基于多目标规划与时间序列的原材料订购运输问题求解

摘　要

对于建筑材料领域的实际工程场景而言，生产原材料的订购与运输计划设计对于工业生产成本控制有着重要经济意义。而这一问题需要考虑企业订购量、供应商供货量、转运方损失率等多方角色之间的关系并对此进行优化，是一个组合的优化问题。对于这一类问题，通常采用多种模型组合的方式进行求解。

问题一要求根据 402 家供货商在 240 周内的历史数据进行评价，确定 50 家最重要的供应商。对历史数据构造评价的统计参量作为决策的考虑因素以后，使用 **TOPSIS-熵权法**为系统客观赋权并综合评价，得到排名为前 50 的供货商。

问题二在问题一的基础上提出优化问题，可以将其分解为三步优化。在进行优化之前，由于要安排接下来 24 周的订购计划与运输计划，故对目标供货商和转运商的历史数据应用 **ARIMA 模型**进行建模并预测。第一次优化是订购商数量的优化，将其抽象为 **0-1 规划**求解，得到最少供应商为 23 家。第二次优化是成本最小的优化，但目标成本有购买成本和运输存储成本两部分构成，属于**多目标优化**，可以通过引入不同权重将其转换为单目标优化求解。第三次优化是损失最小的优化，决策变量是每个供货商和每个承运商之间的选择关系，是一个矩阵形式的 **0-1 规划**。对于这一规划问题，除了使用传统的贪心策略求解外，还使用了**遗传算法**进行优化求解，得到结果差异不大证明算法的可行性。

问题三在问题二的基础上增加了一个目标，即尽量多地采购 A 类和尽量少地采购 C 类原材料。可以使用 A 类总量与 C 总量的比值衡量，在问题二模型的基础上扩展为一个三目标规划模型，但沿用其思想将其转化为单目标规划求解。其余操作与问题二相同。

问题四在问题三的基础上改变约束条件，让产能不再是定值。首先，可以利用 23 家商家供货量与库存的换算关系得到库存，然后利用 **ARIMA 模型**对库存序列进行拟合与预测。得到预测结果经线性拟合，证明其与时间存在相当明显的线性关系。故利用此线性方程替换问题三中原有产能值并重新求解。

问题的结果已经保存到附录 A 和 B 中，模型形式简洁，不同模型在不同部分承担不同作用，协同配合，达成求解目标，展示出较为良好的效果。

关键词：TOPSIS-熵权法　ARIMA 模型　多目标优化　0-1 规划模型　遗传算法

一、 问题重述

1.1 问题背景

某建筑和装饰板材企业生产使用的原材料主要是三类木质纤维和其他植物素纤维材料，包括 A,B,C 三种。该企业每年以 48 周为生产周期，并且需要提前制定 24 周的原材料订购和转运计划。以产能需求为目标确定原材料供应商和每周的原材料订购数量，之后确定转运商来承运，由转运商将供应商提供的每周的原材料供货量运输到该企业仓库储存。

已知该企业每周的产能达到$2.82 \times 10^4 m^3$，生产$1\mathrm{m}^3$产品需要消耗 A 类原材料$0.60\ m^3$，或 B 类原材料$0.66\ m^3$，或 C 类原材料$0.72\ m^3$。原材料的特殊性决定了供应商收到订购单后实际供货量可能会多于或少于订货量。正常生产要求该企业要尽可能保持不少于满足两周生产需求的原材料库存量，因此企业总是全部收购供应商提供的原材料。

在实际运输过程中，原材料会因为各种原因损耗，最后转运商实际运送到达企业仓库的原材料数量称为“接收量”，并且接收量不会超过供货量。规定每家转运商的运输能力上限为$6000\ m^3$/周。一家供应商每周提供的原材料尽量由一家转运商运输。

原材料的采购成本是影响企业生产效益的直接因素，实际中 A 类和 B 类原材料分别比 C 类高 20 %和 10 %。但三类原材料运输和储存的单位费用是相同的。

1.2 问题提出

根据附件 1、2 分别给出的该企业近 5 年 402 家供应商订货量及供货量数据和 8 家转运商的运输损耗率数据，研究以下问题：

1. 对附件 1 中 402 家供应商的供货特征进行量化分析，建立反映保障企业生产重要性的数学模型，并由此确定 50 家最重要的供应商。
2. 参考问题 1，该企业为了满足生产需求至少需要选择多少家供应商？对这些供应商制定未来 24 周最经济的原材料订购方案和在此基础上的损耗最少的转运方案。并对订购方案和转运方案的实施效果进行分析。
3. 出于压缩成本，计划尽可能多地采购 A 类和尽可能少地采购 C 类材料，同时希望转运商的转运损耗尽量少。制定并分析新的订购和转运方案并评价。
4. 该企业经过技术改造具备提高产能的潜力。根据供应商和转运商的实际情况，确定每周产能提高多少，给出订购和转运方案并评价。

二、 问题分析

2.1 问题一的分析

问题一需要我们根据附件 1 数据来对这些供货商进行量化分析后选出最重要的 50 家，所以这里是一个评价类问题。评价类问题常用的方法有层次分析法（AHP）、模糊综合判别法、灰色关联分析（GRA）、主成分分析（PCA）和优劣解距离法（TOPSIS）等[1]，但这些方法侧重点各有不同，前两者属于主观赋权评价法，在该问题中由于缺乏运输订购类问题中相关变量的权重分配研究，进行应用时权重会带有一定主观性和随机性，不利于量化。灰色关联分析更多用于分析序列之间的相关性而非重要性，主成分分析更多则用于降维减少变量维度，并且要求构建的主成分有充分的可解释性，在这一问题中不能很好地达到我们的目的，而 TOPSIS 方法是多目标决策分析中一种

常用的有效方法，是一种趋近于理想解的排序法。它根据有限个评价对象与理想化目标的接近程度进行排序，在现有的对象中进行相对优劣的评价[2]，故我们选用 TOPSIS 方法进行求解。

在求解评价类问题时需要合理选取一些评价指标，由于时间序列系不应该直接使用，这里需要结合这一系列时间序列的统计学参量构建评价指标。

2.2 问题二的分析

问题二是在问题一的基础上对供货商的规模进一次压缩，实际上可以抽象为三层优化问题：对供货商数量的最小化、成本数值的最小化和损失的最小化。

第一重优化：我们对这 50 家商家在接下来 24 周内的供货量进行时间序列预测，并按照原材料的消耗率将历史数据和预测数据均折算为产能的损耗，由高到低对每个供货商提供的产能进行累加，总和超过生产需求时即可得到最少需要的供应商数量及名称。另外，也可以将供应商问题抽象为一个 50 维变量的 0-1 规划问题，以产能作为目标函数进行最优化求解。

第二重优化：由于要求每周方案最经济，将购买成本和生产运输成本二者作为决策目标，我们使用多目标规划方法先得到每周需要订购的原材料 ABC 的数量，接着对这些材料按供货商种类进行分配，得到未来 24 周每周最经济的订购方案。

第三重优化：以运输损失作为目标函数，决策变量为 23×8 的矩阵，矩阵元素表示第 i 家供货商是否选择第 j 家运输商。这一方法可以采用贪心策略编程求解但需要事先进行人工的分析；也可以直接采取遗传算法求解。

2.3 问题三的分析

问题三在问题二的基础上增加了一项目标，要求尽量多地采购 A 类和尽量少地采购 C 类原材料，以减少转运及仓储的成本，对我们多目标规划的求解又增加了难度。这里我们用 AC 总量的比值来描述新的目标，这样一来相对于问题二这就是一个三目标规划（购买成本，运输存储成本和比值）。但我们仍然沿用问题二的想法，通过取不同的参数进行组合将其转化为一个单目标问题，只不过这里不同于问题二的是转化的单目标规划不再是一个线性规划而是非线性规划。

其余步骤可以沿用问题二的模型求解。

2.4 问题四的分析

问题四设置产能会随着时间增长，那么可以将动作分解为两步：第一步，预测产能随着时间变化的增量；第二步，改变问题三中产能破坏其定值，然后再次沿用模型迭代求解。

第一步，求解增量，可以分析库存的规律。我们利用目标商家的 240 日历史数据折算叠加求出每日供给的产能，再结合库存关系沿时间轴累加转换为库存容量，利用 ARIMA 模型预测库存的变化规律。求出预测值以后进行函数拟合找到随着时间变化的规律形式，可以先从线性方程开始尝试。

