

生产企业原材料的订购与运输策略

摘要

本文研究生产企业原材料订购与运输策略,根据原材料供应商和转运商往年数据,结合产能、成本要求,制定采购以及运输策略,根据不同要求,规划制定企业未来 24 周最优的原材料订购和转运方案。

针对问题一:利用 402 家供应商过去 250 周的订货和供货数据,提取供货商评价特征,综合考虑了供货能力、供货稳定性、供货类别三项指标,建立供货稳定性评价函数为 $f(t)$,对 A、B、C 原材料进行加权,利用 Borda-排序法建立供应商综合评分体系 S,并对得分前 50 的供应商进行特征分析,分为稳定型供应商和突变型供应商,。

针对问题二:按得分排序,以累积转化产能 2.82 万为阈值,选取前 23 家供应商满足企业生产。先制定最经济采购方案,再制定最优转运方案,为两个单目标规划问题。对 23 家供应商订货量进行规划,稳定供应商过去 250 周供货量符合正态分布,采用蒙特卡洛算法,突变供应商采用遗传算法预测,突变概率为 $\frac{1}{12}$

目标函数原料成本,迭代循环求解最优值,制定最经济订购方案。周平均最经济订购费用为:20825。以转运历史数据非 0 平均值作为指标,对转运商进行排序,采用随机抽样法生成未来 24 周每家转运商转运商的损耗率。转运方案采用逆向选择,转运商按照排名,选择多家供应商进行转运。损耗量最小为目标,利用动态规划进行决策,生成的最优转运方案。

针对问题三:从供给侧调整原材料订购比例,优先选择 A 类材料供应商合作,重新选取了 31 家供货商保障生产。该问题为多目标规划,目标为采购成本、转运成本、损耗成本、原料 A、C 订购量差值,以产能和转运能力为约束,进行多目标规划模型求解,转运方案选择思路同第一问,使得损耗率小的转运商转运数量优先达到最大或接近最大。解得的最优采购以及转运方案,24 周的原料和损耗总成本为 533642。

针对问题四:依据得分和产品类别,以平均供货累积和 48000 为阈值,选取 134 家供货商保障生产。供应商均作为稳定型供应商以简化模型,采用蒙特卡洛算法生成订购表,一家转运商可选取多家供应商提供转运,平均损耗率小的转运商优先选择 A 类原料供应商,利用动态规划模型求解最佳转运计划,计算总成本 S,迭代 10000 次寻找最优订购以及转运方案。计算周平均产能为 7.4 万立方米,可得出企业每周的产能可以提高 4.93 万立方米。

关键词: Borda-排序 蒙特卡洛算法 动态规划 多目标规划

一、问题重述

1.1 问题背景

某建筑和装饰板材的生产企业在生产中主要用到 A、B、C 三种不同类型的木质及其他的植物纤维材料。相关材料由订购的原材料供应商（也称“供应商”）进行提供，并由第三方的物流公司（也称“转运商”）进行运输，将货物运送到企业仓库进行储存和使用。该企业每年规划使用 48 周进行生产，考虑到需要保持生产活动的持续性，需要在生产活动的前 24 周制定各类原材料的订购和运输计划。

该企业每周的产能为 2.82 万立方米，每生产一立方米的产品需要消耗 0.6 立方米 A 类原材料、0.66 立方米 B 类材料、0.72 立方米 C 类原材料。由于原材料的特殊性，供货商可能存在不能完全按照企业提出的订货量需求进行供应的情况，既实际的供货量可能高出或者低于订货量。故在保证企业正常生产的前提下，该企业最好可以保持库存的原材料足够维持企业两周的生产所需，（三类原材料的储存和运输单价均相同）为此，企业通常对供货商提供的原材料全部收购。

在实际的运输过程中，原材料存在着一定的损耗（ $(\text{损耗量}/\text{供货量}) \times 100\%$ 的值称为“损耗率”），故企业仓库实际收到的由转运商所运输的货物数量称为“接收量”。每家转运商的货物转运能力均为 6000 立方米/周。在一般情况下，一家转运商一周只能运输一家供货商的货物。

原材料的采购成本将直接影响到企业的生产效益，且在实际中，A 类与 B 类原材料的采购单价分别比 C 类原材料的高 20% 与 10%。

1.2 问题提出

根据以上的背景及附件数据，需要解决以下四个问题：

1. 根据附件一，对文件中给出的 402 家供货商的供货特征进行量化分析，建立起基于保证企业正常生产重要性的数学模型，并在此基础上确定 50 家最重要的供货商，在论文中列表给出结果。
2. 参考上一问的结果，求解该企业至少需要选定多少家供应商进行原材料供应方可满足生产需要。并针对这些供货商，为企业制定未来 24 周每周的最经济的原料订购方案，并由该方案制定运输损耗最小的转运方案，最后分析方案的实际实施效果。
3. 该企业为了节省开支，压缩生产成本，现计划对 A 类原材料加大采购力度，同时对 C 类原材料的采购进行缩减，以此来减少运输及储存的成本，同时希望转运商在转运过程中对商品的损耗尽可能的减少。请制定新的订购方案及转运方案，最后分析方案的实际实施效果。
4. 该企业在技术改造后具备了提高产能的潜力。请根据现有的原材料供应商和转运商的实际情况，确定该企业每周产能的提高量，最后给出未来 24 周的订购和转运方案。

二、问题假设

假设 1: 各供货商和转运商的运营情况稳定, 短时间不会出现破产情况;
假设 2: 忽略转运过程中出现交通事故导致原材料大规模损失的情况;
假设 3: 企业仓库足够大, 能够全部接收供货商的原材料;
假设 4: 假设订货与供货都在一周内完成;
假设 5: 题目所给数据真实可靠;

三、符号说明

D_k	k 类原材料所占权重
p_k	k 类原材料生产单位产品的消耗
x_1	企业订货量
x_2	供应商供货量
S	供应商综合得分
$\min(\text{Fiprice})$	每周最终总订购额
P_n	企业第 n 周产能
$wastage_i$	转运商第 i 周运输损耗率
$Tramount_{ji}$	j 转运商第 i 周转运总量
Z_{\max}	转运商的最大转运能力

注: 未列出符号及重复符号以出现处标注为准

四、问题分析

4.1 问题一分析

为保障企业生产, 对于供应链这一环最重要的就是保障原材料的供应。对于一家供应商来说, 考量指标为供货能力和供货稳定性。原料供应能力表现为单次供应量的多少, 供货稳定性表现为供应量与预定量的差值。当差值为负, 即供应量小于预定量, 企业就有由于原材料供应不足而导致产能不足的情况出现的可能性, 故该种情况从根本上阻碍了企业生产, 对企业影响较大。当差值为正, 即供应量大于预定量, 造成库存积压, 增加成本, 但不会影响企业的产能需求, 且在本题中不考虑原料贮藏成本, 所以此种情况对于企业生产影响较小。故建立函数 $f(\Delta x)$ 表征供应商的供货稳定性。将建立评价模型对供应商进行评价, 由于企业制造产品所需原料量不同, 产品类别也直接影响了企业生产, 则针对不同原材料进行加权处理, 权重为 D_k 。

故选取供货能力、供货稳定性和供货类别为供应商的供货特征, 对供应商特征进行量化分析, 建立组合评价模型。对每家供应商依据已有的订货量、供货量数据进行评分, 得分为 S。S 与供货能力、供货稳定性和权重呈正相关。企业应该选取有着较强的供货能力、供货稳定性较好的供应商来保证企业稳定生产。

4.2 问题二分析

问题二首先要求选择足够数量的供应商进行原材料供应,也就是各家供应商每周提供的原材料至少在理论上可以满足该企业每周的产能需求。在问题一建立的评价体系下,得到了排名前 50 的满足企业生产要求的供应商 ID,那么这些“优良”供应商就是我们的重点考虑对象。

在对这些供应商五年来的供货曲线分析后,发现由这些供应商的供货特点可以将其分为三大类:第一是持续供货的稳定型供应商,第二是长时间不供货,但存在极大的供货峰值的突变型供应商(且其峰值的出现周期大约为 12 周一次),第三是既存在持续供货阶段,又具有较大供应能力的综合型供应商。在对供应商按照其提供的 A、B、C 类材料分类后,对每个供应商的供应量取周平均值(该供应商的总供货量/订货天数),并将供应量换算为对应的产能数据,按照产能从高到低的顺序相加,直至达到 2.82 万立方米的每周产能需求,便可求得供应商的最小数量。

题目还要求为企业制定未来 24 周中每周的最经济原材料订购方案,则该方案的目标为最经济,既购买成本最小化,同时满足约束条件。故该问题转化为多层次单目标规划问题。其中,稳定型和综合型供应商的供货能力稳定,故作为长期供货目标进行选择。但由于原材料的特殊性导致的发货误差(该误差不可控,难以预测),存在导致库存量不足而影响生产的可能性,故突变型供应商作为产能不足预警出现时的补充,最后得到最经济的原材料订购方案。

在最优订购方案产生后,对附件二中各转运商的损耗率数据进行分析,得到转运商的优劣及其排名,则考虑让较优的转运企业优先选择订单,并尽可能的多运输原材料,以达到原材料损耗最小的目标。

4.3 问题三分析

该问要求尽可能的减少采购 C 类原材料,多采购 A 类原材料,则通过分析生产单位产品需要的三类原材料的量以及对应的成本价与运输估计费用(只与原材料量的多少有关)得到三类原材料对于该企业生产单位产品的经济数据。

在确定最优订购方案之前,首先在各家供应商的周平均供货值对应的产能之和满足企业的周产能需求前提下,优先对提供经济数据好的原材料的供应商进行选择。

在供应商选择完毕后,该题需要在订购方案最优的情况下,同时满足转运消耗最少,既订购方案的选定与转运方案的选定同时进行,总目标为订购原材料的费用与转运损耗带来的经济损失的和最小,故将该问题转化为单目标动态规划问题。

4.4 问题四分析

该企业具备提高产能的潜力,即产能不设上限,供货商为 402 家,通过计算,原料供给充足。则产能的提升取决于转运能力,那么让 8 家转运商均全力转运,以提升企业产能。产能与原料类别有关,由于原材料 A、B、C 生产单位产品所需的量依次降低。则应尽可能多的采购料 A,其次为 B 和 C。优先用转运损耗较小的转运商转运原料 A,后调节原材料比例以及损耗量来提升产能。

五、问题一的模型建立与求解

5.1 问题分析

供应商的原材料供应能力对保障企业的正常生产具有重要影响，根据题目对产能的要求进行分析，供应商的考量指标为供货能力和供货稳定性，供货稳定性表现为供应量与预定量的差值。当差值为负数时，可能导致企业产能不足，故该情况一定会妨碍企业生产；当差值为正数时，有可能造成产品积压导致库存量上升，但至少不会影响企业的产能需求，且该问不考虑储存费用的影响，则该种情况对企业产生的影响较小。

故建立函数 $f(\Delta x)$ 表征供应商的供货稳定性，建立评价模型对供应商进行评价。其次原材料类别也直接影响企业生产，则针对不同原材料进行加权处理。

选取供货能力、供货稳定性和供货类别为供应商的供货特征，对供应商特征进行量化分析，建立组合评价模型，得到供应商排名，企业择优进行选择。



图 1 问题一解决流程图

5.2 模型建立

设供货稳定性评价函数为 $f(t)$ $t = \frac{\Delta x}{x_1}$ 其中：

$$\begin{cases} x_1 \text{ 代表企业订购量} \\ x_2 \text{ 代表供应商供货量} \\ \Delta x \text{ 代表供应量与订购量的差值} (\Delta x = x_2 - x_1) \end{cases}$$

依据 Δx 对于供应能力的影响，当差值为负，即供应量小于预定量，从根本上阻碍了企业生产，有出现企业产能不足的风险；当差值为正，即供应量大于预定量，易造成库存积压，增加生产成本，由于本题中不考虑原料储存成本，则该情况对于企业生产影响较小。

根据此特点, $f(t)$ 选用指数函数模型。当 $\Delta x < 0$ 时, 采用指数段的突变部分, 当 $\Delta x > 0$ 时, 采用指数函数中缓慢变化的部分。

$$f(t) = \begin{cases} -Aa^t + C, & t < 0 \\ Bb^t + D, & t \geq 0 \end{cases} \quad (a, b \in (0, 1))$$

确定不同类别原料权重 D_k :

由于企业对于生产每立方米产品需消耗 A 类原材料、B 类原材料和 C 类原材料需求不同, 则产品对于不同原材料的依赖性不同。制造每立方米产品需消耗原材料越少, 则产品对于原材料的依赖性越高, 原材料对应权重越高; 制造每立方米产品需消耗原材料越多, 则产品对于原材料的依赖性越高, 原材料对应权重越低。(依赖性特指: 缺少同样的量, 对于产能的影响。) 据此特点, 三类原材料权重 D_k 之比为:

$$D_1 : D_2 : D_3 = \frac{1}{p_1} : \frac{1}{p_2} : \frac{1}{p_2} \quad (1)$$

在式中, p_k 指制造每立方米产品需消耗 k 类原材料的量。

建立供应商综合评价模型

设供应商综合得分为 S, S 与供货能力、供货稳定性和权重呈正相关; 设

$$S = \sum_{i=1}^n f(t) \cdot x_2 \cdot D_k \quad (2)$$

n 代表企业对供应商订货量不为 0 的总周数;

x_2 指供应量, 代表供应商供货能力;

5.3 模型求解

求解供货稳定性评价函数 $f(t)$:

$$f(t) = \begin{cases} -Aa^t + C, & t < 0 \\ Bb^t + D, & t \geq 0 \end{cases} \quad (a, b \in (0, 1)) \quad (3)$$

