

2021 年高教社杯全国大学生数学建模竞赛题目

生产企业原材料的订购与运输

摘 要

本文针对生产企业原材料的订购和转运方案的特性，考虑供求关系、配比关系、经济性以及企业自身协议完成度等因素，通过对四个问题的具体分析，适当地合理假设得到每个问题的解决方案。

问题一对数据进行分析，建立了 4 个能有效反映供货商供货能力和供货风险的指标。这 4 个指标分别为供货次数，平均供货量，供货达标率和总供货量这四个指标。通过 TOPSIS 熵权法，结合 4 个指标，建立多指标评价模型对每个供货商的重要性程度进行评价。基于模型的评价结果，本文筛选出了 50 家最重要的供货商。

问题二依据企业总产能上限筛选最优订购转运方案对 50 家企业近 5 年的总产能进行排序，在总产能误差为 0.3% 的限定条件下筛选，最终选择 37 家供应商。将附件 1 中 240 周 402 企业的订货量与供货量的差取绝对值，依次对上述连续两周所求的绝对值求均值后从小到大排序，选取位于前 12 位的连续两周，即选定 24 周，并得到订购方案见附件 A。通过附件 2 中数据分析各转运商的损耗率，并将损耗率为 0 的转运商去除，将转运数据规整化、标准化、有效化。将转运商的平均损耗率由小到大进行排序，计算出 24 周每一周转运商数量。

问题三依据对不同原材料的倾向要求限筛选最优订购转运方案对生产 A、B、C 三种原料的供应商进行分组，在每一组内对供货商按照每周平均供货量从 A 到 C 近五年每周的产能由大到小排序。根据多 A 少 C 的原则，以企业最大产能为标准，筛选出 32 家供应商企业。利用问题二中同样的方法制定出新的订购方案及转运方案见附件 A 和 B。

问题四依据供应商的材料供应量选定企业扩大的最大产能方案通过计算 402 家企业年平均产能，选出最大的年份的平均产能作为提高产能计算的依据，计算出每周的产能最大值为 40525.5 立方米，增长率为 43.6%。

关键词：TOPSIS 熵权法，0-1 规划，转运方案，实施方案

一、问题重述

某生产企业每年按 48 周安排生产，需提前制定 24 周的原材料订货和转运计划。计划的制定需要根据产能要求确定每周的原材料供货商和相应每周的原材料订货数量，并确定第三方物流公司，委托其将原材料转运至企业仓库。该生产企业所用的原材料分为 A、B 和 C 三种类型，其采购单价不同，均为独立用于生产产品，生产一立方米产品所需消耗的量不同，三种原材料的运输和储存的单位费用则是相同的。该企业每周产能为 2.82 万立方米，且要求尽可能保持不少于满足两周生产需求的原材料库存量。供货商实际供货量可能多于或少于订货量，企业对供货商实际供货的原材料总是全部收购。

每家转运商的运输能力为 6000 立方米/周。在原材料的实际转运过程中会有一些的损耗，不同的转运商产生的损耗程度不同。一家供货商每周的供货的原材料应尽量由一家转运商运输。

附件 1 给出了该企业近 5 年 402 家原材料供货商每周的订货量和供货量数据；附件 2 给出了 8 家转运商近 5 年每周的运输损耗率数据。现完成如下问题：

1. 量化分析附件 1 中 402 家供货商的供货特征，建立反映保障企业生产重要性的数学模型，从而确定 50 家最重要的供货商。

2. 参照问题一所得的供货商重要性，制定未来 24 周每周最经济的原材料订购方案及损耗最少的转运方案，并对方案的实施效果进行分析。

3. 考虑每周所订货的三种原材料的组成结构，压缩生产成本，制定新的订购方案，对新方案的实施效果进行分析，并与原方案对比。

4 企业具备提高产能的潜力，根据现有供货商和转运商的实际情况，分析其规律，确定该企业每周产能可提高多少，制定新的订购方案，对新方案的实施效果进行分析。

二、问题分析

问题一，要求基于供货商的历史订货与供货数据进行量化分析，探寻供货商的供货特征，并建立反映保障企业生产重要性的数学模型，筛选出 50 家最重要的供货商。首先，对附件 1 的数据进行预处理，确保预处理后的数据不存在错误数据。接着针对预处理后的数据，进行探索性数据分析，观察 402 家供货商在供货能力，供货稳定性上是否存在明显的差异化特征。基于数据分析的结果，构建能够有效反映供货商供货能力与供货稳定性的指标，对指标进行分析和筛选。结合筛选过后的指标，建立多指标评价模型，采用熵权法对 402 家供货商进行重要性程度评价，基于评价结果，选择最重要的 50 家供货商。

问题二，在问题一已经筛选出 50 家重要企业的的基础上采用 0-1 整数规划进一步的分析。计算这所有企业的五年内的供货总量排序，从其中由高到低进行排序，排名越高即供应商提供材料的能力越强。现已知如果需要满足企业产能，需要所有家供

应商 240 周的供货量总共为 240×28200 立方米，设定相应固定的约束条件，则筛选过程中需要剔除的供应商应该尽可能对剩余供应商所能提供的总供货量的影响较小。则将从权重最小的开始进行去除，如果对所提供总产能变化在约束条件之内，则继续上述步骤，如果有较大变化则停止筛选剔除，剩余的即为所求的至少需要选择能够尽可能满足企业产能的供应商 37 家。

关于未来 24 周最经济的订购方案的选择，在上一步进行最少供应商的选择之后，需要对每一家供应商的信用以及供求关系进行判断。对每一家供应商每一周供货量与订货量的差的取绝对值，再对连续两周的绝对值取平均值进行从小到大的排序。其中排除特殊情况，需要剔除掉有所选出的两周之间有重合项的供应商。以每一家供应商为基准选出 12 个排名较为靠前的绝对值之差，即 24 周的选择，该供应商的 24 周最经济的订购方案。

关于制定损耗最少的转运方案，首先要对转运商的数据进行预处理，因为所有的转运商在运输的过程中都会存在一定的损耗率，但是通过附件 2 中给出的有关转运率的信息可知，某一些转运商在该周的转运损耗率为 0，因此可以推断该转运商在该周并没有转运计划，在制定转运方案的时候应事先避免安排这些转运商。