第二步，破坏上限，将模型中产能与库存的约束条件中产能改为随时间变化的量，变化规律遵循第一步的方程。随后即可迭代求解。

三、 模型假设

1、假设初始时期有足够的库存，即至少拥有两周的原材料库存量。
2、可以使用时间序列的预测结果进行合理估计。
3、假设供应商信誉度较高，不会出现供货断层的情况。

四、 符号说明

符号	说明
R	归一化的决策矩阵
r_{ij}	决策矩阵中第 i 个决策项第 j 个指标
w_j	熵权法求解的第 j 个指标的权重系数
R^+, R^-	正理想解和负理想解
D^+, D^-	决策项到正理想解和负理想解的距离
p_{ij}	对于指标 j，第 i 类所占比例
e_j	指标 j 的熵值
n	指标 j 有多少类
y_t	一个时间序列
ε_t	白噪声序列
μ	在问题一表示均值，时间序列中表示常数
σ	标准差
λ	常数系数
α, β	时间序列项的系数
A, B, C	A 类、B 类、C 类的订货总量（C 在问题一也表示相似度）
P	C 类产品的购买单价
M	运输单价
N	存储单价
$e(t)$	折算后第 t 日产能
r_i	第 i 家承运商损失率
T	运输策略的指派 0-1 矩阵

（注：若有变量未列入表内或与表内释义不一致请以原文解释为主）

五、 模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

（1）变量的构造

由于提供的数据均为时序数据，不可能直接使用时间序列的历史数据直接进行决策，需要构造其相应的描述统计量。这里我们使用均值、标准差、最大值、最小值和

极差这五维向量来描述一个变量的时间序列较为合适[1,2]。而我们的序列除了供货量、订货量两个时间序列系以外，我们还构造了供货量与订货量的差额作为缺货量构造统计量，因为平均缺货量越少、波动越小就表示供应越稳定。

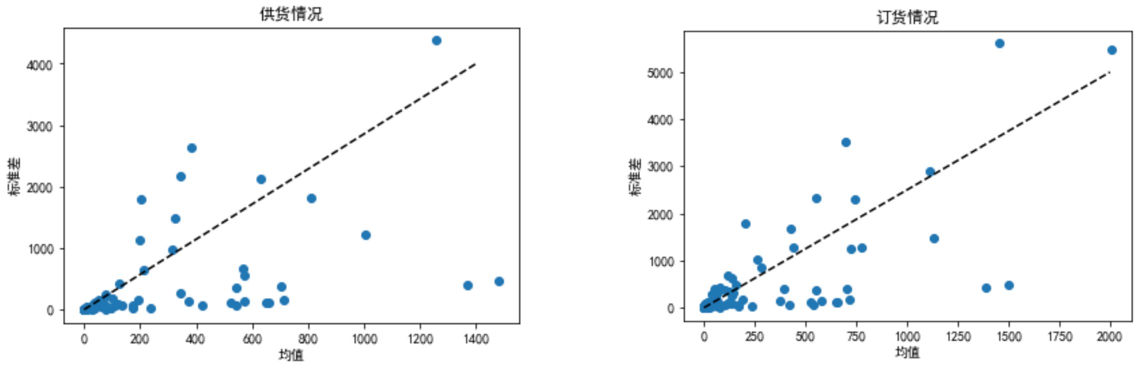

图 1　供货量和订货量的均值方差图像

图 1 为这 402 家企业的供货量与订货量的均值-方差图像，每个点到原点连线的斜率表示变异系数。从图 1 可以看到供货量和订货量的变异系数虽然相对分散，但也是有一定范围的。且二者的分散位置也大致相同，故我们考虑在此基础上构造出一个新的描述统计量差额，即订货量与供货量二者之差，因为该企业对供应商实际提供的原材料总是全部收购且供应商每周提供的原材料也不能影响企业的正常生产，所以这里差额大于 0 和小于 0 均不太合适。由图 2 可以看出，差额的变异系数较为集中，平均差额值波动越小，说明供应商供应能力越稳定，我们主要考虑这些供应商。

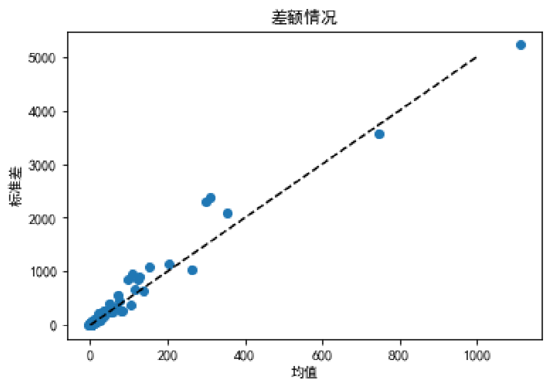

图 2　差额的均值与方差

针对三个时间序列系（每个时间序列系包含 402 家的情况），每个时间序列系构造的向量为：

$$v(x)=[\mu,\sigma,\min(x),\max(x),\max(x)-\min(x)]^T \tag{1}$$

由于多数商家很多情况下数值都是 0，这里我们取非 0 数值的最大值与最小值。针对每家供应商提供的原材料类型（A，B 或 C），我们将其原料消耗情况和购买价格作为两个额外的参考因素，构建出了评价体系的所有变量。

考虑维度有：供货均值、供货标准差、供货最大值、供货最小值、供货极差、订货均值、订货标准差、订货最大值、订货最小值、订货极差、差额均值、差额标准差、差额最大值、差额最小值、差额极差、价格、消耗量，一共 17 个维度用于决策，下面对这一决策矩阵进行求解

（2）TOPSIS 综合评价与熵权分析法

由问题一的分析我们需要构建反映重要性程度的数学模型，并且模型评价的决策指标有多个，是由供货量和订货量的历史数据构建的一系列统计变量，属于多指标综合评价类问题。评价类问题常用的方法有层次分析法（AHP）、模糊综合判别法、灰色关联分析（GRA）、主成分分析（PCA）和优劣解距离法（TOPSIS）等[3]，但这些方法侧重点各有不同，前两者属于主观赋权评价法，在该问题中由于缺乏运输订购类问题中相关变量的权重分配研究，进行应用时权重会带有一定主观性和随机性，不利于量化。灰色关联分析更多用于分析序列之间的相关性而非重要性，主成分分析更多则用于降维减少变量维度，并且要求构建的主成分有充分的可解释性，在这一问题中不能很好地达到我们的目的，而 TOPSIS 方法是多目标决策分析中一种常用的有效方法，是一种趋近于理想解的排序法。它根据有限个评价对象与理想化目标的接近程度进行排序，在现有的对象中进行相对优劣的评价[4]，故我们选用 TOPSIS 方法进行求解。其基本思想为，对原始决策方案进行归一化，然后找出最优方案和最劣方案，对每一个决策计算其到最优方案和最劣方案的欧几里得距离，然后再计算相似度。若方案与最优方案相似度越高则越优先。基本流程如下

Step1：根据归一化得到的决策矩阵（这里我们选取 min-max 归一化）和权重向量构造规范化权重矩阵 R

$$R=[w_j r_{ij}]_{402\times 17} \tag{2}$$

Step2：确定正理想解 R^+和负理想解 R^-,其中J^+,J^-分别表示效益型指标和成本型指标：

$$R^+=\left\{\max_{j\in J^+} r_{ij},\min_{j\in J^-} r_{ij}\right\} \qquad R^-=\left\{\max_{j\in J^+} r_{ij},\min_{j\in J^-} r_{ij}\right\} \tag{3}$$

Step3：计算各评价对象 i（i=1,2,3⋯，402）到正理想解和负理想解的欧几里得距离 D_i：

$$D_i^+=\sqrt{\sum_{j=1}^{n}[r_{ij}-R_j^+]^2} \qquad D_i^-=\sqrt{\sum_{j=1}^{n}[r_{ij}-R_j^-]^2} \tag{4}$$

Step4：计算各评价对象的相似度 C_i

$$C_i=\frac{D_i^-}{D_i^+ + D_i^-} \tag{5}$$

可以看到，相似度是与负理想解与两理想解距离之和的比值，若占比越大则说明离负理想解越远，越优先选择。

Step5：根据 C_i 大小排序可得到结果。

权重向量的构建是 TOPSIS 应用的核心，需要尽可能削弱其主观性。这里我们使用熵权法构建权重向量。熵权法基于信息论，基于信息论的熵值法是根据各指标所含信息有序程度的差异性来确定指标权重的客观赋权方法，仅依赖于数据本身的离散程度[5]。熵用于度量不确定性，指标的离散程度越大，说明不确定性越大，则最终熵值越大，该指标值提供的信息量越多，则权重也相应越大。

根据信息论中对熵的定义[6]，熵值 e 的计算如下所示

$$e_j = -\frac{\sum_{i=1}^{n} p_{ij} \ln p_{ij}}{\ln n} \tag{6}$$

式子(4)中 p_{ij} 代表对于某一个属性 j，第 i 类占样本的比例。n 为属性 j 的取值数量。所以权重系数 w 定义为：

$$w_j = \frac{1 - e_j}{\sum_{i=1}^{m} (1 - e_i)} \tag{7}$$

这样，我们构建了 TOPSIS 综合评价与熵权分析法的综合模型。

5.1.2 模型的求解

首先对决策矩阵进行 min-max 规约化：

$$x_1 = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{8}$$

随后，利用熵权法求解出权重向量。经熵权法求解，该 17 项指标的权重分别为 7.21e-02, 8.19e-02, 8.60e-02, 7.523e-02, 8.72e-02, 6.43e-02, 6.39e-02, 6.05e-02, 7.87e-02, 6.11e-02, 7.12e-02, 6.42e-02,6.09e-02, 1.23e-02, 6.03e-02, 8.18-05, 8.31e-05(7.21e-02 表示 7.21 的 0.01 倍以此类推）。可以看到，权重最高的三项分别为订货量的标准差、订货量最大值和订货量极差。订货量的权重普遍更高，差额权重较低，一个明显的现象是原料的价格和消耗占比最低。

