

生产企业原材料的订购与运输

摘要

为保障生产企业持续稳定的发展，本文建立了基于 CRITIC 方法的综合评价模型来帮助企业筛选高质量合作供应商，并根据企业生产需求、供应商和转运商实际情况制定未来半年的订购和运输方案。同时展望未来，对相同原材料市场下企业最大提升产能给出相应分析。

针对问题一，通过对 402 家供应商的供货特征进行量化分析，本文抽象出供货诚信度（下文称“诚信度”）、企业订货依赖度（下文称“依赖度”）和原材料供应类别初始评价指标。并使用综合评价模型来计算各供应商的重要性得分。基于 CRITIC 方法得出诚信度与依赖度的权重分别为**0.458982**和**0.541017**，并对 402 家供应商的重要性进行评估，排名前 50 的作为最重要的供应商。

针对问题二，在问题一的结论的基础上，建立起 0-1 整数规划模型并确定关于供货能力、供需关系、损耗率、库存与产能、库存剩余量的约束条件，确定该企业在考虑满足生产需求的情况下，至少需要选择 21 家供货商供货原材料才满足生产需求。在这 21 家供货商的基础之上，规划出成本最小的订购模型与损耗最少的转运方案。首先对数据进行预处理，分析得转运损耗率波动小，因此认为转运商未来 24 周的转运损耗率为前 240 周去除 0 元素后的平均损耗率，并且设未来 24 周各供货商的供货能力为去 0 元素后的平均供货量，然后以供货商的选择、转运商与供货商的对应关系、当周使用量为决策变量，抽象出总生产量、总剩余量和各材料剩余量应大于 0 等条件作为约束条件，通过多次迭代遗传算法，根据每周剩余量不同的差异进行动态规划，逐周求解得出最优的订货以及转运方案。

针对问题三，问题三也是求优化方案，由于外在市场环境未发生较大变化，因此选择在问题二原有约束条件的基础上，选择与 AC 订购量和损耗经济有关的目标函数，建立单目标规划模型。应用线性规划求解该模型得到企业订购方案。在该问的结果中 A 的订购量有显著上升、C 的订购量显著下降。对于转运方案，我们则采取按不同损耗率分配不同成本单价原材料的策略，求得新的转运方案。

针对问题四，通过分析得知该企业每周的产能提升潜力主要取决于每周收到的原材料的等效产能。故以每周产能最大为目标函数，综合考虑运输总量、供货量、损耗率、库存量等约束条件，建立 0-1 整数规划和线性规划混合规划模型。模型求解结果显示每周选择的供货商都是全部商家，且全部供货商提供原材料的总体积未达到最大运输体积，即该企业产能的提升主要受到现有供货商供货能力的限制，最终通过多次逐周混合规划即可得到该企业未来 24 周最大产能。最后对结果分析，可对假设进行验证，并证明了模型的合理性。

关键词：混合规划 动态规划 单目标优化 多目标优化 Topsis 综合评价法

一、问题重述

1.1 问题背景

某建筑和装饰板材的生产企业所用原材料总体可分为 A、B、C 三种类型。原材料的采购成本直接影响到企业的生产效益，实际中 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。由于该企业每年按 48 周安排生产，需要提前制定 24 周的原材料订购和转运计划。

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。在实际转运过程中，原材料会有一定的损耗，每家转运商的运输能力为 6000 立方米/周。通常情况下，一家供应商每周供应的原材料尽量由一家转运商运输。

由于原材料的特殊性，供应商不能保证严格按订货量供货，实际供货量可能多于或少于订货量。为了保证正常生产的需要，该企业要尽可能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实际提供的原材料总是全部收购。

1.2 问题重述

根据题目背景以及附件数据来研究以下问题：

- 1) 量化分析 402 家供应商的供货特征，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，并在论文中列表给出结果。
- 2) 基于问题 1，选择能满足生产的需求条件下最少供应商数量。针对这些供应商，制定该企业未来 24 周内每周最经济的原材料订购方案和损耗最少的转运方案。并对订购方案和转运方案的实施效果进行分析。
- 3) 在尽量多地采购 A 类和尽量少地采购 C 类原材料的基础上，制定新的订购方案及转运方案以减少转运、仓储的成本以及转运商的转运损耗率。并分析该方案的实施效果。
- 4) 根据现有原材料的供应商和转运商的实际情况，确定该企业每周产能的可提高量，并给出未来 24 周的订购和转运方案。

二、问题分析

对于生产企业原材料的订货与运输问题，在附录数据中给出了 402 家供货商近 5 年的订货与供货的数据，题目要求据此来制定未来 24 周的订货与转运方案。因此我们首先对这 402 家供货商进行对生产企业的重要性评估，再将运输以及存储成本纳入考虑，建立数学模型寻找实际情况下最优的原材料订购方案以及运输方案。

2.1 问题一的分析

为建立反映保障企业生产重要性的数学模型，本文通过对附件一中 402 家供应商的供货特征进行量化处理，即对供应商的供货能力进行综合评价，确定了供货诚信度（下文称“诚信度”）、企业订货依赖度（下文称“依赖度”）和原材料供应类别三个一级评价指标。其中，诚信度使用订供偏差度矩阵经过 TOPSIS 法计算表征；依赖度则用近 5 年不同原材料的

总订货量，以及历史每周订货量进行归一化进行表征；原材料供应类别用单位产品消耗 A、B、C 三类原材料的采购及运输和储存成本来表征。基于模糊算法改进的 TOPSIS 法，采用 CRITIC 综合评价方法确定不同指标的权重，本文可以对附件 1 中 402 家供应商进行综合评价，为每个供应商的供货能力打分，将排名前 50 的供应商作为最重要的 50 家供应商。

2.2 问题二的分析

问题一筛选出了 50 家最重要的供货商。问题二在问题一的结论的基础上，通过 0-1 线性规划并确定约束条件，确定该企业在考虑满足生产需求的情况下，至少需要选择 $n(n \leq 50)$ 家供货商供货原材料才满足生产需求。在这 $n(n \leq 50)$ 家供货商的基础之上，规划出成本最小的订购模型与损耗最少的转运方案。首先对数据进行预处理，分析得转运损耗率波动小，以所选择的供应商的供货量、转运商的转运平均损耗率和转运原材料的种类为决策变量，抽象出总接受量、剩余量、使用量等变量进行约束条件得考虑，通过多次迭代遗传算法，根据每周剩余材料数量的不同的差异，逐周求解得出最优的订货以及转运方案。

2.3 问题三的分析

问题三在问题二的模型基础之上，增添了新的需求：尽可能多的采购 A 并尽可能地少采购 C。同时这也是一个最优化模型，因此我们对问题二中建立的模型进行了修改：随问题三的需求改变了目标函数，使其与 A、C 的订购量以及损失经济有关；但总体的约束条件不变，并对其进行求解。改变后的新优化方案中 A 的订购量显著上升、C 的订购量显著下降；再通过调整目标函数，求出相关的转运方案，因此可认为模型在一定范围内是合理的。

2.4 问题四的分析

对于问题四进行分析和拆解，发现对目标函数与约束条件进行分析即可将其转化为按时间顺序进行 24 次的线性规划。由于企业已具备了提高产能的潜力，为满足企业产能提升的需求，在选定的 50 家最重要的供货商并不符合提升产能的需求时，考虑将供货商选取范围进一步扩大至全部的 402 家供货商，同时由于产能提升潜力的存在，企业产能能够提高，但是由于原产能以及供货商供给的供货量等约束条件，使得企业提升产能的能力有上下限，限制了产能提高的范围。通过限制企业每周产能的因素为转运商每周的总转运量，据此重新求解符合问题四的订购方案。

三、基本假设

1. 假设数据中给出的供货商均合理存在，因此不进行数据剔除。
2. 假设供货商的供货能力以及转运商的运输能力在未来 24 周并没有变化巨大的变化，供货商的供货能力有限，可以使用近年的数据对未来订购以及转运方案进行预测。
3. 假设在上一周该生产企业会按照订购计划订购下一周生产所需的原材料，并在下一周一开始就存储进仓库。
4. 假设该生产企业的每周产能是定值，每周均能恰好完成且不会超额完成。
5. 假设运输成本只与原材料的体积有关，而与运输路程等条件无关。

6. 假设每周的供货总量在该周内由转运商全部完成，不会出现延迟运输的情况。
7. 假设在企业通过技术改造具备了提高产能的潜力时，此时制定的订购与转运方案使企业对原材料的接受量增大，并将该量与产能提高量相匹配。

四、符号说明

(此处应该画一个表格，三段表捏)

符号	含义说明	单位
N_o	企业订货量	m^3
N_s	供货商的供货量	m^3
B	偏移度矩阵	-
D_H	供货诚信度	-
D_R	企业订货依赖度	-
D_T	原材料供应类别	-
x_i	企业是否向第 <i>i</i> 家供货商订货	-
y_{ij}	第 <i>i</i> 家供货商是否选择第 <i>j</i> 家转运商来运送原材料	-
E_i	第 <i>i</i> 家供货商的供货能力	m^3
F_i	供应商的供货产能能力	m^3
A	企业总接受原料量	m^3
R	每周企业原料剩余量	m^3
U	企业每周原料的使用量	m^3
M	企业等效产能	m^3
LS	损耗经济	元· m^3

表格 1：符号说明

注：表中未出现的符号均在附录中有详细说明。

五、问题一 供应商的重要性评估

5.1 建立评价模型

5.1.1 数据预处理

首先我们对附件 1 中的订货量与供货量数据进行分析，可知每家供货商只供给 ABC 三种原材料中的一种。然后分别计算企业对每家供货商的近 5 年的总订货量以及近 5 年的平均每周订货量，每家供货商的近 5 年总供货量以及近 5 年的平均每周供货量。

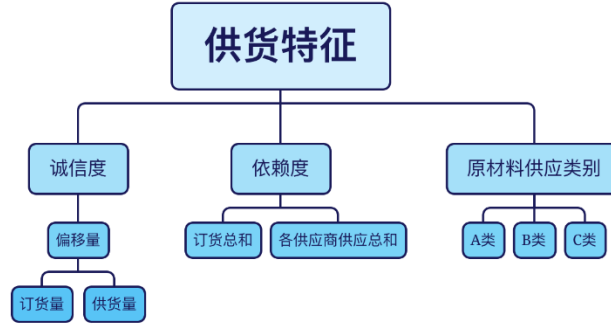
通过对数据的预处理，发现“供货远大于订货”的情况并不存在。一定情况下，供货可能大于供货，但是不会超过 $10m^3$ 。

5.1.2 指标确定

根据参考文献可知，评价指标用于刻画被评价对象的特征或属性。每一项所选取的评判指标均从不同方面反映评价对象所具有的某种特征大小的一个度量。而在多个评价指标构成的指标体系时，构建评价指标体系的原则在于：系统性、科学性、可比性、可测性和独立性。

根据此原则，进行数据预处理之后，我们选择了如下的评价指标用以反映供货商的供货特征。

- ◆ D_H 供货诚信度：表征不同供货商及时完成企业订货订单量的程度，使用订单的完成度决定。
- ◆ D_R 企业订货依赖度：表征企业对于不同供货商的不同重视程度，通过派发给不同供货商的订单量大小来决定。
- ◆ D_T 原材料供应类别：表征该供货商供货的原材料的类型。



