

生产企业原材料的订购与运输

摘要

本文主要研究关于生产企业在订购原材料、运输原材料和生产产品的过程中，在不同限制条件下，分析企业原材料订购方案和转运方案的问题。在解决问题的过程中，通过构建出 RFM 模型、SARIMAX 模型、多目标优化模型、退火算法等，并与 Python 编程语言相结合，求解出模型中所涉及的问题并给出了相应的结论，下面对具体问题进行概括。

针对问题一，对附件一中所给的该企业在近五年具体的原材料数据进行可视化处理，根据处理后的数据来量化分析供应商们的供货特征，对此问题基于 RFM 模型建立出 *ITQR* 数学模型，并且结合 K-Means 聚类对数据进行标准化处理，最终得到了 50 家最重要的供应商的结果。

针对问题二，以供应商每次供货的平均供货量为基础，计算得到 *A*、*B*、*C* 三种原材料的生产产品数量情况，以满足该企业每周的产能为基准，求出所需最少供应原材料的供应商数目；此后再利用 SARIMAX 模型非线性拟合出未来 24 周该企业原材料购买数量的折线图，结合拟合所得数据、产出比例和转运损耗率来制定出最经济的订购方案和损耗最少的转运方案。

针对问题三，结合目标函数、决策变量和约束条件来建立多目标优化模型的方式来进行解决。为了最为贴合题设要求，假设足够多的决策变量来综合分析，设计好决策变量后根据题目来建立目标函数，确定约束条件，最终与退火算法相结合来制定出最为节省生产成本的订购方案和转运方案。

针对问题四，利用多目标规划问题和 SARIMAX 模型非线性拟合，分析计算得到在不超过每家转运商每周最大的转运量 6000 立方米且损耗率较低的情况下，该企业每周可提升的产能。

关键词：Python RFM 模型 SARIMAX 模型 多目标优化模型 退火算法

一、问题重述

1.1 背景知识

原材料是企业生产的重要组成部分，是企业经营的必要条件，一般是由原料和材料组成，材料是加工后的原料，企业可以更好地进行生产加工。除此之外，为了保证企业生产的正常运行，企业还需提前存储一些原材料备用。

某生产企业所用原材料主要是木质纤维和其他植物素纤维材料，总体可分为 A、B、C 三种类型。该企业收购原材料的流程具体划分为：企业首先向供货商订购所需某种数量的原材料，原材料供应商供货后，第三方物流公司将供应商所提供原材料转运至企业仓库，至此企业完成一次原材料采购。

1.2 具体问题

1.问题一

附件一中给出了企业向 402 家供货商的订货数据和供货商的供货数据，据此数据对供货商的供货特征进行量化分析，在对数据处理分析后，建立数学模型和某些指标用以确定最重要的 50 家供应商。

2.问题二

因为问题一中已经确定最重要的 50 家供货商，可参考此数据，分析出在该企业满足生产需求的条件下，至少选择的供货商数量，再基于此数据为企业制定未来 24 周每周最经济的原材料订购方案，根据订购方案再制定出损耗最少的转运方案，并对两个方案进行分析。

3.问题三

为了压缩企业的生产成本和减少转运及仓储的成本，该企业打算多购买 A 类原材料和少采购 C 类原材料，同时尽量减少转运商的转运损耗率，在此目标下制定新的订购方案及转运方案，并分析这两个方案的实际效果。

4.问题四

若该企业生产技术进行升级，可以满足提高每周产能的条件，在已知原材料的供应商和转运商条件下，建立模型确定该企业每周产能可提升多少，并给出未来 24 周的订购和转运方案。

二、问题分析

2.1 针对问题一的分析

附件一中给出了该企业在近五年每周向供应商分别订购A、B、C三种原材料的数据以及供应商向企业供货量的详细数据，我们对附件一中的数据进行可视化处理，量化分析供货特征，确定关键因素作为衡量供应商重要性的指标，并依据此建立数学模型，在确定 50 家最重要的供货商后，在论文中列表给出结果。

2.2 针对问题二的分析

首先，以附件一中的原始数据、企业的每周产能 2.82 万立方米和问题一中所确定的 50 家最重要供应商为基础，对数据进行分析处理后，以每次平均供货量为基础，求出最少能满足产能的供货商数目；再将处理后的原始数据进行分类处理，将三种原料的采购量绘制成图，再利用 SARIMAX 模型非线性拟合模拟出未来 24 周每周所需的 A、B、C 原材料购买总和，综合供应商供货量、非线性拟合模拟出的未来 24 周的购买数据、题目所给原材料 A、B、C 的价格比例和产出比例等等因素，为企业制定未来 24 周每周最经济的原材料订购方案，依据订购方案和尽量减少货物损耗，制定转运方案，并对实际效果进行分析。

2.3 针对问题三的分析

该企业为了压缩成本，计划减少转运的成本和存储的成本。对此，我们建立了多目标优化模型，具体考虑的目标变量有：未来 24 周中每周三种原材料的采购量、三种原材料的价格， $T_1 - T_8$ 共 8 家转运商转运原材料时的转运损耗率以及企业每周最基本的产能。根据这些决策变量和题目所给的约束条件来构建所需多目标优化模型并结合退火算法解决所需求解问题。

2.4 针对问题四的分析

我们根据附件一和附件二中的原材料供应商的供应量和转运商的转运损耗率的数据，对该企业新的生产能力进行预测模拟，建立 SARIMAX 模型来进行预测，确定在未来的 24 周，在原本条件下将该企业的产能提高至更高的效率来生产数量更大的产品。

