况下，第一周买产量为 $56400m^3$ 的原材料，第一周以后买 $28200m^3$ 的原材料。但在实际预测模型中，发现在未来 24 周中，有几周即使 402 家商家都选上，仍不足以提供 $28200m^3$ 的产量，在这时就被迫需要在前一周预留更多的量（超过两周）。上周储量和本周生产极限应具有以下关系：

$$STORE(j-1) - 28200 + Limit(j) > 56400$$

同时，上一周的储量增加，又会使储藏费增加。上一周对储量进行决策时，收到往后生产量的影响，符合动态规划模型。数学过程如下：

目标序列：$[tar_1, tar_2, ..., tar_2 4]$

在初始状态下，$tar_1 = 56400$；$tar_i = 28200$，$i > 1$。

最大生产量序列：$[Limit(1), Limit(2), ..., Limit(24)]$

实际生产量：$[x_1, x_2, ..., x_{24}]$

从第 24 周往回推导。j 从 24 开始到 1：

对于第 j 周，若 Limit(j) ≥tar_j，则说明第 j 周的产量充足，不受前一周储量影响, 那么只要刚好完成本周产量就好，x_j=28200。

若 Limit(j) ≤tar_j，则说明第 j 周的产量不足，$x_j = Limit(j)$。前一周的储存需要增加，至少要是:

$$tar_(j-1) = tar(j) - Limit(j)$$

如此，我们即可以保证每周生产开始前，现有的原材料能满足两周的生产，又能将储存费达到最低。

7.2.2 贪心算法求解购买费最低

本题在购买材料上具有明显的优先级。存在：

$$P(A) > P(C) > P(B)$$

理由如下：

生产 $100m^3$ 产品，若使用 A 材料，则需要订购 $\frac{100\times0.6}{\times0.985} = 60.9m^3$，根据假设，材料只储存一周，储存费加购买费为 7370。若使用 B 材料，则需要订购 $\frac{100\times0.66}{\times0.985} = 65.1m^3$，根据假设，材料只储存一周，储存费加购买费为 7437。若使用 C 材料，则总费用为 7382。提供相同的产量，A 最经济，其次是 C，最后是 B。照此购买得到购买表格 (以表 4 提供后 4 周为例)：

订货总量	Week21	Week22	Week23	Week24
A	11815	9755	16920	5694
C	6126	8599	0	6623
B	0	0	0	6277

表 4: 最后四周的订货计划

我们得到了总共需要的 A,B,C 材料总量。接下来将这些量分配给对应的供应商，但不超过供应商的供货上限。得到的订货单如附件。

7.2.3 贪心算法求解损耗最低策略

根据附录，可知 8 个转运商的转运情况。可以发现，这些转运商都或多或少的存在一些极端值，但数量较少。如果去掉极端值，则转运货损率趋于平稳。先按照平均损耗率排名 (如表 4，运货量为 0 的不计入样本)。在排名靠前的转运商转运量少于 6000 时，优先使用排名靠前的转运商。不过，如果极端值具有周期性，则必须考虑。比如 T6 虽然平均损耗率很低，但我们发现其从第六周开始，每 24 周会出现以此损耗率为 0.05 的极端情况，所以在考虑第六周的转运方案时，不选择使用 T6。同理，在第 17 周不使用 T8。虽然 T2 在过往的第 12 周损耗率也出现 0.05，但 200 周只出现一次这样的情况，可以认为是突发事件而在第 12 周继续信任 T2

在确定了转货商优先级后，我们考虑转货原材料优先级。由于生产 $1m^3$ 的产品，需要 $0.6m^3$ A,$0.66m^3$ B, $0.72m^3$, 在物质的价值层面，明显有：

$$P(A) > P(B) > P(C)$$

, 所以我们优先给 A 分配货损率低的转运商，然后是 B，C 则用货损率相对较高的转运商运。由于在第一部分已经分别得出了 A,B,C 三种材料的转运量，方法是：我们让供应 A 货的供应商先选货损率低转运商，从第一家 A 的供应商开始转运（转运量不可超过供应商上限预期），当排名靠前的转运商转运量达到 6000，就看排名 +1 的下一个。如果总转货量等于第一部分得出的 A 购买量，就退出看 B 材料的供应商，方法与看 A 相同。最终得到的转货方案如附件。

转运商	T3	T6	T2	T8	T4	T1	T7	T5
rate	0.002	0.005	0.009	0.01	0.016	0.019	0.021	0.029

表 5: 供应商货损率与排名

8 问题四的模型建立与求解

现在企业已具备提升产能的潜力，需预测产能提升的最大可能值。假设企业有能力购买附件内的所有材料，但原材料购买能力情况仍旧受到供应商最大供货能力和转运商承载力的限制。依照这些限制条件算出每周可能得到的最多原材料后，根据“保持两周产能的仓储”原则，可反推出产能提升量。