问题三，为了压缩企业的生产成本，尽可能多的选择 A 材料作为生产产品的原材料，即选择供应 A 类产品的供应商。利用问题一中筛选出的重要合作供应商，将生产 A、B、C 的厂家进行聚类分析并按照 A、B、C 三个组类的顺序排列，在计算每个厂家 240 周的总供应量，计算各自供应商每周供应量的均值，并在分别三个组类中按供应量均值从大到小排序。在数据处理完成之后，进行供应商的选择，筛选的指标主要分为两个。其一是秉持着多 A 少 C 的原则挑选；其二是最后选定的总量，即每次从 A 到 B 到 C、从供应大到供应量小的进行筛选的时候应该尽可能接近该企业每周的极限产能，直至筛选至某一家时刚刚超过企业每周总产能即将其退回。依据数学模型公式为其设置新的订购方式和转运方式所用的方法基本上与第二问一致。

问题四，为了计算企业每周产能可以提高多少，则需要通过现有的供应商供货量的数据进行求解。经过数据分析可知供应商提供的材料在某些周数的时候可以超过原有的企业最大产能，而有一些周数的时候可以符合现有企业最大产能。企业每周产能的上限提高则可以接受更多供应商所供应的原材料。将 A、B、C 分类求和后进行以年为单位求平均，分析出每年每类供货商的产能均值，计算每年所有提供原材料供应商的产能总量，最终得到 5 年每年对应的供应量并进行排序挑出其最大值

三、模型假设

(1) 假设一家供应商每周供应的原材料尽量由一家转运商运输，但可以允许一家供应商由数量较少的转运商转运。

(2) 假设企业对供应商实际提供的原材料总是全部收购。

(3) 假设在前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，并且原材料可以在下一周一开始就进入仓库储存起来。

(4) 假设该生产企业在一周内的产能均可以完成且不会超额完成。

(5) 假设运输费用只取决于所装材料的体积，与运输路程的长短无关。

(6) 假设每周的供货总量在该周内由转运商全部转运完成，不会出现遗留到下一周运输的情况。

四、符号说明

符号	说明	符号	说明
W	该企业每周接受量所能获得产能	W_0	该企业每周至少应接受原材料量
Y_j	B 类供应商 $j \in (1, 2, 3, \dots, 134)$	Z_k	C 类供应商 $i \in (1, 2, 3, \dots, 122)$
x_i	A 类供应商对应供货量	y_j	B 类供应商对应供货量
z_k	C 类供应商对应供货量	a	单位立方米消耗 A 类原材料量
b	单位立方米消耗 B 类原材料量	c	单位立方米消耗 C 类原材料量
m	周次, 取值范围为 $[1, 240]$ 且为整数	V_i	供应商选取情况 (只取 0 或 1)
I	每周排名序列	N	供应商序列
Δd	每家供应商每周供应量与订购量的差值的绝对值	T_A	转运商, 其中 $1 \ll \alpha \ll 8$ 且 $\alpha \in \mathbb{Z}$
S_{am}	每周每家转运商运输损耗率	A_p	近五年供应商数据分组后 A 类供应商
B_p	近五年供应商数据分组后 B 类供应商	C_p	近五年供应商数据分组后 C 类供应商
μ	每周转运效率比值	W_{0p}	近五年数据分组后各组数据对应的产能
X_i	A 类供应商其中 $i \in (1, 2, 3, \dots, 146)$		

五、模型建立与求解

5.1.1 问题一模型的建立

熵权法是一种客观赋值方法。在具体使用的过程中，熵权法根据各指标的变异程度，利用信息熵计算出各指标的熵权^[1]，再通过熵权对各指标的权重进行修正，从而得到较为客观的指标权重。

一般来说，若某个指标的信息熵指标权重确定方法之熵权法越小，表明指标值得变异程度越大，提供的信息量越多，在综合评价中所能起到的作用也越大，其权重也就越大。

相反，若某个指标的信息熵指标权重确定方法之熵权法越大，表明指标值得变异程度越小，提供的信息量也越少，在综合评价中所起到的作用也越小，其权重也就越小。

不同的指标代表含义不一样，有的指标越大越好，称为越大越优型指标。有的指标越小越好，称为越小越优型指标，而有些指标在某个点是最好的，称为某点最优型指标。为方便评价，应把所有指标转化成越大越优型指标。

设有 m 个待评对象， n 个评价指标，可以构成数据矩阵：

$$X = (X_{ij})_{m \times n}$$

设数据矩阵内元素，经过指标正向化处理过后的元素为 $(x'_{ij})'$

- 越小越优型指标：C, D 属于此类指标

$$x'_{ij} = \max(x_{ij}) - x_{ij}$$

其他处理方法也可，只要指标性质不变即可

- 某点最优型指标：E 属于此类指标

设最优点为 a ，当 $a=90$ 时 E 最优。

$$x'_{ij} = 1 - \frac{|x_{ij} - a|}{\max(|x_{ij} - a|)}$$

5.1.2 模型的求解

因为每个指标的数量级不一样，需要把它们化到同一个范围内再比较。标准化的方法比较多，这里仅用最大最小值标准化方法。设标准化后的数据矩阵元素为 r_{ij} ，

由上可得指标正向化后数据矩阵元素为 $(x'_{ij})'$

$$r_{ij} = \frac{x'_{ij} - \min(x'_j)}{\max(x'_j) - \min(x'_j)}$$

数据处理后可以构成数据矩阵 $R = (r_{ij})_{m \times n}$ ，对某个指标 r_j ，信息熵为 E_j

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^m p_{ij} \ln p_{ij}$$

其中

$$p_{ij} = \frac{r_{ij}}{\sum_{j=1}^n r_{ij}}$$

为避免 p_{ij} 零元素的出现出现计算错误, 归一化最低区间可以从 0.002 开始。如果某个指标的信息熵 E_j 越小, 就表明其指标值的变异程度越大, 提供的信息量也越大, 可以认为该指标在综合评价起到作用也越大。

根据以上原理, 分析供货商的订货与供货数据, 构建能够反映供货商供货特征的指标。本文构建供货次数, 平均供货量和总供货量和供货达标率用于反映供货商供货能力; 以这 4 个指标, 建立多指标评价模型, 以模型的输出结果作为供货商重要性的体现。

a. 供货次数

供货次数间接反映供货商的接到订货的次数, 生产企业在过去的 5 年向该供货商的订货次数, 能够直接反映生产企业对该供货商的依赖程度。由假设 2 可知, 生产企业对该供货商的依赖程度能够有效反映生产企业对该供货商供货能力的认可度, 即供货次数越多的供货商, 具有更强的供货能力。

b. 平均供货量

平均供货量在一定程度上反映该供货商的供货能力。同时结合供货总量也可以得到各供货商对于企业的供货次数。可以间接反映出该供货商与企业的业务联系, 进而反映出该生产企业与该供货商是否存在长久合作关系。若供货商的供货次数越多, 则说明供货商与企业和合作关系越紧密, 越可能成为高质量供货商。

c. 总供货量

由于该企业对供货商实际提供的原材料总是全部收购, 供货商为实现利益最大化, 一定会保持最大生产效率。因此最大供货量直接反映了供货商供货能力的上限, 同样是衡量供货商供货能力的一个重要指标。

d. 供货达标率

对于企业而言, 供货量过多会导致成本增加, 供货量过少影响产能释放, 因此供货量需要与订单量保持较高的一致性。即供货商的供货量需要保持在企业给予其订单量的一定范围内波动。波动值越小则说明供货商的供货能力越稳定, 越受企业的重视。