图表 1：问题一的多指标评价模型

在此处为得出 D_H ，我们引入了偏移度矩阵 B ， B 中的每一个元素表征 N_o 与 N_s 的偏移程度。

5.1.3 指标计算

1. D_H 诚信度

诚信度的计算，基于偏移度矩阵的计算。计算偏移度矩阵的公式如下所示：

$$B = \begin{cases} \frac{|N_s - N_o|}{N_o}, & N_o \neq 0 \\ 0, & N_o = 0 \end{cases} \quad (1-1)$$

此处使用了 TOPSIS 法进行多指标评价。通过构造评价问题的正理想解和负理想解计算每个方案到理想方案的相对贴近度，来对方案进行排序，从而得出诚信度。下面给出正理想解以及负理想解的表达式：

$$\begin{cases} B_j^* = \max B_{ij} \\ B_j^0 = \min B_{ij} \end{cases} \quad (1-2)$$

根据前面所进行的数据预处理，可知供货量的偏移不会超出订货量的 2 倍以上，因此本文规定 TOPSIS 的负理想解是元素均为 1 的行向量，其列数与偏移度矩阵相同。然后再计算被评价对象到正理想解与负理想解的欧式距离：

$$Dis_{i+} = \sqrt{\sum_{j=1}^n (B_{ij \max} - B_{ij})^2}, \quad Dis_{i-} = \sqrt{\sum_{j=1}^n (B_{ij \min} - B_{ij})^2} \quad (1-3)$$

最后，计算各方案的综合评价（即诚信度）从而给出排序：

$$D_{H_i} = \frac{Dis_{i-}}{Dis_{i+} + Dis_{i-}}, \quad D_{H_i} \in [0, 1] \quad (1-4)$$

D_{H_i} 在 0 到 1 之间取值，越接近 1 说明诚信度越高，反之，越接近 0 说明诚信度越低。

2. D_R 依赖度

企业订货依赖度，分为 ABC 三种原材料分别计算近 5 年不同类型的总订货量，得到 Sum_A 、 Sum_B 、 Sum_C 。设第 i 号供货商的近 5 年的总供货量为 N_{O_i} , $i = 1, 2, \dots, 402$ （所提供的

数据中共有 402 家供货商)，所供原材料类别是 k , $k \in A, B, C$ 。因此近 5 年不同类型的总订货量可表述为 Sum_k 。据上述假设条件，可以得不同供货商得供货量占比 R_i 为：

$$R_i = \frac{N_{oi}}{Sum_k}, k \in \{A, B, C\} \quad (1-5)$$

然后使用 R_i 进行归一化，得出依赖度 D_R 的表达式：

$$D_{R_i} = \frac{R_i}{\max\{R_j | j = 1, 2, \dots, 402\}} \quad (1-6)$$

3. D_T 原材料供应类别

原材料的供应类别，每一家供应商只供应 ABC 中的一种原材料，属于不同供应商的固有属性，因此在计算量化指标权重的时候不进行考虑，使其权重为 0。

$$D_T = \{A, B, C\} \quad (1-7)$$

5.2 模型求解

5.2.1 算法展示

在该问题中, CRITIC 法是一种比熵权法和标准离差法更好的客观赋权法。因此本文在进行量化指标进行得分评价时，将以上的指标通过 CRITIC 法来计算各自权重。CRITIC 法基于评价指标的对比强度和指标之间的冲突性来综合衡量指标的客观权重。考虑指标变异性大小的同时兼顾指标之间的相关性，并非数字越大就说明越重要，完全利用数据自身的客观属性进行科学评价。

设最终衡量供应商重要性得分为 S_i , w_H 与 w_R 分别为 D_H 与 D_R 的权重：

$$S_i = w_H D_{H_i} + w_R D_{R_i} \quad (1-8)$$

CRITIC 法对权重的确定以对比度和冲突性为基础。对比强度指同一个指标各个评价方案之间取值差距的大小，以标准差 σ 的形式来表现，记 D_H 的标准差为 σ_H , D_R 的标准差为 σ_R 。标准差越大，说明波动越大，即各方案之间的取值差距越大，权重会越高；冲突性采用 D_H 与 D_R 的相关系数 r 进行表示，若两个指标之间具有较强的正相关，说明其冲突性越小，权重会越低。对 D_H 与 D_R 的样本标准差进行求解：

$$\begin{cases} \overline{D_H} = \frac{1}{n} \sum_{i=1}^n D_{H_i}, \overline{D_R} = \frac{1}{n} \sum_{i=1}^n D_{R_i} \\ \sigma_H = \sqrt{\frac{\sum_{i=1}^n (D_{H_i} - \overline{D_H})^2}{n-1}}, \sigma_R = \sqrt{\frac{\sum_{i=1}^n (D_{R_i} - \overline{D_R})^2}{n-1}} \\ n = 402 \end{cases} \quad (1-9)$$

其信息量为：

$$C_H = \sigma_H(1-r), C_R = \sigma_R(1-r) \quad (1-10)$$

计算 D_H 与 D_R 的权重：

$$w_H = \frac{C_H}{C_H + C_R}, w_R = \frac{C_R}{C_H + C_R} \quad (1-11)$$

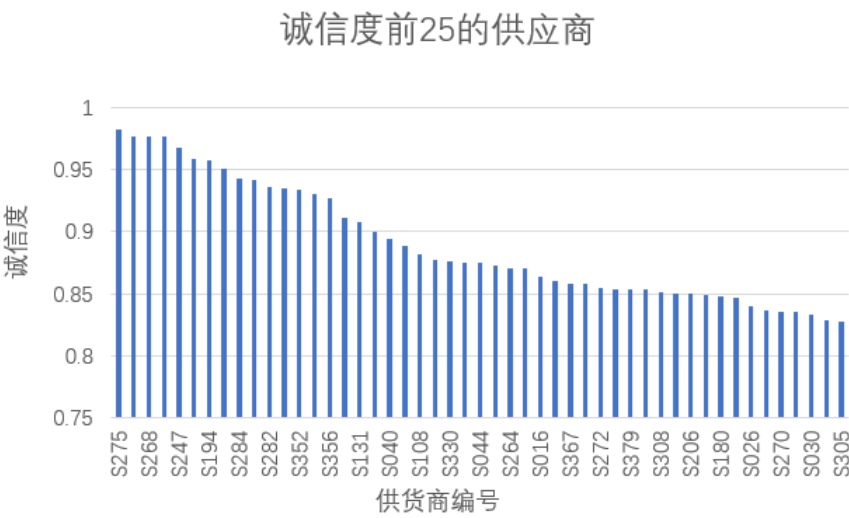
在标准差一定时，指标间冲突性越小，权重也越小；冲突性越大，权重也越大；另外，当两个指标间的正相关程度越大时（相关系数越接近 1），冲突性越小，这表明这两个指标在评价方案的优劣上反映的信息有较大的相似性。

5.2.2 求解结果展示

在确定了能有效描述供货商的供货特征的指标后，构建供货重要性评价模型，通过 TOPSIS 方法求解出诚信度指标，使用 CRITIC 方法求解出各个指标的权重，建立基于 CRITIC 的多指标评价模型，该得分即反映出供货商对于企业的重要性。该得分越高，则说明供货商的重要性越高。

1. 计算 D_H

将附件 1 中的订货量与供货量各输入为矩阵，经过计算得到 D_H 。 D_H 范围在 0.53~0.99 之间，可认为 402 家供货商中没有过于失信的供货商存在，这也符合实际情况。对于一直不及时供货的供应商，通常企业不会再与其合作。



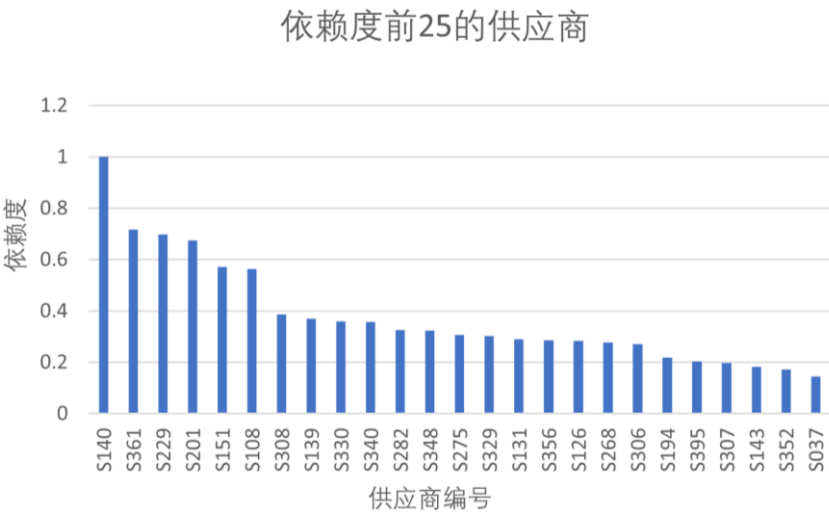
图表 2：诚信度前 25 的供应商

2. 计算 D_R

计算近 5 年不同类型的总订货量（单位为 m^3 ）为：

Sum_A	Sum_B	Sum_C
2056903	1917241	1855279

然后计算依赖度。部分结果如下图所示：



图表 3：依赖度前 25 的供货商

3. 综合指标进行量化分析

我们通过 CRITIC 方法计算出来相关的中间过程值：

σ_H	σ_R	r	w_H	w_R
0.089184	0.105124	0.6432	0.458982	0.541017

从权重的值不难看出，诚信度与依赖度的重要度并没有相差太多。因此我们得到重要性得分的表达式为：

$$S_i = 0.458982D_{H_i} + 0.541017D_{R_i}$$

5.2.3 重要性评估结果展示

以下给出的是评估得分最高的 50 家供货商，他们被认为是最重要的。

供货商 ID	重要性得分	供货商 ID	重要性得分
S140	0.835163	S143	0.478551
S361	0.814666	S031	0.477534
S229	0.806457	S040	0.446897
S201	0.712203	S364	0.4454
S151	0.710435	S294	0.43933
S108	0.71007	S346	0.429576
S340	0.633088	S367	0.424993
S275	0.616648	S307	0.419492
S329	0.612334	S055	0.40939
S282	0.606333	S300	0.408344
S308	0.600144	S037	0.408242
S268	0.598673	S033	0.40253
S330	0.596366	S044	0.401641
S306	0.594036	S264	0.39971
S356	0.579538	S298	0.399559
S131	0.573206	S218	0.396758
S194	0.557087	S016	0.396615
S348	0.522358	S303	0.394997
S352	0.522083	S225	0.393635
S247	0.50914	S272	0.392519
S139	0.507241	S379	0.391812
S126	0.50722	S358	0.3917
S395	0.487559	S362	0.391596
S284	0.485367	S397	0.390133
S365	0.483794	S206	0.390062

表格 2：50 家最重要的供货商

六、问题二 动态规划最优订购与转运方案

6.1 建立满足生产需求的 0-1 线性规划模型

6.1.1 目标函数的建立

在问题一中，本文通过评价供货商的重要性选择出了最重要的 50 家供货商，通过整理着 50 家供货商每周的供货量发现，排在靠后的供应商每周供货量基本为个位数，他们每周向企业供应的原材料中占比非常低。若在考虑原材料在路上的损耗，可以得出重要性前 50 的供货商的供货量基本达到 402 家供货商的总供货量的 90%以上。鉴于上述实际情况，我们在制定新的订购方案时只考虑前 50 名最重要的供货商。