三、问题假设

- 1.假设所给数据真实有效。
- 2.假设该企业仓库容量可容纳所有原材料。
- 3.假设该企业每周生产的需求至少为每周的产能。
- 4.假设所得数据小数过多时，默认数据取小数点后两位。
- 5.假设转运商转运时三种原材料可以分量转运。
- 6.假设在计算过程中，允许小数舍去时与数据有略微差异；
- 7.假设在转运过程中， A 、 B 、 C 三类原材料可同时转运；
- 8.假设在第四问中该企业在提高产能过程中，不考虑成本。

四、符号说明

符号	说明
I	订货时间间隔
T_1	企业向供应商订货次数
Q_1	企业向供应商订货总量
T_2	供应商实际供应次数
Q_2	供应商供货总量
R	供货商的供货率
x_{ij}	表示第 i 周第 j 类原材料的生产成本
l_i	表示第 i 类原材料能生产出产品的产出比
T_i	表示第 i 家转运商
r_{ij}	表示第 i 家转运商在第 j 周转运的转运损耗率
Q_{ij}	表示第 i 家转运商在第 j 周转运的原材料数量
Y_A	表示 A 类原材料的购买量
Y_B	表示 B 类原材料的购买量
Y_C	表示 C 类原材料的购买量

五、模型的建立与求解

5.1 问题一的分析与求解

1.对问题一的分析

首先对附件一进行数据分析，附件一中给出了该企业在近五年、共 240 个周的时间内，每周向 402 家供应商分别订购 A 、 B 、 C 三种原材料的具体数据以及 402 家供应商向企业供货量的详细数据，再对附件一中的数据进行可视化处理，量化分析供货商们的供货特征。对此，我们依据订货时间间隔 I 、企业向供应商订货次数 T_1 、企业向供应商订货总量 Q_1 、供应商实际供应次数 T_2 、供应商供货总量 Q_2 以及最为体现供货商供货效率的供货率 R 这六个指标作为衡量供应商重要性的指标，基于 RFM 模型建立出 $ITQR$ 数学模型，利用 K-Means 聚类对数据进行标准化处理，即可对问题一进行分析求解。

2.对问题一的求解

利用 PyCharm Community Edition、Visual Studio Code 等软件，导入附件一中的具体数据后，结合 Python、JavaScript 语言对数据进行编程可视化处理，结果如下图所示：

依据在我们问题一分析中依据订货时间间隔 I 、企业向供应商订货次数 T_1 、企业向供应商订货总量 Q_1 、供应商实际供应次数 T_2 、供应商供货总量 Q_2 以及供货率 R 这六个指标，基于 RFM 模型建立出 $ITQR$ 数学模型。一般情况下 RFM 模型分析所采用的是属性分箱方法，但此问题中我们划分指标较多，较为复杂，因此，我们决定利用 K-Means 聚类来对数据进行分析，从而识别出最为重要的 50 家供应商。

由于附件一中的数据并未给出我们所需的六个关键指标，我们需要对附件一中的数据进行计算处理，得出所需指标，具体计算方式如下表所示：

表 5.1 所需指标的计算方式

名称	计算方式
订货时间间隔 I	结束时间—最近一次订货时间（单位：周）
企业向供应商订货次数 T_1	企业向供应商订货的次数总和
企业向供应商订货总量 Q_1	企业向供应商订货的货物总量
供应商实际供应次数 T_2	企业向供应商订货后供应商实际供应次数总和
供应商供货总量 Q_2	供应商实际供应货物的总和
供货率 R	计算公式为： $R = \frac{\text{供货总量 } Q_2}{\text{订货总量 } Q_1}$

根据题目所给附件一中的数据 and 表 5.1 所示的计算方法，我们对数据进行编程处理，得到 402 位供货商的六个指标数据，由于数据较多，完整数据附着于附录/支撑材料中，所得结果（部分截图）如下图 5.1 所示：

供应商ID	订货次数	供货次数	订货量	供货量	最近一次订货时间间隔/week	供货率
S001	91	25	231	49	4	0.212121212
S002	95	71	309	273	0	0.883495146
S003	199	191	14279	13138	0	0.920092443
S004	103	33	713	64	2	0.089761571
S005	114	107	6538	6912	0	1.057204038
S006	55	13	462	30	7	0.064935065
S007	240	240	6716	6948	0	1.034544372
S008	56	15	94	41	0	0.436170213
S009	80	19	715	31	0	0.043356643
S010	88	32	576	170	2	0.295138889
S011	43	32	84	85	1	1.011904762
S012	68	12	308	29	2	0.094155844
S013	29	20	44	44	6	1
S014	27	16	32	28	18	0.875
S015	54	15	27883	28	2	0.001004196
S016	22	17	38	37	31	0.973684211
S017	67	25	194	138	2	0.711340206
S018	72	31	113	66	2	0.584070796
S019	58	12	277	109	0	0.393501805
S020	89	33	183	72	2	0.393442623

图 5.1 所有供货商的六个指标数据

其中，当供货率 $R < 1$ 时，越接近 0 则说明供货量越低于订货量；

当供货率 $R = 1$ 时，则说明供货量与订货量持平；

当供货率 $R > 1$ 时，则说明供货量多于订货量。

综上所述，当订货时间间隔 I 越短、企业向供应商订货的次数 T_1 越多、企业向供应商订货总量 Q_1 越大、供应商实际供应次数 T_2 越多、供应商供货总量 Q_2 越多、供货率 R 越接近于 1 或者大于 1 时，我们认为该供应商是较为重要的供应商。