$f(t)$ 代表供货稳定性, 取范围为 $[-1, 1]$ 。

当预定量=供货量时, 即 $x_1 = x_2$ 时, $t=0$, 供货稳定性最佳, $f(0)=1$ 。

当有订货没有供货时, 即 $x_1 \neq 0$, $x_2=0$ 时, $t=-1$, 供货稳定性最差, 为 $f(-1)=-1$

当预定量<供货量时, 即 $x_1 < x_2$ 时, 呈指数型缓慢变化曲线段;

当预定量>供货量时, 即 $x_1 > x_2$ 时, 呈指数型迅速变化曲线段。

根据以上特征, 结合原料对于生产企业产能影响规律, 解得:

$$f(t) = \begin{cases} -4 \cdot 0.8^t + 4, & t < 0 \\ 0.5^t, & t \geq 0 \end{cases} \quad (4)$$

图像为：

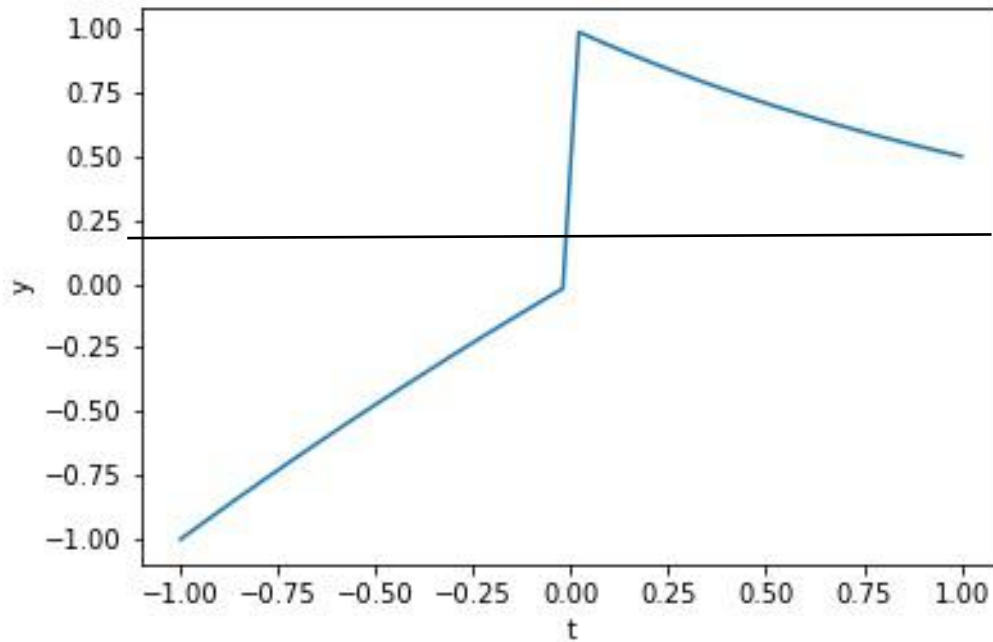


图 2 供货稳定性评价函数图

求解不同类别原料权重 D_k ：

$$D_1 : D_2 : D_3 = \frac{1}{p_1} : \frac{1}{p_2} : \frac{1}{p_2} \quad (5)$$

代入数据得：

$$D_1 : D_2 : D_3 = \frac{1}{0.6} : \frac{1}{0.66} : \frac{1}{0.72} \quad (6)$$

化简并归一化，取值为：

$$D_1 = 0.95$$

$$D_2 = 0.86$$

$$D_3 = 0.79$$

求解供应商综合评价模型：

$$S = \sum_{i=1}^n f(t) \cdot x_2 \cdot D_k$$

代入 $f(t)$ 和 D_k 得出供应商评价模型。

对 402 家供应商进行综合评价，进行得分排名，选出 50 名最重要供应商。
50 名最重要供应商如下表：

表 1 50 名供应商 ID

S229	S131	S307	S040	S086
S361	S308	S143	S338	S210
S140	S330	S395	S364	S003
S108	S356	S247	S367	S114
S282	S268	S374	S055	S273
S151	S306	S284	S346	S189
S275	S348	S126	S080	S078
S329	S352	S037	S294	S292
S340	S201	S031	S218	S074
S139	S194	S365	S244	S291

其中，通过对所选择重要企业进行分析可得，绘制出 50 家供应商的供货量方差与中位数可得，对于保障企业生产比较重要的企业大致分为两类：

第一类：可以长期稳定对企业进行供货，供货稳定性高的供应商，这类供应商方差较小，中位数较高（以 S361 为代表）

第二类：供货能力极强，可以满足企业超大订货需求的供应商，这类供应商与企业不进行稳定合作，却可以对企业提供关键帮助，供应紧急订单，这类供应商供货数据方差极大，中位数极小。（以 S140 为代表）

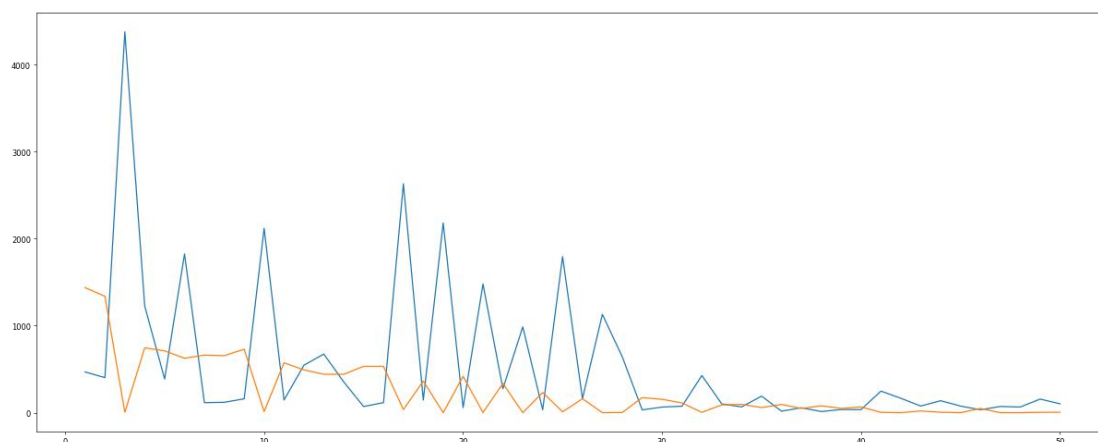


图 3 50 家供应商方差与中位数数据图

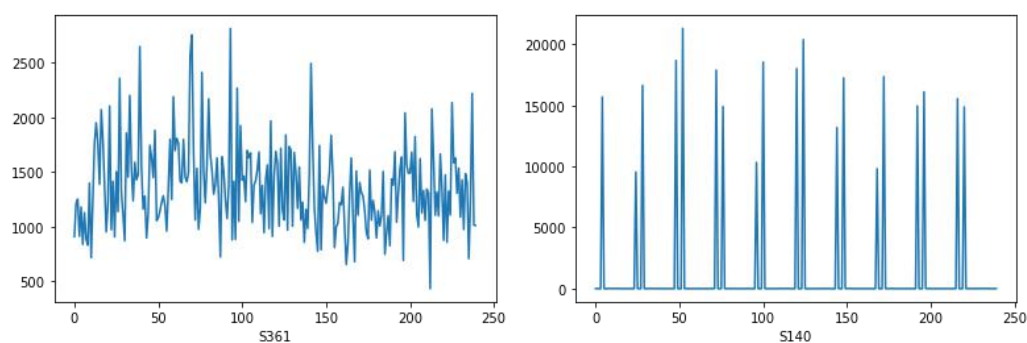


图 4 两类供应商的发货历史数据图（左一右二）

六、问题二的模型建立与求解

6.1 问题分析

题目要求求得最小的供应商数量，既在该数量的供应商进行供货的情况下，该企业的每周产能需求得以满足。又考虑到由于在订购方案欠佳的情况下，即使所选的供应商供货充足，也可能导致一些离散的产能不足的情况出现，故该问题求解时忽略订购方案不同带来的影响，只要所选各供应商的周平均供应量的总和达到企业的每日产能，则从理论上，这些供应商能够满足企业的生产需求，从而求得最小的供应商数量。

在对该企业未来 24 周的采购计划进行规划时，考虑到供应商的不同供应特点，我们将突变型供应商与稳定型、综合型供应商作为两大类考虑，对后者进行规划以保证供应总量的稳定性，突变型供应商作为产能储备不足预警情况出现时的补充，从而得到最优的订购方案。

在最优订购方案产生后，对转运商的相关数据进行分析，得到转运商的优劣，则最优的转运思路既为让较优的转运商优先进行转运选择，并尽可能饱和其转运能力，则该问题转化为损耗最小化的动态规划问题。

6.2 模型建立

6.2.1 确定供应商的最小数量

在第一问中得到的 50 家最重要的供应商中，对每个供应商的供货量对应的产能求周平均值 $\overline{w_j}$ （j 为附件一中供应商 ID），对其由大到小顺序排序，得到序列 $\overline{w_{j,i}}$ （i 为排序的序号），则该企业产能的周平均值为 \overline{W} ，当满足以下条件时：

$$\overline{W} = \sum_{i_{\max}}^{N-1} \overline{w_{j,i}} < 28200 \quad (7)$$

$$\overline{W} = \sum_{i_{\max}}^N \overline{w_{j,i}} \geq 28200 \quad (8)$$

则（ $i_{\max} - N$ ）的值为所求的供应商最小数量。

6.2.2 最优订购方案规划问题模型建立

目标函数的确立：

该问的目标是确立前 24 周最经济的原材料选购方案，由于 A、B、C 类原材料的订购价格不同，且企业对原材料的生产种类不做改变，则最优化目标可以确立为从不同供应商中订购的 A、B、C 类原材料的订购价格之和最小，目标函数如下：

$$\min(\text{Fiprice}) = 1.2 \sum x_A + 1.1 \sum x_B + \sum x_C \quad (9)$$

（设 C 类原材料的单位价格为 1，Fiprice 为每周最终的总订购额， x_k k=A,B,C 为

k 类原材料每周的总订购量)

约束条件:

每周的订购方案应该保证至少不会影响企业的正常产能, 并且企业每周的最大原材料数量不得超过八家转运商的单周最大转运能力, 约束条件如下:

$$\text{转运能力约束: } \sum x_A + \sum x_B + \sum x_C \leq 48000 \quad (10)$$

为满足库存至少够两周的生产所需, 我们考虑在第一周就订购两周生产所需的原材料, 故:

$$\text{第一周的产能约束为: } P_1 = \frac{\sum x_{A1}}{0.6} + \frac{\sum x_{B1}}{0.66} + \frac{\sum x_{C1}}{0.72} \geq 28200 \times 2 \quad (11)$$

(x_{kn} 为 k 类原材料第 n 周的订购量, P_n 为企业第 n 周产能 $k=A、B、C; n=1、2、3$)

第一周的剩余产能:

$$P_{\text{剩1}} = P_1 - 28200 = \frac{\sum x_{A1}}{0.6} + \frac{\sum x_{B1}}{0.66} + \frac{\sum x_{C1}}{0.72} - 28200 \quad (12)$$

第二周的产能约束为:

$$P_2 + P_{\text{剩1}} = \frac{\sum x_{A2}}{0.6} + \frac{\sum x_{B2}}{0.66} + \frac{\sum x_{C2}}{0.72} + \frac{\sum x_{A1}}{0.6} + \frac{\sum x_{B1}}{0.66} + \frac{\sum x_{C1}}{0.72} - 28200 \quad (13)$$

(12) (13) 联立得:

$$P_2 + P_1 \geq 3 \times 28200 \quad (14)$$

同理得第三周的产能约束为:

$$P_3 + P_2 + P_1 \geq 4 \times 28200 \quad (15)$$

第 n 周的产能约束为:

$$P_n + P_{n-1} + \dots + P_1 \geq (n+1) \times 28200 \quad (16)$$

6.2.3 最优转运方案模型的建立

转运商数据分析:

通过对附件二中各家转运商 5 年来的运输损耗率作图如下: (以转运商 1、4 为代表)

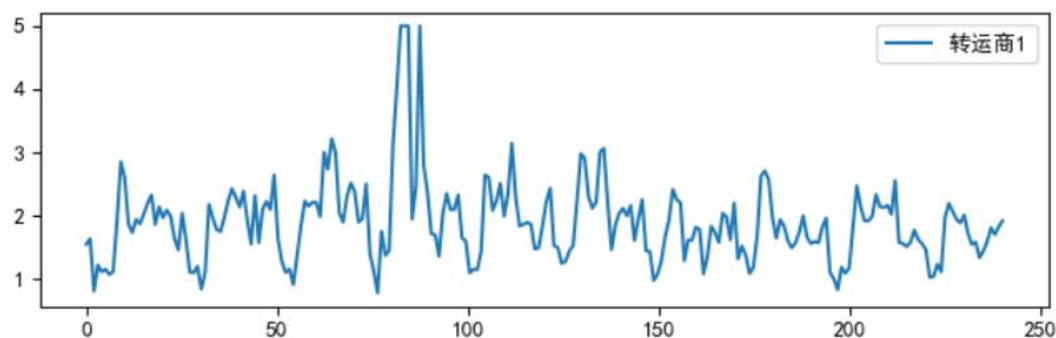


图 5 转运商 1 的运输损耗率图

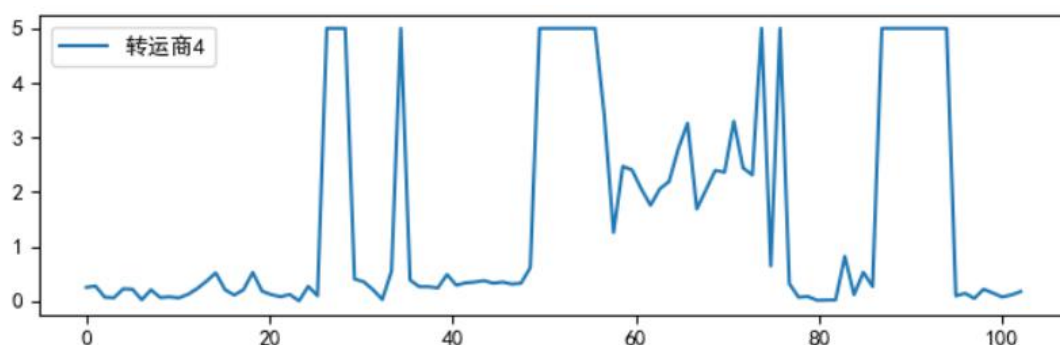


图 6 转运商 4 的运输损耗率图

可以发现，以转运商 1、4 为代表的各家转运商 5 年来的运输损耗率并没有显著的统计学分布规律。但对于不同的转运商而言，其损耗率的大小存在明显差异。下图为转运商 2 的运输损耗率图：