8.1 问题四模型的建立

假设提升后的每周产能是 Q，按照前面的分析第一周采购量对应产能是 2Q，其余周为 Q，那么购买量对应产能的累计序列是 $\{2Q, 3Q, 4Q, 5Q, \ldots, 25Q\}$。第 j 周的最大接收量是 r_j，那么第 j 周的最大累计接收量是 $R_j = \sum_{n=1}^{j} r_j$。为解此题，我们需要求 $\{2Q, 3Q, 4Q, 5Q, \ldots, 25Q\} = \{R_1, R_2 \ldots, R_{24}\}$ 的最优解。

分析发现，每周 402 家供应商的总供货量通常小于 8 家转运商的最大载额，那我们对于转运商选择的目标函数和上题相同，即为最小化转运损失，仅选择平均损耗最小的转运商。由于潜在产能的上限上移，我们对于供货商的选择可以更加自由。由于我们希望优先使用平均损耗最少的转运商，因此可以预测最优的前 2-3 家在每周都会被选择。因此对于最优的 T2, T3, T6，我们取其去噪后的平均折损率 $\rho = \frac{Sum(\rho_i)}{Count(\rho_i \neq 0)}$ ，其余转运商的折损率利用 ARIMA 模型拟合估测。

8.1.1 模拟退火求每周产能最优解

模拟退火是一种常见于最优化求解的贪心算法，起初通过随机横跳并逐渐缩小振幅的方式定位全局最优解，随着时间推移温度逐渐下降，逐渐趋于稳定。在求出最优转运商选择策略后，我们把模拟退火运用于未来每周最优产能的求解。定义目标函数

$$min \parallel \boldsymbol{R} - \boldsymbol{Q} \parallel_2$$

，约束条件

$$\boldsymbol{R} - \boldsymbol{Q} > 0, \forall R_i - Q_i \in \boldsymbol{R} - \boldsymbol{Q}$$

，设置初始值为 (28000, 45000) 之间的随机数，算法如下

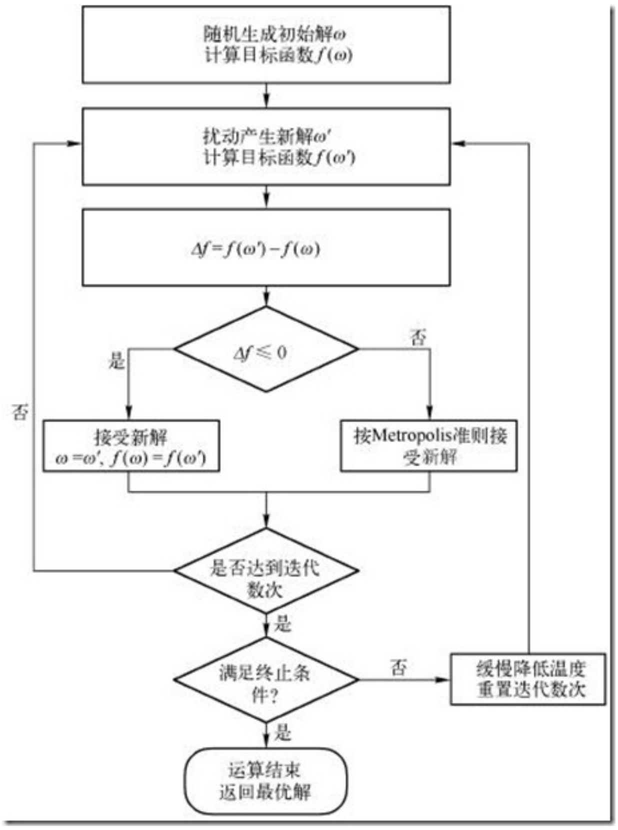
我们发现，此方法的收敛速度较快，每次运行的结果略有浮动。结合目标函数曲线图和多次计算的平均结果，我们将未来每周产能设为定值（现实应用中可理解为期望值），所得结果约 38000。

8.1.2 SLSQP 等 Scipy 附带优化模型求解

Python 的 Scipy 库自带部分适用于于线性和非线性优化问题的求解方法，例如 SLSQP(序列最小二乘规划)、QNM（拟牛顿法）等。此类模型的共同特点是以求出对称正定矩阵的最优解作为搜索方向 [4]，所得最优解由于目标函数具有单峰性，两种方法所得结果均为 37000 左右。按照上述方式确定采购计划，未来 24 周可获得的产能如图。由图可见，大部分时间的周产能都在 40000 上下 10000 都振幅内波动，少数时间出现的空缺能够较为容易的被前面的周期填补，最后平均每周都产能也在 37000-38000 之间。