得到正理想解和负理想解后，我们计算欧几里得距离与相似度并排序得到排名，部分求解结果如表 1 所示：

表 1　TOPSIS 法求解结果展示

排序	供应商	综合得分指数	排序	供应商	综合得分指数
1	S201	0.668609675	26	S015	0.086927763
2	S140	0.598096937	27	S247	0.083900219
3	S348	0.566423448	28	S143	0.083861295
4	S151	0.47791268	29	S352	0.083819062
5	S108	0.353884531	30	S157	0.079462789
6	S229	0.350277596	31	S365	0.063387796
7	S126	0.337654902	32	S338	0.061429623
8	S374	0.326004089	33	S096	0.058539968
9	S361	0.319243361	34	S064	0.05747053
10	S139	0.280988732	35	S161	0.052221295
11	S275	0.248423142	36	S097	0.046077171
12	S330	0.232512075	37	S284	0.045928945
13	S307	0.224336505	38	S210	0.044678912
14	S308	0.217695184	39	S074	0.044486821
15	S395	0.186560254	40	S040	0.044292648
16	S306	0.184224303	41	S047	0.043869251
17	S268	0.181187906	42	S335	0.042879996
18	S329	0.153877563	43	S364	0.042741535
19	S160	0.131649011	44	S086	0.042684796
20	S037	0.127219767	45	S346	0.04193686
21	S340	0.120615725	46	S055	0.041894357
22	S282	0.120206216	47	S158	0.03562641
23	S194	0.116720083	48	S246	0.034877318
24	S356	0.105903917	49	S367	0.032174442
25	S131	0.096223838	50	S031	0.029509677

表中选取排名前 50 的供应商得分及名称，即为综合考虑了多种因素后，为了保障企业生产重要性确定的 50 家最重要的供应商。排序以后，我们选择出 S201 等 50 家供货商，这些供货商无论是订货量还是供货量都比较稳定，平均差额基本都不大，波动较小。

5.2 问题二模型的建立与求解

5.2.1 模型的建立

（1）利用 ARIMA 模型进行时间序列的分析与预测

为制定接下来 24 周的计划，需要对接下来 24 周一些变量进行预测。这里我们可以预测 A 类、B 类、C 类的总供货量、总订货量，也可以分别对每家进行预测。但经观察，总供货量与订货量存在若干异常值，不易构建模型。对于这一情况，我们的策略是先进行细粒度预测，即针对每一家供货商进行预测以后进行求和叠加。

时间序列的预测方法有很多，例如广义线性模型(GLM)、自回归移动平均模型(ARMA)等[7]，近年来还诞生了基于机器学习算法（如支持向量机[8]）的时间序列预测方法和基于深度学习的循环神经网络时间序列模型(如 LSTM[9]、GRU[10]等)。由于数

据量只有 240 周，数量并不大，使用神经网络和机器学习容易可能陷入参数训练不充分或者过拟合等问题，故这里我们选用差分自回归移动平均模型(ARIMA)预测。

ARIMA 模型是统计模型中最常见的一种用来进行时间序列预测的模型，只需要考虑内生变量而无需考虑其他外生变量，但要求序列是平稳序列或者差分后是平稳序列[11]。对 402 家企业的供货量和订货量均进行 Dickey-Fuller 平稳性检验[12]，结果显示序列均为平稳序列通过检验，可以使用 ARIMA 模型。

ARIMA 模型包含 3 个部分，即自回归（AR）、差分（I）和移动平均（MA）三个部分。对其每一个部分，都有其递推公式定义。

①自回归模型：

对于 p 阶自回归模型（AR），其递推公式形如：

$$y_t = \mu + \sum_{i=1}^{p} \alpha_i y_{t-i} + \varepsilon_t \tag{9}$$

②移动平均模型：

对于 q 阶移动平均模型（MA），其递推公式形如：

$$u_t = \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \nu \tag{10}$$

③差分模型：

$$\begin{gathered} \nabla^{(1)} y_t = y_t - y_{t-1} \\ \nabla^{(d)} y_t = \nabla^{(d-1)} y_t - \nabla^{(d-1)} y_{t-1} \end{gathered} \tag{11}$$

即由自回归模型阶数 p、差分阶数 d 和移动平均阶数 q 就可以确定 ARIMA 模型的基本形式，我们将其简记为 ARIMA(p,d,q)

$$y_t = \sum_{i=1}^{p} \alpha_i \nabla^{(d)} y_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t \tag{12}$$

对于模型最优参数选择，我们使用 AIC 准则法分析。AIC 准则由 H. Akaike 提出，主要用于时间序列模型的定阶，而 AIC 统计量的定义如下所示[13]，其中 L 表示模型的最大似然函数：

$$AIC = -2\ln L + 2(p+q+d) \tag{13}$$

当两个模型之间存在较大差异时，差异主要体现在似然函数项，当似然函数差异不显著时，模型复杂度则起主要作用，从而参数个数少的模型是较好的选择。一般而言，当模型复杂度提高时，似然函数 L 也会增大，从而使 AIC 变小，但是参数过多

时，根据奥卡姆剃刀原则，模型过于复杂从而 AIC 增大容易造成过拟合现象。AIC 不仅要提高似然项，而且引入了惩罚项，使模型参数尽可能少，有助于降低过拟合的可能性。

（2）利用离散优化模型进行供应商数量的精简

在之前我们选择的 50 家商家中，我们对其每一家商家在接下来 24 周内的供货量进行预测，并按照原料的消耗率将历史数据和预测数据均折算为产能。由于是要求至少多少家，那么考虑产能求和的最大值超过需求即可。而生产商除了每周的生产量还需要保证库存量，所以这一问题可以被抽象为一个选择问题。

对于一个 50 维的向量 v，每一维都只能是 0 或 1，表示是否被选择，与折算后的产能向量 e 相乘可以得到产能总和函数 E

$$E(t)=ve(t)=\begin{bmatrix} v_1 & v_2 & \cdots & v_{50}\end{bmatrix}\begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_{50}(t)\end{bmatrix} \tag{14}$$

现在需要求解的问题即可简化为在保证正常生产需求情况下，所需要的供应商数量尽可能减少，如下所示

$$\begin{aligned} &\min_{v} \sum v \\ &s.t.\begin{cases} \max\limits_{t}\ E(t)\geqslant 28200\times 2 \\ v_i=0,1\ \ \forall i\in[1,50] \end{cases} \end{aligned} \tag{15}$$

5.2.2 模型的求解

问题二我们分为以下三个小问题进行求解，确定最少需要的供应商数量，得到每周最经济的原材料采购方案和对这些采购方案进行损耗率最小的转运方案。这三个小问题后面的均需要基于前面的问题结果来进行分析，故求解步骤如下：

（1）最少供应商的选择

由前面的分析可得，在最少供应商的求解过程中使用到的是 0-1 非线性整数规划模型，依据式（15）我们采用 Dinkelbach 算法，求解得到在满足生产需求的情况下，我们最少需要的供应商为 23 家，其中有 8 家生产 A 类材料，6 家生产 B 类材料，9 家生产 C 类材料。

另外，我们还设计了一种简易算法，按照原材料的消耗率将历史数据和预测数据均折算为产能的损耗，由高到低对每个供货商提供的产能进行累加，总和超过生产需求时即可得到最少需要的供应商数量及名称。

最终结果差异不大，故选择 23 家供应商作为我们的结果。

（2）时间序列预测的分析结果

对这 23 家供应商的订货量和供货量分别利用 AIC 准则选择合适的参数组合进行预测，然后对 8 家转运商的损失率同样利用 AIC 准则进行预测，可以得到其预测结果如下。图 3 为供应商的 24 周订货量和供货量的预测结果和最优模型参数