本题要求需要进一步缩小满足生产需求的供货商集合，在问题一中所求出的 50 家供货商中，以选择的供货商每周能够提供的材料能满足产能需求作为约束条件，建立 0-1 规划模型，进而寻找供货商数目最小的供货商集合。设 x_i 作为 0-1 决策变量，决定企业是否向第 i 家供货商订货。 $x_i = 1$ 表示向供应商订货， $x_i = 0$ 则表示不向该家供应商订货。因此供货商选择的集合可用下式表达：

$$\{N_i | \sum_{i=1}^{50} x_i = 1\} \quad (2-1)$$

因此确定目标函数为如下表达式：

$$\min \sum_{i=1}^{50} x_i \quad (2-2)$$

6.1.2 约束条件的选取

由于要求选择可满足生产要求的最少供应商数目，并给出相应的最优订购方案以及转运方案。本文选取了以下的约束条件。

供货商给到的供货量是原材料的数量，但是使用不同类别原材料制造产品所需要的量是不同的。依据近 5 年的供货量数据，求出不同供应商近 5 年来每周平均供货量。将该常数定义为供货商每周的供货能力，第 i 家供货商的供货能力表示为 E_i 。

由于所有供货商的供货需要经过转运到达企业，所以其总供货量应满足企业本周的生产的需要和库存不少于两周原材料的需要，因此存在供给需求的等式关系。本文为了方便计算，我们将不同种类原材料的供货量均转化为对应的产能。设第 i 家供应商供货产能为 F_i ，供给需求关系式如下：

$$F_i = \begin{cases} \frac{E_i}{0.6}, & D_{T_i} = A \\ \frac{E_i}{0.66}, & D_{T_i} = B \\ \frac{E_i}{0.72}, & D_{T_i} = C \end{cases} \quad (2-3)$$

根据上述的供应商的供货产能能力 F_i ，由于该企业每周产能至少达到 $2.82 \times 10^4 m^3$ ，本文根据该条件得到了如下的约束条件，使得每周的产能不小于 $2.82 \times 10^4 m^3$ 。

$$\sum_{i=1}^{50} x_i F_i \geq 28200 \quad (2-4)$$

6.1.3 模型建立

根据上述选择的目标函数(2-2)以及约束条件(2-4)，线性规划方程如下。该线性方程的求解即为至少选择的供应商数量，因此对目标函数求解最小值。

$$\min \sum_{i=1}^{50} x_i$$
$$s. t. \sum_{i=1}^{50} x_i F_i \geq 28200$$

(2-5)

6.2 线性规划模型结果展示

供货商 ID	平均供货量/周	供货商 ID	平均供货量/周	供货商 ID	平均供货量/周
S201	3097.269	S151	816.3519	S308	572.0664
S229	1482.051	S275	660.251	S348	490.8548
S140	1517.146	S329	652.1745	S307	465.8084
S361	1371.242	S340	714.7082	S126	554.7176
S395	1024.905	S139	696.3716	S268	542.1581
S108	1015.684	S330	575.9043	S356	539.9264
S282	705.2343	S131	575.8122	S306	526.1021

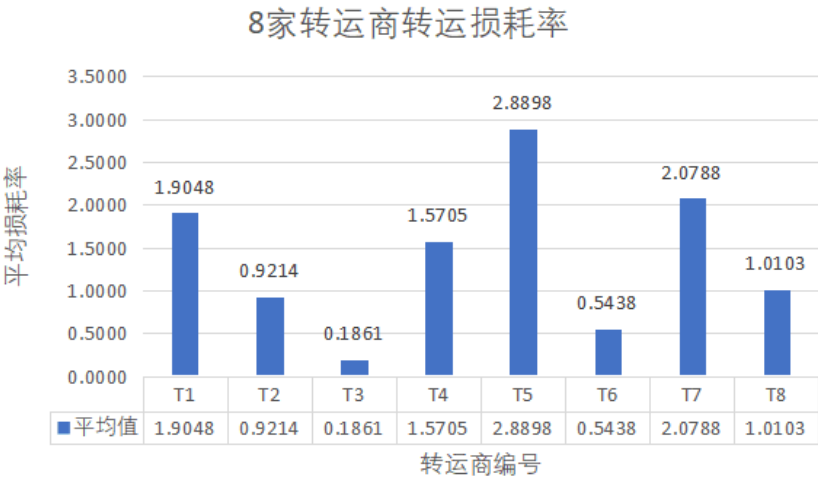
表格 3：选出最少 21 家供应商及提供的原材料平均每周供货量

6.3 建立最优化订购与转运方案模型

6.3.1 数据预处理

根据上述基于 0-1 规划的订购方案模型，得出满足生产需求的情况下最少需要的 21 家供应商。基于上述选择的 21 家供应商，本文开始制定未来 24 周的最经济的订购方案以及损耗最少的转运方案。

由于附件二中关于转运损耗率的数据中有大量数据为 0，在实际情况下，转运过程中一定会存在损耗，所以本文将转运损耗率为 0 的数据视为该转运商本周没有进行转运任务，且将最大转运损耗率 5%视为正常值。据此我们首先计算了关于转运商进行了转运任务周的平均转运损耗率，数据如下所示：



图表 4：各转运商的平均转运损耗率

6.3.2 约束条件的确定

为了制定未来 24 周每周最经济的订购方案，即要求订购的原材料总量在采购、运输及存储三方面的成本花销为最小，在制定方案的同时也需要保证该订购方案满足企业的生产需求，结合已知条件和假设，该规划模型需要满足以下约束：

1. 总接收量

有关不同商家的损耗率，通过对附件二的转运损耗率计算标准差发现（方差十分小，因此使用标准差），转运商的损耗率波动幅度均较小，精度在 $10^{-2} \sim 10^{-3}$ ：

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
0.00114	0.01372	0.01727	0.15626	0.31912	0.03708	0.05310	0.02337

表格 4：不同转运商转运损耗率的标准差

因此使用近 5 年各转运商的平均损耗率作为未来 24 周该转运商的运输损耗率。由于在前文中选择的都是供货数量较稳定的供货商，因此使用不同供应商近 5 年来每周平均供货量作为能够订货的总量，即前文的第 i 家供货商的供货能力 E_i 。实际情况下，企业每周对于不同原材料的总接受量与供货商的供货量、所选择的运货损耗率有关。设该实际总接受量为 A ，则得到下式：

$$A_k(t) = \sum_{D_T \in k} \sum_{j=1}^8 E_i x_i (1 - L_j(t)) y_{ij}, \quad k \in D_T \quad (2-6)$$

其中， k 表示是原材料供应类别， E_i 表示第 i 家供货商的供货能力， x_i 是决策是否在该家供货商订货， $L_j(t)$ 表示在第 t 周内第 j 家转运商的转运损耗率， y_{ij} 决策第 i 家供货商是否选择第 j 家转运商来运送原材料，选择该家转运商则为 1，不选择则为 0；通过上述变量即可表示出未来 24 周每周的不同原材料类别的实际总接收量。

2. 总生产量

该企业每周要保证 $2.82 \times 10^4 m^3$ 的产品被生产，设第 t 周不同原材料的总使用量为 $U_k(t)$ ， $k \in \{A, B, C\}$ 。则下面关于不同原材料生产出总产品量的表达式：

$$\begin{cases} \frac{U_A(t)}{0.6} + \frac{U_B(t)}{0.66} + \frac{U_C(t)}{0.72} = 28200 \\ U_k(t) \geq 0, k \in D_T \end{cases} \quad (2-7)$$

3. 使用量 U 与剩余量 R 之间关系

设每周的原材料剩余量为 $R(t)$ ，在数值上等于该周的企业对原材料的总接收量与上周剩余量的总和，再减去该周的产能需要的原材料总量后剩余的原材料数量。如下式所示：

$$R_k(t) = A_k(t) + R_k(t-1) - U_k(t), \quad k \in D_T \quad (2-8)$$

4. 原材料库存

该企业尽可能保持不少于满足两周生产需求的原材料库存。因此有如下约束条件：

$$R_A(t) + R_B(t) + R_C(t) \geq 56400 \quad (2-9)$$

5. 运输条件限制

每家转运商的运输能力为每周 $6 \times 10^3 m^3$ 。考虑到转运商只有 8 家，远大于供应商数量，因此一家转运商被认为可以运输多家供应商提供的原材料供应量。所以第 j 家转运商每周转运来自不同供应商供应的原材料总量不大于 $6 \times 10^3 m^3$ ，约束表达式如下：

$$\sum_{i=0}^{21} E_i x_i y_{ij} \leq 6000, j = 1, 2, \dots, 8 \quad (2-10)$$

6. 供货商对应的转运商唯一

虽然每家转运商可以运输不同供应商供应的原材料，但是由于题目要求，每个供货商提供的原材料应当仅由一家转运商进行运输，因此有如下限制条件：

$$0 \leq \sum_{j=1}^8 y_{ij} \leq 1 \quad (2-11)$$

6.3.3 目标函数

据上述的约束条件，要选择最经济的订购方案以及损耗最少的转运方案。在选择订购方案时，不需要考虑转运以及存储的费用，因此可得到如下表达式：

$$\min \sum_{k \in D_{T_i}} \sum_{i \in k} E_i x_i C_k \quad (2-12)$$

其中 C_A ， C_B ， C_C 分别是 A，B，C 的原材料购买价格。

选择转运方案时，在最经济订购方案的条件下，根据题中要求要使选择的转运方式所损耗的材料最少，则在之前的要求下改变(2-13)即可。得到最小损耗方案的目标函数如下：

$$\min \sum_{k \in D_{T_i}} \sum_{i \in k} \sum_{j=1}^8 E_i x_i C_k L_j(t) y_{ij} \quad (2-13)$$

6.3.4 模型建立

考虑到每周的规划模型其约束受到前一周订货方案的影响，主要体现在每周接收量的差异以及原材料消耗量的差异，导致的库存变化，即每周的订货方案安排是受到上一周生产消耗情况的影响的。因此在求解订货与转运方案时，逐周对上式模型进行求解，先求出第 $t-1$ 周的订购方案，在基于订购方案得到本周的转运方案，更新第 $t-1$ 周的库存情况，作为第 t 周的求解基础。

在求解订购方案时，基于题目信息设定初始条件。第一周时仓库的库存是充足的，即每初始库存中已有原材料可生产 $2 \times 2.82 \times 10^4 m^3$ 产品。在订购方案的模型中，考虑了第 t 周供货商 S_i 供货时的转运损耗率 $L_j(t)$ 。

基于上述对模型的简化，本文通过遗传算法对每周的订购方案和转运方案模型进行逐周求解，遗传算法将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时，相对一些常规的优化算法，通常能够较快地获得较好的优化结果。

因此基于目标函数(2-12,13)以及约束条件(2-6,7,8,9,10,11)建立非线性规划的订购方案模型为：

$$\begin{aligned}
& \min \sum_{k \in D_{T_i}} \sum_{i \in k} E_i x_i C_k, \quad \min \sum_{k \in D_{T_i}} \sum_{i \in k} \sum_{j=1}^8 E_i x_i C_k L_j(t) y_{ij} \\
& \text{s. t. } \left\{ \begin{aligned}
& A_k(t) = \sum_{D_{T_i} \in k} \sum_{j=1}^8 E_i x_i (1 - L_j(t)) y_{ij} \\
& \frac{U_A(t)}{0.6} + \frac{U_B(t)}{0.66} + \frac{U_C(t)}{0.72} = 28200 \\
& R_k(t) = A_k(t) + R_k(t-1) - U_k(t) \geq 0 \\
& R_A(t) + R_B(t) + R_C(t) \geq 56400 \\
& \sum_{i=0}^{21} E_i x_i y_{ij} \leq 6000, j = 1, 2, \dots, 8 \\
& 0 \leq \sum_{j=1}^8 y_{ij} \leq 1 \\
& U_k(t) \geq 0, k \in D_T
\end{aligned} \right. \quad (2-14)
\end{aligned}$$