基于这种判断，同时为了方便后续进行 K-Means 聚类分析，我们对已经分

类处理后的数据进行标准化处理，数据标准化处理结果如下图 5.2 所示：

	聚类名称	聚类个数	Z订货次数	Z供货次数	Z订货量
0	群1	186	-0.33258756	-0.505998728	-0.224630309
1	群2	19	2.170315657	2.219974167	3.868664133
2	群3	30	-0.952851849	-0.581530816	-0.279106028
3	群4	73	1.426980098	1.494409777	-0.002363023
4	群5	94	-0.584667054	-0.42244399	-0.246571271
	聚类名称	聚类个数	Z供货量	Z最近一次订货时间间隔/week	Z供货率
0	群1	186	-0.25678044	-0.182489896	-0.919639866
1	群2	19	3.93858019	-0.504925901	0.752653525
2	群3	30	-0.26250213	2.908006429	0.91512536
3	群4	73	0.035442238	-0.513988693	0.84970843
4	群5	94	-0.231745545	-0.065769208	0.715639595

图 5.2 数据标准化处理

在对数据进行标准化后，我们对群体进行分类，根据所得六个指标数据和 K-Means 聚类将其分为五个群体，为了更直观的观察数据，我们对数据做雷达图可视化处理，所得五个群体雷达图如下 5.3 图所示：

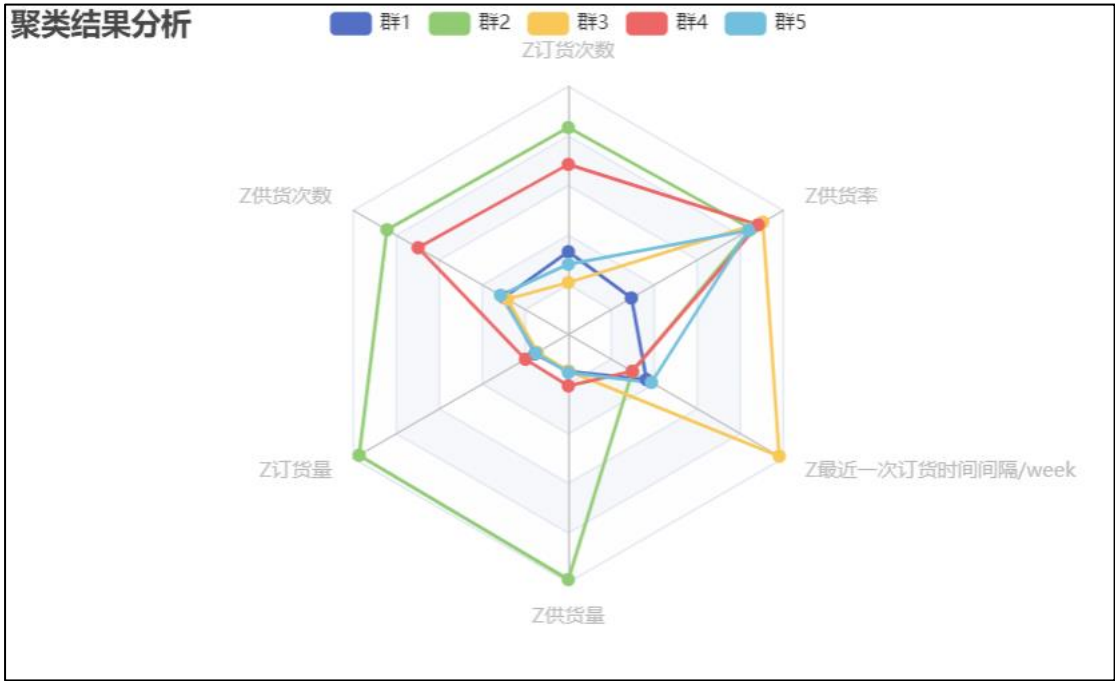


图 5.3 K-Means 聚类所得五个群体

在图中，我们可以直观的看出群体 2 除去最近一次订货时间外，其余五个指标都较为完美，可将群体 2 中所有 19 位供应商视为较为重要的供应商；除去群体 2，虽然群体 4 的订货量和供货量与群体 2 相比较少，但群体 4 的供货次数和订货次数较多，并且在次数较多的情况下供货效率极高，因此我们也将其视为较为重要的供应商群体；由于群体 4 所含供应商数较多，我们按照较为重要的供货

率 R 进行排行，筛选出群体 4 中最为重要的 31 位供应商。

综上所述，在我们对 402 家供货商进行量化分析后、建立数学模型后，在此基础上确定的 50 家最重要的供应商如下表 5.2 所示：

表 5.2 50 家最重要的供应商

序号	供应商 ID	序号	供应商 ID	序号	供应商 ID	序号	供应商 ID	序号	供应商 ID
1	S201	11	S229	21	S023	31	S115	41	S037
2	S348	12	S131	22	S269	32	S314	42	S114
3	S140	13	S340	23	S260	33	S291	43	S307
4	S151	14	S194	24	S150	34	S086	44	S338
5	S308	15	S329	25	S202	35	S197	45	S239
6	S330	16	S275	26	S074	36	S129	46	S054
7	S139	17	S268	27	S066	37	S245	47	S143
8	S108	18	S306	28	S146	38	S258	48	S046
9	S356	19	S282	29	S210	39	S088	49	S263
10	S361	20	S138	30	S376	40	S055	50	S003

5.2 问题二的分析与求解

1.问题二的分析

为了确定该企业在满足每周生产需求的条件下，至少需要的供货商数目，我们对附件一中的数据分析处理，计算得出 402 家供应商每次供应量的均值，以每次供货的平均供货量为基础，列出等式求出 A、B、C 等原材料的生产产品数量情况，最终确定供货量至少满足每周生产需求 2.82 万立方米的供货商数目。

依据附件一处理后的数据可绘制出 A、B、C 三种材料的每周需求数量折线图，我们发现利用 SARIMAX 模型非线性拟合与实际购买量较为相符，因此可以

通过SARIMAX模型非线性拟合绘制出未来 24 周原材料购买数量的折线图，根据拟合所得数据、供应商供货量和原材料A、B、C的价格比例和产出比例等等条件，制定出最佳订购方案，再根据附件二中 $T_1 - T_8$ 共 8 个转运商的转运损耗率，利用MDS多维标度分析选择损耗率最低转运商进行转运，从而制定出损耗最少的订购方案。