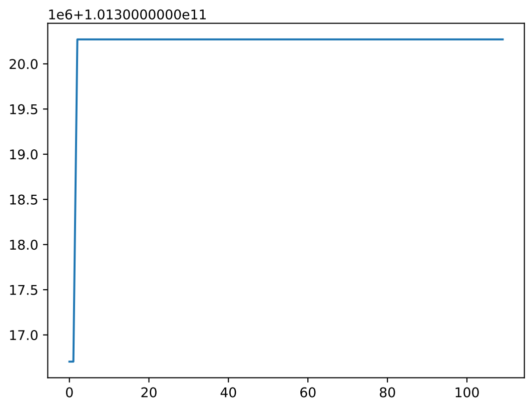
9 模型评价与改进

本模型利用过往数据，较好模拟了供应量和转运损耗率的周期性，能够帮助未来的生产获得较为稳定的收货量，也达成了仓储的优化。不足之处在于周期内各时间点的数据主要依赖于上一年和



0

图 4: 模拟退火算法流程图



0

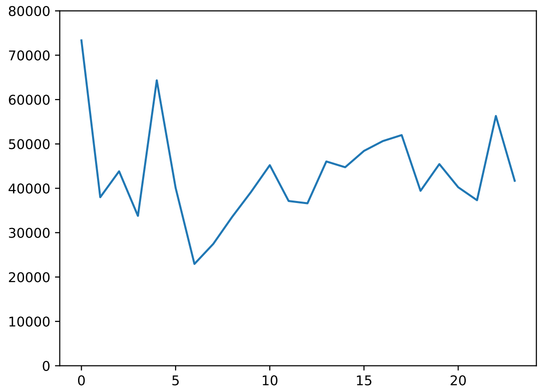
图 5: 模拟退火收敛情况

下一年同周期的平均值，且难以估计极端值的再次出现。此外，问题四模拟退火所求的最优解存在 1000 以内的误差，后续可尝试改变梯度并结合遗传算法改进解法。

10 参考文献

[1] 黄卫来, 张子刚, 刘运哲.(1998). 产量柔性下的最优生产批量和原材料订购决策模型. 系统工程 (01),44-50. doi:CNKI:SUN:GCXT.0.1998-01-007.

[2] 唐雨婷.(2021). 供应链环境下目标成本管理研究. 广西质量监督导报 (04),165-166.



0

图 6: 未来 24 周预测产能

[3] 陈晖, 罗兵 杨秀苔.(2007). 一种考虑原材料库存成本的变质物品 EPQ 模型. 中国管理科学 (03),93-97. doi:10.16381/j.cnki.issn1003-207x.2007.03.015. doi:CNKI:SUN:GXZL.0.2021-04-078.

[4] Selva Prabhakaran. (2021). ARIMA Model –Complete Guide to Time Series Forecasting in Python. Retrieved from https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

[5]Sarkar, T. (2020, December 07). Optimization with SciPy and application ideas to machine learning. Retrieved from https://towardsdatascience.com/optimization-with-scipy-and-application-ideas-to-machine-learning-81d39c7938b8