6.4 订购与转运方案模型求解

6.4.1 结果展示

通过遗传算法模型，对于订购方案以及转运方案求出结果，详细结果见附件 A、附件 B。

6.4.1 订购成本分析

根据题干知，原材料的成本单价之比为：

$$A:B:C = 1.2:1.1:1$$

因此此处并没有给出实际的单价，只给出了不同原材料的成本单价之比。假设 C 成本单价为1 元/ m^3 ，相应 A、B 的成本单价为1.2 元/ m^3 、1.1 元/ m^3 。因此我们在考虑购买成本时可以把三种原材料的购买成本归纳成一个总订购成本的指标。依据附件 A 中的订购方案，分别求得 24 周每周的原材料总购买成本，见下表。

周数	总订购成本	周数	总订购成本
W1	20879.21735	W13	20879.21735
W2	20879.21735	W14	20249.94434
W3	20245.72256	W15	20879.21735
W4	20879.21735	W16	20879.21735
W5	20324.4997	W17	20290.19154
W6	20879.21735	W18	20879.21735
W7	20337.05923	W19	20320.24729
W8	20879.21735	W20	20879.21735
W9	20320.24729	W21	20353.11522
W10	20320.24729	W22	20879.21735
W11	20879.21735	W23	20879.21735
W12	20879.21735	W24	20245.8239

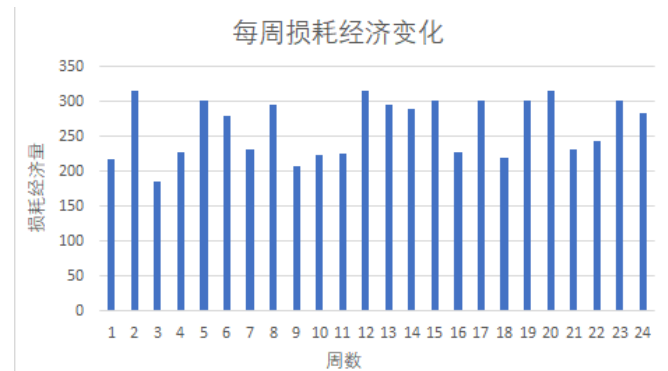
图表 5：各周原材料的购买成本

根据表格情况，可以得出最低的总订购成本 20245.72 元左右，最高的总订购成本在 20879.21 元左右。所得到的总订购成本相对来说平稳，波动不大。

6.4.2 损耗率分析对比

依据上述假设的成本单价，因此可以通过成本单价乘以每周的原材料损耗量得到更为精确的损耗经济成本，这也与上面的购买成本相对应。

依据附件 B 中的运输方案，分别求得 24 周内每周的原材料损耗量，绘制成柱形图见下表。分析可得损耗经济一直维持在稳定的较低状态，与转运损耗最少方案的目的相同。



图表 6：每周损耗经济变化

七、问题三 单目标规划新订购方案

7.1 建立评价模型

7.1.1 模型分析

题目引入压缩成本的考虑，要求减少企业在转运以及存储环节所花费的成本。因此提出要求尽可能多地采购 A 类原材料，尽可能少地采购 C 类原料。由于 A 类原料用于制造产品的用量是最少的，C 类原料用于制造产品的用量是最多的，因此可知，企业希望使用较少体积的原材料来进行产品的生产，所购买的原材料体积减少，在转运和存储环节花销的成本就会随之变少。

7.1.2 目标函数的构造

在问题三中，本文提出了损耗经济的概念，它在数值上等于不同原材料的成本单价乘以相应的损耗总量。损耗经济的指标将不同原材料的成本纳入考虑范围，相比于仅考虑整体原材料总的损耗量更为精确。设损耗经济为 LC 。

1. 增 A 减 C 的订购量

考虑到现实的情况，本文不考虑在多买 A 少买 C 的情况下订购 B 的量不变的情况。三种原材料的订购量应是相互关联的，所以不做上述微小可能性的分析。本文考虑的目标函数与损耗经济、A 与 C 的订购量相关。此处设不同原材料的订购量分别为 V_A 、 V_B 、 V_C 。由于本体中要求增 A 减 C，因此目标函数应与 B 的订购量无关。为简化分析，在此对目标函数主观赋权如下：

$$\min 0.6LS - 0.2V_A + 0.2V_C \quad (3-1)$$

目标函数选择求取最小化，因此对于 V_A 的权重选取负值； V_A 与 V_C 的权重应同等重要，因此将权值相等。

7.1.4 基于单目标优化的模型

在本题中若使用多目标规划，其迭代时间会较长且产生不可行解，因此我们考虑采取单目标规划。考虑到问题三与问题二均为优化模型，问题三是在问题二的模型上做了一些修改，增添了多订购 A 原材料少订购 C 原材料的需求。因此本问约束条件与问题二中的约束条件相同。基于目标函数(3-1)以及约束条件(2-6,7,8,9,10,11)建立单目标的订购模型为：

$$\begin{aligned} & \min 0.6LS - 0.2V_A + 0.2V_C \\ & \text{s. t.} \begin{cases} A_k(t) = \sum_{D_{T_i} \in k} \sum_{j=1}^8 E_i x_i (1 - L_j(t)) y_{ij} \\ \frac{U_A(t)}{0.6} + \frac{U_B(t)}{0.66} + \frac{U_C(t)}{0.72} = 28200 \\ R_k(t) = A_k(t) + R_k(t-1) - U_k(t) \geq 0 \\ R_A(t) + R_B(t) + R_C(t) \geq 56400 \\ \sum_{i=0}^{21} E_i x_i y_{ij} \leq 6000, j = 1, 2, \dots, 8 \\ 0 \leq \sum_{j=1}^8 y_{ij} \leq 1 \\ U_k(t) \geq 0, k \in D_T \end{cases} \end{aligned} \quad (3-2)$$

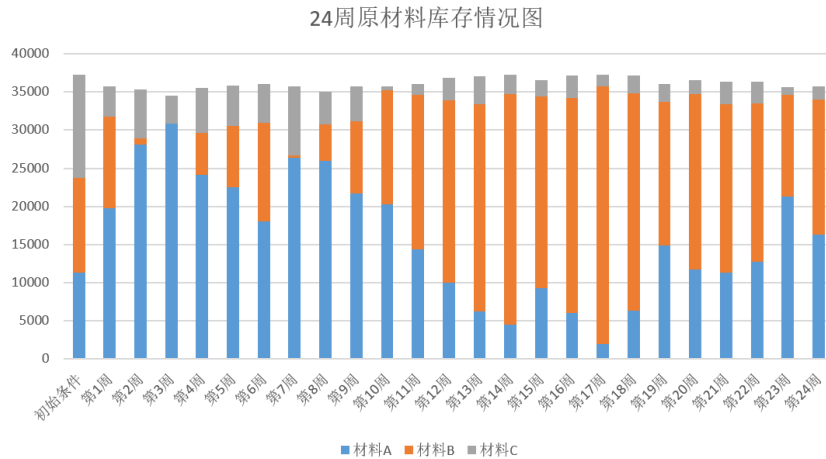
7.2 模型求解

求解结果详见附件 A。

7.2.1 结果展示与分析

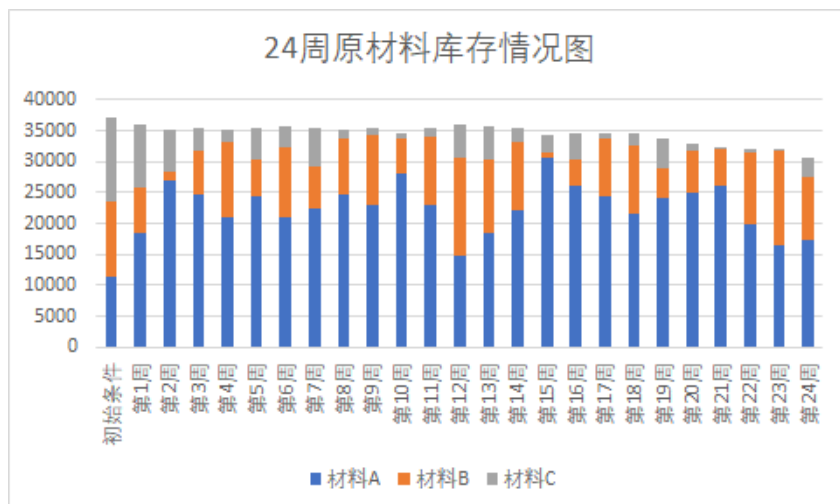
将基于问题三新需求所构建的模型，在此称为新模型。下文将对新模型与问题二原有模型构建出的结果进行比较。

下图是问题二模型所构建出的在最优方案下关于预测未来 24 周内不同类别的原材料每周的剩余量：



图表 7：问题 2 所得不同原材料库存情况

下图是问题三模型所构建出的在最优方案下关于预测未来 24 周内不同类别的原材料每周的剩余量：



图表 8：问题 3 不同原材料库存情况

可以分析，该模型有效地提高了 A 的订购量并减少了 B 的订购量。同时 B 的订购量也发生了相应的变化，B 的订购量显著减少，这也与前面模型的建立分析相符合：三种原材料的订购量应是相互关联的，B 的订购量不变的情况的可能性很低。剩下的相关方案详见附件 A，附件 B。

八、问题四 产能扩大生产

该问中提及：生产企业通过技术改造已经具备了产能提升的潜力，即确保该生产企业有足够的应对供货商的不稳定性，即不会出现库存的积压导致成本急剧上升的情况；同时由于供货商的增加，生产企业的常规最大产能，即 2.82 万立方米，是更加容易满足的。基于如上分析，本文在问题四的模型中将考虑全部的 402 家供货商。

8.1 建立评价模型

8.1.1 模型建立

问题四可以分为进货和投入生产这两个相互独立的部分。题干中描述为通过技术改造来提高产能，因此每周的订货量会增加，库存量也会随之提升。订货量的大幅提升对于产能提高有显著影响，然而运输商所能运送原材料的体积有限，这对产能的提升上限有所限制。

在考虑该企业产能的提升限度时，应考虑稳定状态下的订供货情况。由于企业在产能升级时需要扩大库存容量，此时的产能不能阶跃式提高，而是需要一段生产时间后才达到稳定。达到稳定后可以认为每周企业收到的原材料的等效产能就是其新的最大产能，在第 t 周收到原材料的等效产能可为：

$$M(t) = \frac{U_A(t)}{0.6} + \frac{U_B(t)}{0.66} + \frac{U_C(t)}{0.72} \quad (4-1)$$

最大产能的目标函数即为：

$$\max M(t) \quad (4-2)$$

8.1.2 约束条件的构造

1.总运输量的限制

从问题一中，已筛选出的最重要的前 50 家供应商，在问题二以及问题三的成本优化模型中均取得了较好的效果。由于本问着重于提高产能，那么对于供应商的选择应被放宽，选择更多的供应商来获得更多的供货量。故本问中所需原材料可在放宽条件下的 402 家供货商中选择需要运输的原材料总量，即供应商供应产品总量。

$$A'_k(t) = \sum_{i \in k} E_i x_i, k \in D_T \quad (4-3)$$

其中，k 表示不同的材料种类。

每家转运商的运输能力为每周转运 $6 \times 10^3 m^3$ ，共有 8 家转运公司。因此每周总运输量应不大于 8 家转运商的总运输能力，即：

$$\sum_{k \in D_T} A'_k(t) \leq 48000 \quad (4-4)$$

2.最值产能限制：

由于本题模型的主要目的是提高产能，所以对具备产能极大提升潜力的生产企业，考虑将其原产能 $2.82 \times 10^4 m^3$ 设为每周产能的最小值，规定每周生产量不得少于原产能 $2.82 \times 10^4 m^3$ 。