2.问题二的求解

根据题目条件可知，该企业每周的产能为 2.82 万立方米，若需满足该企业每周的生产需求，则需分析求得最少需要的供货商数目来供应不同数量的A、B、C等原材料，使得这些数量的原材料经过加工生产后的产品产量总和等于或者多于 2.82 万立方米。因此，我们采用所有供应商供应数量均值与对应比例的运算结果累加，与目标结果相对比，直到结果大于等于目标数目，这时的供应商数目则就是该企业至少需要的供应商数量。

由附件一中的企业的订货量数据确定企业在近五年向每家供应商的订购次数总和，并且根据附件一中供货量也可计算得出近五年各类原材料、各个供应商的供货总量，我们将处理后的数据进行均值计算，将平均每次供货量的计算结果进行从大到小的排序，排序结果如下图 5.4 所示：

供应商ID	材料分类	订货次数	供货次数	订货量	供货量	平均每次供货量
S201	A	49	28	348699	81989	1673. 244898
S229	A	240	240	359885	354887	1478. 695833
S140	B	220	219	481103	302047	1372. 940909
S361	C	240	240	333452	328080	1367
S108	B	240	240	271445	240950	1003. 958333
S395	A	84	74	105436	75843	902. 8928571
S151	C	240	240	266510	194498	810. 4083333
S340	B	240	240	172000	171426	714. 275
S282	A	240	240	168531	169340	705. 5833333
S139	B	224	222	177748	151862	677. 9553571
S275	A	240	240	158150	158553	660. 6375
S329	A	240	240	156350	156518	652. 1583333
S131	B	240	240	139381	137512	572. 9666667
S308	B	240	240	186278	136998	570. 825
S330	B	240	240	172980	136652	569. 3833333
S356	C	240	240	132612	130307	542. 9458333

图 5.4 402 家供货商均次供货量排序

若按平均每次供货量从小到大排序，所得供应商与第一问所得 50 家供应商

相对比，其中 50 家供应商中群体 2 的 19 家的供应商完全吻合，群体四的 31 家供应商中有 24 家完全相吻合，数据相吻合比率达到 86%，因此这两者数据也可相互印证。

根据图 5.4 中数据，再将所有供应商供应数量均值中 A 类原材料数量除以 0.6，B 类原材料结果除以 0.66，C 类原材料结果除以 0.72，即可求得这下原材料可生产的产品总和，代码实现如下图 5.5 所示：

```
50 # #拿到26个每次平均供货量
51 New_Data= Data2.iloc[:26]
52 # New_Data.to_csv('26.csv',encoding='gbk')
53
54 A= (New_Data[New_Data['材料分类']=='A']['平均每次供货量']/0.6).sum()
55 print(A)
56 ...
60
61 B = (New_Data[New_Data['材料分类']=='B']['平均每次供货量']/0.66).sum()
62 C = (New_Data[New_Data['材料分类']=='C']['平均每次供货量']/0.72).sum()
63 print(B)
64 print(C)
65 print(A+B+C) #28424.794666507616
66
```

图 5.5 部分代码实现截图

当从大到小供应商的个数为 26 家时，所供应的 A、B、C 原材料可生产的企业产品数量分别为 12846.96、8306.52、7271.31（单位：立方米），这些原材料可生产的企业产品总数为 28424.79（单位：立方米），计算结果如下图。。所示：

```
Run: 问题二每周拟合B x 第二问五十个全部找出来 x
D:\PythonXiangMu\venv\Scripts\python.exe D:/Py
12846.960218484186
8306.52212055621
7271.312327467219
28424.794666507616
Process finished with exit code 0
```

图 5.6 代码所得计算结果截图

计算得出可生产的企业产品总数为 28424.79（单位：立方米），所得结果略微超出该企业每周的 2.82 万立方米的产能，因此该企业应至少选择 26 家企业才可能满足企业每周生产的需求。

在确定最少满足生产的供应商数目后，针对问题一、问题二所确定的供应商和附件一中所给的 240 周企业的订购数据，对附件一中的 240 周的购入原材料数量进行分析处理，绘制出该企业在 240 周内每周 A、B、C 三种材料的数量折线图，如下图 5.7 所示：



图 5.7 240 周购买三类原材料数量折线图

根据该企业在 240 周分别购买三类原材料的数据，观察该折线图我们可以看出，这三张折线图在 240 周的时间序列内呈现出一定的循环或周期性，不同区间内的走势趋势大致相同。

由于原材料的采购成本可以直接影响到企业效益，而 C 类原材料价格最为便宜，A 类和 B 类原材料的价格分别比 C 类原材料的价格高出 20%和 10%，假设 C 类原材料成本为 10，则 A 类原材料价格为 12，B 类原材料价格为 11，同时，题目也说明 0.6 立方米的 A 类原材料可以生产出 1 立方米产品，0.66 立方米的 B 类原材料可以生产出 1 立方米产品，0.72 立方米的 C 类原材料也可以生产出 1 立方米产品。

综述这些条件，B 类原材料性价比最低，应该尽量少订购该类原材料，A 类原材料虽然价格最高，但生产量也较高，可以多购入 A 类原材料，为了企业原材料的订购方式最为经济，应该尽可能多的订购 A 类原材料，适中订购 C 类原材料，尽可能少的订购 B 类原材料。

因此，我们利用附件一中的数据，采用季节性 SARIMAX 模型来建立时间序列，用以模拟未来 24 周的购买原材料的数量。根据图 5.7 所得折线图，利用 SARIMAX 模型非线性拟合出未来 24 周该企业的 A、B、C 原材料购买量的曲线示意图，从而为企业制定未来 24 周每周最经济的原材料订购方案，A 类原材料的非线性拟合折线图如下图 5.8 所示：