11 附录

TOISIS 综合评价

```
import pandas as pd
import csv
import numpy as np
np.set_printoptions(suppress=False)

supply = pd.read_csv('Data/109供货.csv')
demand = pd.read_csv('Data/109订货.csv')
supply = supply.values
demand = demand.values

id = supply[:,0]
product = supply[:,1]
supply = supply[1:,2:]
demand = demand[1:,2:]
surplus = supply - demand

# Accomplishment
nonzeros = np.zeros(np.shape(demand))
finished = np.zeros(np.shape(demand))
for i in range(np.shape(demand)[0]):
    for j in range(np.shape(demand)[1]):
        if demand[i][j] != 0:
            nonzeros[i][j] =1
        if surplus[i][j] >0:
            finished[i][j] =1
finish = np.sum(finished, axis=1)
nonzeros = np.sum(nonzeros, axis=1)
finish_rate = finish / nonzeros
# print(finish_rate)

# Supply
tot_supply = np.matrix(np.sum(supply, axis=1)).T
print(tot_supply)

# Supply Error
err_rate = np.zeros(np.shape(demand))
for i in range(np.shape(demand)[0]):
    for j in range(np.shape(demand)[1]):
        try:
```

```
            err_rate[i,j] = abs(surplus[i,j]/demand[i,j])
        except:
            pass
err_average = np.sum(err_rate, axis = 1) / np.count_nonzero(err_rate, axis=1)
err_average = np.max(err_average) - err_average #正向化

# 大额订单比例
price = demand
nonzeros = np.zeros(np.shape(demand))
bigs = np.zeros(np.shape(demand))
for i in range(np.shape(demand)[0]):
    for j in range(np.shape(demand)[1]):
        if demand[i][j] != 0:
            nonzeros[i][j] =1
        if supply[i][j] > 500:
            bigs[i][j] =1
big_tot = np.sum(bigs, axis=1)
num_purchase = np.sum(nonzeros)
big_rate = big_tot / num_purchase
# print(big_rate)
print(np.shape(big_rate))

def z_stand(data):
    mean = data - np.mean(data)
    n = np.shape(data)[0]
    std = np.sqrt(np.var(data))*np.sqrt(n/(n-1))
    new = mean / std
    return new
def stand(data):
    min = np.min(data)
    max = np.max(data)
    return (data-min)/(max - min)
tot_supply = z_stand(tot_supply).T * 0.5
finish_rate = np.matrix(stand(finish_rate)).T * 0.25
err_average = np.matrix(stand(err_average)).T * 0.125
big_rate = np.matrix(stand(big_rate)).T * 0.121
# zero1 = stand(data[:,3])
# print(np.shape(demand1))
data = np.hstack((finish_rate, err_average, big_rate, tot_supply))
print(data)

#余弦标准化
```

```
ss = np.sqrt(np.sum(np.square(data),axis = 0).astype('float'))
Z = data / ss
data = Z

maxs = np.max(data, axis = 0)
mins = np.min(data, axis = 0)
D1 = np.sqrt(np.sum(np.square(data-maxs), axis=1).astype('float'))
D0 = np.sqrt(np.sum(np.square(data-mins), axis=1).astype('float'))
# print(D1,D0)
score = np.array(D0 / (D1 + D0))
print(score)
indexes = np.argsort(score, axis = 0)[:51]
result = id[indexes]
# print(result)
dataframe = pd.DataFrame(result)
dataframe.to_csv(r'Data/Result1(1).csv')
% 代码段
```

模拟退火

```
import matplotlib.pyplot as plt
import math
from random import random
import numpy as np

"""
fun(x)内的列表表示此前动态规划结果下的各周累计产能
"""

def func(x):
    datalist = np.array([73351.91090182167, 111339.89459363623,
        155177.3072630262, 188958.04212824057, 253307.98703481574,
        293421.4847445094, 316365.5306642972, 343833.8928474991,
        377378.3558840396, 416533.2896341347, 461752.1959745696,
        498887.22383573174, 535502.7684939676, 581554.3592197088,
        626300.7612415648, 674732.1778451478, 725377.4025729486,
        777374.433939069, 816799.994877443, 862258.5386438754,
        902497.5432791817, 939807.6232049897, 996114.3526803257,
        1037806.3160686874])
    prod = np.array([n for n in range(2,26)])*x
    err = np.sum(np.square(datalist - prod))
    return err

```

```
def main():
    plot_obj_func()
    T_init = 100  # 初始最大温度
    alpha = 0.90  # 降温系数
    T_t = 1e-3  # 最小温度，即退出循环条件
    T = T_init
    x = random() * 15800 +28200  # 初始化x，在28200和45000之间
    y = func(x)
    results = []  # 存x，y
    while T > T_t:
        x_best = x
        y_best = y
        flag = 0
        # 每个温度迭代50次，找最优解
        for i in range(50):
            delta_x = random() - 1  # 自变量进行波动
            # 自变量变化后仍要求在[0,10]之间
            if 28200 < (x + delta_x) < 40000:
                x_new = x + delta_x
            else:
                x_new = x - delta_x
            y_new = func(x)
            if (y_new > y or math.exp(-(y - y_new) / T) > random()):
                flag = 1  # 有新值被接受
                x = x_new
                y = y_new
                if y > y_best:
                    x_best = x
                    y_best = y
        if flag:
            x = x_best
            y = y_best
        results.append((x, y))
        T *= alpha

    print('最优解 x:%f,y:%f' % results[-1])
    iter_plt(results)

def plot_obj_func():
    X1 = [i for i in range(20000,50000,10)]
    Y1 = [func(x) for x in X1]
    plt.plot(X1, Y1)
    plt.show()
```

```python


def iter_plt(results):
    X = [i for i in range(len(results))]
    Y = np.array([results[i][1] for i in range(len(results))])
    plt.plot(X, Y)
    plt.show()

if __name__ == '__main__':
    main()
```

其余题目主要解题过程为 Excel 处理，故不在此附录