由于在本问中运输损耗不作为优化目标，因此进行部分松弛操作，在前两问中通过计算得到，所有转运商的转运损耗率均处于 1%~5% 之间，因此不同转运商的损耗率之间的微小差别对最大产能的影响很小，所以在第四问中采用八家转运商的平均损耗率来对损耗率进行估计，得到平均总损耗率 1.39%。设平均总损耗率为 $\bar{L} = 1.39\%$ 。

为了防止每周的生产消耗的材料数量过大导致库存减少，不满足储货需求以及后期的产能要求，需要对产能最大值限制。假设运送的所有原材料均为使用体积最少的 A 类材料，减去运输中损耗部分，此时产能为 $\frac{6000 \times 8(1-\bar{L})}{0.6} = 80000(1-\bar{L})$ ，即为产能的最大值，可知每周将所有运输来的原材料进行生产，产能也不会超过上述值，因此可得：

$$28200 \leq M \leq 80000(1-\bar{L}) \quad (4-5)$$

3.库存数量限制

库存量至少要为产能的 2 倍才能保证每周生产的稳定性，企业要求库存量至少为两周的生产原材料量。

$$R_k(t) = (1-\bar{L})A'_k(t) - U_k(t) + R_k(t-1) \geq 0, k \in D_T \quad (4-6)$$

8.1.3 基于线性规划的产能提升模型

结合目标函数(4-2)以及约束条件(4-3,4,5,6)得到包括 0-1 整数规划之内的混合线性规划模型：

$$s. t. \begin{cases} \max M(t) = \frac{U_A(t)}{0.6} + \frac{U_B(t)}{0.66} + \frac{U_C(t)}{0.72} \\ A'_k(t) = \sum_{i \in k} E_i x_i, k \in D_T \\ \sum_{k \in D_T} A'_k(t) \leq 48000 \\ 28200 \leq M \leq 8000(1-L) \\ R_k(t) = (1-\bar{L})A'_k(t) - U_k(t) + R_k(t-1) \geq 2M, k \in D_T \\ U_A(t), U_B(t), U_C(t) \geq 0 \end{cases} \quad (4-7)$$

其中，需要注意的是对于 t 从 1 到 24 共 24 个取值对应着 24 个不同的混合线性规划，

因此有 24 个最大原材料的等效产能。

8.2 模型求解

在第四问中订购方案和转运方案模型虽然存在一定的更新，但是从模型结构以及约束条件的角度看，模型并未产生本质上的变化，因此同样可以考虑使用遗传算法对模型进行求解，所需要的限制条件以及目标函数如上文所示，求解结果如下：

时间	最大总产能	时间	最大总产能
W1	31282.86423	W13	37924.17728
W2	33513.83007	W14	37941.37211
W3	35001.14064	W15	37952.83533
W4	35992.68101	W16	37960.47747
W5	36653.70793	W17	37965.57224
W6	37094.39254	W18	37968.96875
W7	37388.18228	W19	37971.23309
W8	37584.04211	W20	37972.74265
W9	37714.61533	W21	37973.74902
W10	37801.66414	W22	37974.41993
W11	37859.69668	W23	37974.86721
W12	37898.38504	W24	37975.1654

表格 5：24 周最大总产能

8.3 结果分析

根据所得结果，将得到的最大总产能按周序绘制为如下折线图。

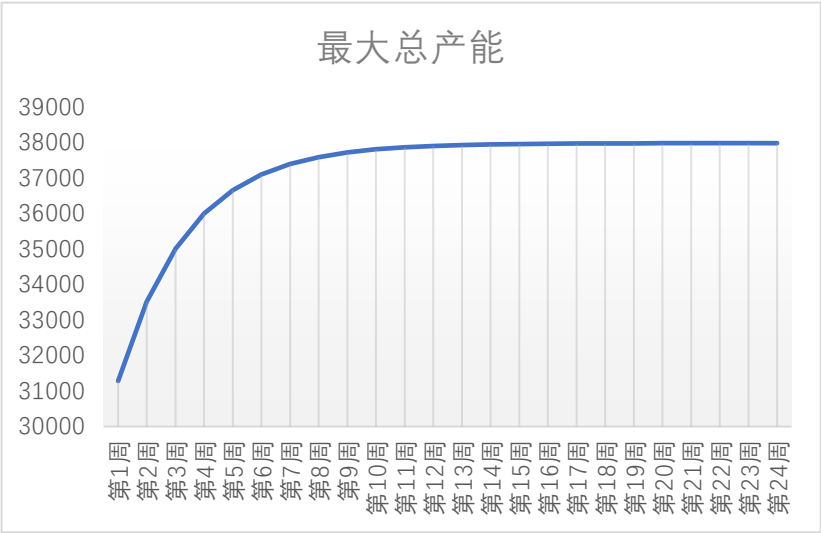


表 2 24 周最大总产能变化图

初步分析该曲线为增速有快减慢，最后值趋于 0 的曲线。表示着最大产能是一个由少到多，并逐渐稳定在一个定值附近的形势。由于在逐周计算最大产能时，根据实际情况将原材料初始库存量设为原最大产能的两倍，即 $5.64 \times 10^4 m^3$ 。在第1~11周左右，由于库存原材料数量的限制，即库存材料量至少要保证两倍的产能，导致计算所得的最大产能较小。

而在逐周的进货材料积累下，库存量逐渐充裕，最大产能逐渐升高，在库存材料量逐渐达到每周可提供材料最大产能的两倍后，最大产能趋向于稳定。这也与前面分析由于企业在产能升级时需要扩大库存容量，此时的产能不能阶跃式提高，而是需要一段生产时间

后才达到稳定相符合。后面趋向定值的原因是：由于生产产品的 A 原料所消耗的体积较少，在转运量固定的情况下，会选择更多的购入 A 原料。

同时，经过计算发现，402 家供货商所提供的供货总量小于转运商所能转运的体积，因此后面限制产能提升的主要原因是供货商的供货量，因此制约了生产企业产能的提升。

九、模型评价

9.1 优点分析

一、多种评价指标方式协同确定

本文在确定评价指标时，进行了全面细致的考虑，贴合实际情况。关注对于生产企业来说供应商的重要程度。诚信度以及依赖度的指标都是供应商相对于该生产企业的供货特征，并不能象征性代表供应商的总体特征。同时，在分析订购以及转运方案时，详细地考虑到了各种约束条件，符合实际情况。

二、规划模型中创新性地加入迭代约束

由于迭代关系的加入，当某一周订货量过大导致剩余量过大时，下一周的订货量会自动减小以减小下一周的剩余量，这使得模型更加稳定。除此之外，逐周对模型进行分析也使得下一周的方案是根据上一周的结果而产生的，使模型具有了动态特性，在更好贴合实际的情况下完成预测。

三、问题解决后结果分析反映了一定实际意义

问题四得到每周能达到的最大产能后，我们发现运输量仍然没有达到边界，而最大产能却趋于定值，由此分析得出该企业最大产能主要受供货商的供货能力的约束，当供货商能力提高后才会收到最大运输量的约束，这对企业提高产能的计划有较强的指导意义。

9.2 缺点分析

一、优化订购方案时舍弃低排名的供货商

在问题二、三的求解中，本文为了模型的简化，依据问题一所建立的重要性评价模型进行供应商的选择，舍弃了重要性不高的供应商。实际上在选择供应商时应该同时考虑大订货量和小订货量的供应商，本文进行求解时，进行的处理是在问题一中得分较低的小型供应商们在问题二、三的最佳订购方案中基本考虑其订单量。在实际情况中，并不一定时最优解。

二、遗传算法带来不确定性影响

遗传算法在更真实更科学的同时，同样也有一些问题，最主要的一点即随着迭代次数的不同以及每次计算完一周的方案后得到的是一组最优解的集合，为了下面几周的计算，需要在其中随机选取一组最优解再进行下一次的迭代，使得最终得出的方案具有不确定性，每次求得的方案可能不一样。

十、参考文献

- [1]张霞,何南.综合评价方法分类及适用性研究[J].统计与决策,2022,38(06):31-36.DOI:10.13546/j.cnki.tjyj.2022.06.006.
- [2]宋玥.运筹学在企业运输成本优化方面的应用[J].商,2016,(10):3.
- [3]武慧栋,冯琴荣.基于遗传算法的覆盖近似空间最优近似求解算法[J].山西师范大学学报(自然科学版),2022,36(01):10-15.DOI:10.16207/j.cnki.1009-4490.2022.01.002.
- [4]鲁春阳,文枫,杨庆媛,陈琳琳,宗会明.基于改进 TOPSIS 法的城市土地利用绩效评价及障碍因子诊断——以重庆市为例[J].资源科学,2011,33(03):535-541.
- [5]崔杨,郭福音,仲悟之,赵钰婷,付小标.多重不确定性环境下的综合能源系统区间多目标优化调度[J/OL].电网技术:1-12[2022-08-08].DOI:10.13335/j.1000-3673.pst.2021.1612.

附录

详细的符号说明:

符号	含义说明	符号	含义说明
N_i	第 <i>i</i> 家供应商	N_s	供货商的供货量
T_j	第 <i>j</i> 家转运商	N_o	企业订货量
D_H	供货诚信度	L_j	第 <i>j</i> 家转运商的转运损耗率
B	诚信度偏移量矩阵	Dis_i	偏移量 <i>B</i>
D_R	企业订货依赖度	Sum_A	近 5 年 A 原料总订货量
Sum_B	近 5 年 B 原料总订货量	Sum_C	近 5 年 C 原料总订货量
R_i	近 5 年第 <i>i</i> 家供应商供应量占某原料总订货量的比例	y_{ij}	第 <i>i</i> 家供货商是否选择第 <i>j</i> 家转运商来运送原材料
D_T	原材料供应类别	r	诚信度与依赖度的相关系数
S_i	供应商重要性得分	C_H	CRITIC 诚信度的信息度
w_H	重要性得分中诚信度所占权重	C_R	CRITIC 依赖度的信息度
w_R	重要性得分中依赖度所占权重	σ_H	诚信度的标准差
x_i	企业是否向第 <i>i</i> 家供货商订货	σ_R	依赖度的标准差
E_i	第 <i>i</i> 家供货商的供货能力	F_i	供货商的供货产能能力
A	企业总接受原料量	LS	损耗经济
R	每周企业原料剩余量	U	企业每周原料的使用量
M	企业等效产能	\bar{L}	平均总转运损耗率

表格 6: 符号说明

Matlab 代码:

问题一&问题二代码:

主程序:

```

clc,clear

%Qestion 1

%导入根据类型排序后的订购量和供货量
load 根据类型排序后的订购量和供货量及对应的供货商 id.mat
global headA tailA headB tailB headC tailC rows cols E leastChoices CA CB
CC s lastRest1 lastRest2
%syms CA CB CC
CA=1.2; CB=1.1; CC=1;
[rows cols]=size(order);
headA=1;tailA=146;    %1 到 146 行为 A 材料供货商
headB=147;tailB=280; %147 到 280 为 B 材料供货商
headC=281;tailC=402; %281 到 402 为 C 材料供货商

%求诚信度和依赖度
[DH,DR]=get_DH_DR(order,supply);
%求诚信度和依赖度的权重
[WH,WR]=get_WH_WR(DH,DR);
%score 为他们的综合评分
score=WH*DH+WR*DR;
%找出前 50 名在列向量中的下标,存储在 goodTrader 的第一列
goodTrader=zeros(50,2);