用权重公式 $w_{ij} = \frac{(1-E_j)}{\sum_{j=1}^n (1-E_j)}$ 以及得分公式 $S_i = \sum_{j=1}^n w_j r_{ij}$ 算出最后结果:

表 1 各指标所占权重

指标	供货次数	平均供货量	供货达标率	总供货量
权重	0.065	0.45	0.45	0.035

根据得分进行排名，筛选前 50 个供应商：

供应商 ID	分数	供应商 ID	分数	供应商 ID	分数
S229	100	S348	26.0365	S055	6.7669
S361	92.4457	S352	25.0812	S346	6.5412
S140	85.1096	S143	23.3217	S080	5.4132
S108	67.8923	S201	23.0968	S294	5.3018
S151	54.802	S307	22.0279	S086	5.0502
S340	48.3003	S395	21.3648	S244	4.6154
S282	47.7125	S247	15.9697	S210	4.4147
S275	44.6727	S037	14.2755	S218	4.3553
S329	44.0992	S374	13.8635	S003	3.6944
S139	42.7871	S126	13.389	S074	3.6699
S131	38.7433	S284	13.1232	S114	3.0725
S308	38.5984	S365	11.7238	S273	2.6647
S330	38.5009	S031	11.6043	S292	2.5734
S356	36.7129	S040	8.983	S189	2.4979
S268	36.5661	S338	8.4769	S208	2.4599
S306	35.5262	S364	8.0976	S078	2.4024
S194	28.557	S367	7.4134		

5.2.1 针对供应商的选取模型

针对问题二，先在 402 家供应商中选取满足题目要求的供应商，再在近五年的数据进行对未来 24 周订购方案的模拟。问题二可以描述为：有 n 家供应商和 1 个企业，对各类产品 A、B、C，其对应单价分别为 a 、 b 、 c ，每家单周供货量为 w_i （单位：立方米），企业一周所能接收的总供货量 W （单位：立方米）。先需选取供应商为企业供应原材料，使企业尽可能满足一周的需求。对于各个供应商，只有两个选择，即选择其为该企业供应原材料或者不选择，不可多次选择，也不能选择一部分进行运输供应。

采用 0-1 整数规划，假设 v_i 、 v_j 、 v_k 分别表示供应 A、B、C 三类原材料的供应商 n 选取情况，当 $v_i = 0$ 或 $v_j = 0$ 或 $v_k = 0$ 时，表示没有选择该供货商为企业供应原材料；当 $v_i = 1$ 时，表示选择该供货商为企业供应原材料。

产能是指在计划期内，企业参与生产的全部固定资产，在既定的组织技术条件下，

所能生产的产品数量，或者能够处理的原材料数量。根据近五年 ABC 类材料的总供货量，可以计算出该企业五年内的总产能，通过实际总产能和理想总产能计算得出总产的误差约为 0.3%，按排名累加 240 周的供应量如果累加后和累加前的差值小于总产能误差为 0.3%，我们则认为该家供应商是必须的供应商，据此选取供应商。

根据问题描述，设计出如下约束条件和目标函数。

约束条件：

$$\left\{ \begin{array}{l} W - \left(\frac{\sum x_i v_i}{a} + \frac{\sum y_j v_j}{b} + \frac{\sum z_k v_k}{c} \right) \leq \eta \\ v_i, v_j, v_k \in \{0,1\} \end{array} \right\}$$

$$\text{目标函数：} \max(\sum w_i v_i + w_j v_j + w_k v_k)$$

其中， $1 \leq i \leq n$ ， η 为约束参数， v_i 表示供应商选取情况（只取 0 或 1）每家单周供货量为 w_i ，问题二归结为寻找一个满足约束条件并且是目标函数达到最大的解向量

$V = (v_1, v_2, \dots, v_n)$ 。 W 为接收的总供货量。

5.2.2 针对供应商供应方案的选取

先计算出每家供应商每周对该企业供应量与该企业每家供应商每周的订购量的差值的绝对值，记作 $\Delta d_{nm} = |\text{供货量} - \text{订货款}|$

取连续 2 周的差值的平均值，即 $\Delta \bar{d}_{nm} = \frac{1}{2}(\Delta d_{nm} + \Delta d_{n(m+1)})$ ，作为选取订购方案的参照标准，差值越小说明该供货商信用良好，依次对上述连续两周所求的绝对值求均值后从小到大排序，选取位于前 12 位的连续两周，即选定 24 周在已知订购和供货历史数据中预测未来该企业的订购方案。

令 T_a 为转运商，每周每家转运商的运输损耗率为 S_{am} ，其中 $1 \leq a \leq 8$ ， $1 \leq m \leq 240$ 。

计算每家转运商的平均损耗率：

$$\bar{S}_a = \frac{\sum S_{am}}{\sum m}$$

根据所得的平均损耗率 \bar{S}_a ，将次作为参考标准，尽可能多的选择 \bar{S}_a 较小的转运商参与未来 24 周每周的转运方案中。

5.2.3 模型的求解

参考问题一中选取的 50 家最重要的供应商，利用问题一中的排序（按照每周各供应供货量排序，取平均名次），计算出每家供应商 240 周为该企业所供应的原材料

总和 W 。若该企业每周均有 A 类原材料产能，根据每立方米产品需消耗 A 类原材料 0.6 立方米，得：

$$w_0 = 28200 \times 0.6 = 16920 \text{（单位：立方米）}$$

设定约束参数 $\eta = 28200 \times 240 \times 0.3\%$ ，现为了使企业尽可能满足生产需求，由于每立方米产品只需消耗 A 类原材料 0.6 立方米，比 B、C 类消耗的都要少，因此企业每周所需要达到的接收量至少为 w_0 。根据约定参数，在 50 家供应商中一一排除，计算出排除各家供应商后，剩余供应商所能提供的供货量与 W 作差，是否小于约束条件。

现规定如下^[2]：

（1）若所得差值小于约束条件，则被视为排除该家供应商后，对最后企业产能无影响，即该家供应商每周对企业的供应量不会影响企业产能；

（2）若差值大于约束条件，则被视为重要供应商，作为最后的选择，将其对企业的供应量计入最后的供应量中。以此为条件约束，至少选取 37 家供应商才能满足该企业原材料的供应去需求，选取结果如下表 2 所示：