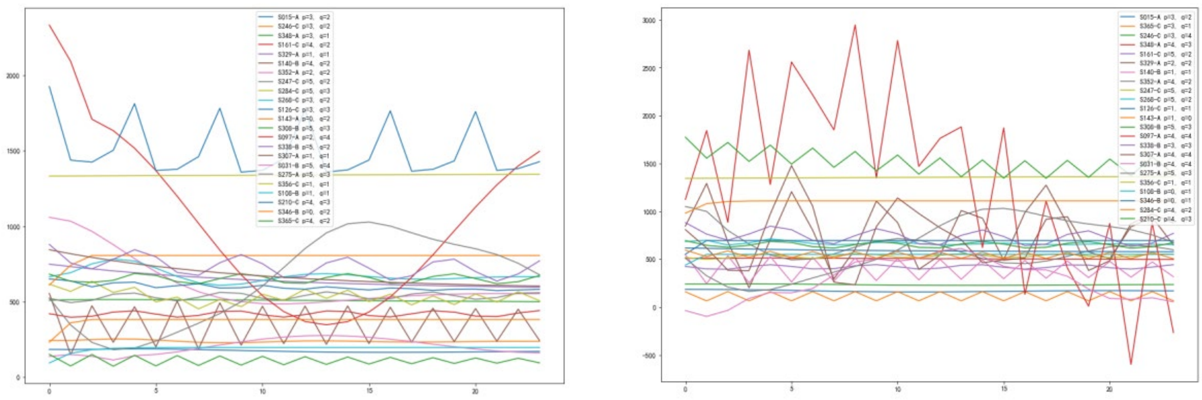

图 3　23 家供应商供货量与企业订货量预测

图 4 表示 8 家转运商的模型拟合效果，贴合较近。红色曲线代表每一家真实的损耗率，蓝色曲线代表我们通过时间序列预测得到的损耗率。分析这些时间序列的残差可以发现，误差项均满足正态分布，说明我们的序列建模是较为成功的。

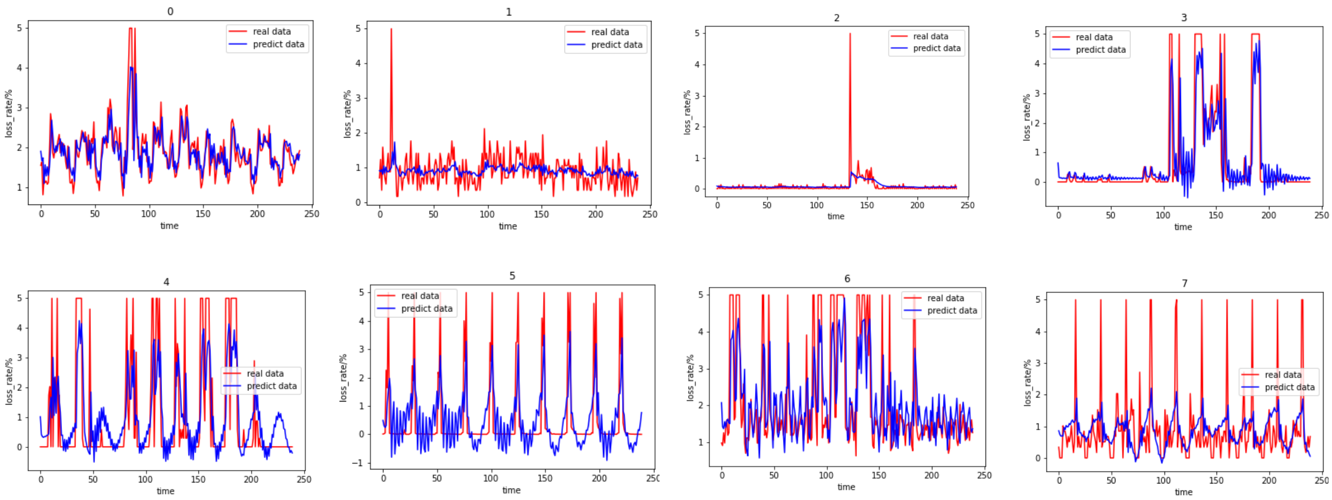

图 4　8 家转运商损耗率的预测值与真实值对比

图 5 表示选择的 23 家供应商对三种原材料在未来 24 周供货量总和的预测和该企业在未来 24 周三种原材料订货量总和的预测。可以看到，C 的水平往往最低，AB 的变化是不规则的。但 A 的变化周期较长，在 24 周内的供货量呈现先降低后增高的趋势，但订货量波动较大。

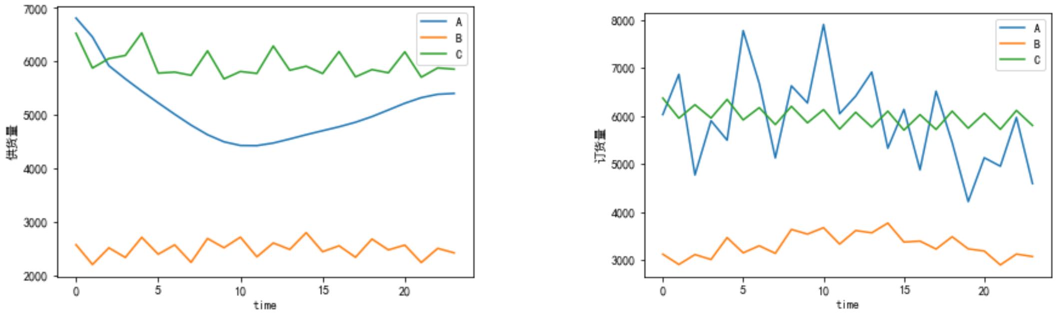

图 5　供应商供货量与企业订货量总和的预测

（3）每周最经济的原材料订购方案的确定

因为 ABC 三种原材料的运输和储存的单位费用相同，故设每立方米原材料的运输费用为 M=1000 元，每立方米原材料的储存费用为 N=500 元，原材料 C 的单价为 P=100 元每立方米，由于 AB 原材料分别比 C 的单价高 20%和 10%，所以 AB 原材料的单价分别为 1.2P=120 元每立方米，1.1P=110 元每立方米。

我们的思路为先对 ABC 三种材料的总量进行求解，得到每一周的总量，然后每一周内再对这些材料，根据 23 家供应商类别的不同再次进行原材料的分配，进而得到每周最经济的订购方案。由历史数据可以得到每一周中三种材料的供应商的供货量标准差 σ_A， σ_B， σ_C，且供货量与订货量的差额是随机误差。

这一问题本质上是一个多目标组合规划的问题。决策变量是 ABC 的三个总量，约束条件一方面是产能与库存的关系，另一方面是决策变量要在我们预测值附近（由于历史数据通过了 K-S 检验具备正态性，我们认为数据应该满足正态性原则）决策目标有两个，一个是购买成本，单价有差异；另一个是运输与存储的成本，单价无差异。但在求解多目标优化过程中我们常常引入不同的权重系数来将其转化为一个单目标优化的问题，这里将其组合为一个优化目标。后续将讨论改变 P、M、N 的值对结果的影响。问题被简化为一个线性规划模型。

设每周用于三种材料的采购，运输和储存的总费用为 W，目的是使 W 最小。以第一周为例，设我们预测得到的三种原材料总和分别为 A_1，B_1，C_1，从而我们列出线性规划模型如下

$$\min\ W=(1.2A+1.1B+C)\cdot P+(A+B+C)\cdot(M+N)$$

$$s.t.\begin{cases}\dfrac{A}{0.6}+\dfrac{B}{0.66}+\dfrac{C}{0.72}\geqslant 2.82\times10^4\times2\\ A(t)-3\sigma_A\leqslant A\leqslant A(t)+3\sigma_A\\ B(t)-3\sigma_B\leqslant B\leqslant B(t)+3\sigma_B\\ C(t)-3\sigma_C\leqslant C\leqslant C(t)+3\sigma_C\end{cases}\tag{16}$$

从而求解得到三种原材料第一周的订购总量，同样的方法可以求出后续 24 周每周各种材料的总订购量，如表 2 所示（仅展示出部分结果，完整结果见支撑材料）

表 2　三种原材料前 5 周的订购总和

原材料种类	第一周 W_1	第二周 W_2	第三周 W_3	第四周 W_4	第五周 W_5
A	6814	6461	5923	5678	5448
B	2582	2211	2523	2343	2719
C	6528	5879	6056	6115	6536

得到每种材料的总量后，我们需要将其和 23 家供应商进行匹配，以确定企业向每家供应商的订购方案。以第一周的 8 家 A 类供应商为例，设 A 类材料的供应商分别为 a_i（i=1,2,⋯,8），供货不准确率分别为 f_i（i=1,2,⋯,8），不准确供货量为 uq，目的是使 uq 绝对值最小。我们由此建立的线性回归模型如下：

$$\min\ uq=\left|\sum_{i=1}^{8}a_i\cdot f_i\right|$$

$$s.t.\begin{cases}\displaystyle\sum_{i=1}^{8}a_i=A\\ 0<a_i\leqslant 0.8A\end{cases}\tag{17}$$

从而求解得到第一周中，企业对所需要的 A 类产品这 8 家供应商的订购量如下表 3 所示：

表 3　第一周 8 家 A 类材料供应商的订货量

供应商	S015	S097	S143	S275	S307	S329	S348	S352
A	184	4650	611	524	845	749	510	1059

按照同样的方法，即可得到第一周中 BC 材料所对应的供应商，企业分别的订货量，和未来 24 周内，企业对每家供应商最经济的原材料订购方案。所有结果均已填充至支撑材料中附件 A 中。