```

```

goodTrader(:,1)=getGoodTrader(score);

%Question 2

%获得 402 个供货商的供货能力（平均供货量）向量
E=getE(order,supply); %得到每个供货商平均供货量
cE=zeros(402,1); %cE 为将元素数值转换为对应能生产的产品数后的向量
cE(headA:tailA)=E(headA:tailA)/0.6;
cE(headB:tailB)=E(headB:tailB)/0.66;
cE(headC:tailC)=E(headC:tailC)/0.72;

%前 50 名供货商的平均供货量对应能生产的产品数存储在 goodTrader 第二列
for i=1:50
    goodTrader(i,2)=cE(goodTrader(i,1));
end
sorted_GoodTrader=sortrows(goodTrader,-2);
sorted_GoodTrader(:,3)=zeros(50,1);
sorted_GoodTrader(1,3)=sorted_GoodTrader(1,2);
%求前缀和（包括该元素）
for i=2:50
    sorted_GoodTrader(i,3)=sorted_GoodTrader(i-
1,3)+sorted_GoodTrader(i,2);
end

%最少选择的供货商
leastChoices=sorted_GoodTrader(find(sorted_GoodTrader(:,3)<=28200),1);
leastChoices=[leastChoices;sorted_GoodTrader(length(leastChoices)+1,1)];
id_leastChoices=id(leastChoices);
fprintf('最少选择%d 家供应商，他们的编号如下:\n',length(leastChoices))
disp(id_leastChoices')

%最经济订购方案求解

%还要算出当周 rest
s=[1.904769167,0.921370417,0.186055556,1.570482353,2.889825301,0.54376111
1,2.078833333,1.010282759]/100;
nowRest1=zeros(25,3);
x1=zeros(24,21);
y1=zeros(21,8,24);
use1=zeros(24,3);
fval1=zeros(24,1);
nowRest1(1,:)= [28200*0.6*2/3,28200*0.66*2/3,28200*0.72*2/3];
lastRest1=zeros(1,3);
nowRest2=zeros(25,3);
x2=zeros(24,21);
y2=zeros(21,8,24);
use2=zeros(24,3);
fval2=zeros(24,1);
nowRest2(1,:)= [28200*0.6*2/3,28200*0.66*2/3,28200*0.72*2/3];
lastRest2=zeros(1,3);
% fg1=figure();
% set(fg1,'position',get(0,'ScreenSize'));
for i=1:24
    % [x1(i,:),y1(:, :, i),use1(i,:),leastCostFval(i),nowRest1(i+1,:)] = least
CostProg(nowRest1(i,:),s);
    lastRest1=nowRest1(i,:);
    lastRest2=nowRest2(i,:);
    [x1(i,:),y1(:, :, i),gax1,gafval1,use1(i,:),nowRest1(i+1,:)] ...

```

```

        ,x2(i,:),y2(:,i),gax2,gafval2,use2(i,:),nowRest2(i+1,:)]=gaProg
    ());
    fval1(i,1)=gafval1(1,:);
    fval2(i,1)=gafval2(1,:);
end
figure(1)
bar([1:24]',fval1(:,1),'FaceColor',[0 0.4470 0.7410]);
title('每周订购花费随时间变化图像')
xlabel('Week')%横轴标签
ylabel('订购花费')%左纵轴标签
figure(2)
bar([1:24]',fval2(:,1),'FaceColor',[0.6350 0.0780 0.1840]);
title('每周损耗经济随时间变化图像')
xlabel('Week')%横轴标签
ylabel('损耗经济')%左纵轴标签

```

应用 Topsis 法计算诚信度和依赖度的函数代码：

```

%求诚信度和依赖度

function [DH,DR]=get_DH_DR(order,supply)
global headA tailA headB tailB headC tailC rows cols
offset=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        if order(i,j)==0
            continue;
        end
        offset(i,j)=abs(supply(i,j)-order(i,j))/order(i,j);
    end
end

%topsis 计算诚信度

%正理想解为全 0 的行向量
PIS=zeros(1,cols);
%负理想解所有元素为整个矩阵中的最大值的行向量
NIS=zeros(1,cols);
NIS(1,:)=max(offset,[],'all');
%经过验证负理想解就是全 1 的行向量

%计算各个供货商的综合评价价值

%计算各个被评价对象到 PIS 的欧氏距离
dPIS=zeros(rows,1);
for i=1:rows
    dPIS(i,1)=sqrt(sum((offset(i,:)-PIS(1,:)).^2));
end

%计算各个被评价对象到 NIS 的欧氏距离
dNIS=zeros(rows,1);
for i=1:rows
    dNIS(i,1)=sqrt(sum((offset(i,:)-NIS(1,:)).^2));
end

%计算各评价对象的综合评价价值(诚信度)
DH=zeros(rows,1);

```

```

DH=dNIS./(dPIS+dNIS);

%计算依赖度

sumOrder=sum(order,2); %计算每个供货商的总订购量
sumOrderInType=zeros(3,1); %A,B,C 三种材料的总订购量，sumOrderInType(1)表示
A 的总订购量
sumOrderInType(1)=sum(sumOrder(headA:tailA));
sumOrderInType(2)=sum(sumOrder(headB:tailB));
sumOrderInType(3)=sum(sumOrder(headC:tailC));

%计算向该供应商订购的货物数在该类型中的总订购货物中的占比
R=zeros(rows,1);
R(headA:tailA)=sumOrder(headA:tailA)/sumOrderInType(1);
R(headB:tailB)=sumOrder(headB:tailB)/sumOrderInType(2);
R(headC:tailC)=sumOrder(headC:tailC)/sumOrderInType(3);

%得到依赖度 DR
maxR=max(R);
DR=R/maxR; %依赖度
end

```

CRITIC 法计算诚信度和依赖度的权重代码:

```

%CRITIC 法求诚信度和依赖度的权重
function [WH,WR]=get_WH_WR(DH,DR)
SD=std([DH,DR],0); %方差
r=corrcoef([DH,DR]); %相关系数矩阵
W=SD.*(sum(1-r));
W=W./sum(W);
WH=W(1);WR=W(2);
End

```

给打分后的供货商排序的函数代码:

```

function goodTrader=getGoodTrader(score)
[val,idx]=sort(score,1,'descend');
goodTrader=idx(1:50);
end

```

计算 402 个供货商的供货能力的函数代码:

```

%计算 402 个供货商的供货能力

function E=getE(order,supply)
global rows cols
E=zeros(rows,1);

%只考虑供货量/订货款>=0.9 的供货数据，找到这些数据的平均值就是该供货商的供货能力
rate=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        if order(i,j)==0
            rate(i,j)=0;
            continue;
        end
        rate(i,j)=supply(i,j)/order(i,j);
    end
end

```

```

end
end
for i=1:rows
    fd=find(rate(i,:)>=0.9);
    tmp=[];
    for j=1:length(fd)
        tmp=[tmp,supply(i,fd(j))];
    end
    if isempty(tmp)
        continue;
    end
    E(i)=mean(tmp);
end
end

```

使用遗传算法对问题二模型求解的函数：

```

function
[x1,y1,gax1,gafval1,use1,nowRest1,x2,y2,gax2,gafval2,use2,nowRest2]=gaProg()
%function [x,y,use,fval,nowRest]=gaProg()
%tx(1)~tx(21)是 x 矩阵
%{
[tx(22)~tx(42);
tx(43)~tx(63);
tx(64)~tx(84);
tx(85)~tx(105);
tx(106)~tx(126);
tx(127)~tx(147);
tx(148)~tx(168);
tx(169)~tx(189);]
为 y 矩阵
%}
%tx(190)~tx(192)是 useA,useB,useC
global E leastChoices headA headB headC tailA tailB tailC s lastRest1
lastRest2
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
ns=42;
nvar=192;
A=zeros(ns,nvar);
B=zeros(ns,1);
%每个供货商的转运商唯一
for i=1:21
    A(i,(22+(i-1)):21:(169+(i-1)))=-1;
    B(i)=0;
end

for i=1:21
    A(21+i,(22+(i-1)):21:(169+(i-1)))=1;
    B(21+i)=1;
end
Aeq=zeros(1,192);
Aeq(190:192)=[1/0.6,1/0.66,1/0.72];
Beq=[28200];
lb=zeros(192,1);

```

```

ub=ones(192,1);
ub(190:192)=2*28200;
intcon=[1:189];
options=optimoptions('ga');
options.FunctionTolerance=1e-6;
options.MaxStallGenerations=200;
Fun=@gaFun_LeastPay;
nonlcon=@gaNonlcon1;
[gax1,gafval1]=ga(Fun,nvar,A,B,Aeq,Beq,lb,ub,nonlcon,intcon,options);
Fun=@gaFun_LeastLoss;
nonlcon=@gaNonlcon2;
[gax2,gafval2]=ga(Fun,nvar,A,B,Aeq,Beq,lb,ub,nonlcon,intcon,options);

%
[gax,fval]=gamultiobj(Fun,nvar,A,B,Aeq,Beq,lb,ub,nonlcon,intcon,optimoptions('gamultiobj'));

x1=gax1(1,1:21)';
y1=[gax1(1,22:42)',gax1(1,43:63)',gax1(1,64:84)',gax1(1,85:105)',gax1(1,106:126)',gax1(1,127:147)',gax1(1,148:168)',gax1(1,169:189)'];
use1=gax1(1,190:192)';
nowRest1=lastRest1;
for i=1:length(SA)
    for j=1:8

nowRest1(1)=nowRest1(1)+E(leastChoices(SA(i)))*gax1(SA(i))*y1(SA(i),j)*(1-s(j));
    end
end
nowRest1(1)=nowRest1(1)-use1(1);
for i=1:length(SB)
    for j=1:8

nowRest1(2)=nowRest1(2)+E(leastChoices(SB(i)))*gax1(SB(i))*y1(SB(i),j)*(1-s(j));
    end
end
nowRest1(2)=nowRest1(2)-use1(2);
for i=1:length(SC)
    for j=1:8

nowRest1(3)=nowRest1(3)+E(leastChoices(SC(i)))*gax1(SC(i))*y1(SC(i),j)*(1-s(j));
    end
end
nowRest1(3)=nowRest1(3)-use1(3);

x2=gax2(1,1:21)';
y2=[gax2(1,22:42)',gax2(1,43:63)',gax2(1,64:84)',gax2(1,85:105)',gax2(1,106:126)',gax2(1,127:147)',gax2(1,148:168)',gax2(1,169:189)'];
use2=gax2(1,190:192)';
nowRest2=lastRest2;
for i=1:length(SA)
    for j=1:8

nowRest2(1)=nowRest2(1)+E(leastChoices(SA(i)))*gax2(SA(i))*y2(SA(i),j)*(1-s(j));
    end
end
end