表 2 选取的 37 家供应商

供应商	供应商	供应商
S229	S330	S037
S361	S356	S374
S140	S268	S126
S108	S306	S284
S151	S194	S365
S340	S348	S031
S282	S352	S040
S275	S143	S338
S329	S201	S364
S139	S307	S367
S131	S395	S055
S308	S247	S346
		S080

5.2.4 关于订购方案的设计

针对上述方式所选取的 37 家供应商，调取各家供应商近五年的订购量和供货量，通过计算得到每家供应商连续 2 周相应的 $\Delta \bar{d}_{nm}$ ，为保证该企业所定制的未来 24 周订购方案原材料最经济，需对所得数值由小到大进行排序。经计算排序后得出相应原材料订购方案，结果在附件 A 中给出。

5.2.5 关于转运方案的设计

运用预处理完后的转运商运输损耗率数据，参考问题一中对各供应商的排序方式，对附件 2 中所给定的各家转运商进行排序，结果如下表所示：

表 38 家转运商的平均损耗率商

次序	转运商 ID	平均损耗率
1	T3	0.186
2	T6	0.544
3	T2	0.921
4	T8	1.010
5	T4	1.570
6	T1	1.905
7	T7	2.079
8	T5	2.890

根据上述制定的订购方案，结合题目中已知条件：每家转运商每周可转运 6000 立方米原材料。计算出未来 24 周每周所需要的转运商家数，相应结果在下表给出：

表 4 未来 24 周所需的转运商家数

周次	转运商家数	周次	转运商家数	周次	转运商家数
第 01 周	3	第 09 周	2	第 17 周	2
第 02 周	4	第 10 周	3	第 18 周	2
第 03 周	3	第 11 周	3	第 19 周	3
第 04 周	4	第 12 周	3	第 20 周	5
第 05 周	4	第 13 周	2	第 21 周	3
第 06 周	3	第 14 周	3	第 22 周	4
第 07 周	3	第 15 周	3	第 23 周	5
第 08 周	4	第 16 周	6	第 24 周	6

再根据各家转运商的排序结果，按照顺序选择相应家数的转运商对原材料制定损耗率最少的转运方案^[3]。转运方案如下表所示：

表 5 针对于不同转运商每周的安排

转运商 ID	周次	转运商 ID	周次	转运商 ID	周次
T3	第 09 周	T3	第 01 周	T3	第 02 周
T6	第 13 周	T6	第 03 周	T6	第 04 周
	第 17 周	T2	第 06 周	T2	第 05 周
	第 18 周		第 07 周	T8	第 08 周
			第 10 周		第 22 周
			第 11 周		
			第 12 周		
			第 14 周		
			第 15 周		
			第 19 周		
			第 21 周		
转运商 ID	周次	转运商 ID	周次		
T3	第 20 周	T3	第 16 周		
T6	第 23 周	T6	第 24 周		

T2		T2			
T8		T8			
T4		T4			
		T1			

5.3.1 模型的建立

针对问题三，为了压缩企业的生产成本，尽可能多的选择 A 材料作为生产产品的原材料，即选择供应 A 类产品的供应商。利用问题一中筛选出的 50 家供应商，将生产 A、B、C 的厂家进行聚类分析并按照 A、B、C 三个组类的顺序排列，在计算每个厂家 240 周的总供应量，计算各自供应商每周供应量的均值，并在分别三个组类中按供应量均值从大到小排序。在数据处理完成之后，进行供应商的选择，筛选的指标主要分为两个。其一是秉持着多 A 少 C 的原则挑选；

其二是最后选定的总量，即每次从 A 到 B 到 C、从供应大到供应量小的进行筛选的时候应该尽可能接近该企业每周的极限产能，直至筛选至某一家时刚刚超过企业每周总产能。依据数学模型公式为其设置新的订购方式和转运方式所用的方法基本上与第二问一致。

在问题二的基础之上，为了减少企业生产成本、转运成本、储存成本，尽量多地采购 A 类、尽量少地采购 C 类原材料。根据问题一中的参数：A 类材料的供应商每周为企业供应的原材料为 X_i （单位：立方米）；供应 B 类材料的供应商每周为企业供应的原材料为 Y_i （单位：立方米）；供应 C 类材料的供应商每周为企业供应的原材料为 Z_k （单位：立方米）。令该企业每周产能为 W_E ，结合企业每周需要达到的产能综合，得：

$$W' = \frac{\sum X_i}{a} + \frac{\sum Y_i}{b} + \frac{\sum Z_k}{c}$$

$$\Delta w = \frac{W'}{\text{周数}} - W_E$$

由于三类原材料运输和储存的单位费用相同，为了达到减少企业生产成本，转运成本，储存成本的效果，要尽可能求得 i_{max} 、 k_{min} 、 Δw_{min} 且 $\Delta w \geq 0$ 。其中 W_E 为周计划产能 2.82 万立方米。 i, k 分别表示选取材料 A, C 的数量。

5.3.2 模型的求解

根据题目中的已知条件相应带入数据计算：

1：将 402 家供应商按照供应的原材料类别进行分类，结合问题一的模型，计算出最为重要的 50 家供应商，将这些供应商分为 A、B、C 三类；

2: 计算出每家供应商 240 周平均每周对该企业的供应量,并按照分类将其平均每周供应量从大到小排序;

3: 先对 A 类供应商平均每周供应量进行叠加,与企业每周所需供应量作对比,计算得出仅 A 类供应商为企业供应原材料无法达到所需,并在此前提下保证了选择更多的 A 类原材料供应商;

4: 计算出剩余产能需求,由供应 B 类原材料的供应商供应剩余部分原材料;

5: 通过计算求解,选取 B 类供应商供应部分剩余原材料,并判断是否能够满足供应需求,通过计算得出无法满足;

6: 计算出 A 类和 B 类供应后剩余产能需求,最后剩余极少数材料供应需求由供应 c 类原材料的供应商供应;

7: 通过计算求解,选取 C 类供应商供应剩余原材料,计算出需要的 C 类供应商的家数及相应数据。

以此为供应方案依据,选择相应供应商为企业供应原材料,满足 i_{\max} 、 k_{\min} 由于要减少仓储的成本,因此供货量应刚刚好或多出的越少越好,即 Δw_{\min} 且 $\Delta w \geq 0$ 的条件,其中 Δw_{\min} 表示多生产出的部分。

$$\left\{ \begin{array}{l} W' = \frac{\sum X_i}{a} + \frac{\sum Y_j}{b} + \frac{\sum Z_k}{c} \\ \min \Delta w = \frac{W'}{\text{周数}} - W_E \geq 0 \\ i_{\max} > k_{\min} \geq 0 \end{array} \right.$$