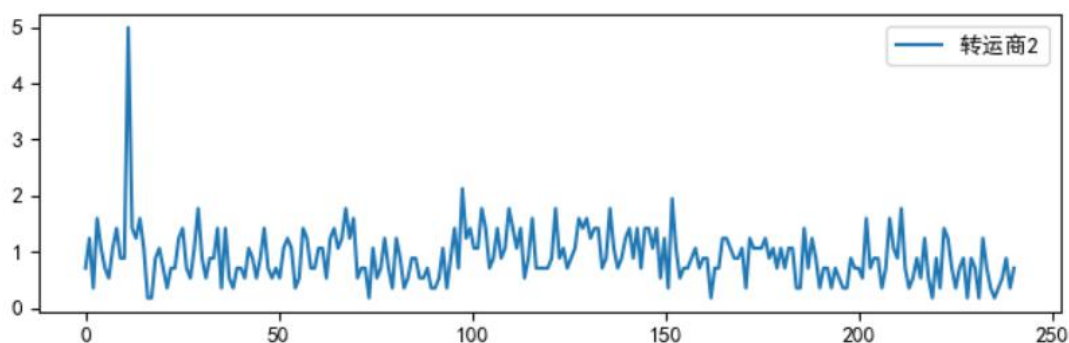


图 7 转运商 2 的运输损耗率图

将转运商 1 与转运商 2 的运输损耗率图像对比，可以明显发现转运商 2 的损耗率峰值数量更少，且大部的运输损耗率均小于转运商 1 的，故转运商 2 的转运可靠性（与原材料的运输损耗成反比）大于转运商 1 的。

对于转运商 3 来说，其运输损耗率在五年的时间内均居于较低水平，只有一个损耗率峰值出现，说明该转运商的转运能力也较为出色。下图为转运商 3 的运输损耗率图：

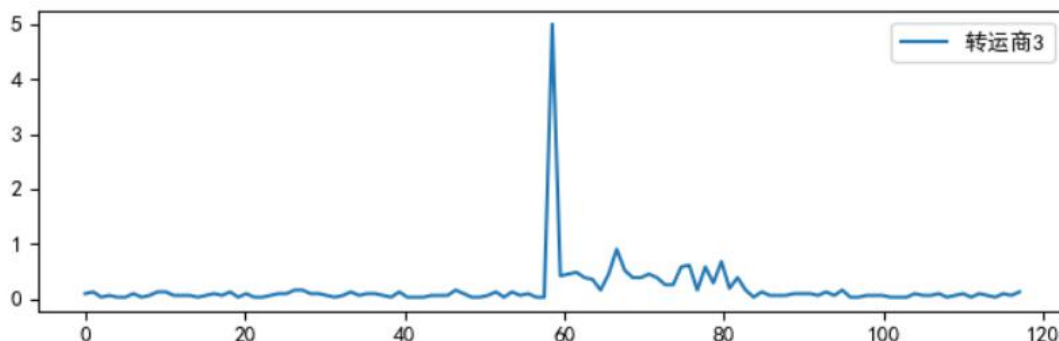


图 8 转运商 3 的运输损耗率图像

综合以上分析，我们选取各转运商 5 年来的运输损耗率周平均值（计算公式为：5 年内企业的运输损耗率总和/运输天数）来反映转运商转运水平的优劣。

目标函数的确定：

该问题要求在最佳订购方案已经确定的情况下，确定最佳的转运方案，每家转运商的周平均转运损耗率为：

$$\overline{wastage} = \frac{\sum_{i=1}^{t_{\text{供货}}} wastage_i}{t_{\text{供货}}} \quad (17)$$

（ $wastage_i$ 为第 i 周的运输损耗率， $t_{\text{供货}}$ 为供货总天数）

各转运商的优劣排名与 $\overline{wastage}$ 值的大小成反比。由于题目要求的时间周期为 24 周，附件二中给出了 5 年的损耗率数据，故我们考虑，某转运商在未来 24 周内出现的损耗率数据与过去 5 年的数据具有一定的映射关系。我们通过对过去五年非 0 的转运损耗率随机抽样，作为转运商未来 24 周的转运损耗率。则转运损耗最小化的目标函数如下：

$$\min(\overline{Tramount}_{\text{总}}) = \sum_{j=1}^8 wastage_{ji} Tramount_i \quad (18)$$

（ $\overline{Tramount}_{\text{总}}$ 为对应周总的原材料运输损失量 $wastage_{ji}$ 为第 j 个转运商在第 i 周的损失率 $Tramount_i$ 为对应转运商第 i 周的转运总量 $j=1,2,\dots,8, i=1,2,\dots,24$ ）

约束条件：

在最优订购方案已经确定的情况下，问题求解需要满足的约束条件为：

$$Tramount_{ji} \leq 6000 \quad (19)$$

（ $Tramount_{ji}$ 为第 j 家转运商在第 i 周的转运总量）

其次考虑到转运商的转运能力差异，应当尽可能饱和优秀转运商的转运能力，既 $\lim_{\rightarrow} \frac{|Tramount_{ji} - 6000|}{6000}$ 的值尽可能小。

6.3 模型求解

6.3.1 确定供应商的最小数量问题求解

将附件一数据依照要求带入上式计算，可得供应商的最小数量为 23 个，下面为得到的 23 家供应商的相关数据：

表 2 求得的 23 家供应商数据表

供应商 ID	材料分类	供应商平均产能	累计产能
S229	A	2464.493	2464.493
S361	C	1898.611	4363.104
S140	B	2089.712	6452.816
S108	B	1521.149	7973.965
S282	A	1175.972	9149.938
S151	C	1125.567	10275.5
S275	A	1101.063	11376.57
S329	A	1086.931	12463.5
S340	B	1082.235	13545.73
S139	B	1036.459	14582.19
S131	B	868.1313	15450.32
S308	B	864.8864	16315.21
S330	B	862.702	17177.91
S356	C	754.0914	17932
S268	C	751.0764	18683.08
S306	C	729.7222	19412.8
S348	A	793.9948	20206.8
S352	A	618.2708	20825.07
S201	A	4880.298	25705.36
S194	C	586.603	26291.97
S307	A	762.1442	27054.11
S143	A	574.9097	27629.02
S395	A	1708.176	29337.2

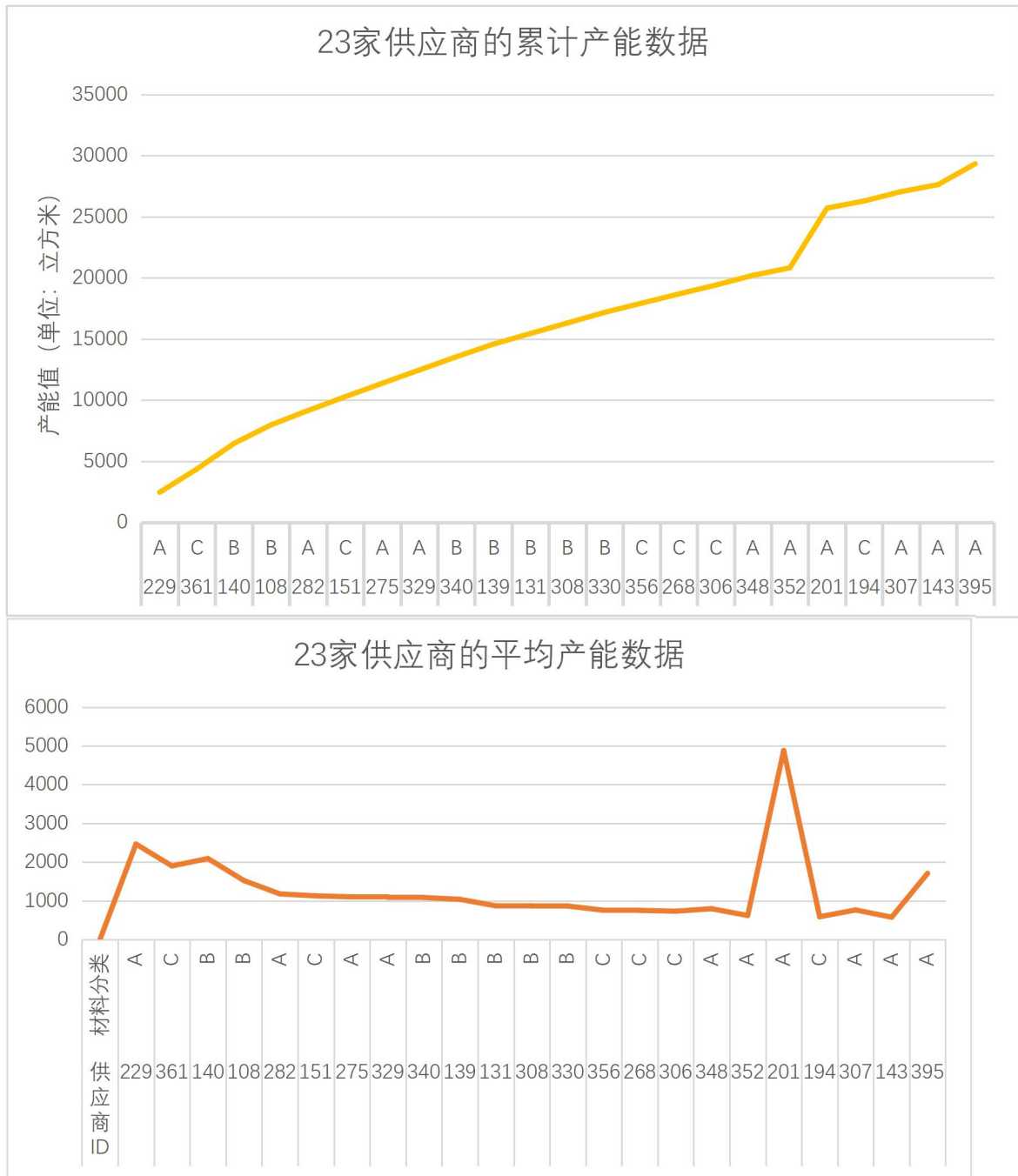


图 9 23 家供应商的相关数据（图中最下的数据为供应商 ID，均省略 S）

6.3.2 最优订购方案规划问题模型求解

供应商按照其供货特点，分为两大类，第一类由稳定型与综合型供应商组成，第二类由突变型供应商组成。对附件一中供应商的历史供货数据进行分析后，将 6.3.1 中 23 家供应商依照供货类型进行分类，可得下表：

表 3 供应商的供应类型分类表

供应类型	供应商 ID
稳定型	S229 S361 S282 S275 S329 S340 S131 S356 S268 S306 S352 S194 S143 S395

综合型	S108 S201 S348 S330 S308 S151
突变型	S140 S139 S307

将第一问得到的 50 家最重要公司 5 年来的供货量随时间变化的图像画出，可以清晰的得到三类供应商的供货特点如下：

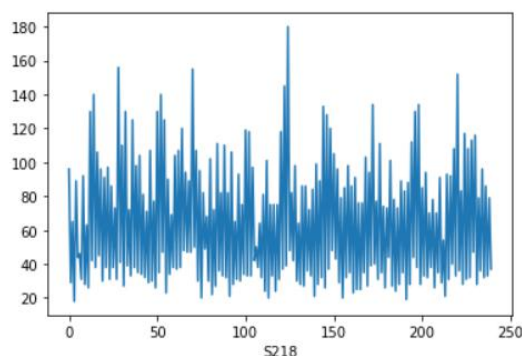


图 10 稳定型供应商供货量数据图像

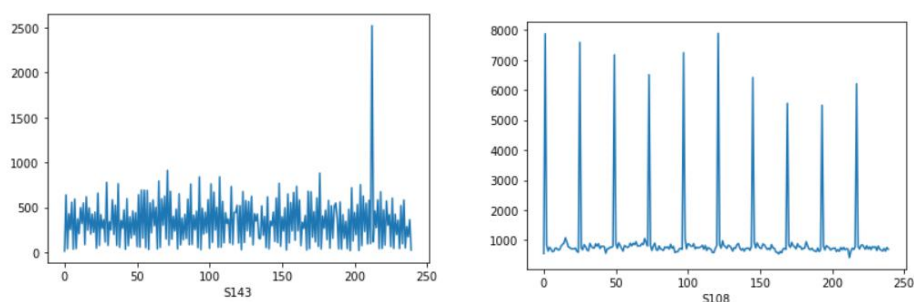


图 11 综合型与突变型供应商供货量数据图像（左综右突）

转运商选择供应商供货，采用动态规划模型求解。将制定转运方案分为 N 次单个转运商匹配问题，转运商按照排名先后相对供应企业进行选择。

对于综合型供应商，可以发现，其在五年的时间内，出现供货量突变的次数极少，故在实际的订购过程中，这种微弱的突变性对订购方案的选择产生的影响极小，将该类型供应商其他时间的供货量的中位数代替其突变值，即可将综合型供应商视为稳定型供应商。

