赛区评阅编号（由赛区组委会填写）：

2023 高教社杯全国大学生数学建模竞赛

承　诺　书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称“竞赛章程和参赛规则”，可从 http://www.mcm.edu.cn 下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、“贴吧”、QQ 群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）： C

我们的报名参赛队号（12 位数字全国统一编号）： 0000

参赛学校（完整的学校全称，不含院系名）： 电子科技大学

参赛队员 (打印并签名)：1.

2.

3.

指导教师或指导教师组负责人 (打印并签名)：

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期： 2023 年 07 月 14 日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号:　　　　　　　　　　　　　　全国评阅编号:
（由赛区填写）　　　　　　　　　　　　　（全国组委会填写）

2023 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）:

评阅人						
备注						

送全国评阅统一编号:
（赛区组委会填写）

(请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。)

建材市场供应、转运、生产体系研究

摘要

建筑和装饰板材的企业往往有多家材料供应商，企业、供应商、运输商共同构成了生产体系。本文主要给出了基于 TOPSIS 算法的**供货商评价体系**，利用 ARIMA 模型预测供货量，然后通过求解规划问题确定了生产所需的**最少供应商**，然后改变约束条件，得到了不同约束条件下最优的**订货方案和转运方案**。最后将提高产能转化为**多重背包问题**并使用粒子群算法求解，在不计成本的情况下得到了**企业最大产量**。

针对问题一，为了建立保障企业生产的供应商评价体系，我们从**供货能力**、**供货平稳度**等角度出发，建立了五个指标以综合反映供应商各方面能力。然后，我们利用**TOPSIS（优劣解距离法）**算法计算各供应商的综合分数并得到前 50 名，其中各指标的权重由**变异系数法**得到。排名显示，前 5 名供应商都具有较好的供货能力、稳定的供货水平和与订货量匹配的能力，而那些**供货量水平不稳定**的供货商更容易排名靠后。

针对问题二，首先需要确定保证生产的最少的供应商数量，然后再制定订货方案以最小化生产成本，最后根据生产方案制定最优转运方案。我们首先通过 **ARIMA** 模型预测供货商供货量，然后构建 **0-1 规划模型**，确定约束条件为供应商最少、成本最小，使用**分支定界法**来求解。结果显示最少需要 14 家供应商，最优方案的平均每周成本为 10 万 1813，大部分是存储成本，平均运输损耗为 46.76 单位产品。

针对问题三，需要在问题二的模型里加上采购 **A 材料最少，C 材料最多**的约束。我们利用材料和产品的比率关系将多目标优化转化成**单目标优化问题**，然后应用问题二中模型求解，结果显示重新计算后 A 材料平均每周采购占比 64.25%，C 材料平均每周采购占比 3.38%，存储成本因为 A 材料过多有明显上升，证明在有材料偏好的情况下企业不能实现成本最小化。

针对问题四，要求在不计成本的情况下求出最大产能提升空间，根据问题二，三的数据，在每周供货量和转运能力确定的情况下，我们将问题转化为 0-1 多重背包问题，然后建立了**二进制粒子群算法**以求每周最大材料数，再通过求解**线性规划方程**解出产能的可行域，最后通过二分法搜索最大产能。结果显示，第 24 周库存与产能的关系为 $E_{24} = -22\theta + 96.2754$，产能最大为 4.1859 万立方米。

对问题四中的灵敏度分析显示，粒子群算法有较好的鲁棒性，参数不会明显影响每周的最佳转运方案。

关键字： **TOPSIS**，**ARIMA**，**规划问题**，**分支定界法**, **粒子群算法**

一、 问题背景

现代企业为保障生产体系正常运转，往往会选择多家供应商。每家供应商的都有其特点，它们的供货种类，供货能力，能够稳定供货的时间等并不一定相同。企业在实际生产时会考虑这些特点并给各供应商按照重要性排名，然后订货。生产体系的另一个关键要素是转运的成本，企业必须合理设计转运方案以减少生产成本，而当企业想要提高产能时，它也必须综合考虑供应商供货特点和转运成本。

二、 问题分析

2.1 问题一

题目要求建立一个综合评价供应商的体系，这个体系应该考虑供货商的各个方面以保障企业的生产能力。企业在寻找供货商时，主要关注某些重要指标如供货商供货能力，供货量是否总是与订货量一致，供货能力是否可以长期保持以及企业对于供货商的偏好程度。我们主要建立了五个指标评估供应商：供货种类，最大供货能力，供货平均值，供货平稳度，商家偏好程度，最大供货时长。在建立指标后，我们使用 TOPSIS 算法对供货商进行评估和排名，其中指标的权重可以由变异系数法计算。

2.2 问题二，三

问题二需要基于问题一里的指标和评价模型，建立订货模型以计算至少要多少家供应商才能保持生产，然后考虑材料，库存成本给出订货方案，根据订货方案确定转运方案以减小转运损耗。问题三在问题二的基础上要求尽量多地采购 A 类和尽量少地采购 C 类原材料。首先需要预测供货商的供货量，转运商的损耗率，然后根据题目不同的要求确定不同的约束条件，考虑到一家供货商只能选择一家转运商，一家供货商只有供货与不供货两种状态，所以这是一系列 0-1 规划问题，可以使用分支定界法求解。

2.3 问题四

在忽略材料和库存成本的情况下，企业的产能实际上取决于转运损耗。在确定的一周，八家转运商能够转运的总量是一定的供货商的供货量是确定的，每周的最多的购买材料方案也是确定的，这样 24 周的最大产能都可以求出，对于这个产能序列可以利用倒推最大产能。为最大化每周的产能，需要尽可能多的利用转运商转运能力，在确定转运能力和确定供货量和货物价值的条件下，我们将其转化为多重背包问题。这是一个 np