```



```

nowRest2(1)=nowRest2(1)-use2(1);
for i=1:length(SB)
    for j=1:8

nowRest2(2)=nowRest2(2)+E(leastChoices(SB(i)))*gax2(SB(i))*y2(SB(i),j)*(1-s(j));
    end
end
nowRest2(2)=nowRest2(2)-use2(2);
for i=1:length(SC)
    for j=1:8

nowRest2(3)=nowRest2(3)+E(leastChoices(SC(i)))*gax2(SC(i))*y2(SC(i),j)*(1-s(j));
    end
end
nowRest2(3)=nowRest2(3)-use2(3);
end

```

订购经济最小化的目标函数:

```

%tx(1)~tx(21)是 x 矩阵
%{
[tx(22)~tx(42);
tx(43)~tx(63);
tx(64)~tx(84);
tx(85)~tx(105);
tx(106)~tx(126);
tx(127)~tx(147);
tx(148)~tx(168);
tx(169)~tx(189);]
为 y 矩阵
%}
%tx(190)~tx(192)是 useA,useB,useC

function y=gaFun_LeastPay(tx)
global E leastChoices headA headB headC tailA tailB tailC CA CB CC
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
y(1)=E(leastChoices(SA(:)))*tx(SA(:))*CA+E(leastChoices(SB(:)))*tx(SB(:))*CB+E(leastChoices(SC(:)))*tx(SC(:))*CC;
end

```

订购经济最小化的非线性限制条件函数:

```

function [c,ceq]=gaNonlcon1(x)
global E leastChoices headA headB headC tailA tailB tailC s lastRest1
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
c(1)=x(190)-lastRest1(1);
for i=1:length(SA)
    for j=1:8
        c(1)=c(1)-E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-1)*(1-s(j));
    end
end

```

```

end
c(2)=x(191)-lastRest1(2);
for i=1:length(SB)
    for j=1:8
        c(2)=c(2)-E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-1)*(1-s(j));
    end
end
c(3)=x(192)-lastRest1(3);
for i=1:length(SC)
    for j=1:8
        c(3)=c(3)-E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-1)*(1-s(j));
    end
end
c(4)=2*28200+x(190)/0.6+x(191)/0.66+x(192)/0.72-lastRest1(1)/0.6-lastRest1(2)/0.66-lastRest1(3)/0.72;
for i=1:length(SA)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-1)*(1-s(j)))/0.6;
    end
end
for i=1:length(SB)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-1)*(1-s(j)))/0.66;
    end
end
for i=1:length(SC)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-1)*(1-s(j)))/0.72;
    end
end
for j=1:8
    c(4+j)=-6000;
    for i=1:length(leastChoices)
        c(4+j)=c(4+j)+E(leastChoices(i))*x(i)*x(22+(j-1)*21+i-1);
    end
end
ceq=[];
end

```

损耗经济最小化的目标函数：

```

%tx(1)~tx(21)是 x 矩阵
%{
[tx(22)~tx(42);
tx(43)~tx(63);
tx(64)~tx(84);
tx(85)~tx(105);
tx(106)~tx(126);
tx(127)~tx(147);
tx(148)~tx(168);
tx(169)~tx(189);]
为 y 矩阵
%}

```

```

%tx(190)~tx(192)是 useA,useB,useC

function y=gaFun_LeastLoss(tx)
global E leastChoices headA headB headC tailA tailB tailC CA CB CC s
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
%y(1)=E(leastChoices(SA(:)))*tx(SA(:))*CA+E(leastChoices(SB(:)))*tx(SB
(:))*CB+E(leastChoices(SC(:)))*tx(SC(:))*CC;
y(1)=0;
for i=1:length(SA)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SA(i)))*tx(SA(i))*CA*s(j)*tx(22+(j-
1)*21+SA(i)-1);
    end
end
for i=1:length(SB)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SB(i)))*tx(SB(i))*CB*s(j)*tx(22+(j-
1)*21+SB(i)-1);
    end
end
for i=1:length(SC)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SC(i)))*tx(SC(i))*CC*s(j)*tx(22+(j-
1)*21+SC(i)-1);
    end
end
end
end

```

损耗经济最小化的限制条件函数：

```

function [c,ceq]=gaNonlcon2(x)
global E leastChoices headA headB headC tailA tailB tailC s lastRest2
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
c(1)=x(190)-lastRest2(1);
for i=1:length(SA)
    for j=1:8
        c(1)=c(1)-E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-
1)*(1-s(j));
    end
end
c(2)=x(191)-lastRest2(2);
for i=1:length(SB)
    for j=1:8
        c(2)=c(2)-E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-
1)*(1-s(j));
    end
end
c(3)=x(192)-lastRest2(3);
for i=1:length(SC)
    for j=1:8
        c(3)=c(3)-E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-
1)*(1-s(j));
    end
end
end

```

```

c(4)=2*28200+x(190)/0.6+x(191)/0.66+x(192)/0.72-lastRest2(1)/0.6-
lastRest2(2)/0.66-lastRest2(3)/0.72;
for i=1:length(SA)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-
1)*(1-s(j)))/0.6;
    end
end
for i=1:length(SB)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-
1)*(1-s(j)))/0.66;
    end
end
for i=1:length(SC)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-
1)*(1-s(j)))/0.72;
    end
end
for j=1:8
    c(4+j)=-6000;
    for i=1:length(leastChoices)
        c(4+j)=c(4+j)+E(leastChoices(i))*x(i)*x(22+(j-1)*21+i-1);
    end
end
ceq=[];
end

```

问题三:

主函数代码:

```

clc,clear

%Qestion 1

%导入根据类型排序后的订购量和供货量
load 根据类型排序后的订购量和供货量及对应的供货商 id.mat
global headA tailA headB tailB headC tailC rows cols E leastChoices CA CB
CC s lastRest1
%syms CA CB CC
CA=1.2; CB=1.1; CC=1;
[rows cols]=size(order);
headA=1;tailA=146; %1 到 146 行为 A 材料供货商
headB=147;tailB=280; %147 到 280 为 B 材料供货商
headC=281;tailC=402; %281 到 402 为 C 材料供货商
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);

%求诚信度和依赖度
[DH,DR]=get_DH_DR(order,supply);
%求诚信度和依赖度的权重
[WH,WR]=get_WH_WR(DH,DR);
%score 为他们的综合评分
score=WH*DH+WR*DR;

```

```

%找出前 50 名在列向量中的下标,存储在 goodTrader 的第一列
goodTrader=zeros(50,2);
goodTrader(:,1)=getGoodTrader(score);

%Question 2

%获得 402 个供货商的供货能力（平均供货量）向量
E=getE(order,supply); %得到每个供货商平均供货量
cE=zeros(402,1); %cE 为将元素数值转换为对应能生产的产品数后的向量
cE(headA:tailA)=E(headA:tailA)/0.6;
cE(headB:tailB)=E(headB:tailB)/0.66;
cE(headC:tailC)=E(headC:tailC)/0.72;

%前 50 名供货商的平均供货量对应能生产的产品数存储在 goodTrader 第二列
for i=1:50
    goodTrader(i,2)=cE(goodTrader(i,1));
end
sorted_GoodTrader=sortrows(goodTrader,-2);
sorted_GoodTrader(:,3)=zeros(50,1);
sorted_GoodTrader(1,3)=sorted_GoodTrader(1,2);
%求前缀和（包括该元素）
for i=2:50
    sorted_GoodTrader(i,3)=sorted_GoodTrader(i-
1,3)+sorted_GoodTrader(i,2);
end

%最少选择的供货商
leastChoices=sorted_GoodTrader(find(sorted_GoodTrader(:,3)<=28200),1);
leastChoices=[leastChoices;sorted_GoodTrader(length(leastChoices)+1,1)];
id_leastChoices=id(leastChoices);
fprintf('最少选择%d 家供应商，他们的编号如下:\n',length(leastChoices))
disp(id_leastChoices')

%最经济订购方案求解

%还要算出当周 rest
s=[1.904769167,0.921370417,0.186055556,1.570482353,2.889825301,0.54376111
1,2.078833333,1.010282759]/100;
nowRest1=zeros(25,3);
x1=zeros(24,21);
y1=zeros(21,8,24);
use1=zeros(24,3);
fval1=zeros(24,1);
nowRest1(1,:)=[28200*0.6*2/3,28200*0.66*2/3,28200*0.72*2/3];
lastRest1=zeros(1,3);
% fg1=figure();
% set(fg1,'position',get(0,'ScreenSize'));
totLoss=zeros(24,1);
for i=1:24
    %[x1(i,:),y1(:,:,i),use1(i,:),leastCostFval(i),nowRest1(i+1,:)] = least
CostProg(nowRest1(i,:),s);
    lastRest1=nowRest1(i,:);

[x1(i,:),y1(:,:,i),gax1,gafval1,use1(i,:),nowRest1(i+1,:)] = gaProg_T3();
    totLoss(i)=0;
    for ii=1:length(SA)
        for j=1:8

```

```

totLoss(i)=totLoss(i)+E(leastChoices(SA(ii)))*x1(SA(ii))*CA*s(j)*y1(SA(ii),j);
    end
end
for ii=1:length(SB)
    for j=1:8

totLoss(i)=totLoss(i)+E(leastChoices(SB(ii)))*x1(SB(ii))*CB*s(j)*y1(SB(ii),j);
        end
    end
    for ii=1:length(SC)
        for j=1:8

totLoss(i)=totLoss(i)+E(leastChoices(SC(ii)))*x1(SC(ii))*CC*s(j)*y1(SC(ii),j);
            end
        end
        fval1(i,1)=gafval1(1,:);
    end

orderx=zeros(402,24); %订购方案
transy=zeros(402,8,24); %转运方案
for i=1:24

orderx(id_leastChoices(find(x1(i,:)==1)),i)=E(leastChoices((find(x1(i,:)==1))));
end
tmp=[E(leastChoices),E(leastChoices),E(leastChoices),E(leastChoices),E(leastChoices),E(leastChoices),E(leastChoices),E(leastChoices)];
for i=1:24
    transy(id_leastChoices(:,i)=y1(:, :, i).*tmp;
end

```

使用遗传算法进行规划求解代码：

```

clc,clear

%Question 1

%导入根据类型排序后的订购量和供货量
load 根据类型排序后的订购量和供货量及对应的供货商 id.mat
global headA tailA headB tailB headC tailC rows cols E leastChoices CA CB
CC s lastRest1
%syms CA CB CC
CA=1.2; CB=1.1; CC=1;
[rows cols]=size(order);
headA=1;tailA=146; %1 到 146 行为 A 材料供货商
headB=147;tailB=280; %147 到 280 为 B 材料供货商
headC=281;tailC=402; %281 到 402 为 C 材料供货商
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);

%求诚信度和依赖度