对于突变型供应商，采用遗传算法进行预估其订购量，从 S108 供应商供货量数据图像可以看到，其突变时间具有明显的周期性，集中体现出每 12 周一次的趋势，故定义突变型供应商产生突变，那么设定其既能够接收大额订单并及时供货的概率为 $\frac{1}{12}$ 。对该类企业的供货突变值进行数据统计后，发现其大部突变值均超过了 6000 立方米/周（单个转运商的周转运最大值），则我们对选中的突变型原材料供应商的突变值均取 2000-6000 立方米/周，以便于在选择供货商时，避免出现向单家供应商发出超过单个转运商的周转运最大值的订单。且该类供应商只接收该突变值区间内的订单。

企业生产的第一周，由于没有储备，为保证接下来企业产能生产的稳定性，故企业在该周订购的原材料应该保证产能达到 28200×2 立方米，在 2.3.1 中得到的 23 家供应商中，S108 供应商为综合型企业，其具备较强的短期大量供应能力，故第一周选取 S108 企业，并向其订购 6000 立方米的原材料，以保证订购量的总量规模。

对于稳定型供应商，每周供货量比较稳定，采用蒙特卡洛算法，随机生成订购量，每周该公司宜向其订购 5 年内区间为 0 到单周最大供货量之间的随机订单，订单随机数生成符合正态分布，以正态分布概率随机生成其订购量。并且突变型企业以 $\frac{1}{12}$ 的突变概率参与订单的接收，在上文给出的约束条件下，产生第一次的最优订购方案，得到一个最小收购成本

$$\min(Fiprice) = 1.2 \sum x_A + 1.1 \sum x_B + \sum x_C \quad (20)$$

的值，再以同样的方法进行二次，三次...n 次运算，每次都会得到一个最新的最小收购成本，并对最小收购成本进行更新，直至迭代次数达到 1000 次，则停止迭代，取最新的值为该周的最小收购成本，得到最优订购方案，进而进入下一周的规划。

求解思路如下图所示：

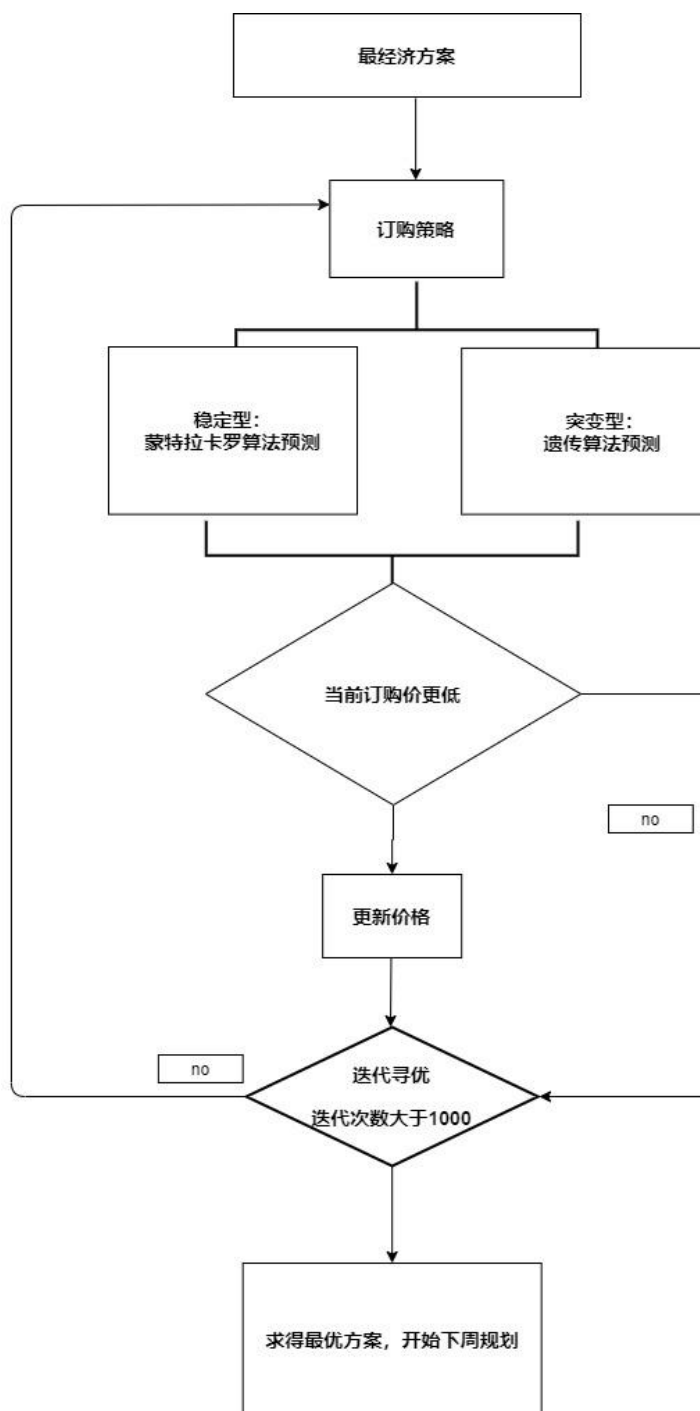


图 12 最优订购方案规划问题求解思路图

在进行大量的实验寻优后，我们发现稳定型供应商的订单均在其 5 年内供应峰值的 40%-90%之间，故将每个供货商的随机订单的订购区间修改为该供应商 5 年内供应峰值的 40%-90%之间，可以发现，其迭代求解次数大大减少，求解效率提高。

6.3.3 最优转运方案模型的求解

转运商转运水平的求解：

将附件二中转运商的运输损耗率带入 6.2.3 的对应计算公式中，求得八家转

运商的周平均运输损耗率如下图所示：

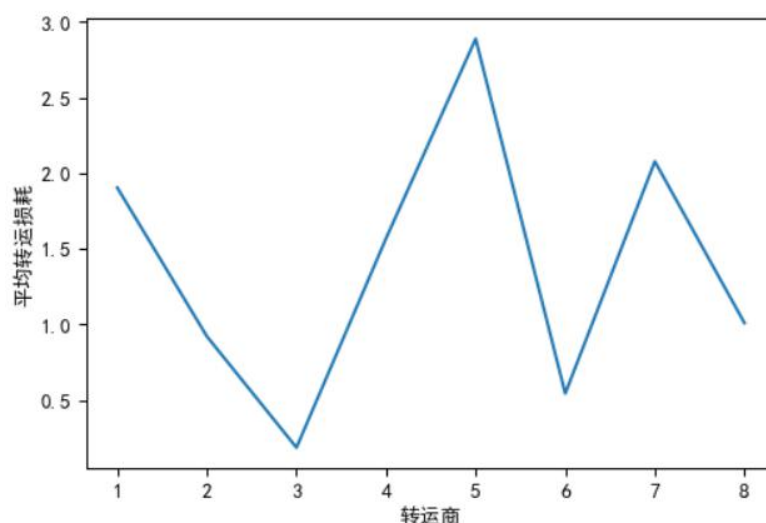


图 13 转运商的周平均运输损耗率

可得转运商的优劣顺序为（由优到劣顺序排列）：3、6、2、8、4、1、7、5

转运方案的求解：

Step1:

找到最优转运商，从 6.3.2 中找到该周的最优订购计划，对订购的原材料量按从大到小进行排序。

Step2:

首先将目前所剩的订单中原材料订购量最大的订单交给当前的最优转运商，判断转运商当前的剩余转运能力是否能够接收该订单，如果剩余转运能力足够，则接收该订单；若剩余转运能力不足以接收，则转向考虑仅次于该订单的订单，直至当周计划中原材料订购量最小的订单。

Step3:

若该周订单仍有剩余，则重复以上两步的步骤，直至该周的订单清零，则此时的转运方案既为该周最优的转运方案。

6.4 订购方案和转运方案的实施效果分析

问题二为两次单目标规划，第一个目标是每周订购方案最经济，即满足产能的条件下，运材料购买费用最经济。通过模型建立与求解可以得出，原材料购买费用最少为 21825。

当理想情况下，即每周原材料对应产能刚好满足产能要求，准备两周的原料储备粮，且损耗率不计的情况下，原材料购买费用应该为：21150~21326。费用接近，在误差范围内，说明该方案为最经济的原材料订购方案。

第二个目标为损耗率最小，根据损耗率定义，不考虑原材料种类。通过转运方案规划得到最小损耗方案，订购材料实际对应每周平均产能为：29940 立方米。订购材料得满足两周的原料储备量，所以订购材料实际对应每周平均产能大于 2.82 万立方米。

在供应商全部可以按时按量供应的情况下,可以满足产能要求,保障企业生产。由于三类原材料运输和储存的单位费用相同,在订购方案最优情况下,运输存储费用为常数。满足要求前提下,原材料费用最低,损耗率最小,分别达到最优,所以该订购方案和转运方案的实施效果良好。

七、问题三的模型建立与求解

7.1 问题分析

在尽量多采购 A 类,少采购 C 类的要求下,分析三类原材料对于生产单位产品的经济数据,得到原则上企业选择原材料类型的倾向。以各供应商的周平均供货量对应的产能之和满足企业的周产能需求为前提,对原材料供给侧进行修改,既重新筛选符合要求的供应商。

由于题目要求在订购方案产生的同时,对转运方案进行规划,达到每周原材料订购费用与转运损耗造成的经济损失之和最小的目标,将问题转化为在该目标下的单目标动态规划问题。

7.2 模型建立

7.2.1 原材料供应商的选择

由于生产单位产品对三类原材料的需求,以及原材料价格各不相同等因素的影响,企业应当在经济角度考虑优先选择哪类原材料进行生产。下表为生产单位产品之于三类原材料的相关数据:(此处定义 M 与 N 分别为单价与运输+储存价格的单位)

表 4 生产单位产品之于三类原材料的相关数据表

原材料类别	所需原材料量	单价	生产单位产品的成本	运输价格+储存价格
A	0.6 立方米	1.2M	0.72M	0.6N
B	0.66 立方米	1.1M	0.726M	0.66N
C	0.72 立方米	1M	0.72M	0.72N

分别将三类原材料的生产单位产品的成本与运输+储存价格相加,得到该原材料的对于产品生产的经济数据,该数据越大,说明订购该原材料进行生产的成本更高。从该表中可知,欲生产单位产品,选择 A 类原材料与 C 类原材料的生产成本相同,但 A 类原材料的运输+储存价格更低,故企业选择 A 类原材料进行生产优于选择 C 类材料,与题目要求吻合,故生产产品的原材料选择为: A 类最优, B 类次之, C 类最差。

与问题二类似,进行订购规划的前提是所选择供应商的周平均供货量的产能可以满足企业的周产能需求。在问题一求得的 50 家供应商中,以供应商供应原材料类型的不同将其分为 A 类、B 类、C 类供应商,首先在 A 类供应商中按照周平均供货量由高到低进行选择,其次是 B 类、C 类供应商,直至其供货量之和对应的产能达到企业的周产能需求,这时选择出的供应商即纳入本题最优订购方案规划时考虑的供应商群体中。

7.2.2 约束条件的确定

该问的约束条件与第二问中订购与转运方案的约束条件相同，均为所制定的订购计划中，各供应商的周原料供应量之和能够满足企业的周产能需求，且单个供应商周接收订单的原材料订购总量小于等于其最大周转运能力（6000 立方米）

7.2.3 目标函数的确立：

考虑在某订购方案下的原材料花费最少和对应于该订购方案的最优转运方案损耗造成的损失最小同时满足，则将两部分的支出（运输损耗视为额外支出）相加，之和最小的规划方案即为所求的方案。故该动态规划问题的目标函数如下：

$$\begin{aligned} \text{cost1} &= 1.2 \sum_{j=1}^N x_{jA} + 1.1 \sum_{j=1}^N x_{jB} + 1.0 \sum_{j=1}^N x_{jC} \\ \text{cost2} &= 1.2 \sum_{j=1}^N \sum_{i=1}^{24} E_{ji} x_{ji} + 1.1 \sum_{j=1}^N \sum_{i=1}^{24} E_{ji} x_{ji} + 1.0 \sum_{j=1}^N \sum_{i=1}^{24} E_{ji} x_{ji} \quad (21) \\ \min(\text{cost}) &= \min(\text{cost1} + \text{cost2}) \end{aligned}$$

（ x_{jk} , $k = A, B, C$ 为 j 供应商 24 周供应的 k 类原材料总量； x_{ji} 为 j 供应商在第 i 周的原材料供应量； E_{ji} 为供应商 j 在第 i 周选择的转运商转运时的随机损耗率； N 为纳入考虑的供应商总数）

该目标函数的 cost1 为原材料订购的费用， cost2 为转运商在转运过程中因原材料损耗而产生的损失费用， $\min(\text{cost})$ 为所求支出最小的目标函数。

7.3 模型求解

原材料供应商的确定：

在对问题一中筛选出的 50 家供应商按原材料供应类别进行分类后（且每类供应商内以周平均供应量由大到小顺序排列）根据 7.2.1 中依照 A 类、B 类、C 类供应商顺序进行选择的原则进行供应商筛选，直至选出的供应商的周平均供应量满足企业的周产能需求，共筛选出 31 家供应商进行原材料的供应。