难问题，可以考虑使用二进制粒子群算法求解。得到每周最大采购方案之后将其转化为规划问题解出最大产能的满足约束的解集，在解集里使用二分法搜索解。

三、模型假设

为简化模型，分析问题本质，我们提出如下假设并给出其的依据：

1. 在确定订货方案时，假设企业在第一周已经有了满足两周生产需求的库存。
   依据: 题目要求制定一个 24 周的计划，那么合理假设在做计划之前，库存量能够保障生产需求，这与订货计划的要求一致。
2. 假设 C 材料的单价，各材料的运输成本、储存成本是一样的。
   依据： 这三个价格是可以看作相互独立的，为方便模型计算取了同样的值，这并不影响模型的效果，而模型也仍然可以处理三个价格都不相同的情况。

四、符号说明

表 1 符号说明

符号	意义
M	供货商最大供货能力
V	供货商供货平稳度
L	供货商最大供货时长
T	每周某种材料购买总量
Q	材料转换为产品的比率
P	供货商供给量
α	转运损耗率
E	库存

五、 模型建立

5.1 问题一：基于供货指标和 TOPSIS 算法的供应商评价模型

5.1.1 供应商指标构建

供应商的地位高低取决于多种因素，其中比较重要的指标有：供应的材料，供货能力，供货平稳度，商家偏好程度和能稳定供货的最大持续时间。我们通过题目所给数据量化了这些指标，然后使用模型计算出每个供应商的重要性，每个供应商的重要指标结构如图：

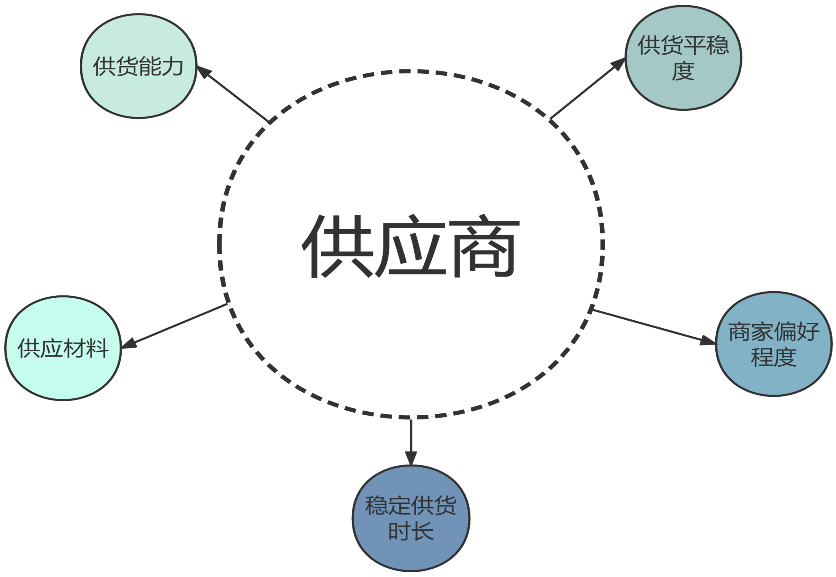


图 1　供应商重要指标

供货种类　供货商只供应 A，B，C 三种材料，随时间变化，商家采购每种材料的意愿是不同的。那些生产商家偏好的材料的供货商的地位应该更高，我们通过商家对不同材料的采购量和所有材料采购量，计算每一类材料在后续模型计算中的权重，这里以 A 材料为例，定义 T_{Aj} 每一周材料为 A 的总量

$$T_{Aj} = \sum_{i=1}^{402} x_{ij}, x \in A \tag{1}$$

然后比较 T_{Aj}，T_{Bj}，T_{Cj}，这个值最大的材料在当周赋权重为 2，第二大的材料赋为 1，最小的赋为 0，将这个权重记为 z。A 材料全局权重 TW 为

$$TW_A = \frac{\sum_{j=1}^{240} z_{Aj}}{\sum_{j=1}^{240}(z_{Aj} + z_{Bj} + z_{Cj})} = \frac{\sum_{j=1}^{240} z_{Aj}}{720} \tag{2}$$

这里在得到每周各材料的权重 z 后，将它们加起来并除以总的权重之和，即 $(0+1+2)\times 240$。计算得 $TW_A = 0.4542, TW_B = 0.3861, TW_C = 0.1597$

最大供货能力　供货商的供货能力被定义为供货商单次最大供货量，这个指标代表了供货上限，供货能力越强，商家更有可能把大额订单交给它以防止供货能力不足带来交割意外，以及减少多个供货商带来的转运损耗。为后续计算方便，将供货按生产产品的比率等价替换为生产产品数。假设第 i 家供应商在第 j 周的供应量为 x_{ij}，订货量为 y_{ij}，定义供货能力为 M，

$$M = \max x_{ij} \tag{3}$$

供货平均值　供货平均值反映了一个供货商的平均供货能力，供货商的供货能力并不是一直保持一致，有可能会下滑。供货平均值描述了供货商在一个长期的时间范围内的供货能力。我们只选取了有订单时的供货值计算平均值。

供货平稳度　供货商在供货中常常存在供货量与订货量不相同的情况，供货量少于订货量会阻碍企业生产，而多于订货量时企业又不得不收购多余部分而受到损失。与平均供货量不同，这个指标主要描述供货时供货匹配订货量的能力。所以企业期望与供货能力稳定，不偏离订货量过多的供货商合作。由此，我们定义供货商的供货平稳度 V 为

$$V_i = \sum_{j=1}^{240} \frac{\left(x_{ij} - y_{ij}\right)^2}{y_{ij}} / 240 \tag{4}$$

V_i 表示第 i 家供货商的供货平稳度，这里利用残差的平均数计算。

商家偏好程度　由于各种原因，在各指标相同的情况下商家会更倾向于某些供货商，这可能与供货商品牌，双边关系，转运成本甚至季节有关。我们根据题目所给数据去刻画这种倾向性，对于某供货商，主要根据商家在该供货商的订货次数计算。