以此为条件约束,选取以下 32 家供应商才能满足该企业原材料的供应去需求,选取结果如下表 6 所示:

表 6 选择的 32 家供应商

供应商	供应原材料类别	供应商	供应原材料类别
S229	A	S143	A
S201	A	S352	A
S282	A	S157	A
S348	A	S015	A
S275	A	S096	A
S329	A	S097	A
S395	A	S208	A
S307	A	S140	B
供应商	供应原材料类别	供应商	供应原材料类别
S108	B	S040	B
S308	B	S055	B
S139	B	S364	B
S330	B	S367	B
S340	B	S346	B
S131	B	S361	C

S031	B	S151	C
S338	B	S356	C

在供应商的选择环节已经筛选处理 32 家供应商，与问题二处理的方法类似。

5.3.3 关于订购方案的设计

引入 $\Delta \bar{d}_{nm}$ 表示供应商的供货效果用以为第一个筛选条件。为保证该企业所定制的未来 24 周订购方案原材料最经济，引入 A 类材料供货商平均每周供货量作为第二个筛选条件，A 类材料供货商平均每周供货量越大说明 A 类材料越多，C 类材料的需求量越少。综合已经判断出的两个筛选条件，计算排序后得出相应原材料订购方案，结果在附件 A 中给出。

5.3.4 关于转运方案的设计

在制定完成相应的订购方案的设计后，结合题目中的每周最大能进行转运材料的限定空间大小，希望转运商的转运损耗率尽量少计算出前 24 周每周所需要的转运商家数，相应结果在下表：

表 7 前 24 周每周所需转运商数

周次	转运商家数	周次	转运商家数	周次	转运商家数
第 01 周	2	第 09 周	2	第 17 周	2
第 02 周	4	第 10 周	3	第 18 周	2
第 03 周	3	第 11 周	3	第 19 周	2
第 04 周	4	第 12 周	3	第 20 周	5
第 05 周	4	第 13 周	2	第 21 周	3
第 06 周	3	第 14 周	3	第 22 周	4
第 07 周	2	第 15 周	3	第 23 周	5
第 08 周	4	第 16 周	6	第 24 周	6

再根据各家转运商的排序结果，按照顺序选择相应家数的转运商对原材料制定损耗率最少的转运方案。转运方案如下表所示：

表 8 针对于不同转运商每周的安排

转运商 ID	周次	转运商 ID	周次	转运商 ID	周次
T3	第 01 周	T3	第 03 周	T3	第 02 周
T6	第 07 周	T6	第 06 周	T6	第 04 周
	第 09 周	T2	第 10 周	T2	第 05 周
	第 13 周		第 11 周	T8	第 08 周
	第 17 周		第 12 周		第 22 周
	第 18 周		第 14 周		
	第 19 周		第 15 周		
			第 21 周		
转运商 ID	周次	转运商 ID	周次		

T3	第 20 周	T3	第 16 周		
T6	第 23 周	T6	第 24 周		
T2		T2			
T8		T8			
T4		T4			
		T1			

5.4.1 数据预处理

将近五年供应商的供货量数分为五个部分进行处理,在每一年中针对不同的供应材料种类分类,即每年分为 A、B、C 三类。

5.4.2 关于最大产能的计算

将近五年的数据按照年份分为 5 组,即 240 周数据每 48 周为一组,并对每 组数据按照供货商的供货类别进行分类,分别令为 A_p 、 B_p 、 C_p ,其中 $1 \leq p \leq 5$ 且为整数。根据不同类别,分别计算出每年每类原材料所对应的供货量,再结合单位消耗量,得出对应的产能并求均值,记作 W_{0p} 近五年数据分组后各组数据对应的产能单位为立方米,即:

$$W_{0p} = \frac{\sum \bar{A}_p}{a} + \frac{\sum \bar{B}_p}{b} + \frac{\sum \bar{C}_p}{c}$$

观察所得的 W_{0p} 值^[4],挑选出产能最大的值作为标准,计算该企业产能的提高效率:

$$\varphi = \frac{W_0 - \left(\frac{\sum \bar{A}}{a} + \frac{\sum \bar{B}}{b} + \frac{\sum \bar{C}}{c} \right)}{W_0} \times 100\%,$$

其中 W_0 表示近五年总计划产能: 28200×240

代入附件 1 中所提供的数据信息,计算得出: (单位: 立方米)

表 9 每种材料每周的平均供应量

	第一组	第二组	第三组	第四组	第五组
$\sum \bar{A}_p$	7192.7	10123.1	9397.8	8664.0	8836.3
$\sum \bar{B}_p$	6671.6	8029.0	7962.7	8631.0	8181.9
$\sum \bar{C}_p$	7062.9	8271.7	8121.3	6607.0	7437.6
W_{0p}	31905.8	40525.5	39007.5	36693.6	37454.0

φ	13.14%	43.7%	38.3%	30.1%	32.8%
-----------	--------	-------	-------	-------	-------

计算得出该企业每周的产能可以提高 13.14%到 43.7%。

可以发现该企业每周的产能都有提高的潜力，因此只需要压缩生产成本，以减少转运及仓储的成本，同时希望转运商的转运损耗率尽量少，即第三问的求解结果同样适用于本问。

六、模型评价与推广

6.1 模型的优点

(1) 问题一的模型利用 TOPSIS 法, 对 4 个重要指标赋予权重, 得到各供货商质量评分, 指标的选取全面地考虑了已知信息, 例如对度量供货连续性指标时通过熵权法构建评价体系, 使得模型更加可靠:

(2) 问题二的对订货与运输方案构造从问题一中的高质量供货商出发, 以保障企业生产的稳定性为前提, 对订货与运输方案的制定尽可能全面地考虑其影响因素, 且尽量避免主观因素的影响;

(3) 问题三的订货与运输方案以问题二的模型为基础, 数据量大量减少且具有一定的说服力:

(4) 问题四的订货与运输方案同样以问题二的模型为基础, 进一步放宽约束条件。

6.2 模型的缺点

在建模过程中, 对部分约束的研究还不够深入, 未能分析其影响结果的逻辑。

七、参考文献

- [1]朱崇贤.企业物资供应链构建分析[J].中国商论,2021(16):114-116.
- [2]叶红霞.制造企业库存管理问题和对策研究[J].大众投资指南,2021(13):155-156.
- [3]张雨晨. 建筑工程项目群材料运输问题研究[D].沈阳工业大学,2020.
- [4]陶连城. 基于 JSHOP2 的任务动态规划系统研究与实现[D].北京邮电大学,2020.