总支出量最少（既目标函数最优解）的求解：

Step1:

在筛选出的 31 家供应商中，随机生成 24 周的订购方案。

Step2:

判断生成的订购方案是否满足产能与转运能力的约束条件，若不满足，则返回 Step1，若满足，则按照 6.2.3 中寻求最优转运方案的思路为生成的订购方案寻求转运方案。

Step3:

j 供应商在第 i 周的供应量可以由生成的订购方案确定，可以由匹配的最优转运方案找到 j 供应商在第 i 周的转运企业，从而得到 P_{ji} 。既一旦订购方案与和

其匹配的转运方案生成，就可以求得该方案下目标函数的最小值。并对该值进行动态更新。

Step4:

判断迭代次数是否大于 1000 次，若不够，则返回 Step1;若够，则此时的订购方案与转运方案为所求最优方案。

6.4 订购方案和转运方案的实施效果分析

1、问题二中每周采购不同类别原材料周平均数量如下：

A 类：8550 立方米每周 B 类：6757 立方米每周 C 类：4141 立方米

问题三中每周采购不同类别原材料周平均数量如下：

A 类：10525 立方米每周 B 类：7037 立方米每周 C 类：1563 立方米

通过数据对比可以得出：问题三中采购方案 A 远多于 C，有效减少了转运以及仓储成本。

2、本题为最经济且损耗率最低，建立动态规划模型。按照成本费用核算公式：
成本费用=原材料费用+损耗费用+运输存储费用

在最优情况下得出 24 周原材料成本费用总和为：533672

当采用问题二中分布规划计算的出最小成本费用为：544786

注：记原料 C 的单价为 1

通过对比，在同时考虑订购以及转运的情况下，调控 A、B、C 采购比例，建模求出最优采购和转运方案，更有利于提高工厂产能和较少损耗和转运费用。

八、问题四的模型建立与求解

8.1 问题分析

在最大产能不设上限的情况下，对 402 家供货商供应的原材料产能计算了，以判断理论产能供应限制。

另外，产能的提升取决于转运能力，则 8 家转运商均全力转运，以提升企业产能。且由第三问对于三类原材料的分析可得，应尽可能多的采购料 A，其次为 B 和 C。优先选用转运损耗较小的转运商转运原料 A，通过调节原材料比例以及损耗量来提升产能。

8.2 模型建立

8.2.1 数据处理

对供应商重新进行评价，选用第一问中综合评价模型，根据综合得分对全部供应商进行排序。通过对排名情况以及该供应商过去五年供货量进行数据分析可

得，排名靠后的供应商平均供货量很小，对于企业产能影响极小。故优先选用排名靠前的供货商进行供应。

8.2.2 重要供应商选择

首先计算所有供应商的平均供货量，作为供应商供应能力的评价值 \bar{x}_i ，接着按照排名顺序计算累加平均供应能力 $\sum \bar{x}_i$ 。已知共有 8 家转运商，每家转运商均以最高转运能力运转，最大转运能力为：

$$Z_{\max} = 6000 \times 8 = 48000 \quad (22)$$

当累加平均供应能力 $\sum \bar{x}_i > 50000$ 时，以上部分供应商可以满足转运能力最大的要求，将以上供货商选择为企业的重要供货商。

根据原材料种类对于产能的影响，应尽可能多的采购料 A，其次为 B 和 C。对以上重要供货商进行二次排序，在综合得分排名基础上，按照 A、B、C 的顺序进行二次数据排序，此数据即为重要供应商以及排名。并依据过去 250 周供货数据，以供货方差为依据，将供应商划分为稳定型供应商和突变型供应商。

8.2.3 模型简化

1、由于产能不受限制，则对于原料需求量激增，所有转运商的转运能力均用于制造产品，来达到长期稳定高效生产，则不考虑满足两周生产需求的原材料库存量要求。

2、由于供应商分为稳定型供应商和应急性供应商，根据应急性供应商的特点，无法长期供应，可以短期内供应大量值。由于产能不受限制，则应急型供应商需要长期进行原料补充。在此问题中，将应急型供应商供应数据进行平稳化处理，使其既满足供应商生产原材料规律，又有利于进行原材料补充，使得最大限度达到最大转运能力，以增加企业产能。

8.2.3 寻找最优订购以及转运方案

1、初始化重要供应商可能的订货量：

取随机值，由于产能不受限制，则对于稳定型企业的订货量应大于过去订货的平均值，以增加原料供给量。初始化采用随机数法给出，根据过去 250 周供货数据，供货商的供货最大值很可能偏离整体数据，为了避免极端数据的影响，则选择中位数作为衡量指标。随机数选择中位数和 1.5 倍中位数之间的随机整数，以此值作为该供应商初始化数据。按照此规则对第一周选择的所有重要供应商进行初始化。

2、寻找最优转运方案：

为了达到产能最大，采用逆向选择思维，将供应商选择转运商转变为转运商选择供应商。

首先计算 8 家转运商的平均损耗，用来衡量该转运商的转运能力，进行排序。平均损耗率最小的转运商优先进行选择，依照供应商排序表的顺序依次选择，直至最大限度的满足最大转运量。

算法思路如下：

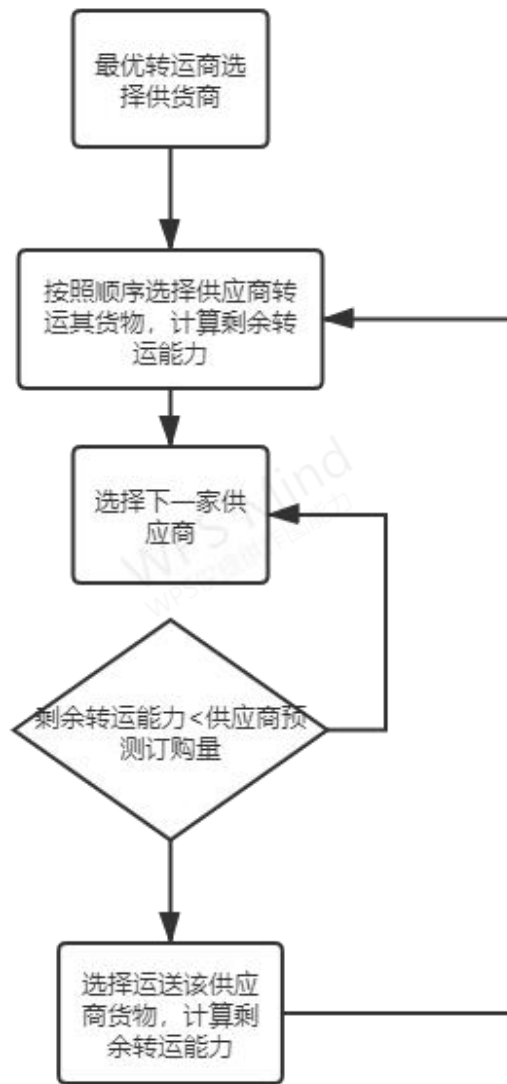


图 14 最优转运商的选择思路流程图

当剩余转运能力为 0 或者对所有供货商进行遍历完成之后，该转运商转运方案制定，由下一家转运商对剩余供货商进行选择。

按照此算法，平均损耗量小的转运商将优先运送 A 产品，损耗量最小，产能可达到最大，寻找出最优转运方案

。

3、寻找全局最优解：

在所给范围内，对重要供应商可能的订货量重新预测，，每次预测都按照供应商选择算法寻找最优配送方案，计算并记录每一周产能，据此求出该企业周平均产能。

目标函数即为周平均产能：

$$\bar{W}_{max} = \frac{\sum_{i=1}^{24} \frac{(\sum X_p - \sum X_q)}{M_p}}{24} \quad (23)$$

$\sum X_p$ 指的是所选择的供应商一周内不同原材料供应量

$\sum X_q$ 为不同原材料的损耗量

M_p 为对应原材料的产品转化率

计算迭代 10000 次，即可找出使得产能最大的订购方案及转运方案。可提升产能即为：

$$\Delta W = \bar{W}_{max} - W_0 \tag{24}$$

W_0 ：为目前企业每周的产能，为常数 2.82 万立方米

8.3 模型求解

8.3.1 重要供货商选择

依照模型求解得到的转运商数据见支撑材料，下图为达到要求的截取部分：

材料分类 供应商平均供应能力 累计供应能力			
供应商ID			
S229	A	3191.98	3191.98
S361	A	2629.66	5821.64
S140	B	2765.28	8586.92
S108	C	2079.56	10666.5
S282	A	1549.82	12216.3
...
S104	B	11.7754	47898.3
S228	B	19.2707	47917.5
S112	B	51.6964	47969.2
S394	A	27.9335	47997.2
S017	C	10.8782	48008

图 15 供应商相关数据图

由表可得，前 134 家供应商即可满足最大运力要求（在表中，供应商 S017 为第 134 家供应商），所以选择这 134 家供应商即可满足要求。

8.3.2 最佳采购以及转运方案求解

通过多次程序实验，当订购方案迭代 10000 次，就可以找到最佳的订购以及转运方案。

周数	平均产能	周数	平均产能	周数	平均产能	周数	平均产能
1	77482	7	77273	13	82157	19	76881
2	74899	8	79464	14	75774	20	79549
3	77772	9	77445	15	78266	21	77145

4	79811	10	75969	16	76471	22	76530
5	75126	11	76904	17	78666	23	75991
6	81203	12	77647	18	75658	24	74802

周 7 平均产能为：7.4 万立方米

3、平均每周可提升产能：4.93 万立方米

九、模型评价

9.1 模型优点

- 1、数据特征分析充分，根据以往转运商订货供货数据特点，考虑到生产过程中实际情况，将供应商分为稳定型供应商和实际型供应商，选择保障企业生产的重要供应商时，同时选择了两种供应商，并制定不同的订货策略，可以有效应对突发情况，更符合实际生产过程。
- 2、损耗率采用随机抽样，在企业在选择转运商时，无法确定具体损耗率，只能通过以往数据进行评价，使得结果具有一定的随机性，更符合现实情况。
- 3、采用蒙特卡洛算法和遗传算法，分别针对两种供应商，使得订购方案更科学。
- 4、转运方案制定采用动态规划，兼顾效率与准确性。

9.2 模型缺点

- 1、在处理突变型供应商的供货数据时，具有一定的主观性。
- 2、模型考虑的影响因素不够全，如仓储费用随时间的变化。
- 3、多目标规划过程中，寻优复杂度较高，模型运行效率较低。

9.2 模型改进

- 1、算法合理调整参数，以提高运行效率
- 2、优化算法结构，提高寻优准确性。

十、参考文献

- [1] 司守奎. 数学建模算法与应用[M]. 北京：国防工业出版社. 2016-2.
- [2] 葛显龙, 王伟鑫, 李顺勇. 智能算法及应用[M]. 陕西：西安交通大学出版社 2017-08.

附录

附录一：50 家企业综合得分排名表

企业名称	综合得分	企业名称	综合得分	企业名称	综合得分	企业名称	综合得分
S229	316917.2435	S356	96052.94439	S037	35852.67231	S086	10582.73309
S361	225420.1526	S268	95603.50132	S031	34851.93137	S210	9818.638453
S140	224845.9604	S306	93901.01957	S365	31968.99077	S003	9038.902209
S108	195778.5333	S348	83542.05506	S040	30305.73908	S114	8490.263903
S282	156587.1989	S352	76396.70446	S338	23961.95622	S273	8350.85536
S151	145125.0488	S201	75193.20529	S364	21733.90959	S189	8208.411534
S275	140854.36	S194	74765.6373	S367	21661.38842	S078	7153.26208
S329	139104.1277	S307	73326.15308	S055	19579.47875	S292	6910.421161
S340	137802.0682	S143	69338.65548	S346	17308.15821	S074	6907.022773
S139	124401.6012	S395	67788.08345	S080	16284.18955	S291	6873.309978
S131	106515.8348	S247	41520.27762	S294	12436.92084		
S131	104000.9205	S374	38473.73904	S218	12286.25891		
S330	103851.5722	S284	36199.1097	S244	10623.68767		

附录二：关键代码：

第一问关键代码

#材料分类系数

Dtype = list(data1['材料分类'])

Ddata = [0.95,0.86,0.79]

D = []

for i in Dtype:

 if i == 'A':

 D.append(Ddata[0])

 elif i == 'B':

 D.append(Ddata[1])

 elif i == 'C':

 D.append(Ddata[2])

 else:

 continue

#构造趋势函数

def f(x):

 if x < 0:

 return 4-4*(0.8)**x

 elif x >= 0:

 return (0.5)**x

 else:

 pass

#供货量与订货量的差值比企业订货量

```

temp4=temp3/temp1
temp4 = temp4.fillna(0)

#计算供货商每周得分
roworder = 0
ttp = []
for row in temp3.itertuples():
    lineorder = 0
    rowlist = []
    for i in list(row)[1:]:
        #计算供应商每周对应综合得分
        tp =
f(np.array(temp4)[roworder][lineorder])*abs(np.array(temp2)[roworder][lineorder])*
D[roworder]
        rowlist.append(tp)
        lineorder +=1
    roworder +=1
    ttp.append(rowlist)

#计算供货商总得分
d1 = []
for i in ttp:
    d1.append(sum(i))

#构造供货商总得分表
df1 = pd.DataFrame(d1)

#构造问题一供货商综合得分表
tttp = ttp.copy()
count = 0
for i in tttp:
    ct = 0
    i.append(Dtype[count])
    i.append(d1[count])
    count +=1
index = temp1.index.values
columns = data1.columns.values[2:]
index = list(index)
columns = list(columns)
columns.append('材料分类')
columns.append('得分')
w1_defen = pd.DataFrame(tttp,index=index,columns=columns)

#供货商综合得分表排序后得到最优的 50 个供货商
df1_data = df1.sort_values(by=[0],ascending=False).head(50)

#第二问关键代码