（4）每周损耗最少的转运方案的确定

在前面已确定的每周每家的订购方案后，由于要求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运输，且每家转运商的转运能力为 6000m^3 每周，所以我们就对第一周中企业对 23 家供应商的采购量，进行分块相加，根据贪心策略优先选择损失率最低的承运商，并使其中某几家材料订购量之和最接近 6000，再依据 8 家转运商的损耗率，来对这些材料进行转运分配。下面以第一周为例

第一周 a_1~a_8，b_1~b_6，c_1~c_9，企业对其的采购量分别为[184,4650,611,524,845,749,510,1059,139,95,555,683,880,232,651,421,1926,244,523,666,617,1332,151]，我们进行分块求和的情况为

转运商 T_3：①232+421+510+524+555+617+651+683+749+1059=6000

转运商 T_4：②95+523+611+666+845+1332+1925=5996

转运商 T_6：③184+880+244+4649=5957

转运商 T_5：④151+138=290

此外，我们还可以利用遗传算法进行全局式自动化求解，无需过多分析过程。事实上，这一问题可以抽象为一个 0-1 规划问题，变量在一个 23 行 8 列的矩阵中排布，要求损失最小。约束条件为每一家转运量不超过 6000。依据此，我们建立的模型如下

$$\min\ L=(x_1\ \cdots\ x_{23})\begin{pmatrix}T_{1,1} & \cdots & T_{1,8}\\ \vdots & \ddots & \vdots\\ T_{23,1} & \cdots & T_{23,8}\end{pmatrix}\begin{pmatrix}r_1\\ \vdots\\ r_8\end{pmatrix}$$

$$s.t.\begin{cases}\sum_{j=1}^{8}T_{i,j}=1,\ \forall i\in[1,23]\\ \sum_{i=1}^{23}x_iT_{i,j}\leqslant 6000,\ \forall j\in[1,8]\end{cases} \tag{18}$$

对于这一 0-1 规划问题，传统的算法例如匈牙利法、蒙特卡洛方法已经难以求解因为问题的变量维数太高。随着进化计算与群体智能算法的成熟，有越来越多的智能算法被用于函数优化问题的求解，其中，遗传算法是一类借鉴生物界自然选择和自然遗传机制的随机搜索算法，通过模拟生物的遗传、变异等自然现象搜索函数极值[14]。其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有更好的全局寻优能力；不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向[15,16]。

遗传算法借鉴了生物学的概念，首先需要对问题进行编码，通常是将函数编码为二进制代码以后，随机产生初始种群作为初始解。随后是遗传算法的核心操作之一——选择，通常选择首先要计算出个体的适应度，根据适应度不同来采取不同选择方法进行选择，常用方法有适应度比例法、期望值法、排位次法、轮盘赌法等[17]。

在自然界中，基因的突变与染色体的交叉组合是常见现象，这里也需要在选择以后按照一定的概率发生突变和组合。不断重复上述操作直到收敛，得到的解即最优。遗传算法基本思想如图 6 所示：

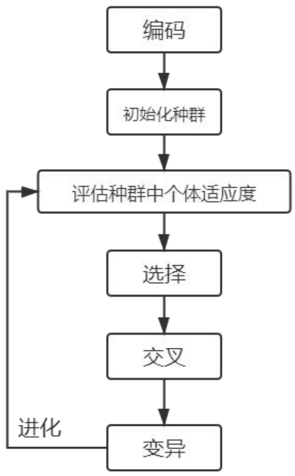

图 6　遗传算法的基本思想

最终通过遗传算法，设置迭代次数为 500 轮，最终结果如图 7 所示。比对承载的分配方案，两种方法求解差异不大，说明遗传算法的使用是可行的。

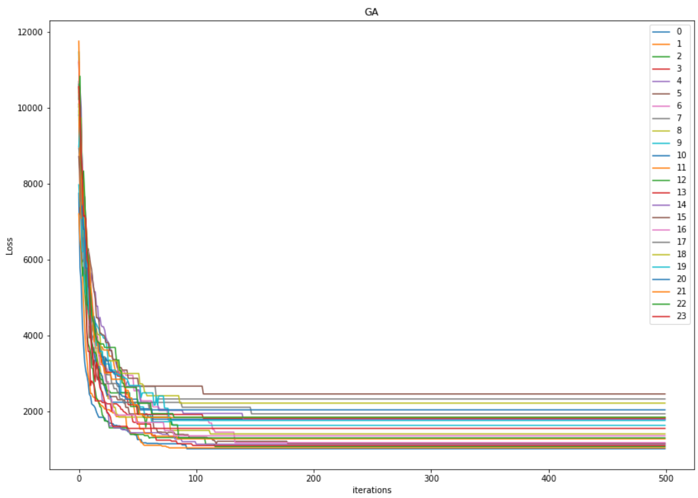

图 7　遗传算法的迭代曲线

对于我们在问题二中建立的最经济采购方案与损耗率最小转运方案，我们通过对三种材料的单位运输成本 M，单位储存成本 N 与单价 P 进行不同的随机取值，利用蒙特卡洛模拟了 100000 次，发现不同取值下最优结果出现的位置都集中在最优解附近，且最优运输方案通过了遗传算法的评估，说明我们建立模型的鲁棒性较好。

5.3 问题三模型的建立与求解

5.3.1 模型的建立

问题三相比于问题二需要多考虑一个因素：尽量多地采购 A 类和尽量少地采购 C 类原材料，以减少转运及仓储的成本。对于每一周，这一变量我们用 A 与 C 的比值来描述。那么这样一来我们的多目标规划模型就有三个目标：

$$\min\ W_1=(1.2A+1.1B+C)\cdot P$$
$$\min\ W_2=(A+B+C)\cdot(M+N)$$
$$\min\ W_3=\lambda\cdot\frac{A}{C}$$
$$s.t.\begin{cases}\dfrac{A}{0.6}+\dfrac{B}{0.66}+\dfrac{C}{0.72}\geqslant 2.82\times10^4\times2\\ A(t)-3\sigma_A\leqslant A\leqslant A(t)+3\sigma_A\\ B(t)-3\sigma_B\leqslant B\leqslant B(t)+3\sigma_B\\ C(t)-3\sigma_C\leqslant C\leqslant C(t)+3\sigma_C\end{cases}\tag{19}$$

5.3.2 模型的求解

对于这一问题我们仍然沿用问题二的思想，将其简化为一个单目标规划问题，此时目标函数为 min W=$(1.2A+1.1B+C)\cdot P+(A+B+C)\cdot(M+N)+\lambda\dfrac{A}{C}$。只是这里已经不再是一个线性规划了，我们考虑使用蒙特卡洛模拟来进行该非线性规划的求

解，得到三种原材料第一周的订购总量，同样的方法可以求出后续 24 周每周各种材料的总订购量，如表 4 所示（仅展示出部分结果，完整结果见支撑材料）

表 4　三种原材料前 5 周的订购总和

原材料种类	第一周 W_1	第二周 W_2	第三周 W_3	第四周 W_4	第五周 W_5
A	9145	10008	9584	9693	9008
B	2571	2210	2523	2328	2733
C	6521	5913	6043	6119	6540

对于最佳订购方案的求解，可以沿用问题二中式(18)，从而求解得到第一周中，企业对所需要的 A 类产品这 8 家供应商的订购量如下表 5 所示

表 5　第一周 8 家 A 类材料供应商的订货量

供应商	S015	S097	S143	S275	S307	S329	S348	S352
A	193	4646	616	532	855	740	510	1053

按照同样的方法，即可得到第一周中企业分别的订货量，和未来 24 周内，企业对每家供应商最经济的原材料订购方案。所有结果均已填充至支撑材料中附件 A 中。

对于最佳转运方案，我们将前面得到结果带入式(18)，设置迭代轮数上限为 200 次，结果已经写入附件 B 中，遗传算法的迭代情况如图 8 所示

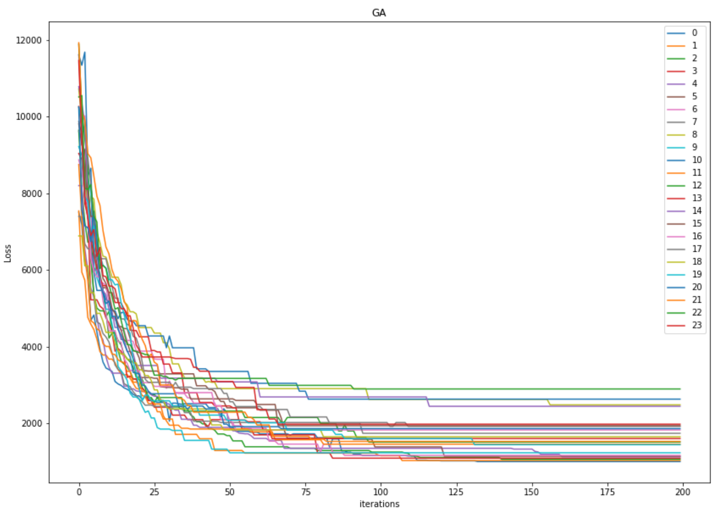

图 8　遗传算法的迭代曲线

对于我们在问题三中建立的最经济采购方案与损耗率最小转运方案，我们通过对 AC 材料的采购占比情况 λ 进行一定正范围内的随机取值，同样使用利用蒙特卡洛模拟，且通过时间序列模拟了 10 周内的情况，发现不同取值下最优结果出现的位置都集中在最优解附近，预测结果与求解最优值间的误差为 6.2%，在可以接受的范围之内，说明我们建立模型的稳定性较好。