最大供货时长　供货商的供给能力可能只能持续一段时间，商家应该更愿意和那些能长期保持供货能力的供货商合作，我们使用最大供货时长 L 来之间刻画这种意愿。

$$L = \max len\left(a_{sub}\right) \tag{5}$$

$len\left(a_{sub}\right)$ 表示子序列的长度。

部分计算结果如下表

表 2 部分供应商指标

供应商 ID	供货量最大值	误差平方值	稳定供货的最大持续时间	订货次数	供应材料
S001	6	0.297	88	4	B
S002	67	0.172	95	4	A
S003	314	0.084	194	11	C
...					
S402	20	0.243	64	5	B

5.1.2 TOPSIS 模型

TOPSIS，优劣解距离法，全称为"Technique for Order Preference by Similarity to Ideal Solution"，是一种多准则决策分析方法。该方法通过计算解决方案与理想解决方案和负理想解决方案的相似性，来排列解决方案的优先级。[1] 这里将原算法将“方案”的概念替换为“供应商”，供应商被视为一个五维向量。首先，利用 402 家供应商和其五个指标组成的矩阵 $G(g_{ij})_{402*5}$，为使指标保持在同一尺度和量纲，进行标准化处理

$$h_{ij} = \frac{g_{ij}}{\sqrt{g_{1j}^2 + g_{2j}^2 + \cdots + g_{402j}^2}} \tag{6}$$

h_{ij} 是处理后的矩阵 H 的元素。

然后，构建加权规范矩阵以确定负理想方案和理想方案。我们使用变异系数法来求各指标的权重。变异系数（Coefficient of Variation，CV）是标准差与平均值的比值，通常以百分比形式表示。它衡量了数据的相对离散程度，可以用来比较具有不同尺度或不同单位的数据集的变异性。较大的变异系数表示数据更为离散，而较小的变异系数表示数据更为集中。也可以用于求取变异系数加权的权重，以反映每个因素或指标对总体的贡献程度。具体过程如下：

1. 计算变异系数：对于每个因素或指标，计算其数据的标准差和平均值，并计算相应的变异系数。计算变异系数的公式如下：

   $$CV = \frac{\sigma}{\mu} \times 100\% \tag{7}$$

   CV 是变异系数，σ 是标准差，μ 是均值。标准差是数据的离散度度量，平均值是数据的集中趋势度量。

2. 标准化变异系数：将所有的变异系数进行标准化处理，使其数值范围在 0 到 1 之间。

使用公式

$$x_{ij}^{'}=\frac{\max x-x_{ij}}{x_{ij}-\min x} \tag{8}$$

将原变异系数映射到 $[0,1]$ 上。

3. 求取权重：将标准化变异系数进行归一化处理，使得所有权重之和等于 1。可以通过使用公式

$$w=\frac{x_{ij}^{'}}{\sum x_{ij}^{'}} \tag{9}$$

计算得指标权重为

表 3　各指标权重

最大供货能力	供货平稳度	商家偏好程度	最大供货时长	供货平均值
0.4517	0.0268	0.0596	0.1529	0.3088

在得到了各指标的权重之后，构建加权规范矩阵 C

$$C=WH \tag{10}$$

确定样品的理想方案 S^{+} 和负理想方案 S^{-}。指标最优的一组数据为理想方案，指标最差的为负理想方案，结合 5.2.1 指标评价的好坏，其理想方案和负理想方案分别为

$$\begin{aligned} S^{+}&=\left\{c_1^{+},c_2^{+},\cdots,c_m^{+}\right\},c_m^{+}=\min\left(c_j\right)\\ S^{-}&=\left\{c_1^{-},c_2^{-},\cdots,c_m^{-}\right\},c_m^{-}=\max\left(c_j\right)\end{aligned} \tag{11}$$

在得到正负理想方案之后，将所有供货商与这两个方案对比，计算相似度，以向量间的欧式距离衡量差别

$$\begin{aligned} d_i^{+}&=\sqrt{\textstyle\sum_{j=1}^{m}\left(z_j^{+}-z_{ij}\right)^2}\\ d_i^{-}&=\sqrt{\textstyle\sum_{j=1}^{m}\left(z_j^{-}-z_{ij}\right)^2}\end{aligned} \tag{12}$$