```

```

#50 家供应商的一周总平均提供
temp5 = dff1.set_index(0).drop([1],axis=1)

tp = temp5.std(axis=1)#具体方差表
tmp = temp5.median(axis=1)#具体中位数

#按顺序提取供应商
index = list(dff1[0])
pj_values = list(mean_temp5)
sum_cn = 0
ct = 0
count = 0
pjcn = []
ljcn = []
for i in mean_temp5:
    sum_cn = sum_cn + (i/D50[ct])
    ljcn.append(sum_cn)
    pjcn.append((i/D50[ct]))
    if sum_cn < 28200:
        count+=1
        ct+=1
        continue
    else:
        break
gys = list(dff1[0].values)[0:count+1]
cpt = Dtype50[0:count+1]
gys23 = pd.DataFrame([gys,cpt,pjcn,ljcn],index=['供应商 ID','材料分类','供应商平均产能','累计产能']).T.set_index('供应商 ID')# 选择的 23 个供应商

#for i in gys:
#    if i in tubian_list
tpw = []
for i in gys:
    tpw.append(np.array(temp5.T[i]))
biao23 = pd.DataFrame(tpw)# 选择的 23 个供应商具体数据

#第一周随机序列
def first_suiji():
    #23 家分类
    tubian_id = ['S140','S139','S307','S395']
    yichang_id = ['S201','S348','S330','S308','S151']
    teshu_id = ['S108']
    wending_id = []
    wending_type=[]

    for i in gys:
        if (i not in yichang_id )and (i not in tubian_id) and (i not in teshu_id):
            wending_id.append(i)

```

```

wending_type.append(gys23.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值
yic = []
yic_type = []
for j in yichang_id:
    addres = gys.index(j)
    yic.append(np.array(biao23.loc[addres]))
    yic_type.append(gys23.loc[j]['材料分类'])
for k in yic:
    for m in k:
        if m > 6000:
            addres = list(k).index(m)
            k[addres] = np.median(k)#异常值使用中位数
yic_suiji_xulie = []
for t in yic:
    yic_max=t.max()
    yic_min=t.min()
    yic_suiji_xulie.append(np.random.randint(yic_max*0.65,yic_max, 1))

#特殊值取行
addres = gys.index(teshu_id[0])
tes = np.array(biao23.loc[addres])

#稳定值取行
wen = []
for j in wending_id:
    addres = gys.index(j)
    wen.append(np.array(biao23.loc[addres]))

#稳定值取随机序列
wen_up = []
wen_down = []
for i in wen:
    wen_up.append(i.max())
    wen_down.append(i.max()*0.65)

wen_suiji_xulie = []
for i in wen_up:
    addres = wen.index(i)
    wen_suiji_xulie.append(np.random.randint(wen_down[addres],i, 1))

#总的稳定值随机序列
wenzong = wen_suiji_xulie+yic_suiji_xulie
wenzong_type = wending_type+yic_type
for i in yichang_id:
    wending_id.append(i)

```

```

#特殊值取随机序列后加上最大值
text = []
tq = []
te_type = gys23.loc[teshu_id[0]]['材料分类']
for i in tes:
    if i != tes[list(tes).index(tes.max())]:
        text.append(i)
        tq.append(i)
    else:
        text.append('s')
text[text.index('s')] = np.array(tq).mean()
tes_up = np.array(text).max()
tes_down = np.array(text).min()
te_suiji_xulie = np.random.randint(tes_up*0.65,tes_up, 24)
tes_max = list(tes).index(tes.max())
#te_suiji_xulie[random.randint(0,24)] = tes_max
#te_suiji = te_suiji_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周
tb = []
tb_type = []
for j in tubian_id:
    tb_type.append(gys23.loc[j]['材料分类'])
    tp = np.linspace(0,0,24)
    addres = np.random.randint(0,24, 2)
    if (addres[0] == 0) or (addres[1] == 23):
        tb.append(np.random.randint(5999,6000, 1))
    else:
        tb.append(0)

return
wenzong,tb,np.array([6000]),wenzong_type,tb_type,te_type,[wending_id,tubian_id,te
shu_id]

```

#后面几周随机序列函数详见支撑材料

```

#第一周随机
result1 = []
mubiao1_list = []
P1 = []
for i in range(1000):
    suiji_list = first_suiji()
    count +=0
    t2 = []
    t2.append(int(suiji_list[2]))
    suiji_data = suiji_list[0]+suiji_list[1]+t2
    suiji_type = suiji_list[3]+suiji_list[4]
    suiji_type.append(suiji_list[5])

```

```

gh = pd.DataFrame([suiji_type,suiji_data]).T
A = gh[gh[0] == 'A'][1].values.sum()
B = gh[gh[0] == 'B'][1].values.sum()
C = gh[gh[0] == 'C'][1].values.sum()
if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200*2)):
    P1.append(A/0.6+B/0.66+C/0.72)
    mubiao1 = 1.2*A+1.1*B+C
    mubiao1_list.append(mubiao1)
    result1.append(suiji_list)
else:
    continue
addres = mubiao1_list.index(min(mubiao1_list)[0])
fliter = []
for i in result1[addres][0]:
    fliter.append(int(i))
for i in result1[addres][1]:
    fliter.append(i)
fliter.append(int(result1[addres][2]))
fliter.append(int(P1[addres]))
fliter.append(int(min(mubiao1_list)))
fliter1 = fliter.copy()

#后面几周随机详见支撑材料
#随机转运的转运商选择结果 24
def suiji_zhuanyun(count):
    zy1 = []
    for i in range(0,count):
        zy1.append(int(np.random.randint(1,9,1)))
    return zy1

#随机损耗对应 1~8 转运商
def suiji_sunhao():
    ss = []
    tpq_suiji = []
    for i in tpq:
        tpq_suiji.append(random.sample(list(np.array(i)),1)[0])
    return tpq_suiji

#对应的概率
sjz = suiji_zhuanyun(23)
sh = suiji_sunhao()
temp = []
for i in sjz:
    temp.append(sh[i-1])
c=0
q = []
for i in temp7.values[0:-2]:#.sort_values(by=k-1)[-1][k-1]
    t = list(i)
    t.append(c)

```

```

q.append(t)
c+=1
temp1 = pd.DataFrame(q)
pm = sh_p[0].values.copy()
tp111 = [] #记录选中供货商值
tp222 = [] #记录选中供货商和
tp333 = [] #记录供货商名字
tp444 = [] #记录转运商名字
#tp555 = 0 #记录检索到哪一位置供应商了
tp666 = [] #记录供货商供货量
for k in range(2,26):
    ghs_tp = temp1.sort_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小
    ghs_mz = temp1.sort_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小
    #zys_tp = temp1.sort_values(k-1)[::-1].iloc[:,-1]#对应损失率
    tp11 = [] #记录选中供货商值
    tp22 = [] #记录选中供货商和
    tp33 = [] #记录供货商名字
    tp44 = [] #记录转运商名字
    #tp55 = 0 #记录检索到哪一位置供应商了
    tp66 = [] #记录供货商供货量
    sjz_t = sjz.copy()
    pm = sh_p[0].values.copy()
    for j in pm:
        #try:
        tp1 = [] #记录选中供货商值
        tp2 = 0 #记录选中供货商和
        tp3 = [] #记录供货商名字
        tp4 = [] #记录转运商名字
        #tp5 = 0 #记录检索到哪一位置供应商了
        tp6 = [] #记录供货商供货量
        count = 0
        yz = []
        for i in ghs_mz:
            try:
                yz.append([i,ghs_tp[count]])
            except:
                yz.append([i,0])
            count+=1
        for i in yz:
            tp2 +=i[1]
            if tp2 <= 6000:
                tp3.append(i[0])
                tp6.append(i[1])
            else:
                tp2 -=i[1]
            tp5+=1
            tp5 = j

```



```

        ghs_tp = [x for x in ghs_tp if x not in tp6]
        pm = pm[1:]
        ghs_mz = [x for x in ghs_mz if x not in tp3]
    tp11.append(tp1)
    tp22.append([tp2])
    tp33.append(tp3)
    tp44.append(tp4)
    #tp55.append([tp5])
    tp66.append(tp6)
    #except:
        #continue
    tp111.append(tp11)
    tp222.append([tp22])
    tp333.append(tp33)
    tp444.append(tp44)
    #tp555.append([tp55])
    tp666.append(tp66)

kkk = []
for k in range(24):
    pm = sh_p[0].values.copy()
    t = []
    for j in pm:
        tp = []
        for i in range(8):
            name = tp333[k][i]
            shuju = tp666[k][i]
            tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set_index(0))
        t.append(tp)

    kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis
=1))
result = pd.concat(kkk,axis=1)
result.to_csv('转运方案 2 终.csv')

```

#第三问关键代码

分步式考虑

#第一周随机序列

```

gys = list(gyswz.index.values)
def first_suiji_3():
    #31 家分类
    tubian_id = ['S140','S307','S395','S139']
    yichang_id = ['S151','S308','S330','S348','S201','S143']
    teshu_id = ['S108']
    wending_id = []
    wending_type=[]

```

```

for i in gys:
    if (i not in yichang_id )and (i not in tubian_id) and (i not in teshu_id):
        wending_id.append(i)
        wending_type.append(gyswz.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值
yic = []
yic_type = []
for j in yichang_id:
    addres = gys.index(j)
    yic.append(np.array(biao31.loc[addres]))
    yic_type.append(gyswz.loc[j]['材料分类'])
for k in yic:
    for m in k:
        if m > 2500:
            addres = list(k).index(m)
            k[addres] = np.median(k)#异常值使用中位数
yic_suiji_xulie = []
for t in yic:
    yic_max=t.max()
    yic_min=t.min()
    yic_suiji_xulie.append(np.random.randint(yic_max*0.65,yic_max, 1))

#特殊值取行
addres = gys.index(teshu_id[0])
tes = np.array(biao31.loc[addres])

#稳定值取行
wen = []
for j in wending_id:
    addres = gys.index(j)
    wen.append(np.array(biao31.loc[addres]))

#稳定值取随机序列
wen_up = []
wen_down = []
for i in wen:
    wen_up.append(i.max())
    wen_down.append(i.max()*0.65)

wen_suiji_xulie = []
for i in wen_up:
    addres = wen_up.index(i)
    wen_suiji_xulie.append(np.random.randint(wen_down[addres],i, 1))

#总的稳定值随机序列
wenzong = wen_suiji_xulie+yic_suiji_xulie
wenzong_type = wending_type+yic_type

```

```

for i in yichang_id:
    wending_id.append(i)

#特殊值取随机序列后加上最大值
text = []
tq = []
te_type = gyswz.loc[teshu_id[0]]['材料分类']
for i in tes:
    if i != tes[list(tes).index(tes.max())]:
        text.append(i)
        tq.append(i)
    else:
        text.append('s')
text[text.index('s')] = np.array(tq).mean()
tes_up = np.array(text).max()
tes_down = np.array(text).min()
te_suiji_xulie = np.random.randint(tes_up*0.65,tes_up, 24)
tes_max = list(tes).index(tes.max())
#te_suiji_xulie[random.randint(0,24)] = tes_max
#te_suiji = te_suiji_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周
tb = []
tb_type = []
#for i in tubian_id:
#    addres = gys.index(i)
#    tubian.append(np.array(biao31.loc[addres]))
for j in tubian_id:
    tb_type.append(gyswz.loc[j]['材料分类'])
    tp = np.linspace(0,0,24)
    addres = np.random.randint(0,24, 2)
    if (addres[0] == 0) or (addres[1] == 23):
        tb.append(np.random.randint(5999,6000, 1))
    else:
        tb.append(0)

return
wenzong,tb,np.array([6000]),wenzong_type,tb_type,te_type,[wending_id,tubian_id,te
shu_id]

#后面几周随机序列详见支撑材料

#随机转运的转运商选择
def sui_ji_zhuanyun_3():
    zy1 = []
    for i in range(0,31):
        zy1.append(int(np.random.randint(1,9,1)))
    return zy1

```

```

# 分步式考虑
#第一周
result1 = []
mubiao1_list = []
P1 = []
for i in range(1000):
    sui_ji_list = first_sui_ji_3()
    count += 0
    t2 = []
    t2.append(int(sui_ji_list[2]))
    sui_ji_data = sui_ji_list[0]+sui_ji_list[1]+t2
    sui_ji_type = sui_ji_list[3]+sui_ji_list[4]
    sui_ji_type.append(sui_ji_list[5])
    gh = pd.DataFrame([sui_ji_type,sui_ji_data]).T
    A = gh[gh[0] == 'A'][1].values.sum()
    B = gh[gh[0] == 'B'][1].values.sum()
    C = gh[gh[0] == 'C'][1].values.sum()
    if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200*2)):
        P1.append(A/0.6+B/0.66+C/0.72)
        mubiao1 = 1.2*A+1.1*B+C
        mubiao1_list.append(mubiao1)
        result1.append(sui_ji_list)
    else:
        continue
addres = mubiao1_list.index(min(mubiao1_list)[0])
fliter = []
for i in result1[addres][0]:
    fliter.append(int(i))
for i in result1[addres][1]:
    fliter.append(i)
fliter.append(int(result1[addres][2]))
fliter.append(int(P1[addres]))
fliter.append(int(min(mubiao1_list)))
fliter1 = fliter.copy()

#后面几周随机函数详见支撑材料
#筛选对应材料的供货商
p = int(P1[addres])
fliter_list = []
P_list = [int(P1[addres])]
CT = 0
for i in range(2,25):
    result2 = []
    mubiao2_list = []
    Pn = []
    p=P_list[CT]
    for i in range(200):
        try:

```

```

suiji_list = suiji_3()
count +=0
t2 = []
t2.append(int(suiji_list[2]))
suiji_data = suiji_list[0]+suiji_list[1]+t2
suiji_type = suiji_list[3]+suiji_list[4]
suiji_type.append(suiji_list[5])
gh = pd.DataFrame([suiji_type,suiji_data]).T
A = gh[gh[0] == 'A'][1].values.sum()
B = gh[gh[0] == 'B'][1].values.sum()
C = gh[gh[0] == 'C'][1].values.sum()
if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200*2)-p):
    Pn.append(A/0.6+B/0.66+C/0.72)
    mubiao2 = 1.2*A+1.1*B+C
    mubiao2_list.append(mubiao1)
    result2.append(suiji_list)
else:
    continue
except:
    continue
addres = mubiao2_list.index(min(mubiao2_list)[0])
fliter = []
for i in result2[addres][0]:
    fliter.append(i[0])
for i in result2[addres][1]:
    tmp = []
    try:
        for j in i:
            if j != 0:
                tmp.append(j[0])
            else:
                tmp.append(j)
        for k in tmp:
            fliter.append(k)
    except:
        fliter.append(i)
fliter.append(int(result2[addres][2]))
fliter.append(int(Pn[addres]))
fliter.append(int(min(mubiao2_list)))
fliter_list.append(fliter)
P_list.append(int(Pn[addres]))
CT +=1

```

#整体订购方案

```

t = [fliter1]
for i in fliter_list:
    t.append(i)
ttp = []
for i in suiji_list[-1]:

```

```

        for j in i:
            ttp.append(j)
ttp.append('原料对应产能')
ttp.append('原材料费用')
dinggoufangan=pd.DataFrame(t,columns=ttp,index=list(np.linspace(1,24,24))).T
t = []
for i in dinggoufangan.values:
    tttt = []
    for j in i:
        tttt.append(int(j))
    t.append(np.array(tttt))
dinggoufangan3 =
pd.DataFrame(np.array(t),index=dinggoufangan.index.values,columns=dinggoufangan.columns.values)
dinggoufangan3.to_csv('订购方案')

temp7 = dinggoufangan3.copy()
data3 = pd.read_excel('附件 2 近 5 年 8 家转运商的相关数据.xlsx')

temp6 = data3.set_index('转运商 ID')
tpq = []
for i in np.array(temp6):
    ttp = []
    for j in i:
        if j != 0:
            ttp.append(j*0.01)
        else:
            continue
    tpq.append(ttp)

ct = 1

for i in tpq:
    x = np.linspace(0,len(i),len(i))
    y = list(i)
    fig = plt.figure(figsize=(8,11),dpi=98)
    na=plt.subplot(4,1,1)
    na.plot(x,y,label='转运商 {}'.format(ct))
    na.legend()
    ct+=1
    plt.show()
#fig.savefig('zhuanyun.jpg')

#计算转运商信息
tp = []
for i in np.linspace(1,8,8):
    tp.append(int(i))
tpq_mean = []
for i in tpq:

```

```

    tpq_mean.append(np.array(i).mean())
plt.figure(dpi=95)
plt.ylabel('平均转运损耗')
plt.xlabel('转运商')
plt.plot(tp,tpq_mean)
plt.savefig('zhunayunpj.jpg')
plt.show()
sh_p = pd.DataFrame([tp,tpq_mean]).T.sort_values(by=1)
#随机转运的转运商选择结果 24
def sui_ji_zhuanyun(count):
    zy1 = []
    for i in range(0,count):
        zy1.append(int(np.random.randint(1,9,1)))
    return zy1

#随机损耗 1~8
def sui_ji_sunhao():
    ss = []
    tpq_suiji = []
    for i in tpq:
        tpq_suiji.append(random.sample(list(np.array(i)),1)[0])
    return tpq_suiji

#对应的概率
sjz = sui_ji_zhuanyun(23)
sh = sui_ji_sunhao()
temp = []
for i in sjz:
    temp.append(sh[i-1])
c=0
q = []
for i in temp7.values[0:-2]:#.sort_values(by=k-1)[::-1][k-1]
    t = list(i)
    t.append(c)
    q.append(t)
    c+=1
temp1 = pd.DataFrame(q)
pm = sh_p[0].values.copy()
tp111 = [] #记录选中供货商值
tp222 = [] #记录选中供货商和
tp333 = [] #记录供货商名字
tp444 = [] #记录转运商名字
#tp555 = 0 #记录检索到哪一位供应商了
tp666 = [] #记录供货商供货量
for k in range(2,26):
    ghs_tp = temp1.sort_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小
    ghs_mz = temp1.sort_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小
    #zys_tp = temp1.sort_values(k-1)[::-1].iloc[:,1]#对应损失率

```

```

tp11 = [] #记录选中供货商值
tp22 = [] #记录选中供货商和
tp33 = [] #记录供货商名字
tp44 = [] #记录转运商名字
#tp55 = 0 #记录检索到哪一位置供应商了
tp66 = [] #记录供货商供货量
sjz_t = sjz.copy()
pm = sh_p[0].values.copy()
for j in pm:
    #try:
    tp1 = [] #记录选中供货商值
    tp2 = 0 #记录选中供货商和
    tp3 = [] #记录供货商名字
    tp4 = [] #记录转运商名字
    #tp5 = 0 #记录检索到哪一位置供应商了
    tp6 = [] #记录供货商供货量
    count = 0
    yz = []
    for i in ghs_mz:
        try:
            yz.append([i,ghs_tp[count]])
        except:
            yz.append([i,0])
        count+=1
    for i in yz:
        tp2 +=i[1]
        if tp2 <= 6000:
            tp3.append(i[0])
            tp6.append(i[1])
        else:
            tp2 -=i[1]
        tp5+=1
        tp5 = j
        ghs_tp = [x for x in ghs_tp if x not in tp6]
        pm = pm[1:]
        ghs_mz = [x for x in ghs_mz if x not in tp3]
    tp11.append(tp1)
    tp22.append([tp2])
    tp33.append(tp3)
    tp44.append(tp4)
    #tp55.append([tp5])
    tp66.append(tp6)
    #except:
    #continue
tp111.append(tp11)
tp222.append([tp22])
tp333.append(tp33)
tp444.append(tp44)

```



```

#tp555.append([tp55])
tp666.append(tp66)

kkk = []
for k in range(24):
    pm = sh_p[0].values.copy()
    t = []
    for j in pm:
        tp = []
        for i in range(8):
            name = tp333[k][i]
            shuju = tp666[k][i]
            tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set_index(0))
        t.append(tp)#3 6 2 8 4 1 7 5
                    #0 1 2 3 4 5 6 7

kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis
=1))
result = pd.concat(kkk,axis=1)
result.to_csv('转运方案 3 分终.csv')
#分步目标值
cs_type = list(gyswz['材料分类'].values)
suihaobiao = []
for i in range(24):
    suihaobiao.append(suiji_sunhao())
ck = suihaobiao[0]
for j in suihaobiao[1:]:
    ck = ck + j
roworder = 0
rowlist = []
for row in fenbu.fillna(0).itertuples():
    ttp = []
    temp = np.array(row[1:])*np.array(ck)+np.array(row[1:])
    roworder+=1
    rowlist.append(temp)
count = 0
tptp = []
for i in cs_type:
    if i == 'A':
        tptp.append(rowlist[count].sum()*1.2)
    elif i == 'B':
        tptp.append(rowlist[count].sum()*1.1)
    else:
        tptp.append(rowlist[count].sum())
    count+=1
#min(np.array(tptp))
sum(tptp)

```

同步式考虑

#第一周随机序列

`gys = list(gyswz.index.values)`

`def first_suiji_3():`

 #31 家分类

`tubian_id = ['S140','S307','S395','S139']`

`yichang_id = ['S151','S308','S330','S348','S201','S143']`

`teshu_id = ['S108']`

`wending_id = []`

`wending_type=[]`

 for i in gys:

 if (i not in yichang_id)and (i not in tubian_id) and (i not in teshu_id):

`wending_id.append(i)`

`wending_type.append(gyswz.loc[i]['材料分类'])`

#异常值取行与去除异常值后加入稳定值

`yic = []`

`yic_type = []`

for j in yichang_id:

`addres = gys.index(j)`

`yic.append(np.array(biao31.loc[addres]))`

`yic_type.append(gyswz.loc[j]['材料分类'])`

for k in yic:

 for m in k:

 if m > 2500:

`addres = list(k).index(m)`

`k[addres] = np.median(k)#异常值使用中位数`

`yic_suiji_xulie = []`

for t in yic:

`yic_max=t.max()`

`yic_min=t.min()`

`yic_suiji_xulie.append(np.random.randint(yic_max*0.65,yic_max, 1))`

#特殊值取行

`addres = gys.index(teshu_id[0])`

`tes = np.array(biao31.loc[addres])`

#稳定值取行

`wen = []`

for j in wending_id:

`addres = gys.index(j)`

`wen.append(np.array(biao31.loc[addres]))`

#稳定值取随机序列

`wen_up = []`

`wen_down = []`

```

for i in wen:
    wen_up.append(i.max())
    wen_down.append(i.max()*0.65)

wen_suiji_xulie = []
for i in wen_up:
    addres = wen_up.index(i)
    wen_suiji_xulie.append(np.random.randint(wen_down[addres],i, 1))

#总的稳定值随机序列
wenzong = wen_suiji_xulie+yic_suiji_xulie
wenzong_type = wending_type+yic_type
for i in yichang_id:
    wending_id.append(i)

#特殊值取随机序列后加上最大值
text = []
tq = []
te_type = gyswz.loc[teshu_id[0]]['材料分类']
for i in tes:
    if i != tes[list(tes).index(tes.max())]:
        text.append(i)
        tq.append(i)
    else:
        text.append('s')
text[text.index('s')] = np.array(tq).mean()
tes_up = np.array(text).max()
tes_down = np.array(text).min()
te_suiji_xulie = np.random.randint(tes_up*0.65,tes_up, 24)
tes_max = list(tes).index(tes.max())
#te_suiji_xulie[random.randint(0,24)] = tes_max
#te_suiji = te_suiji_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周
tb = []
tb_type = []
#for i in tubian_id:
#    addres = gys.index(i)
#    tubian.append(np.array(biao31.loc[addres]))
for j in tubian_id:
    tb_type.append(gyswz.loc[j]['材料分类'])
    tp = np.linspace(0,0,24)
    addres = np.random.randint(0,24, 2)
    if (addres[0] == 0) or (addres[1] == 23):
        tb.append(np.random.randint(5999,6000, 1))
    else:
        tb.append(0)

```

```

        return
wenzong,tb,np.array([6000]),wenzong_type,tb_type,te_type,[wending_id,tubian_id,te
shu_id]

```

#后面几周随机序列生成函数详见支撑材料

#随机转运的转运商选择

```

def sui_ji_zhuanyun_3():
    zy1 = []
    for i in range(0,31):
        zy1.append(int(np.random.randint(1,9,1)))
    return zy1

min_2 = []
dinggou_list3 = []
zhunayun_list3 = []
for mmm in range(8):
    try:
        #第一周
        result1 = []
        mubiao1_list = []
        P1 = []
        for i in range(30):
            sui_ji_list = first_sui_ji_3()
            count +=0
            t2 = []
            t2.append(int(sui_ji_list[2]))
            sui_ji_data = sui_ji_list[0]+sui_ji_list[1]+t2
            sui_ji_type = sui_ji_list[3]+sui_ji_list[4]
            sui_ji_type.append(sui_ji_list[5])
            gh = pd.DataFrame([sui_ji_type,sui_ji_data]).T
            A = gh[gh[0] == 'A'][1].values.sum()
            B = gh[gh[0] == 'B'][1].values.sum()
            C = gh[gh[0] == 'C'][1].values.sum()
            if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200*2)):
                P1.append(A/0.6+B/0.66+C/0.72)
                mubiao1 = 1.2*A+1.1*B+C
                mubiao1_list.append(mubiao1)
                result1.append(sui_ji_list)
            else:
                continue
        addres = mubiao1_list.index(min(mubiao1_list)[0])
        fliter = []
        for i in result1[addres][0]:
            fliter.append(int(i))
        for i in result1[addres][1]:
            fliter.append(i)
        fliter.append(int(result1[addres][2]))
        fliter.append(int(P1[addres]))
        fliter.append(int(min(mubiao1_list)))

```

```

fliter1 = fliter.copy()

#后面几周
#筛选对应材料的供货商
p = int(P1[addres])
fliter_list = []
P_list = [int(P1[addres])]
CT = 0
for i in range(2,25):
    result2 = []
    mubiao2_list = []
    Pn = []
    p=P_list[CT]
    for i in range(150):
        suiiji_list = suiiji_3()
        count +=0
        t2 = []
        t2.append(int(suiiji_list[2]))
        suiiji_data = suiiji_list[0]+suiiji_list[1]+t2
        suiiji_type = suiiji_list[3]+suiiji_list[4]
        suiiji_type.append(suiiji_list[5])
        gh = pd.DataFrame([suiiji_type,suiiji_data]).T
        A = gh[gh[0] == 'A'][1].values.sum()
        B = gh[gh[0] == 'B'][1].values.sum()
        C = gh[gh[0] == 'C'][1].values.sum()
        if ((A+B+C) <= 48000) and
        ((A/0.6+B/0.66+C/0.72)>=(28200*2)-p):
            Pn.append(A/0.6+B/0.66+C/0.72)
            mubiao2 = 1.2*A+1.1*B+C
            mubiao2_list.append(mubiao1)
            result2.append(suiiji_list)
        else:
            continue
    addres = mubiao2_list.index(min(mubiao2_list)[0])
    fliter = []
    for i in result2[addres][0]:
        fliter.append(i[0])
    for i in result2[addres][1]:
        tmp = []
        try:
            for j in i:
                if j != 0:
                    tmp.append(j[0])
                else:
                    tmp.append(j)
            for k in tmp:
                fliter.append(k)
        except:
            fliter.append(i)
    fliter.append(int(result2[addres][2]))