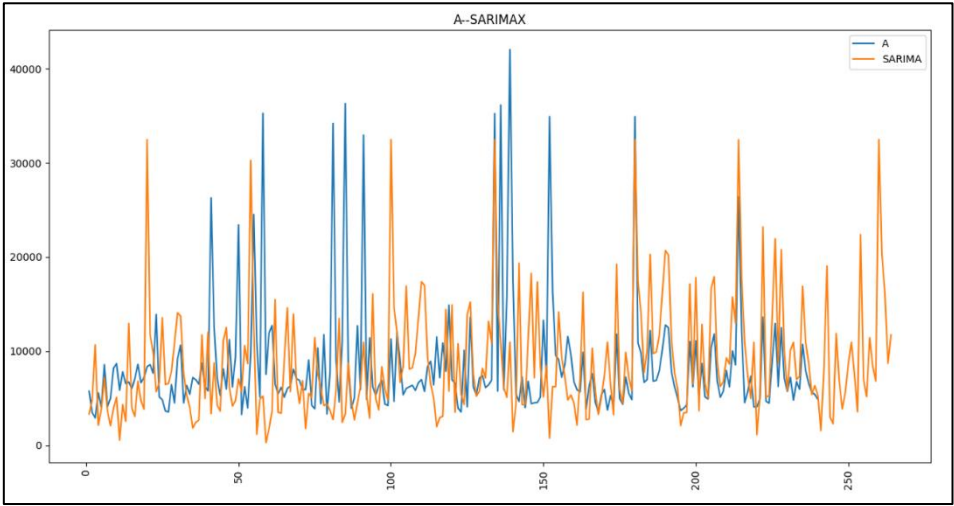


图 5.8 A 类原材料非线性拟合折线图

B 类原材料的非线性拟合折线图如下图 5.9 所示：

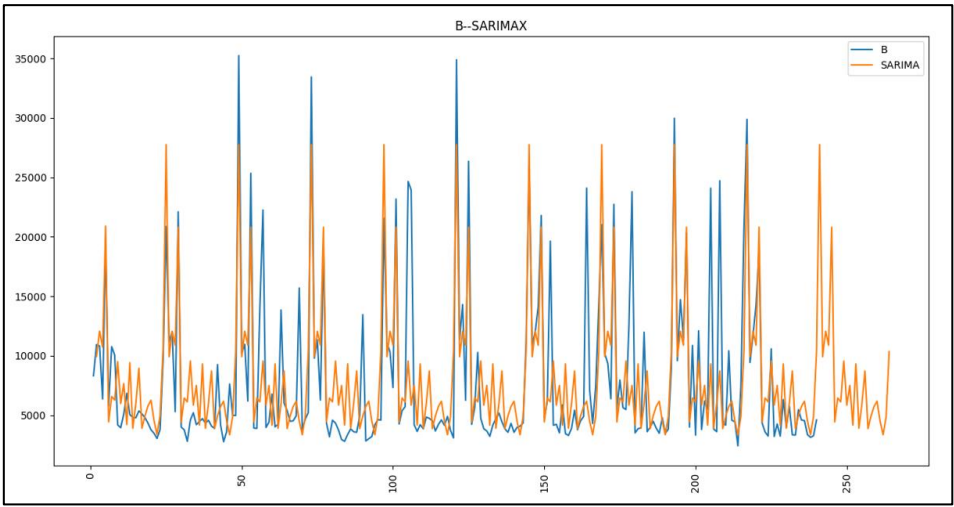


图 5.9 B 类原材料非线性拟合折线图

C 类原材料的非线性拟合折线图如下图 5.10 所示：

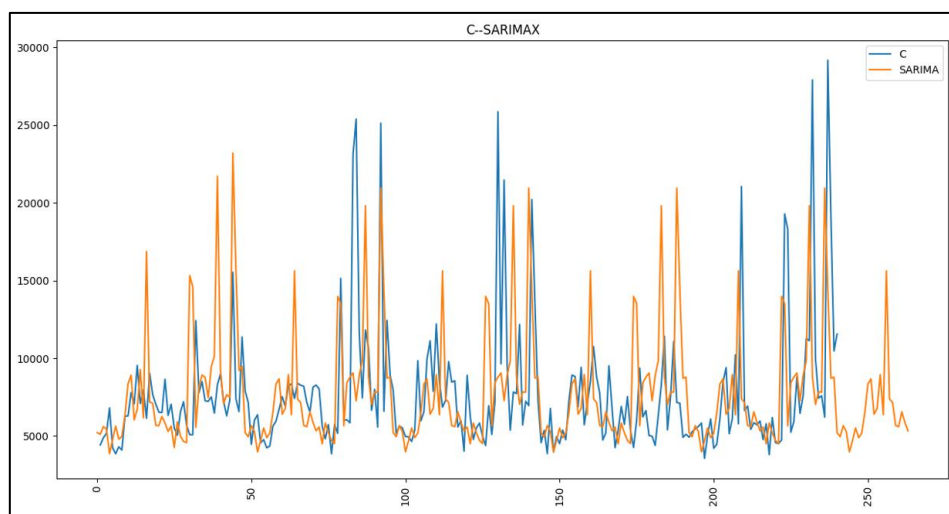


图 5.10 C 类原材料非线性拟合折线图

根据所得 A、B、C 三类原材料的非线性拟合折线图，我们可预测出未来 24 周 A、B、C 三种原材料的应实际购买量，按照每周平均供货量向供货商订购，若订货量过大，则按照供货量最大供货。为了制定最经济的方案，首先选择供货量最大的供应商，若仍需一定数量的原材料，则确定剩余所需货物数量后，在供货商中选择供货能力接近的，最终当供货量多于拟合所得订货量时，这些供货商即是该企业某周所需原材料，当供货量不满足订货量时，我们按照平均供货量来参与计算，将其看作某周的最大供货量。