根据 d_i^{+} 和 d_i^{+} 评分，计算公式如下

$$S_i=\frac{d_i^{-}}{d_i^{+}+d_i^{-}} \tag{13}$$

全部供货商的分数如下图所示，标注了前 50 名：

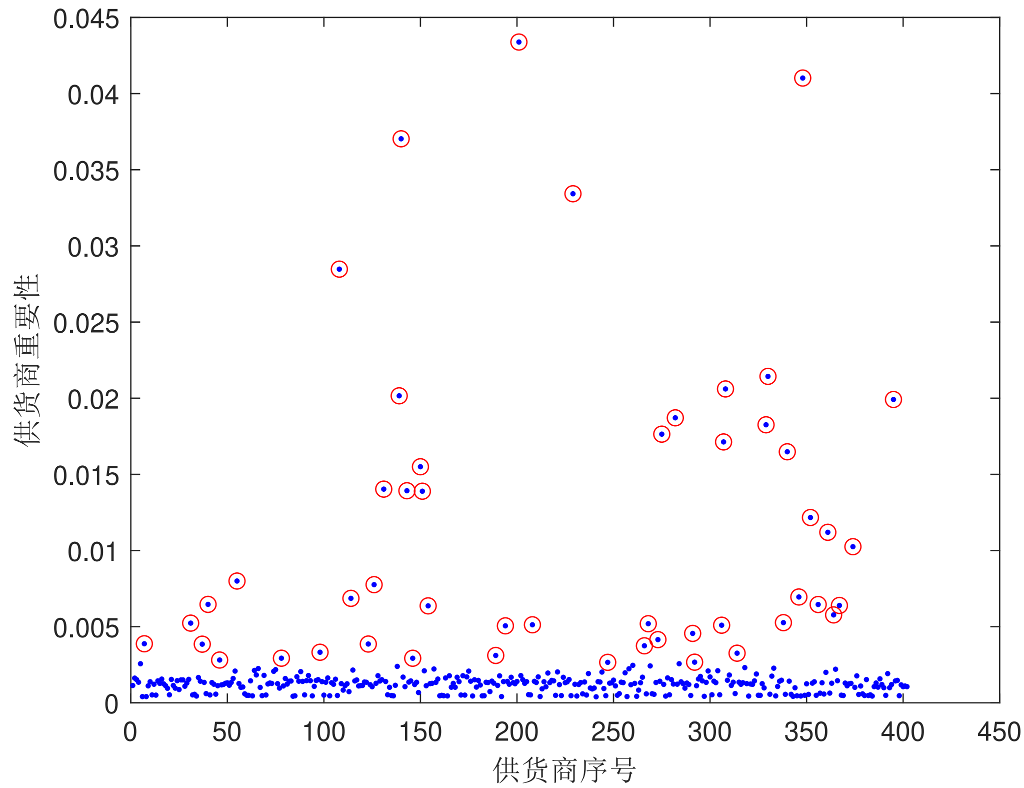


图 2　供应商重要性示意图

我们分析了排名较高的供应商，结果显示企业更重视商家的供货能力和供货稳定性，与订货量的匹配能力也是一个重要的指标。重要性分布图呈现二八分布，这显示明显较强的供应商大概占总数的百分之 20（约 40 家），在实际生产中，企业更愿意选择重要程度更高的供应商，考虑每周的生产所需材料一定，企业可以选择一两家供应能力强和四五家供应能力靠后的供应商供货。前 50 的供应商的具体列表附在附录。

5.2 问题二，三：供货方案和转运方案模型

5.2.1 订货方案设计

首先为确定供货商的每周供货量，我们使用 ARIMA 模型 [2] 根据最大供货量和平均供货量进行预测，这样消除了某些周订货量为 0 的情况，使模型更加严谨。

ARIMA（Autoregressive Integrated Moving Average）模型是一种广泛应用于时间序列预测和分析的统计模型。它可以捕捉到时间序列中的趋势。它由三个部分组成：自回归（AR）部分、差分（I）部分和移动平均（MA）部分。其中，自回归部分使用前期观测值来解释当前值的变化；移动平均部分使用前期误差项来解释当前值的变化；差分部分则是将原始时间序列进行差分，以消除趋势或季节性等因素对数据的影响。ARIMA 模型有三个参数 p, d, q。

1. p 表示 AR 部分阶数，即使用多少个前期观测值来解释当前值。例如，如果 $p=2$，则当前值取决于前两个时刻的观测值。我们将预测序列长度设为 240 周，即在预测计划中第 1 周的数据时，我们会使用前 240 周的历史数据。
2. d 表示差分阶数，即对原始序列进行几阶差分以消除趋势或周期性等影响。例如，如果 $d=1$，则使用当前值减去前一个时刻的值来得到差分值，从而消除了第一阶趋势影响。
3. q 表示 MA 部分阶数，即使用多少个前期误差项来解释当前值。例如，如果 $q=1$，则当前值取决于前一个时刻的误差项。

我们将预测值与前文的供货指标进行综合以修正 ARIMA 模型因为 0 带来的预测失准，

$$F = M + F \tag{14}$$

这个式子将预测值 F 修正为置信数据上限和预测值的和。对于供货商，我们取 $d=1, p=2, q=2$，对于转运商，我们取 $d=1, p=4, q=4$。算法流程图如下：

Algorithm 1 ARIMA 模型预测算法

Require: 时间序列数据 y_t

Ensure: 预测未来值 $\hat{y}_{t+h}$

差分操作：$\Delta y_t = y_t - y_{t-1}$

建立 ARIMA 模型：$\Delta y_t = \phi_1 \Delta y_{t-1} + \cdots + \phi_p \Delta y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \cdots - \theta_q \epsilon_{t-q}$

估计模型参数 $\phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q$ 和误差方差 σ^2

预测未来值：

for $i=1$ to h **do**

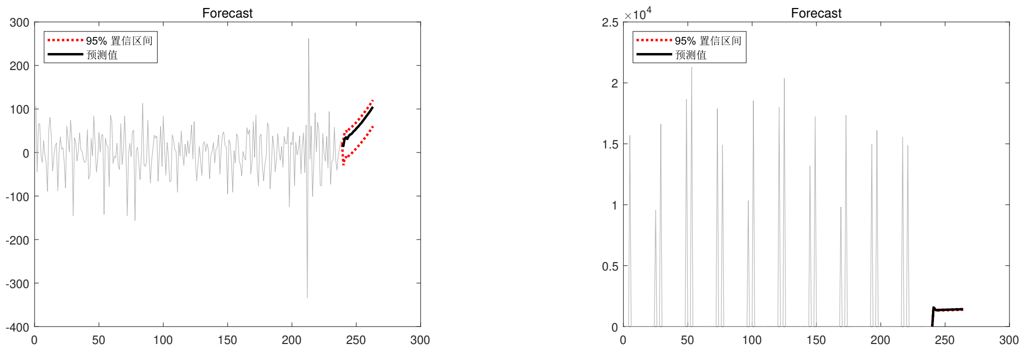
  $\hat{\Delta} y_{t+i} = \phi_1 \Delta y_t + \cdots + \phi_p \Delta y_{t-p+1} - \theta_1 \epsilon_t - \cdots - \theta_q \epsilon_{t-q+1}$