```



```

[DH,DR]=get_DH_DR(order,supply);
%求诚信度和依赖度的权重
[WH,WR]=get_WH_WR(DH,DR);
%score 为他们的综合评分
score=WH*DH+WR*DR;
%找出前 50 名在列向量中的下标,存储在 goodTrader 的第一列
goodTrader=zeros(50,2);
goodTrader(:,1)=getGoodTrader(score);

%Question 2

%获得 402 个供货商的供货能力（平均供货量）向量
E=getE(order,supply); %得到每个供货商平均供货量
cE=zeros(402,1); %cE 为将元素数值转换为对应能生产的产品数后的向量
cE(headA:tailA)=E(headA:tailA)/0.6;
cE(headB:tailB)=E(headB:tailB)/0.66;
cE(headC:tailC)=E(headC:tailC)/0.72;

%前 50 名供货商的平均供货量对应能生产的产品数存储在 goodTrader 第二列
for i=1:50
    goodTrader(i,2)=cE(goodTrader(i,1));
end
sorted_GoodTrader=sortrows(goodTrader,-2);
sorted_GoodTrader(:,3)=zeros(50,1);
sorted_GoodTrader(1,3)=sorted_GoodTrader(1,2);
%求前缀和（包括该元素）
for i=2:50
    sorted_GoodTrader(i,3)=sorted_GoodTrader(i-1,3)+sorted_GoodTrader(i,2);
end

%最少选择的供货商
leastChoices=sorted_GoodTrader(find(sorted_GoodTrader(:,3)<=28200),1);
leastChoices=[leastChoices;sorted_GoodTrader(length(leastChoices)+1,1)];
id_leastChoices=id(leastChoices);
fprintf('最少选择%d 家供应商，他们的编号如下:\n',length(leastChoices))
disp(id_leastChoices')

%最经济订购方案求解

%还要算出当周 rest
s=[1.904769167,0.921370417,0.186055556,1.570482353,2.889825301,0.543761111,2.078833333,1.010282759]/100;
nowRest1=zeros(25,3);
x1=zeros(24,21);
y1=zeros(21,8,24);
use1=zeros(24,3);
fval1=zeros(24,1);
nowRest1(1,:)= [28200*0.6*2/3,28200*0.66*2/3,28200*0.72*2/3];
lastRest1=zeros(1,3);
% fg1=figure();
% set(fg1,'position',get(0,'ScreenSize'));
totLoss=zeros(24,1);
for i=1:24
    %[x1(i,:),y1(:,:i),use1(i,:),leastCostFval(i),nowRest1(i+1,:)] = leastCostProg(nowRest1(i,:),s);
    lastRest1=nowRest1(i,:);

```

```

[x1(i,:),y1(:, :, i),gax1,gafval1,use1(i,:),nowRest1(i+1,:)] = gaProg_T3();
totLoss(i) = 0;
for ii = 1:length(SA)
    for j = 1:8

totLoss(i) = totLoss(i) + E(leastChoices(SA(ii))) * x1(SA(ii)) * CA * s(j) * y1(SA(ii), j);
        end
    end
    for ii = 1:length(SB)
        for j = 1:8

totLoss(i) = totLoss(i) + E(leastChoices(SB(ii))) * x1(SB(ii)) * CB * s(j) * y1(SB(ii), j);
            end
        end
        for ii = 1:length(SC)
            for j = 1:8

totLoss(i) = totLoss(i) + E(leastChoices(SC(ii))) * x1(SC(ii)) * CC * s(j) * y1(SC(ii), j);
                end
            end
            fval1(i, 1) = gafval1(1, :);
end

orderx = zeros(402, 24); %订购方案
transy = zeros(402, 8, 24); %转运方案
for i = 1:24

orderx(id_leastChoices(find(x1(i, :) == 1)), i) = E(leastChoices((find(x1(i, :) == 1))));
end
tmp = [E(leastChoices), E(leastChoices), E(leastChoices), E(leastChoices), E(leastChoices), E(leastChoices), E(leastChoices), E(leastChoices)];
for i = 1:24
    transy(id_leastChoices, :, i) = y1(:, :, i) .* tmp;
end

```

模型的目标函数：

```

%tx(1)~tx(21)是 x 矩阵
%{
[tx(22)~tx(42);
tx(43)~tx(63);
tx(64)~tx(84);
tx(85)~tx(105);
tx(106)~tx(126);
tx(127)~tx(147);
tx(148)~tx(168);
tx(169)~tx(189);]
为 y 矩阵
%}
%tx(190)~tx(192)是 useA, useB, useC

function y = gaFun_T3(tx)

```

```

global E leastChoices headA headB headC tailA tailB tailC CA CB CC s
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
%y(1)=E(leastChoices(SA(:))'*tx(SA(:))*CA+E(leastChoices(SB(:))'*tx(SB
(:))*CB+E(leastChoices(SC(:))'*tx(SC(:))*CC;
y(1)=0;
for i=1:length(SA)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SA(i)))*tx(SA(i))*CA*s(j)*tx(22+(j-
1)*21+SA(i)-1);
    end
end
for i=1:length(SB)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SB(i)))*tx(SB(i))*CB*s(j)*tx(22+(j-
1)*21+SB(i)-1);
    end
end
for i=1:length(SC)
    for j=1:8
        y(1)=y(1)+E(leastChoices(SC(i)))*tx(SC(i))*CC*s(j)*tx(22+(j-
1)*21+SC(i)-1);
    end
end
wei=[0.6,0.2,0.2];
y(1)=wei(1)*y(1);
y(1)=y(1)-
wei(2)*E(leastChoices(SA(:))'*tx(SA(:))+wei(3)*E(leastChoices(SC(:))'*
tx(SC(:))';
end

```

模型的非线性限制条件:

```

function [c,ceq]=gaNonlcon_T3(x)
global E leastChoices headA headB headC tailA tailB tailC s lastRest1
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
c(1)=x(190)-lastRest1(1);
for i=1:length(SA)
    for j=1:8
        c(1)=c(1)-E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-
1)*(1-s(j));
    end
end
c(2)=x(191)-lastRest1(2);
for i=1:length(SB)
    for j=1:8
        c(2)=c(2)-E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-
1)*(1-s(j));
    end
end
c(3)=x(192)-lastRest1(3);
for i=1:length(SC)
    for j=1:8
        c(3)=c(3)-E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-
1)*(1-s(j));
    end
end

```

```

end
end
c(4)=2*28200+x(190)/0.6+x(191)/0.66+x(192)/0.72-lastRest1(1)/0.6-
lastRest1(2)/0.66-lastRest1(3)/0.72;
for i=1:length(SA)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SA(i)))*x(SA(i))*x(22+(j-1)*21+SA(i)-
1)*(1-s(j)))/0.6;
    end
end
for i=1:length(SB)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SB(i)))*x(SB(i))*x(22+(j-1)*21+SB(i)-
1)*(1-s(j)))/0.66;
    end
end
for i=1:length(SC)
    for j=1:8
        c(4)=c(4)-(E(leastChoices(SC(i)))*x(SC(i))*x(22+(j-1)*21+SC(i)-
1)*(1-s(j)))/0.72;
    end
end
for j=1:8
    c(4+j)=-6000;
    for i=1:length(leastChoices)
        c(4+j)=c(4+j)+E(leastChoices(i))*x(i)*x(22+(j-1)*21+i-1);
    end
end
ceq=[];
end

```

问题四：

主函数代码：

```

clc,clear

%导入根据类型排序后的订购量和供货量
load 根据类型排序后的订购量和供货量及对应的供货商 id.mat
global headA tailA headB tailB headC tailC rows cols E leastChoices CA CB
CC s lastRest1
%syms CA CB CC
CA=1.2; CB=1.1; CC=1;
[rows cols]=size(order);
headA=1;tailA=146; %1 到 146 行为 A 材料供货商
headB=147;tailB=280; %147 到 280 为 B 材料供货商
headC=281;tailC=402; %281 到 402 为 C 材料供货商
SA=find(headA<=leastChoices&leastChoices<=tailA);
SB=find(headB<=leastChoices&leastChoices<=tailB);
SC=find(headC<=leastChoices&leastChoices<=tailC);
E=getE(order,supply); %得到每个供货商平均供货量

s=mean([1.904769167,0.921370417,0.186055556,1.570482353,2.889825301,0.543
761111,2.078833333,1.010282759])/100;
nowRest1=zeros(25,3);
x1=zeros(24,402);
M=zeros(24,1);

```

```

use1=zeros(24,3);
fval1=zeros(24,1);
%nowRest1(1,:)=[50000*0.6*2/3,50000*0.66*2/3,50000*0.72*2/3];
nowRest1(1,:)=[28200*0.6*2/3,28200*0.66*2/3,28200*0.72*2/3];
lastRest1=zeros(1,3);
for i=1:24
    %[x1(i,:),y1(:, :, i),use1(i,:),leastCostFval(i),nowRest1(i+1,:)] = least
    CostProg(nowRest1(i,:),s);
    lastRest1=nowRest1(i,:);
    [x1(i,:),M(i),gax1,gafval1,use1(i,:),nowRest1(i+1,:)] = gaProg_T4();
    fval1(i,1)=gafval1(1,:);
end

transy=zeros(402,8); %对应的供货商和转运商的转运矩阵
%rank1~8 转运商对应的 id
rankTrans=[3,6,2,8,4,1,7,5];
%rank1~8 转运商对应的损耗率
[0.186055556,0.543761111,0.921370417,1.010282759,1.570482353,1.904769167,
2.078833333,2.889825301]
cntTrans=0;
for i=1:402
    while sum(transy(:,rankTrans(cntTrans+1)))+E(i)>6000
        cntTrans=cntTrans+1;
    end
    transy(id(i),rankTrans(cntTrans+1))=E(i);
end

```

对模型进行整型线性混合规划求解的函数代码:

```

function [x1,M,gax1,gafval1,use1,nowRest1]=gaProg_T4()
%tx(1)~tx(402)是 x 矩阵
%tx(403)~tx(405)是 useA,useB,useC
%tx(406)是 M
global E headA headB headC tailA tailB tailC s lastRest1
ns=5;
nvar=406;
A=zeros(ns,nvar);
B=zeros(ns,1);
A(1,1:402)=E;
B(1)=48000;
A(2,headA:tailA)=-(1-s)*E(headA:tailA);
A(2,403)=1; B(2)=lastRest1(1);
A(3,headB:tailB)=-(1-s)*E(headB:tailB);
A(3,404)=1; B(3)=lastRest1(2);
A(4,headC:tailC)=-(1-s)*E(headC:tailC);
A(4,405)=1; B(4)=lastRest1(3);
A(5,403:406)=[1/0.6,1/0.66,1/0.72,2];
A(5,headA:tailA)=-((1-s)/0.6)*E(headA:tailA);
A(5,headB:tailB)=-((1-s)/0.66)*E(headB:tailB);
A(5,headC:tailC)=-((1-s)/0.72)*E(headC:tailC);
B(5)=lastRest1(1)/0.6+lastRest1(2)/0.66+lastRest1(3)/0.72;
Aeq=zeros(1,406);
Aeq(403:406)=[1/0.6,1/0.66,1/0.72,-1];
Beq=[0];
lb=zeros(nvar,1);
lb(406)=28211;
ub=zeros(nvar,1);

```

```

ub(1:402)=ones(402,1);
ub(403:405)=6000*8;
ub(406)=80000*(1-s);
intcon=[1:402];
options=optimoptions('ga');
options.FunctionTolerance=1e-6;
%options.MaxStallGenerations=400;
%Fun=@gaFun_T4;
%[gax1,gafval1]=ga(Fun,nvar,A,B,Aeq,Beq,lb,ub,[],intcon,options);

f=zeros(1,nvar); f(406)=-1;
[gax1,gafval1]=intlinprog(f,intcon,A,B,Aeq,Beq,lb,ub);

x1=gax1(1:402)';
use1=gax1(403:405)';
M=gax1(406);
nowRest1=lastRest1;
for i=headA:tailA
    nowRest1(1)=nowRest1(1)+E(i)*x1(i);
end
nowRest1(1)=nowRest1(1)-use1(1);
for i=headB:tailB
    nowRest1(2)=nowRest1(2)+E(i)*x1(i);
end
nowRest1(2)=nowRest1(2)-use1(2);
for i=headC:tailC
    nowRest1(3)=nowRest1(3)+E(i)*x1(i);
end
nowRest1(3)=nowRest1(3)-use1(3);
end

```