若想要指定损耗最少的转运方案，我们首先尝试将 $T_1 - T_8$ 共 8 个转运商的 240 周的损耗率取平均值，但是实际效果一般；随后我们选择利用 MDS 多维数据分析来判断 8 位转运商的转运损耗效率，效果较好，我们以此来作判断依据。

由于题目中未说明不能三种原材料同时运输，我们假设所有转运商均可转运三类原材料，并且可以分量转运。为了满足我们损耗量最小的要求，可以首先安排损耗率最低的转运商进行转运，当损耗率最低的转运商满足 6000 立方米的转运最大值时，依次选取损耗率次之的转运商进行转运，以此类推

要想损耗最少，则需要我们先让损耗率最低的转运商进行转运，由于每个转运商最多转运 6000 立方米的原材料，且在通常情况下，一家供应商每周供应的原材料尽量由一家转运商，所以当运输过程中原材料数量过多或者过少时，过多我们则拆分让不同转运商进行转运，过少我们则让该转运商同时转运其他原材料，通过这种方式来进行转运的原材料损耗量最少。

5.3 问题三的分析与求解

1.问题三的分析

该企业为了压缩成本，计划尽量多采购类和尽量少采购 C 类原材料，可以减少转运的成本和存储的成本。对此，我们计划建立多目标优化模型，具体考虑的目标变量有：未来 24 周中每一周三种材料的采购量、三种原材料的价格， $T_1 - T_8$ 共 8 家转运商转运原材料时的转运损耗率以及企业每周最基本的产能。根据这些决策变量和题目所给的约束条件来构建所需多目标优化模型。

2.问题三的求解

由于该企业计划分类别的采购原材料、将转运商的转运损耗率尽量降低，以减少转运及仓储的成本，从而达到该企业压缩生产成本，最大程度的优化自身的生产利润。

若要建立多目标优化模型，则目标函数、决策变量和约束条件必不可少。根据题干要求，我们假设决策变量 x_{ij} 表示第 i 周第 j 类原材料的生产成本， Y_A 表示 A 类原材料的购买量， Y_B 表示 B 类原材料的购买量， Y_C 表示 C 类原材料的购买量， I_i 表示第 i 类原材料能生产出产品的产出比， T_i 表示第 i 家转运商， r_{ij} 表示第 i 家转运商在第 j 周转运的转运损耗率， Q_{ij} 表示第 i 家转运商在第 j 周转运的原材料数量。

建立多目标优化模型所需的目标函数为：

$$\left\{ \begin{array}{l} I_A Y_A + I_B Y_B + I_C Y_C \geq 2.82 * 10^4 \\ \min \sum_{i=1}^{240} \sum_{j=1}^3 x_{ij} \\ \min \sum_{i=1}^8 \sum_{j=1}^{240} Q_{ij} r_{ij} \end{array} \right.$$

约束条件s.t.有：

- ① 每家转运商每周最多运送 6000 立方米的货物；
- ② 尽可能压缩生产成本；
- ③ 三类原材料运输和储存的单位费用相同；

- ④ 设C类原材料价格为 10，则A类原材料价格为 12, B类原材料价格为 11;
- ⑤ 尽量多地采购 A 类和尽量少地采购 C 类原材料;
- ⑥ 该企业对供应商实际提供的原材料总是全部收购。

根据以上所设的目标函数、决策变量和约束条件来建立多目标优化模型，若想要更加好的求出模型的最优解，将多目标优化模型与退火算法相结合可以更好的达到我们的要求。

退火算法来自于固体退火原理，在固体加热时，内部粒子随着温度升高内能增大变为无序的状态，在慢慢冷却时粒子又渐渐有序，最后常温状态下内能最小，达到基态。若与多目标优化模型相结合，则可以把内能模拟成目标函数数值，在算法计算过程中，模拟得出的目标函数数值进行重复迭代，在算法最终结束时可取到近似最优解，在取到近似最优解时，就是在尽可能压缩成本的情况下，确定的新的订购方案及转运方案。

对多目标优化模型中所设置的决策变量，我们设置 C 类原材料的采购单价为 10，因为 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%，所以我们可得 A 类原材料的采购单价为 12，B 类原材料的采购单为 11，并且我们需要计划尽量多地采购 A 类和尽量少地采购 C 类原材料。

经过 Python 程序的编程处理，导出数据后我们得到了关于转运商、原材料类型、原材料采购总价和采购数量所能生产出的企业产品数量，如下图 5.11 所示：

ID	Type	price	mount
345	2	1320	181.8182
364	3	1500	208.3333
30	2	2090	287.8788
211	3	2100	291.6667
246	3	2500	347.2222
307	2	2970	409.0909
329	2	3190	439.3939
193	3	4300	597.2222
267	3	5300	736.1111
305	3	5300	736.1111
150	3	5400	750
351	1	6000	833.3333
130	2	7260	1000

图 5.11 退火算法处理后的数据

在得到所需数据后，我们模拟数据进行算法处理后，即可得到所得的订购方

案，在订购方案确定后，将转运商的转运损耗率与附件中数据相结合得到新的转运方案。新的转运方案满足在减少转运及仓储的成本的同时，转运商的转运损耗率又尽量少，因为三类原材料运输和储存的单位费用相同，若要减少转运成本，我们计划最大程度的降低转运损耗率，新的订购计划又能将收购的原材料的数量把控为刚好满足每周的产能，这样可以在满足每周最低产能的同时，仓库中也不会有较多的冗余的原材料，这样就也可以降低仓储的成本，较为完美的处理了本问题。