  $\hat{y}_{t+i} = y_t + \hat{\Delta} y_{t+i} + \sum_{j=1}^{i-1} \hat{\Delta} y_{t+j}$

end for

return 输出预测结果 $\hat{y}_{t+1}, \cdots, \hat{y}_{t+h}$

这里给出两家供货商的供给曲线预测：



(a) 供货商 1 **(b) 供货商 2**

图 3 两家供货商的供货量预测

可以看到供货商的供给量并不恒定，其波动规律符合历史数据。得到供给量数据后，为求得满足生产需要的最少的供应商，我们将它转化为如下的线性规划

$$\begin{gathered} \min \sum_{i=1}^{24} x_{ij} \\ s.t. \\ \sum_{i=1}^{50} \frac{x_i P_{ij}}{Q_i} \geq 28200 + 28200 \times 2 \end{gathered} \tag{15}$$

x_{ij} 是一个 0-1 变量，定义为

$$\begin{cases} x_{ij} = 1, \text{第 i 家供货商的货在第 j 周购买} \\ x_{ij} = 0, \text{其他情况} \end{cases} \tag{16}$$

P_{ij} 是第 i 家供应商在第 j 周的供货量，Q_i 是第 i 家供货商生产的材料对应转换为产品的比值。我们使用分支定界法求解这个线性规划问题，分支定界法是一种在解决组合优化问题时常用的算法。它通过将问题的解空间划分为一个个子问题，并逐步缩小搜索范围，以找到最优解或者近似最优解。[3] 具体的流程如下：

1. 初始化：确定要解决的问题类型，并初始化一个空的解集。同时，定义一个上界（或下界）来限制最优解的搜索范围。
2. 划分子问题：将原始问题进行划分，得到多个子问题。这些子问题通常是原始问题的一个子集，但它们的解空间要小于原始问题的解空间。
3. 求解子问题：对每个子问题应用适当的解法进行求解。这里的子问题均为单目标线性优化问题，可以通过常规方法直接求解。
4. 更新上界（或下界）：通过比较每个子问题的解与当前已知的最优解，更新上界（或下界）。如果某个子问题的解优于当前的上界（或下界），则将其作为新的上界（或下界）。
5. 剪枝：根据当前已知的最优解和上下界的比较关系，剪去那些不可能包含最优解的子问题。

6. 迭代：重复步骤 3 到步骤 5，直到找到最优解或者搜索空间为空。
7. 如果解空间为空，给出基于上一迭代过程的近似最优解。

求得最小需要 14 家供应商才能保证生产。而考虑成本最小时，建立规划

$$
\begin{gathered}
\min C=\sum_{t=1}^{24} C_t \\
s.t. \\
C_t=C_{Vt}+G_{Tt}+C_{St} \\
\alpha_{\max}=\max_{1\leq k\leq 24}\left\{\alpha_{ij},\alpha_{ij}\neq 0\right\} \quad (j=1,2,\cdots,8) \\
\frac{E_{At}}{0.6}+\frac{E_{Bt}}{0.66}+\frac{E_{Ct}}{0.72}=28200 \\
\frac{V_{A(t-1)}}{0.6}+\frac{V_{B(t-1)}}{0.66}+\frac{V_{c(t-1)}}{0.72}\geqslant 56400 \\
V_{At}=-E_{At}+V_{A(t-1)}+P_{At}\left(1-\alpha_{\max}\right) \\
V_{Bt}=-E_{Bt}+V_{B(t-1)}+P_{Bt}\left(1-\alpha_{\max}\right. \\
V_{ct}=-E_{ct}+V_{c(t-1)}+P_{ct}\left(1-\alpha_{\max}\right) \\
C_{vt}=P_{At}C_A+P_{Bt}C_B+P_{ct}C_C \\
C_{Tt}=\left(P_{At}+P_{Bt}+P_{Ct}\right)C_T \\
C_{st}=\left(V_{at}+V_{bt}+V_{ct}\right)C_S
\end{gathered} \tag{17}
$$