```

```

        fliter.append(int(Pn[adres]))
        fliter.append(int(min(mubiao2_list)))
        fliter_list.append(fliter)
        P_list.append(int(Pn[adres]))
        CT +=1

#整体订购方案
t = [fliter1]
for i in fliter_list:
    t.append(i)
ttp = []
for i in suiiji_list[-1]:
    for j in i:
        ttp.append(j)
ttp.append('原料对应产能')
ttp.append('原材料费用')

dinggoufangan=pd.DataFrame(t,columns=ttp,index=list(np.linspace(1,24,24))).T
t = []
for i in dinggoufangan.values:
    tttt = []
    for j in i:
        tttt.append(int(j))
    t.append(np.array(tttt))
dinggoufangan3 =
pd.DataFrame(np.array(t),index=dinggoufangan.index.values,columns=dinggoufangan.columns.values)
dinggou_list3.append(dinggoufangan3)

#转运方案
t31 = dinggoufangan3[0:31]
result2 = []
for k in range(24):
    index = list(dinggoufangan3[0:31].index.values)
    ttp = list(t31.iloc[:,k].values)
    ss_ghs = pd.DataFrame([index,ttp]).T.sort_values(by=1).iloc[::-1]
    ghsmv = ss_ghs[0].values #供货商名单
    zysmdv = sh_p[0].values #转运商名字
    ghsmv= ss_ghs[1].values #供货商从大到小排的供货量
    ghsl = []
    for i in zysmdv:
        tp = 0
        ghsl = []
        ct = 0
        ghsmv_tp = []
        for j in ghsmv:
            tp += j
            if tp <= 6000:

```

```

        ghsl.append(ghsmd[ct])
        ghsmdv_tp.append(j)
        ct += 1
    ghsl.append(ghsl)
    ghsmd = [x for x in ghsmd if x not in ghsl]
    zysmdv = zysmdv[1:]
    ghsmdv = [x for x in ghsmdv if x not in ghsmdv_tp]
    count = 0
    ys = pd.DataFrame(index =
ss_ghs[0].values,columns=[1,2,3,4,5,6,7,8])
    count2 = 0
    for i in ghsl:
        for j in i:
            ys.loc[j][(sh_p[0].values)[count]] = ss_ghs[1].values[count2]
            count2 += 1
        count += 1
    result2.append(ys)
    chushi = result2[0]
    for i in result2[1:]:
        chushi = pd.concat([chushi,i],axis=1)
    zhunayun_list3.append(chushi)
    #chushi

#一次循环目标值
cs_type = list(gyswz['材料分类'].values)
suihaobiao = []
for i in range(24):
    suihaobiao.append(suiji_sunhao())
ck = suihaobiao[0]
for j in suihaobiao[1:]:
    ck = ck + j
roworder = 0
rowlist = []
for row in chushi.fillna(0).itertuples():
    ttp = []
    temp = np.array(row[1:])*np.array(ck)+np.array(row[1:])
    roworder+=1
    rowlist.append(temp)
count = 0
tptp = []
for i in cs_type:
    if i == 'A':
        tptp.append(rowlist[count].sum()*1.2)
    elif i == 'B':
        tptp.append(rowlist[count].sum()*1.1)
    else:
        tptp.append(rowlist[count].sum())
    count+=1
min_2.append(sum(tptp))
except:

```

```

pass

address = min_2.index(min(min_2))

dinggou_list3[address].to_csv('wenti3_ghs.csv')

zhunayun_list3[address].to_csv('wenti3_zys.csv')

```

#第四题关键代码

```

#筛选突变进行平稳化处理
x4 = np.linspace(0,402,402)
fc4 = origin4[243].values
adr = []
count = 0
for i in fc4:
    if i > 1300:#取出突变型
        adr.append(count)
        count +=1

#max ---> rand(0.66m~0.33m)/(m/6000)
ttp = []
for i in adr:
    ttp.append(origin4.iloc[i])
cn = pd.DataFrame(ttp).iloc[:,2:242]
roworder = 0
ttp = []
max_l4 = []
for row in np.array(cn):
    max_l4.append(row.max())
cout = 0
tp = []
for row in np.array(cn):
    ct = 0
    for i in row:
        if i > 6000:
            row[ct] =
random.uniform(0.6*max_l4[cout],max_l4[cout])/(max_l4[cout]/6000)
            ct +=1
        cout +=1
    tp.append(row)
values_t = origin4.iloc[:,2:242]
#values_t

#处理后的数据表

```



```

count = 0
tpv = []
for i in np.array(values_t):
    ct = 0
    if count in cn.index.values:
        tpv.append(tp[ct])
        ct +=1
    else:
        tpv.append(i)
    count+=1
temp =
pd.DataFrame(tpv,index=values_t.index.values,columns=values_t.columns.values)
#temp

ghs4 = origin4.copy()
#处理类型成本
Dtype402 = []
for i in np.array(ghs4):
    Dtype402.append(i[1])
Ddata3 = [0.6,0.66,0.72]
D402 = []
for i in Dtype402:
    if i == 'A':
        D402.append(Ddata3[0])
    elif i == 'B':
        D402.append(Ddata3[1])
    elif i == 'C':
        D402.append(Ddata3[2])
    else:
        continue

#取每行平均值
t4 = ghs4.set_index(0).drop([1],axis=1)
you0 = list((t4 == 0).astype(int).sum(axis=1))
sum_t4 = list(t4.sum(axis=1).values)
fei0 = [240 - i for i in you0]
mean_t4 = np.array(sum_t4)/np.array(fei0)
index = list(t1.index.values)
pj_values = list(mean_t4*1.14)
ghs4.iloc[:,242] = pj_values
ghs4 = ghs4.sort_values(by=242)[::-1]
#ghs4 = pd.concat([ghs4[ghs4[1] == 'A'],ghs4[ghs4[1] == 'B'],ghs4[ghs4[1] == 'C']])
#ghs4

sum_cn = 0
ct = 0
count = 0
pjcn = []
ljcn = []
for i in pj_values:

```

```

sum_cn = sum_cn + i
lfcn.append(sum_cn)
pjcn.append(i)
if sum_cn < 48000:
    count+=1
    ct+=1
    continue
else:
    break
gys = t4.index.values[0:len(pjcn)]
cpt = ghs4[1].values[0:len(gys)]
gyslim = pd.DataFrame([gys,cpt,pjcn,lfcn],index=['供应商 ID','材料分类','供应商平均供应能力','累计供应能力']).T.set_index('供应商 ID')
#gyslim = pd.concat([gyslim[gyslim['材料分类'] == 'A'],gyslim[gyslim['材料分类'] == 'B'],gyslim[gyslim['材料分类'] == 'C']])
gyslim.to_csv('重要参数 4.csv')

def onew():
    while True:
        one_w = []
        for i in pjcn:
            a = random.uniform((i*0.8),(i*1.3))
            one_w.append(a)
        if sum(one_w) > 48000:
            return one_w

#概率表
tq = []
for i in range(24):
    for j in sui_ji_sunhao():
        tq.append(j)

kb = []
#订购方案
tp = []
for i in range(24):
    tp.append(onew())
t286 = pd.DataFrame(tp,columns=gyslim.index.values).T
t286.to_csv('订购 4.csv')

temp7 = pd.read_csv('订购 4.csv')
data3 = pd.read_excel('附件 2 近 5 年 8 家转运商的相关数据.xlsx')

temp6 = data3.set_index('转运商 ID')
tpq = []
for i in np.array(temp6):
    ttp = []
    for j in i:
        if j != 0:

```

```

        ttp.append(j*0.01)
    else:
        continue
    tpq.append(ttp)

ct = 1

for i in tpq:
    x = np.linspace(0,len(i),len(i))
    y = list(i)
    fig = plt.figure(figsize=(8,11),dpi=98)
    na=plt.subplot(4,1,1)
    na.plot(x,y,label='转运商 {}'.format(ct))
    na.legend()
    ct+=1
    plt.show()
#fig.savefig('zhuanyun.jpg')

#计算转运商信息
tp = []
for i in np.linspace(1,8,8):
    tp.append(int(i))
tpq_mean = []
for i in tpq:
    tpq_mean.append(np.array(i).mean())
plt.figure(dpi=95)
plt.ylabel('平均转运损耗')
plt.xlabel('转运商')
plt.plot(tp,tpq_mean)
plt.savefig('zhunayunpj.jpg')
plt.show()
sh_p = pd.DataFrame([tp,tpq_mean]).T.sort_values(by=1)
#随机转运的转运商选择结果 24
def suiiji_zhuanyun(count):
    zy1 = []
    for i in range(0,count):
        zy1.append(int(np.random.randint(1,9,1)))
    return zy1

#随机损耗 1~8
def suiiji_sunhao():
    ss = []
    tpq_suiji = []
    for i in tpq:
        tpq_suiji.append(random.sample(list(np.array(i)),1)[0])
    return tpq_suiji

#对应的概率
sjz = suiiji_zhuanyun(23)

```

```

sh = sui_ji_sunhao()
temp = []
for i in sjz:
    temp.append(sh[i-1])
c=0
q = []
for i in temp7.values[0:-2]:#.sort_values(by=k-1)[::-1][k-1]
    t = list(i)
    t.append(c)
    q.append(t)
    c+=1
temp1 = pd.DataFrame(q)
pm = sh_p[0].values.copy()
tp111 = [] #记录选中供货商值
tp222 = [] #记录选中供货商和
tp333 = [] #记录供货商名字
tp444 = [] #记录转运商名字
#tp555 = 0 #记录检索到哪一位置供应商了
tp666 = [] #记录供货商供货量
for k in range(2,26):
    ghs_tp = temp1.sort_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小
    ghs_mz = temp1.sort_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小
    #zys_tp = temp1.sort_values(k-1)[::-1].iloc[:,1]#对应损失率
    tp11 = [] #记录选中供货商值
    tp22 = [] #记录选中供货商和
    tp33 = [] #记录供货商名字
    tp44 = [] #记录转运商名字
    #tp55 = 0 #记录检索到哪一位置供应商了
    tp66 = [] #记录供货商供货量
    sjz_t = sjz.copy()
    pm = sh_p[0].values.copy()
    for j in pm:
        #try:
        tp1 = [] #记录选中供货商值
        tp2 = 0 #记录选中供货商和
        tp3 = [] #记录供货商名字
        tp4 = [] #记录转运商名字
        #tp5 = 0 #记录检索到哪一位置供应商了
        tp6 = [] #记录供货商供货量
        count = 0
        yz = []
        for i in ghs_mz:
            try:
                yz.append([i,ghs_tp[count]])
            except:
                yz.append([i,0])
            count+=1

```

```

for i in yz:
    tp2 +=i[1]
    if tp2 <= 6000:
        tp3.append(i[0])
        tp6.append(i[1])
    else:
        tp2 -=i[1]
        tp5+=1
        tp5 = j
        ghs_tp = [x for x in ghs_tp if x not in tp6]
        pm = pm[1:]
        ghs_mz = [x for x in ghs_mz if x not in tp3]
    tp11.append(tp1)
    tp22.append([tp2])
    tp33.append(tp3)
    tp44.append(tp4)
    #tp55.append([tp5])
    tp66.append(tp6)
    #except:
        #continue
    tp111.append(tp11)
    tp222.append([tp22])
    tp333.append(tp33)
    tp444.append(tp44)
    #tp555.append([tp55])
    tp666.append(tp66)

kkk = []
for k in range(24):
    pm = sh_p[0].values.copy()
    t = []
    for j in pm:
        tp = []
        for i in range(8):
            name = tp333[k][i]
            shuju = tp666[k][i]
            tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set_index(0))
        t.append(tp)#3 6 2 8 4 1 7 5
            #0 1 2 3 4 5 6 7

kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis
=1))
result = pd.concat(kkk,axis=1)
result.to_csv('转运方案 4 终.csv')

#平均产能
data1 = pd.read_excel(r'附件 1 近 5 年 402 家供应商的相关数据.xlsx',sheet_name='
企业的订货量 (m³) ')
type_z = data1['材料分类'].values

```

```

z4 = pd.read_csv('订购 4 终.csv')

z4 = z4.drop(['Unnamed: 0'],axis=1)
z4 = z4.set_index('0')
#z4 = z4.T

Dtype = type_z
Ddata = [1/0.6,1/0.66,1/0.72]
D = []
for i in Dtype:
    if i == 'A':
        D.append(Ddata[0])
    elif i == 'B':
        D.append(Ddata[1])
    elif i == 'C':
        D.append(Ddata[2])
    else:
        continue
t = []
for i in range(24):
    t.append(D)
#temp = /24
z4 = pd.DataFrame(np.array(z4)*np.array(t).T)
res = z4.apply(lambda x: x.sum(),axis=0).T
res.to_csv('重要数据 4.csv')
print('第四问平均产能: {}'.format(res.sum()/24))

```