5.4 问题四的分析与求解

1.问题四的分析

针对问题四，在该企业已经具备了提高产能的潜力的前提下，我们可以根据附件一和附件二中的原材料供应商的供应量和转运商的转运损耗率的数据，对该企业新的生产能力进行预测模拟，建立 SARIMAX 模型来进行预测，将原本的条件的供应商供应的原材料的数量，确定在未来的 24 周，在原本条件下将该企业的产能提高至更高的效率来生产数量更大的产品。

2.问题四的求解

在问题四的前提下，我们假设该企业为了提高产能不计算成本，计算在理想常态下该企业可生产的最大产品数量。在问题三所建立的基础上我们对 402 家供货商在供货过程中的最大供货量、分析在 240 周供货的供货规律，以及供货商供货的每周均值，综合分析这三个数据我们可得到每家供应商最优的供应货量。

在这种条件下，利用附件一和附件二中的数据，采用 SARIMAX 模型来模拟建立未来 24 周的非线性拟合数据，可以较为准确的模拟未来 24 周的购买三类原材料中不同原材料的数量。

利用 SARIMAX 模型非线性拟合出在新的条件下未来 24 周该企业的 A、B、C 原材料购买量的折线示意图，从而为企业制定未来 24 周每周最经济的原材料订购方案，A 类原材料的非线性拟合折线图如下图 5.12 所示：