其中 C_V 是储存成本,C_T 是转运成本,C_S 是购买成本,C_A,C_B,C_C 是对应材料单价。以第一周为例，求解的结果为

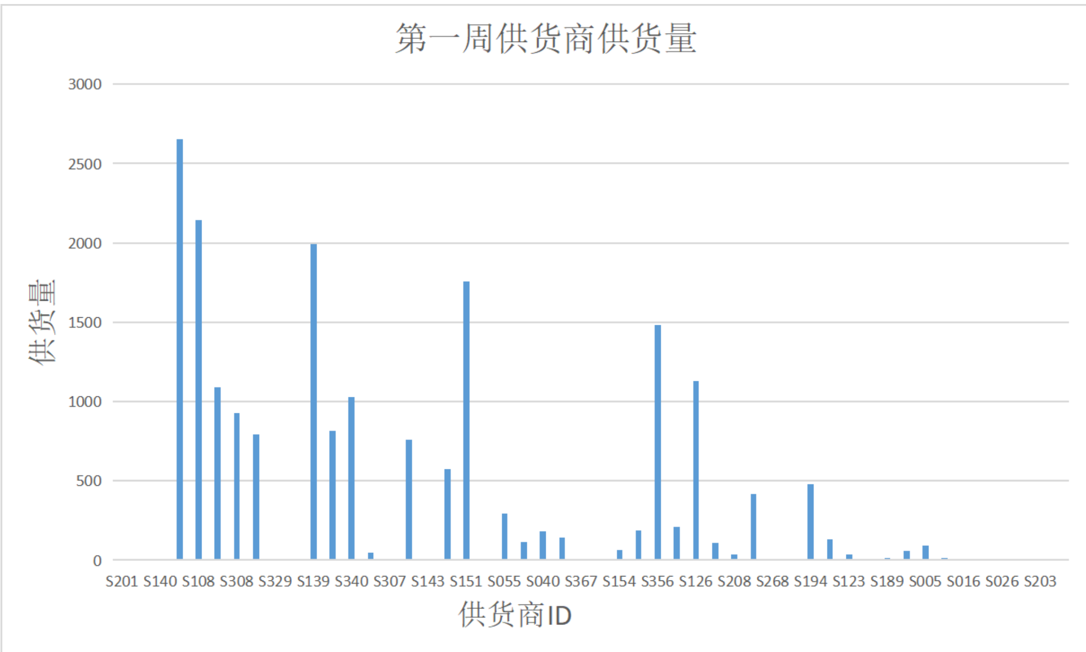


图 4　第一周供货方案

而供货的材料比例为

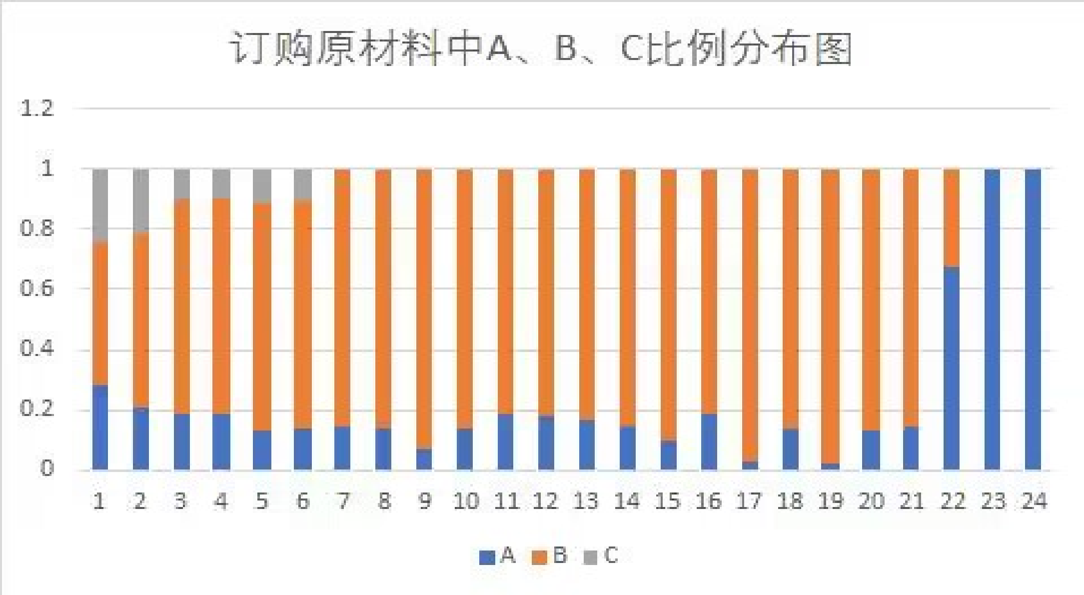


图 5　供货材料比例

分析总的供货结果，每周的平均购买生产 2 万 8227 单位产品的材料，平均成本为 10 万 1813，保证了平均储存量为 5 万 6949 产品材料。可以发现企业更偏好 A 材料，不喜欢 C 材料，推测这是因为 C 材料单价高，用它生产不划算。

5.2.2 转运方案设计

仍然通过 ARIMA 预测转运商损耗率。第一周的转运商损耗率为

表 4　第一周转运商损耗率

转运商 ID	T1	T2	T3	T4	T5	T6	T7	T8
损耗率	1.893	0.821	0.043	1.177	1.607	0.863	1.348	0.973

因为转运方案只针对每周的购买计划，不需要考虑库存影响，这样线性规划被简化为

$$\min W=\sum_{i=1}^{50}\sum_{j=1}^{8} y_{ij}p_i\alpha_i$$
$$s.t.$$
$$\begin{cases} \sum_{i=1}^{50} y_{i1}P_i \le 6000 \\ \sum_{i=1}^{50} y_{i2}P_i \le 6000 \\ \vdots \\ \sum_{i=1}^{50} y_{i8}P_i \le 6000 \\ \sum_{j=1}^{8} y_{ij}=1\ (\text{for } i=1\cdots 50) \end{cases} \tag{18}$$

y_{ij} 也是 0-1 变量，与 x_{ij} 相似的，定义为

$$\begin{cases} y_{ij}=1, \text{第 i 家供货商的货由第 j 家转运商运输} \\ y_{ij}=0, \text{其他情况} \end{cases} \tag{19}$$