附录

第一问 matlab

```
%处理数据
clc;clear
[num,txt,row]=xlsread('附件1 近5年402家供应商的相关数据.xlsx',1);
[num2,txt2,row2]=xlsread('附件1 近5年402家供应商的相关数据.xlsx',2);
%计算供给达标周数
sum_hege=zeros(402,1);
[m,n]=size(num);
for j=1:m
    for i=1:n
        if num(j,i)~=0&&num2(j,i)>=num(j,i)
            sum_hege(j,1)=sum_hege(j,1)+1;%计算供给达标的周数
        end
    end
end
end

num_gongji=sum(num~=0,2);%需要的供货次数
num2_sum=sum(num2,2);%总共货量
num2_mean=mean(num2,2);%平均供货量;
dabiaolv=sum_hege./num_gongji;%供货达标率

data=[num2_gongji num2_sum num2_mean dabiaolv];

%熵权
x=data;%导入数据
[n,m]=size(x);
% 数据的归一化处理
[X,ps]=mapminmax(x',0,1);
ps.ymin=0.002;% 归一化后的最小值
ps.ymax=0.996;% 归一化后的最大值
ps.yrange=ps.ymax-ps.ymin;% 归一化后的极差
X=mapminmax(x',ps);
X=X'; % X为归一化后的数据
% 计算比重
for i=1:n
    for j=1:m
        p(i,j)=X(i,j)/sum(X(:,j));
    end
end
```

```

end
%计算熵值
k=1/log(n);
for j=1:m
    e(j)=-k*sum(p(:,j).*log(p(:,j)));
End
d=ones(1,m)-e; % 计算信息熵冗余度
w=d./sum(d); % 求权值 w
%TOPSIS 综合评价
A=data;%评价矩阵
W=w%权重
[ma, na]=size(A);
for i=1:na
    B(:,i)=A(:,i)*W(i); %得到加权标准化矩阵
end
V1=zeros(1,na); %初始化正理想解和负理想解
V2=zeros(1,na);
BMAX=max(B); %取加权标准化矩阵每列的最大值和最小值
BMIN=min(B);
for i=1:na
    V1(i)=BMAX(i);
    V2(i)=BMIN(i);
end
for i=1:ma
    C1=B(i,:)-V1;
    S1(i)=norm(C1); %S1, S2 分别为离正理想点和负理想点的距离
    C2=B(i,:)-V2;
    S2(i)=norm(C2);
    T(i)=S2(i)/(S1(i)+S2(i));
end
output=T'*100%得出评分

```

第二问 python

```

是笑死 import xlrd
import numpy
file_location = "C:/Users/gaofan/Desktop/数学建模代码/a.xlsx"
data = xlrd.open_workbook(file_location)
sheet = data.sheet_by_index(1)
sheet2=data.sheet_by_index(0)
data1 = [[sheet.cell_value(r,c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]#读取数据

#402 家供应商 240 周供应量总和

```

```

list1 = numpy.zeros((402, 2))#240 周总共的
for i in range(0, 402):
    sum=0
    for j in range(0, 240):
        sum=sum+data1[i+1][j+2]
    list1[i][0]=sum
    list1[i][1]=i+1

#402 家供应商 240 周供应量总和的排序
for i in range (0, 402):#排序
    for j in range(0, 401-i):
        if list1[j][0]<list1[j+1][0]:
            (list1[j][0], list1[j+1][0])=(list1[j+1][0], list1[j][0])
            (list1[j][1], list1[j+1][1])=(list1[j+1][1], list1[j][1])
# 0.3%:240*28200*0.3%=20304(这个不是排名, 是实际值)
qushu = 0
for i in range(0, 402):
    if list1[i][0] < 20304:
        qushu = i + 1
        break

print(qushu)
for i in range(0, qushu):
    print(list1[i][1])

import xlrd
import numpy

file = "C:/Users/gaofan/Desktop/数学建模代码/c.xlsx" # 37 家供应商的表格
data2 = xlrd.open_workbook(file)
sheet1 = data2.sheet_by_index(0)
data_d = [[sheet1.cell_value(r, c) for c in range(sheet1.ncols)] for r in
range(sheet1.nrows)]

sheet2 = data2.sheet_by_index(1)
data_g = [[sheet2.cell_value(r, c) for c in range(sheet2.ncols)] for r in
range(sheet2.nrows)]
list1 = numpy.zeros((37, 239, 4)) # 第三和第四是每个的差值 (有正负那种)
a = b = c = 0

for i in range(0, 37):
    for j in range(0, 239):
        a = data_g[i + 1][j + 2] - data_d[i + 1][j + 2]
        b = data_g[i + 1][j + 3] - data_d[i + 1][j + 3]

```

```

list1[i][j][2] = a
list1[i][j][3] = b
if a >= 0:
    a = a
else:
    a = -a
if b >= 0:
    b = b
else:
    b = -b
c = (a + b) / 2
list1[i][j][0] = c # 差值 (除法)
list1[i][j][1] = j + 1 # 周数

for k in range(0, 37):
    for i in range(0, 239):
        for j in range(0, 238 - i):
            if list1[k][j][0] > list1[k][j + 1][0]:
                (list1[k][j][0], list1[k][j + 1][0]) = (list1[k][j + 1][0],
list1[k][j][0])
                (list1[k][j][1], list1[k][j + 1][1]) = (list1[k][j + 1][1],
list1[k][j][1])
                (list1[k][j][2], list1[k][j + 1][2]) = (list1[k][j + 1][2],
list1[k][j][2])
                (list1[k][j][3], list1[k][j + 1][3]) = (list1[k][j + 1][3],
list1[k][j][3])

list2 = numpy.zeros((37, 12, 3))
for i in range(0, 37):
    for j in range(0, 12):
        list2[i][j][0] = -2

for i in range(0, 37):
    # k=12
    k = 0
    for flag in range(0, 50):
        # for k in range(0, 12):
        b = 0
        for j in range(0, 12):
            # if list1[i][12-k][1]==list2[i][j][0]+1:
            if list1[i][flag][1] == list2[i][j][0] + 1 or list1[i][flag][1]
== list2[i][j][0] - 1:
                b = 1
                break