5. 4 问题四模型的建立与求解

5. 4. 1 模型的建立

产能提升情况可以通过库存积累与消耗反映。初始状态下已经有 2 周库存，根据历史的 240 周内的供货量数据折算为产能求和求得每天所生产的原料需求，减去每周生产需求后累计求和，从而得到每周的剩余库存。在历史数据中存在库存为负的时间段，表示当周生产实际上是比理想产能低的。

我们对产能序列应用 ARIMA 模型预测，首先进行 ACF-PACF 检验，绘制其 ACF-PACF 图像如图 9 所示

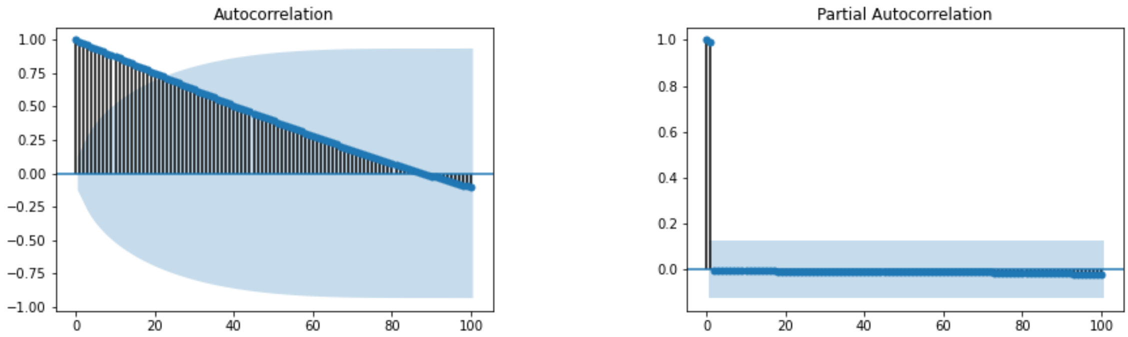

图 9　自相关与偏自相关图像

ACF 图说明的是当前序列值和当前序列过去之间的相关程度；PACF 描述的是残差（在去除滞后已经解释的影响之后）和下一个滞后值之间的相关性。可以看到，在 PACF 图中只有两项不在范围之内，说明我们的模型不应该选择复杂模型。使用 AIC 准则设置 ARIMA 模型参数上限 p 不超过 6，q 不超过 4 进行下一步分析。

我们进行数据拟合与预测的工作，得到结果如图 10：

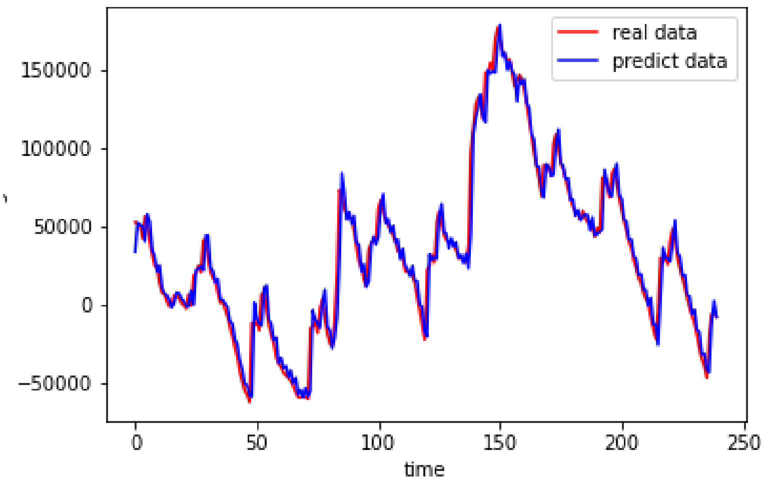

图 10　库存的拟合情况

通过 AIC 准则我们选择的参数组合为(3，0，1)，数据拟合情况非常接近，并且预测趋势确实是上升的。这一上升趋势可以在一定程度上反映产能增长。接着我们分析

其残差图像与残差的统计分布如图 11 所示

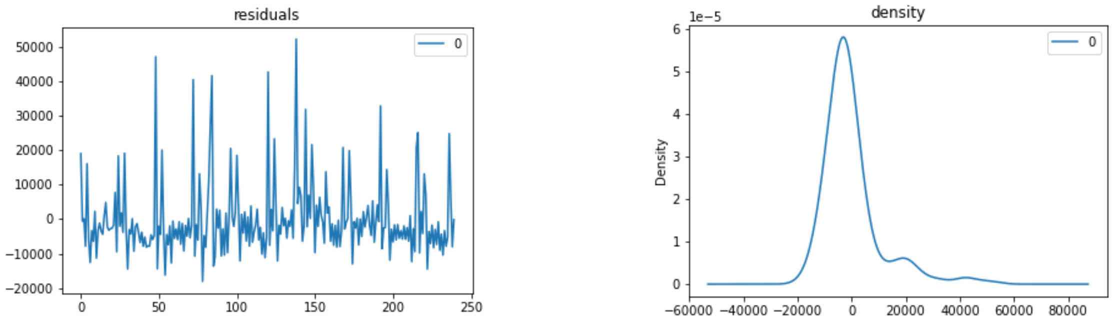

图 11　残差的数值情况与统计分布

5.4.2 模型的求解

可以看到，残差近似为一个均值为 0 的正态分布，说明该序列建模较为成功。我们认为，产能提升与时间呈线性回归模型，即：

$$e(t) = wt + b \tag{20}$$

经拟合，该线性回归方程为$e(t) = 1058.2t + 28200$，R^2=0.9847 呈现高度相关性，说明使用线性模型是很合适的，可以用于预测。在这种情况下我们沿用问题三的模型，但由于产能的提升，每一周模型并不固定，而是呈现线性增长的趋势

$$\min\ W = (1.2A + 1.1B + C)\cdot P + (A + B + C)\cdot(M + N) + \lambda \frac{A}{C}$$

$$s.t.\begin{cases} \dfrac{A}{0.6} + \dfrac{B}{0.66} + \dfrac{C}{0.72} \geqslant (wt + b) \times 2 \\ A(t) - 3\sigma_{A1} \leqslant A \leqslant A(t) + 3\sigma_{A1} \\ B(t) - 3\sigma_{B1} \leqslant B \leqslant B(t) + 3\sigma_{B1} \\ C(t) - 3\sigma_{C1} \leqslant C \leqslant C(t) + 3\sigma_{C1} \end{cases} \tag{21}$$

也就是说，每日的产能线性递增，订货量也会增长。从而求解得到三种原材料第一周的订购总量，同样的方法可以求出后续 24 周每周各种材料的总订购量，如表 6 所示（仅展示出部分结果，完整结果见支撑材料）

表 6　三种原材料前 5 周的订购总和

原材料种类	第一周 W_1	第二周 W_2	第三周 W_3	第四周 W_4	第五周 W_5
A	9573	10680	10396	11201	10853
B	2691	2394	2821	2663	3054
C	6773	6293	6751	6840	7647

最佳订购方案的求解可以沿用问题二中式(16)，从而求解得到第一周中，企业对

所需要的 A 类产品这 8 家供应商的订购量如下表 7 所示：

表 7　第一周 8 家 A 类材料供应商的订货量

供应商	S015	S097	S143	S275	S307	S329	S348	S352
A	205	4735	668	531	932	819	578	1105

按照同样的方法，即可得到第一周中企业分别的订货量，和未来 24 周内，企业对每家供应商最经济的原材料订购方案。所有结果均已填充至支撑材料中附件 A 中。对于最佳运输方案的求解，利用前面求解结果和遗传算法求解，设置迭代上限 200 轮，得到迭代曲线如图 12 所示，转运商选择结果写入附件中