同样使用分支定界法，解得每周损耗情况如下图

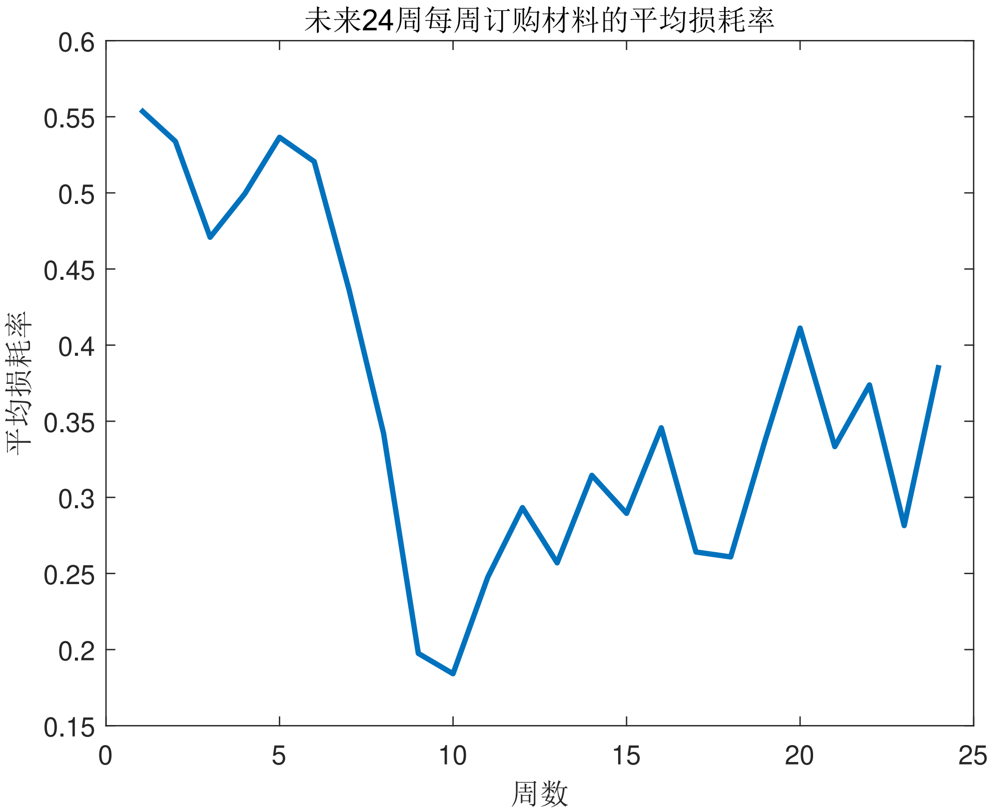


图 6　每周损耗情况

每周平均转运损失 46.76 单位材料，平均需要 2.45 家转运商。平均损耗率控制在 0.55 以下，可以证明实现了转运方案优化。

5.2.3 有材料偏好的情况下的订货转运方案设计

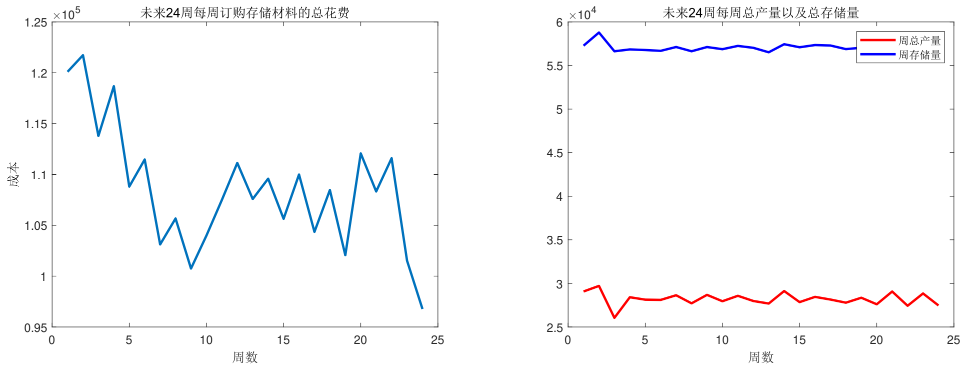
为压缩成本，要求购买最多的 A 材料和最少的 C 材料，我们在原目标的基础上引入两个新的目标

$$\begin{cases} \max P_A \\ \min P_C \end{cases} \tag{20}$$

这是一个多目标优化问题，在有 0-1 规划约束的条件下难以求解，因为单位 A，C 材料可生产的产品数已知，单价已知，根据这些数据我们将两个目标合并在一起。

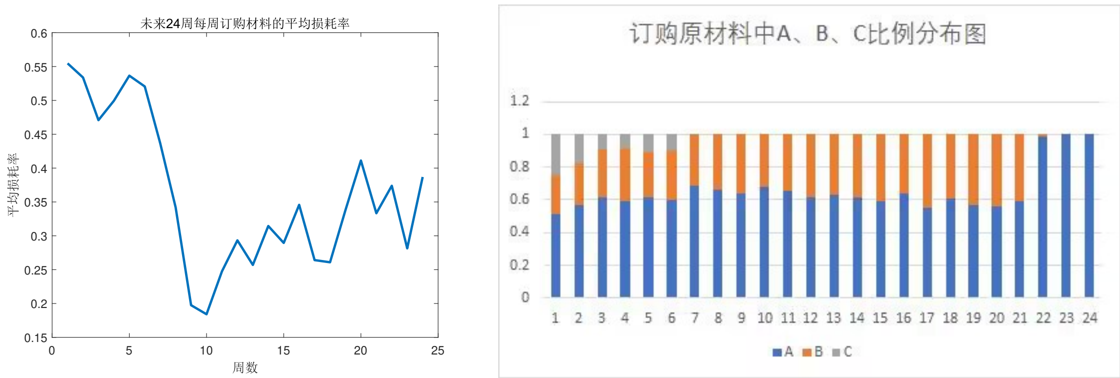
$$\min(\sum_{t=1}^{24}(C_{Vt}+C_{Tt}+C_{St}))\times\frac{Q_C}{Q_A} \tag{21}$$

使用分支定界法，得



(a) 成本折线图 **(b) 产量折线图**

图 7　有材料偏好的购买结果



(a) 损耗折线图 **(b) 材料比例图**

图 8　有材料偏好的转运结果

可以看到损耗率基本与问题二中方案保持一致，A 材料的购买占比上升，C 材料的购买占比下降，产量稍微有下降但仍能满足生产，因为 A 材料购买太多导致存储成本有上升。企业在正常情况下不能只追求购买成本最低，还应该考虑转运成本和储存成本来实现总成本最低。

5.3 问题四：基于的最大产能计算模型

为最大发挥产能潜力，企业应当采购所有供货商的材料并制定合适的转运方案，而在特定的一周，供货商的供货量和转运商的转运能力都是确定的，这样就转化为了一个多重背包问题，等价的描述是：对于不同数量的不同价值材料，现在要将其分配给八个运力不同的转运商，问如何分配使总材料价值最大。