```

```

        if b == 0:
            list2[i][k][0] = list1[i][flag][1]
            list2[i][k][1] = list1[i][flag][2]
            list2[i][k][2] = list1[i][flag][3]
            # list2[i][12-k][0]=list1[i][12-k][1]
            k = k + 1
        if k == 12:
            break
list3 = numpy.zeros((37,24))

for i in range(0,37):
    a=b=c=0
    for j in range (0,12):
        a=list2[i][j][0]
        b=a+1
        c=int(a+2)
        list3[i][j*2]=data_d[i+1][c]
        k=k+1
        c=int(b+2)
        list3[i][j*2+1]=data_d[i+1][c]
list = numpy.zeros((8,2))
for i in range(0,8):
    sum0=0
    sum=0
    for j in range (0,240):
        if data[i+1][j+1]==0 :
            sum0=sum0+1
        else :
            sum=sum+data2[i+1][j+1]
    sum=sum/(240-sum0)
    list[i][0]=sum
    list[i][1]=i+1
for i in range (0,8):
    for j in range(0,7-i):
        if list[j][0]>list[j+1][0]:
            (list[j][0],list[j+1][0])=(list[j+1][0],list[j][0])
            (list[j][1],list[j+1][1])=(list[j+1][1],list[j][1])

for i in range(0,8):
    print(list[i][0],list[i][1])

a=0
list4= numpy.zeros(24)

```

```

for i in range(0,24):#列
    sum=0
    for j in range(0,37):#行
        a = data_y[j+1][i+2]
        sum=sum+a
    list4[i]=sum
for i in range(0,24):#b 代表的是需要选取转运商的家数
    b=list4[i]//6000
    if list4[i]%6000 != 0:
        b=b+1

```

第三问 python

```

import numpy
import xlrd

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/a.xlsx"
data = xlrd.open_workbook(file_location)
sheet = data.sheet_by_index(1)
sheet2 = data.sheet_by_index(0)
data_d = [[sheet2.cell_value(r, c) for c in range(sheet2.ncols)] for r in
range(sheet2.nrows)]

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/b.xlsx"
data = xlrd.open_workbook(file_location)
sheet = data.sheet_by_index(0)
data = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]

```

```

file = "C:/Users/gaofan/Desktop/数学建模代码/e.xlsx"
data1 = xlrd.open_workbook(file)
sheet1 = data1.sheet_by_index(0)
data3_d = [[sheet1.cell_value(r, c) for c in range(sheet1.ncols)] for r in
range(sheet1.nrows)]
sheet2 = data1.sheet_by_index(1)
data3_g = [[sheet2.cell_value(r, c) for c in range(sheet2.ncols)] for r in
range(sheet2.nrows)]

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/f.xlsx"
data3_y = xlrd.open_workbook(file_location)
sheet = data3_y.sheet_by_index(0)
data3_y = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]

```

```

# 订货量算总数(402 家企业 240 周总供应量)
list2 = numpy.zeros((402, 2))

for i in range(0, 402):
    sum = 0
    for j in range(0, 240):
        sum = sum + data_d[i + 1][j + 2]
    list2[i][0] = sum
    list2[i][1] = i + 1

# 订货量排名
for i in range(0, 402): # 排序
    for j in range(0, 401 - i):
        if list2[j][0] < list2[j + 1][0]:
            (list2[j][0], list2[j + 1][0]) = (list2[j + 1][0], list2[j][0])
            (list2[j][1], list2[j + 1][1]) = (list2[j + 1][1], list2[j][1])

num = 0
sum = 0
for i in range(0, 50):
    a = int(list2[i][1])
    if data_d[a][1] == 'A':
        sum = sum + list2[i][0] / 0.6
        num = num + 1
        print(list2[i][1])
        if (sum / 240) > 28200:
            print('$', num + 1, list2[i][1])
            break

for i in range(0, 50):
    a = int(list2[i][1])
    if data_d[a][1] == 'B':
        sum = sum + list2[i][0] / 0.66
        num = num + 1
        print(list2[i][1])
        if (sum / 240) > 28200:
            print('$', num + 1, list2[i][1])
            break

for i in range(0, 50):
    a = int(list2[i][1])
    if data_d[a][1] == 'C':
        sum = sum + list2[i][0] / 0.72
        num = num + 1

```

```

        print(list2[i][1])
        if (sum / 240) > 28200:
            print('$', num, list2[i][1])
            break

list = numpy.zeros((8, 2))
for i in range(0, 8):
    sum0 = 0
    sum = 0
    for j in range(0, 240):
        if data[i + 1][j + 1] == 0:
            sum0 = sum0 + 1
        else:
            sum = sum + data[i + 1][j + 1]
    sum = sum / (240 - sum0)
    list[i][0] = sum
    list[i][1] = i + 1
for i in range(0, 8):
    for j in range(0, 7 - i):
        if list[j][0] > list[j + 1][0]:
            (list[j][0], list[j + 1][0]) = (list[j + 1][0], list[j][0])
            (list[j][1], list[j + 1][1]) = (list[j + 1][1], list[j][1])

for i in range(0, 8):
    print(list[i][0], list[i][1])

# 第三问的 32 家          1
list_1 = numpy.zeros((32, 239, 4)) # 第三和第四是每个的差值（有正负那种）
a = b = c = 0
for i in range(0, 32):
    for j in range(0, 239):
        a = data3_g[i + 1][j + 2] - data3_d[i + 1][j + 2]
        b = data3_g[i + 1][j + 3] - data3_d[i + 1][j + 3]
        list_1[i][j][2] = a
        list_1[i][j][3] = b
        if a >= 0:
            a = a
        else:
            a = -a
        if b >= 0:
            b = b
        else:
            b = -b
        c = (a + b) / 2

```



```

list_1[i][j][0] = c # 差值 (除法)
list_1[i][j][1] = j + 1 # 周数

for k in range(0, 32):
    for i in range(0, 239):
        for j in range(0, 238 - i):
            if list_1[k][j][0] > list_1[k][j + 1][0]:
                (list_1[k][j][0], list_1[k][j + 1][0]) = (list_1[k][j +
1][0], list_1[k][j][0])
                (list_1[k][j][1], list_1[k][j + 1][1]) = (list_1[k][j +
1][1], list_1[k][j][1])
                (list_1[k][j][2], list_1[k][j + 1][2]) = (list_1[k][j +
1][2], list_1[k][j][2])
                (list_1[k][j][3], list_1[k][j + 1][3]) = (list_1[k][j +
1][3], list_1[k][j][3])

# 第三问的          2
list_2 = numpy.zeros((32, 12, 3))
for i in range(0, 32):
    for j in range(0, 12):
        list_2[i][j][0] = -2

for i in range(0, 32):
    # k=12
    k = 0
    for flag in range(0, 50):
        # for k in range(0, 12):
        b = 0
        for j in range(0, 12):
            # if list1[i][12-k][1]==list2[i][j][0]+1:
            if list_1[i][flag][1] == list_2[i][j][0] + 1 or
list_1[i][flag][1] == list_2[i][j][0] - 1:
                b = 1
                break
        if b == 0:
            list_2[i][k][0] = list_1[i][flag][1]
            list_2[i][k][1] = list_1[i][flag][2]
            list_2[i][k][2] = list_1[i][flag][3]

            # list2[i][12-k][0]=list1[i][12-k][1]
            k = k + 1
        if k == 12:
            break