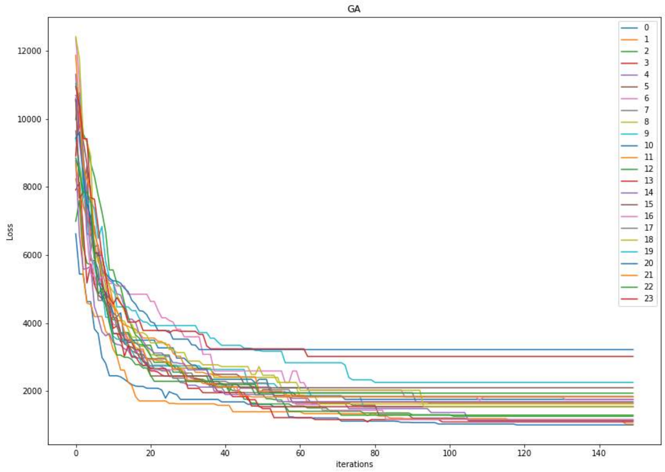

图 12　遗传算法迭代情况

六、 模型的检验与分析

6.1 模型的检验

为衡量模型的准确性，我们针对模型的一些步骤及其结果进行了相关检验：

1. ARIMA 模型平稳性的 Dicky-Fuller 检验：Dicky-Fuller 检验常被用来检验时间序列的平稳性，需要事先判断序列是否平稳。经检验，所有序列均为平稳序列。

2. ARIMA 模型残差的正态性检验：正态性检验的方法有很多，这里采取了 K-S 检验的方式，最终所有序列的残差均通过置信度为 0.05 的正态性检验，接受备择假设认为残差服从正态分布。

3. 遗传算法求解规划问题的结果检验：求解结果经检验均满足约束条件。

6.2 模型的定性分析

我们对每一步骤的模型选择与其它常用模型进行过对比，这里再定性分析一下这些模型的合理性：

1. 评价模型的选取：为了削弱评价权重的主观性这里选取熵权法求解权重，使用 TOPSIS 综合评价方法得到结果较为客观可信。

2. 时间序列模型的选取：这里由于数据量是中规模体量，预测方法没有选用小规模数据常用的拟合与灰色预测，也没有使用大规模数据的机器学习和海量数据的深度学习方法。采用 ARIMA 模型，利用 AIC 准则确定最优参数，模型不仅满足平稳性，残差也是正态的，并且形式简洁防止过拟合。

3. 规划求解算法的选取：使用了传统算法和遗传算法两者进行对比，两种方法的结果差异不大，使得到的结果更具有可信度。

6.3 模型的灵敏度分析

为了进一步衡量模型的鲁棒性与泛化性能，我们做了一些参数相关的灵敏性分析。由于 TOPSIS-熵权分析法中没有自己设定的参数，ARIMA 模型的参数是利用 AIC 准则选取的最优参数组合，这里就主要针对规划问题的权重系数进行了讨论。

我们在问题二中针对多目标规划的两个目标随机选取了 100 组权重参数，但惊奇发现最终的分配方案惊人地一致；同样地，在问题三中针对三个目标选取参数同样不改变分配方案，说明多目标的参数选择对结果是不敏感的。

七、 模型的评价、改进与推广

7.1 模型的优点

综合上述的模型建立与求解，我们认为，我们的模型有如下优点：

1. 利用 TOPSIS-熵权分析的评价模型，权值选取遵从信息论的基本原则，属于客观方法，能够在一定程度上削弱甚至消除主观意义上权重分配造成的偏差，使得评价模型更为客观可信。

2. 利用 ARIMA 模型的时间序列预测，从算法原理上属于时间序列预测中较优的方法。而从参数选择的意义上来看，使用 AIC 准则选择了合适的模型参数，既保障了模型拟合的效果，又降低了参数个数防止过拟合并且模型形式较简洁，训练开销得到了降低。

3. 三层规划组合，层层递进环环相扣，每一层都可以根据上一层的解来设计相应程序自动化求解。

4. 运输方案的求解使用遗传算法，相比传统算法而言提升了速度。并且这一想法可以扩展到任意高维自变量的函数极值求解。同时也使用传统贪心策略进行了对比，优先选择损失率最小的，这样求解的结果与遗传差异不大，进一步验证了结果可靠性。

7.2 模型的缺点

尽管模型已经表现出比较良好的特征，但仍存在一些问题：

1. 利用 ARIMA 模型预测的结果只能考虑内生变量的影响，预测结果是有一定偏差的，并且难以预测发生突发情况导致的订货量、供货量和损失率的变化。

2. 遗传算法在求解过程中可能并不能严格满足约束条件，由于初始化种群的随机性以及变异、重组操作的随机性，得到的结果可能不完全一致。这时需要人工进行一些微调，耗费精力。

7.3 模型的改进

若对模型进行改进，可从以下方面进行改进：

1. 尝试使用不同的群体智能算法对这一约束问题进行求解例如蚁群算法、粒子群算法、萤火虫群算法等。

2. 考虑在时间序列模型预测中加入一些隐藏外生变量或者考虑季节性因素，将其扩展为 SARIMAX 模型进行预测。

7.4 模型的推广

通过这一问题的求解，我们认为这一模型可以在如下方面进行扩展：

1. 未来一段时间的计划制定：指定未来一段时间的计划需要事先估计未来每个时间节点的相关数值，所以可以使用时间序列对历史数据进行建模再迁移到未来时间的预测效应。

2. 供应商选择，订购方案，转运方案的三步策略：使用多种规划方法组合对问题的目标函数、限制条件和决策变量进行建模，能够自动化求解。

3. 遗传算法用于高维自变量对应函数的极值求解：将矩阵每个 0-1 变量都看作自变量的一个维度，然后求解极值。这一做法主要是将二维矩阵展平为一维，这一想法可以迁移到任意高维自变量对应函数的极值求解。

4. 若不限制一家供应商的货物必须由一家转运商承载，那么这一问题就不再是 0-1 规划，可得到的解情况更多，得到的极值也可能更小。仍然可以通过遗传算法等群体智能算法进行优化求解。

八、 参考文献

[1]. 刘雪梅. 上市公司治理结构与盈余管理关系研究[D].华北电力大学（北京）,2008.
[2]. AAAI 2019 : National Conference on Artificial Intelligence. 2019.
[3]. 虞晓芬,傅玳.多指标综合评价方法综述[J].统计与决策,2004(11):119-121.
[4]. Chen C T . Extensions of the TOPSIS for group decision-making under fuzzy environment[J]. Fuzzy Sets & Systems, 2000, 114(1):1-9.
[5]. 程启月.评测指标权重确定的结构熵权法[J].系统工程理论与实践,2010,30(07):1225-1228.
[6]. 谢宏,程浩忠,牛东晓.基于信息熵的粗糙集连续属性离散化算法[J].计算机学报,2005(09):1570-1574.
[7]. 罗凤曼. 时间序列预测模型及其算法研究[D].四川大学,2006.
[8]. 陈顺财. 基于支持向量机的时间序列预测研究[D].兰州理工大学,2008.
[9]. Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[10]. Cho K , Merrienboer B V , Gulcehre C , et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.
[11]. Asteriou D , Hall S G . ARIMA Models and the Box-Jenkins Methodology[M]. 2016.
[12]. Mcleod A I , Hao Y . Dickey-Fuller test[J].
[13].
[14]. 马永杰,云文霞.遗传算法研究进展[J].计算机应用研究,2012,29(04):1201-1206+1210.
[15]. MATLAB 中文论坛. MATLAB 智能算法 30 个案例分析[M]. 北京航空航天大学出 版社, 2010. P17-P22
[16]. 吴渝, 唐红, 刘洪涛. 网络群体智能与突现计算[M]. 科学出版社, 2012.
[17]. 李士勇，李研，林永茂. 智能优化算法与涌现计算[M]. 清华大学出版社，2020，P115-118.