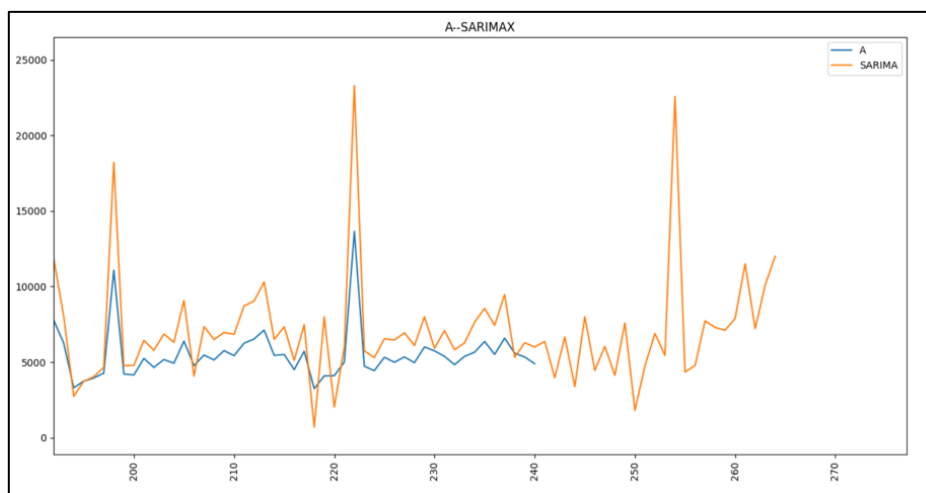


图 5.12 提高产能后 A 类原材料的拟合折线图

B 类原材料的非线性拟合折线图如下图 5.13 所示：

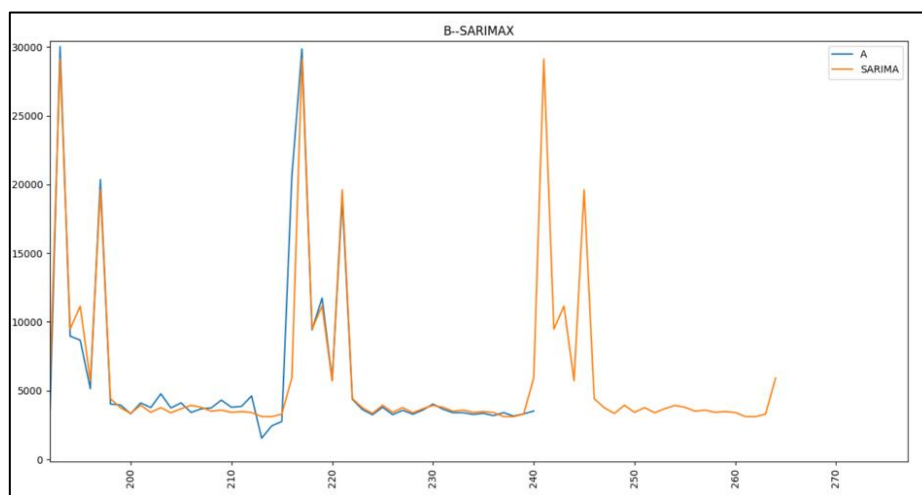


图 5.13 提高产能后 B 类原材料的拟合折线图

C 类原材料的非线性拟合折线图如下图 5.14 所示：

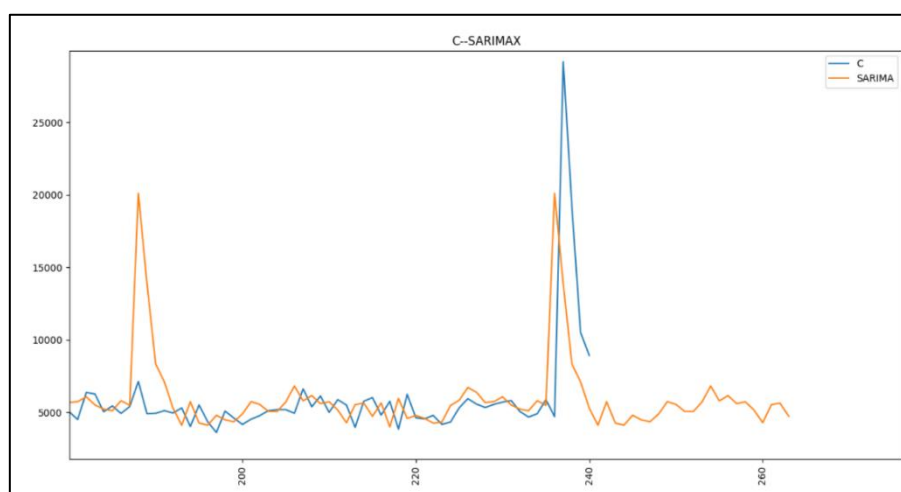


图 5.13 提高产能后 C 类原材料的拟合折线图

六、模型的改进及推广

在解决本道题目时，我们在进行数据处理及建立模型的过程中，由于时间和能力的受限，较为缺乏对问题的分析讨论。虽然在一些方面的也存在对所得到的结果的分析讨论，例如在问题二的解决过程中印证了问题一结果的正确性，但是在问题处理时有很多地方存在着考虑不足或者数据误差的问题，在得到结果时可能会造成结果的失真，但是在最后的实际结果来看，在某些方面也可以较为完美的反映所得到的一般规律，也可从侧面反映出模型的实用价值。

七、参考文献

- [1]姜启源、谢金星、叶俊，《数学模型》，北京，高等教育出版社，2005 年；
- [2]司守奎、孙兆亮，《数学建模算法与应用》（第二版） 北京，国防工业出版社，2016 年；
- [3]张丽霞、侍克斌，《施工网络进度计划的多目标优化》，中国知网，2003 年

论文附录

问题一：

1.Kmeans 聚类与决策变量代码

```
import pandas as pd
import numpy as np

data = =

pd.read_csv(r'C:\Users\233\Desktop\1.csv',encoding='gbk',usecols=[1,2,3,4,5,6])

data = (data - data.mean(axis = 0))/(data.std(axis = 0)) #标准化变换
data.columns=['Z'+i for i in data.columns]

# data.to_excel('tmp/zscoreddata.xls', index = False) #数据写入

# print(data)

#K-Means 聚类算法

from sklearn.cluster import KMeans #导入 K 均值聚类算法
import matplotlib.pyplot as plt

k = 5 #需要进行的聚类类别

#调用 k-means 算法，进行聚类分析

kmodel = KMeans(n_clusters = k, n_jobs = 4,random_state=1) #n_jobs 是并行
数， predict=kmodel.fit(data) #训练模型

#简单打印结果

s = pd.Series(['群 1','群 2','群 3','群 4','群 5'], index=[0,1,2,3,4]) #创建一个序列
s

r1 = pd.Series(kmodel.labels_).value_counts() #统计各个类别的数目
r2 = pd.DataFrame(kmodel.cluster_centers_) #找出聚类中心

r = pd.concat([s,r1,r2], axis = 1) #横向连接（0 是纵向），得到聚类中心对应的
类别下的数目

r.columns = ['聚类名称'] + ['聚类个数'] + list(data.columns) #重命名表头

r.to_csv('聚类.csv',encoding='gbk')

print(kmodel.labels_)
```

```

print(kmodel.cluster_centers_)
data['Type'] = kmodel.labels_+1
data.to_csv(r'OneTypeFinally.csv',encoding='gbk')
#决策变量
New_Data = pd.DataFrame()
New_Data['订货次数'] = Count
New_Data['供货次数'] = Count2
New_Data['订货量'] = Mount
New_Data['供货量'] = Mount2
New_Data['最近一次订货时间间隔/week'] = TimeA
New_Data['供货率'] = Rate

```

问题二 SARIMAX 模型

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

Data = pd.read_csv(r'C:\Users\233\Desktop\ 企 业 分 类 \ 企 业
ABC.csv',encoding='gbk')

Time = [i for i in range(len(Data.iloc[0]))][1:]
Value = Data.iloc[0].values[1:]
y_hat_avg = (pd.DataFrame())
train = pd.DataFrame()
train['Count'] = Value
train['Time'] = Time
test = pd.DataFrame()
test['Count'] = Value[228:]
test['Time'] = Time[228:]
y_hat_avg = pd.DataFrame()

```

```

y_hat_avg['Time'] = [i for i in range(240,504)]    #240, 480

import statsmodels.api as sm

#0,2,1,28

y_hat_avg['Time'] = [i for i in range(1,265)]

fit1 = sm.tsa.statespace.SARIMAX(np.array(Value).astype('float'), order=(0, 0, 0),
seasonal_order=(0, 2, 1,80)).fit()

y_hat_avg['SARIMA'] = fit1.predict(start=240, end=503, dynamic=True)

plt.figure(figsize=(16, 8))

plt.plot(train['Time'],train['Count'], label='A')

# plt.plot(y_hat_avg['Time'],y_hat_avg['SARIMA'], label='SARIMA')

meanPeak =
(26300+23400+24500+35200+34100+36300+32900+35200+36100+42000+34900+3
4900+26400)/13

li=[]

for i in y_hat_avg['SARIMA'].values:

    i = i-2000

    if i > meanPeak:

        i = meanPeak

    if i<0:

        i = (y_hat_avg['SARIMA'].values).mean()

    li.append(i)

print(li[-24:])

print(len(li[-24:]))

plt.plot(y_hat_avg['Time'],li, label='SARIMA')

plt.xticks( rotation=90, fontsize=10)

plt.legend(loc='best')

plt.title('A--SARIMAX')

```

问题三 退火算法

```
from sopt.SGA import SGA

# from math import sin

def func1(x):
    A,B,C = x
    return A+B+C>=28200

if __name__ == '__main__':
    sga = SGA.SGA(func = func1,func_type = 'min',variables_num = 2,
        lower_bound = 0,upper_bound = 2,generations = 20,
        binary_code_length = 10)

    # run SGA

    sga.run()

    # show the SGA optimization result in figure

    sga.save_plot()

    # print the result

    sga.show_result()
```