```

```

# 第三问的          3
list_3 = numpy.zeros((32, 24))

for i in range(0, 32):
    a = b = c = 0
    for j in range(0, 12):
        a = list_2[i][j][0]
        b = a + 1
        c = int(a + 2)
        list_3[i][j * 2] = data3_d[i + 1][c]
        k = k + 1
        c = int(b + 2)
        list_3[i][j * 2 + 1] = data3_d[i + 1][c]

```

```

# 第三问的          4
a = 0
list_4 = numpy.zeros(24)
for i in range(0, 24): # 列
    sum = 0
    for j in range(0, 32): # 行
        a = data3_y[j + 1][i + 2]
        sum = sum + a
    list_4[i] = sum

```

```

# 第三问的          5
for i in range(0, 24):
    b = list_4[i] // 6000
    if list_4[i] % 6000 != 0:
        b = b + 1
    # print(b)
    # print(list_4[i]/(6000*b))

```

第四问 python

```

# -*- coding: utf-8 -*-
import numpy
import xlrd

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/a.xlsx"
data = xlrd.open_workbook(file_location)
sheet = data.sheet_by_index(1)
sheet2 = data.sheet_by_index(0)
data = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in

```

```

range(sheet.nrows)]
data1 = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]
data_d = [[sheet2.cell_value(r, c) for c in range(sheet2.ncols)] for r in
range(sheet2.nrows)]

list3 = numpy.zeros((240, 3)) # 分类看歌类每周的总值

for j in range(0, 240):
    sum = 0
    for i in range(0, 402):
        if data1[i + 1][1] == 'A':
            sum = sum + data_d[i + 1][j + 2]
    list3[j][0] = sum

for j in range(0, 240):
    sum = 0
    for i in range(0, 402):
        if data1[i + 1][1] == 'B':
            sum = sum + data_d[i + 1][j + 2]
    list3[j][1] = sum

for j in range(0, 240):
    sum = 0
    for i in range(0, 402):
        if data1[i + 1][1] == 'C':
            sum = sum + data_d[i + 1][j + 2]
    list3[j][2] = sum

for i in range(1, 6):
    sumA = sumB = sumC = 0
    print(i)
    for j in range((i - 1) * 48, i * 48):
        sumA = sumA + list3[j][0]
        sumB = sumB + list3[j][1]
        sumC = sumC + list3[j][2]
    sumA = sumA / 48
    sumB = sumB / 48
    sumC = sumC / 48
    print(sumA)
    print(sumB)
    print(sumC)
    print(sumA / 0.6 + sumB / 0.66 + sumC / 0.72)

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/a.xlsx"
data4 = xlrd.open_workbook(file_location)
sheet = data4.sheet_by_index(1)
sheet2 = data4.sheet_by_index(0)
data4_g = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]
data4_d = [[sheet2.cell_value(r, c) for c in range(sheet2.ncols)] for r in
range(sheet2.nrows)]

```

```

file_location = "C:/Users/gaofan/Desktop/数学建模代码/g.xlsx"
data4_y = xlrd.open_workbook(file_location)
sheet = data4_y.sheet_by_index(0)
data4_y = [[sheet.cell_value(r, c) for c in range(sheet.ncols)] for r in
range(sheet.nrows)]

```

```

# 第四问的 402 家          1
list4_1 = numpy.zeros((402, 239, 4)) # 第三和第四是每个的差值（有正负那种）
a = b = c = 0
for i in range(0, 402):
    for j in range(0, 239):
        a = data4_g[i + 1][j + 2] - data4_d[i + 1][j + 2]
        b = data4_g[i + 1][j + 3] - data4_d[i + 1][j + 3]
        list4_1[i][j][2] = a
        list4_1[i][j][3] = b
        if a >= 0:
            a = a
        else:
            a = -a
        if b >= 0:
            b = b
        else:
            b = -b
        c = (a + b) / 2
        list4_1[i][j][0] = c # 差值（除法）
        list4_1[i][j][1] = j + 1 # 周数

for k in range(0, 402):
    for i in range(0, 239):
        for j in range(0, 238 - i):
            if list4_1[k][j][0] > list4_1[k][j + 1][0]:
                (list4_1[k][j][0], list4_1[k][j + 1][0]) = (list4_1[k][j +
1][0], list4_1[k][j][0])
                (list4_1[k][j][1], list4_1[k][j + 1][1]) = (list4_1[k][j +
1][1], list4_1[k][j][1])

```

```

        (list4_1[k][j][2], list4_1[k][j + 1][2]) = (list4_1[k][j +
1][2], list4_1[k][j][2])
        (list4_1[k][j][3], list4_1[k][j + 1][3]) = (list4_1[k][j +
1][3], list4_1[k][j][3])

```

```

        # 第四问的                2
list4_2 = numpy.zeros((402, 12, 3))
for i in range(0, 402):
    for j in range(0, 12):
        list4_2[i][j][0] = -2

for i in range(0, 402):
    # k=12
    k = 0
    for flag in range(0, 50):
        # for k in range(0, 12):
        b = 0
        for j in range(0, 12):
            # if list1[i][12-k][1]==list2[i][j][0]+1:
            if list4_1[i][flag][1] == list4_2[i][j][0] + 1 or
list4_1[i][flag][1] == list4_2[i][j][0] - 1:
                b = 1
                break
        if b == 0:
            list4_2[i][k][0] = list4_1[i][flag][1]
            list4_2[i][k][1] = list4_1[i][flag][2]
            list4_2[i][k][2] = list4_1[i][flag][3]
            # print(list4_2[i][k][0])

            # list2[i][12-k][0]=list1[i][12-k][1]
            k = k + 1
        if k == 12:
            break

```

```

# 第四问的                3
list4_3 = numpy.zeros((402, 24))

```

```

for i in range(0, 402):
    a = b = c = 0
    for j in range(0, 12):
        a = list4_2[i][j][0]
        b = a + 1
        c = int(a + 2)
        # print(c)

```

```

        list4_3[i][j * 2] = data4_d[i + 1][c]
        k = k + 1
        c = int(b + 2)
        # print(c)
        list4_3[i][j * 2 + 1] = data4_d[i + 1][c]

# 第四问的          4
a = 0
list4_4 = numpy.zeros(24)
for i in range(0, 24): # 列
    sum = 0
    for j in range(0, 402): # 行
        a = data4_y[j + 1][i + 2]
        sum = sum + a
    list4_4[i] = sum

for i in range(0, 24):
    b = list4_4[i] // 6000
    if list4_4[i] % 6000 != 0:
        b = b + 1
    print(b, end=" ")
print(list4_4[i] / (6000 * b))

```