附录

环境：Windows 10+ intel i7+ NVIDIA GTX1650

软件：jupyter notebook(python 3.8.2)+MATLAB 2017a

附录

```
# 问题一
import pandas as pd
data_dinghuo=pd.read_excel("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）')
data_gonghuo=pd.read_excel("附件1 近5年402家供应商的相关数据.xlsx",sheet_name='供应商的供货量（m³）')
import numpy as np
newdata=data_dinghuo[['供应商 ID','材料分类']]
newdata['供货均值']=data_gonghuo.loc[:,'W001':'W240'].T.mean()
newdata['供货标准差']=data_gonghuo.loc[:,'W001':'W240'].T.std()
newdata['供货最大值']=data_gonghuo.loc[:,'W001':'W240'].T.max()
zuixiao=[]
for s in data_gonghuo['供应商 ID']:
    w=np.array(data_gonghuo[data_gonghuo['供应商 ID']==s].loc[:,'W001':'W240'])
    zuixiao.append(np.min(w[np.nonzero(w)]))
newdata['供货最小值']=zuixiao
newdata['供货极差']=newdata['供货最大值']-newdata['供货最小值']
newdata['订货均值']=data_dinghuo.loc[:,'W001':'W240'].T.mean()
newdata['订货标准差']=data_dinghuo.loc[:,'W001':'W240'].T.std()
newdata['订货最大值']=data_dinghuo.loc[:,'W001':'W240'].T.max()
zuixiao=[]
for s in data_dinghuo['供应商 ID']:
    w=np.array(data_dinghuo[data_dinghuo['供应商 ID']==s].loc[:,'W001':'W240'])
    zuixiao.append(np.min(w[np.nonzero(w)]))
newdata['订货最小值']=zuixiao
newdata['订货极差']=newdata['订货最大值']-newdata['订货最小值']
data_delta=data_dinghuo.loc[:,'W001':'W240']-data_gonghuo.loc[:,'W001':'W240']
data_delta['供应商 ID']=data_dinghuo['供应商 ID']
newdata['差额均值']=data_delta.loc[:,'W001':'W240'].T.mean()
newdata['差额标准差']=data_delta.loc[:,'W001':'W240'].T.std()
newdata['差额最大值']=data_delta.loc[:,'W001':'W240'].T.max()
zuixiao=[]
for s in data_delta['供应商 ID']:
    w=np.array(data_delta[data_delta['供应商 ID']==s].loc[:,'W001':'W240'])
    zuixiao.append(np.min(w[np.nonzero(w)]))
newdata['差额最小值']=zuixiao
newdata['差额极差']=newdata['差额最大值']-newdata['差额最小值']
A=[]
B=[]
```

```
for i in newdata['供应商 ID']:
    if newdata[newdata['供应商 ID']==i]['材料分类'].values[0]=='A':
        A.append(1.2)
        B.append(0.6)

    elif newdata[newdata['供应商 ID']==i]['材料分类'].values[0]=='B':
        A.append(1.1)
        B.append(0.66)

    else:
        A.append(1)
        B.append(0.72)

newdata['价格']=A
newdata['消耗量']=B
def entropyWeight(data):
    data = np.array(data)
    # 归一化
    P = data / data.sum(axis=0)
    # 计算熵值
    E = np.nansum(-P * np.log(P) / np.log(len(data)), axis=0)
    # 计算权系数
    return (1 - E) / (1 - E).sum()
newdata=newdata.replace("A",1)
newdata=newdata.replace("B",2)
newdata=newdata.replace("C",3)

newdata=newdata.loc[:,'供货均值':'消耗量']
def topsis(data, weight=None):
    # 归一化
    data = data / np.sqrt((data ** 2).sum())

    # 最优最劣方案
    Z = pd.DataFrame([data.min(), data.max()], index=['负理想解', '正理想解'])

    # 距离
    weight = entropyWeight(data) if weight is None else np.array(weight)
    Result = data.copy()
    Result['正理想解'] = np.sqrt(((data - Z.loc['正理想解']) ** 2 * weight).sum(axis=1))
    Result['负理想解'] = np.sqrt(((data - Z.loc['负理想解']) ** 2 * weight).sum(axis=1))

    # 综合得分指数
    Result['综合得分指数'] = Result['负理想解'] / (Result['负理想解'] + Result['正理想
解'])
```

```
    Result['排序'] = Result.rank(ascending=False)['综合得分指数']

    return Result, Z, weight

out=topsis(newdata)
topsis=pd.DataFrame(out[0])
topsis['供应商 ID']=data_gonghuo['供应商 ID']
topsis.to_excel("topsis.xlsx")
```

```
#问题二 时间序列预测
import pandas as pd
data_dinghuo=pd.read_excel("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）')
topsis=['S015','S031','S037','S040','S047','S055','S064','S074','S086','S096','S097','S108','S126','S131','S139','S140','S143','S151',
'S157','S158','S160','S161','S194','S201','S210','S229','S246','S247','S268','S275','S282','S284','S306','S307','S308','S329','S330','S335',
'S338','S340','S346','S348','S352','S356','S361','S364','S365','S367','S374','S395']
data=pd.DataFrame([])
for s in topsis:
    data=pd.concat([data,data_dinghuo[data_dinghuo['供应商 ID']==s]])
data=data.reset_index(drop=True)
testseries=data.loc[0,'W001':'W240'].T.values
zhesuan=[]
#data=data_dinghuo
import numpy as np
for s in data['供应商 ID']:
    if data['材料分类'][data['供应商 ID']==s].values=='A':
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.6).reshape(240,))
    elif data['材料分类'][data['供应商 ID']==s].values=='B':
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.66).reshape(240,))
    else:
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.72).reshape(240,))
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
from statsmodels.api import tsa
import matplotlib.pyplot as plt
predict_gonghuo=[]
List=[1,47,27,42,22,36,16,43,28,32,29,13,17,35,11,39,34,2,30,44,12,25,41]
for i in List:
```

```
    y=data.loc[i,'W001':'W240'].T.values
    #x=np.arange(402)
    warnings.filterwarnings("ignore")
    print(i)
    cat=tsa.arma_order_select_ic(y,max_ar=5,max_ma=4,ic='aic')['aic_min_order']    #
AIC 定阶
    print(cat)
    try:
        md=tsa.ARMA(y,cat).fit()
        #x_predict=np.arange(240,264)
        dnext=md.predict(240,263)
        predict_gonghuo.append(dnext)
        print(dnext)
    except:
        pass
#predict_gonghuo=np.array(predict_gonghuo)
predict_gonghuo
```

```
# 问题二 指派问题的遗传算法
from sko.GA import GA
import numpy as np
import pandas as pd
history=[]
def loss(x):
    s=0
    for i in range(8):
        s+= sum(predict[np.where(x==i)])*loss_rate[i]
    return s
for j in range(24):
    try:
        predict=np.array(pd.read_excel("23 家每周的分配量-3.xlsx")['第%d 周
'%(1+j)])
    except:
        predict=np.array(pd.read_excel("23 家每周的分配量-3.xlsx")['第 0%d 周
'%(1+j)])
    loss_rate=np.array(pd.read_excel("损耗率预测.xlsx")[j])
    ucons=[]
    '''
    cons=[]
    for i in range(23):
        cons.append(lambda                                                  x:
(x[i*8]+x[i*8+1]+x[i*8+2]+x[i*8+3]+x[i*8+4]+x[i*8+5]+x[i*8+6]+x[i*8+7])-1)
    '''


    for i in range(8):
```

```
        ucons.append(lambda x: sum(predict[np.where(x==i)])-6000)
    ga = GA(func=loss, n_dim=23, max_iter=150, constraint_ueq=ucons, lb=[0]*23, ub=[7]*23, precision=[1]*23)
    general_best,func_general_best=ga.fit()
    #solve=pd.DataFrame(general_best)
    print(general_best)
    df_GA=pd.DataFrame(general_best,columns=None)
    FitV_history_GA32 = pd.DataFrame(ga.all_history_Y)
    plt_min1 = FitV_history_GA32.min(axis=1)
history.append(plt_min1)
#solve.to_excel("指派/week_%d.xlsx"%j)
#print(np.matmul(predict,general_best))
plt.figure(figsize=(14,10))
for i in range(24):
    plt.plot(history[i].index, history[i], label='max')
plt.legend(np.arange(24))
plt.xlabel("iterations")
plt.ylabel("Loss")
plt.title("GA")
plt.show()
```

问题四 产能预测

```
import pandas as pd
data=pd.read_excel("附件 1 近 5 年 402 家供应商的相关数据.xlsx",sheet_name='供应商的供货量（m³）')
zhesuan=[]
#data=data_dinghuo
import numpy as np
for s in data['供应商 ID']:
    if data['材料分类'][data['供应商 ID']==s].values=='A':
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.6).reshape(240,))
    elif data['材料分类'][data['供应商 ID']==s].values=='B':
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.66).reshape(240,))
    else:
        zhesuan.append(np.array(data.loc[data['供应商 ID']==s,'W001':'W240']/0.72).reshape(240,))
zhesuanhou=pd.DataFrame(zhesuan)
channeng=zhesuanhou.sum()
kucun=leiji.cumsum()+28200*2
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
```

```
from statsmodels.api import tsa
import matplotlib.pyplot as plt
x=np.arange(240)
y=kucun
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
lag_acf = acf(y, nlags=20)
lag_pacf = pacf(y, nlags=20, method='ols')
#fig, axes = plt.subplots(1,2, figsize=(20,5))
plot_acf(x, lags=100)
plt.savefig("自相关.png")
plt.show()
plot_pacf(x, lags=100)
plt.savefig("偏自相关.png")
plt.show()
cat=tsa.arma_order_select_ic(y,max_ar=5,max_ma=3,ic='aic')['aic_min_order']    # AIC
定阶
#原图像在 3.x 附近波动说明用 ARMA 模型适合一点
md=tsa.ARMA(y,cat).fit()
print(md.summary())
yhat=md.predict()
err=np.sqrt(((y-yhat)**2).mean())
print(cat)
print(err)
#画图比较
plt.xlabel('time')
plt.ylabel('storage')
plt.plot(x,y,'r-',x,yhat,'b-')
plt.legend(('real data','predict data'))
plt.savefig("库存.png")
plt.show()
#残差分析
residuals=pd.DataFrame(md.resid)
residuals.plot(title="residuals")
plt.savefig("残差图像.png")
plt.show()
residuals.plot(kind='kde',title="density")
plt.savefig("密度分布.png")
plt.show()
#预测值
x_predict=np.arange(240,264)
dnext=md.predict(240,263)
plt.plot(x_predict,dnext)
plt.xlabel('time')
```

```
plt.ylabel('storage')
plt.savefig('预测图象.png')
plt.show()
```