通过二进制粒子群算法 [4] 求解分配问题，并找到最优解。

Algorithm 2 二进制粒子群优化算法（Binary Particle Swarm Optimization, BPSO）

初始化粒子群中每个粒子的位置和速度，速度为 0-1 序列
for 迭代次数小于 200 **do**
  for 每个粒子 **do**
    计算粒子当前位置的适应度公式为 $V^{k+1} = w \cdot V^k + c_1 \cdot r_1 \cdot \left(\text{Pbest}^{\,k} - X^k \right) + c_2 \cdot r_2 \cdot \left(gbest^k - X^k \right)$
    if 适应度大于个体历史最优适应度 **then**
      更新个体历史最优位置
    end if
    if 适应度大于全局历史最优适应度 **then**
      更新全局历史最优位置
    end if
    根据速度和加速度更新粒子的位置和速度
  end for
end for
返回全局历史最优位置对应的解

这里的适应度即为最大材料价值，我们将粒子的速度的值限制为 0 或 1，表明一家供应商只能选择一家转运商，实现了 0-1 规划。第一周的订货及转运方案如下：

在求出每周的转运方案后，问题转化为一个规划问题，即

$$\begin{gathered} \max \theta \\ s.t. \\ \begin{cases} E_i \geq \theta \\ E_i = R_i + E_{i-1} - \theta \end{cases} \end{gathered} \tag{22}$$

注意到将 E_i 累加可得

$$\begin{gathered} (E_n - E_{n-1}) + (E_{n-1} - E_{n-2}) + \cdots + (E_2 - E_1) = (R_n - \theta) + (R_{n-1} - \theta) + \cdots + (R_2 - \theta) \\ \Rightarrow E_n = \textstyle\sum_{i=1}^{n} R_i - (n-2)\,\theta \end{gathered} \tag{23}$$

绘制图像得 E_n 和 θ 的函数关系

这里为求在约束条件下最大的 θ，使用二分法搜索，在估计区间内取中点检查是否满足约束，如果是，取区间的下半部分为新区间，如果不是，取区间的上半部分为新区间，直到最后的值与求出的最好的值差别小于 0.0001。流程图如下：

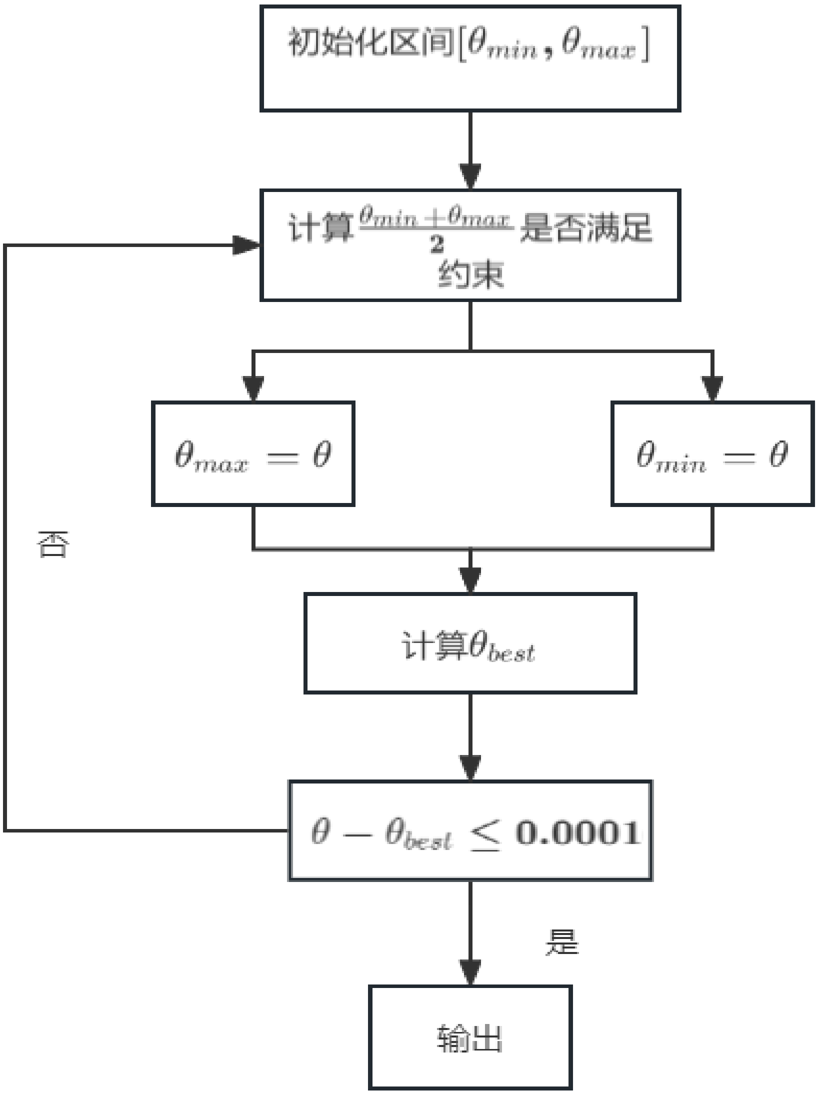


图 9　二分法寻找最优解

解得 θ 最优为 4.1859，即在不考虑成本的情况下，企业的理论最大产能为每周生产 4.1859 万立方米

六、 灵敏性分析

只有问题四中的粒子群算法需要调整关键参数，我们在问题四中参数的基础上乘以一个系数，然后求解观察总材料转运量的变化。以第一周为例，改变粒子群算法中的粒子速度变化率和停止概率，这两个参数控制了粒子群算法的随机性，对 c_1, c_2, ω 使用控制变量法，各变量乘以 0 到 2，发现最大转运量均没有变化，即算法的鲁棒性非常好，改变参数并不会使结果有明显变化，这也证明模型是可依赖的，可推广的。

七、 模型总结

7.1 模型优点

- 分支定界法可以将空间复杂度很高的规划问题转化为若干个空间复杂度不高的式子，让模型可以得到最优解。
- 我们并未使用太多依赖参数的算法，这使得模型具有稳定性，而问题四中粒子群算法检验具有鲁棒性，最优结果的求取对参数并不敏感。

7.2 模型缺点

- 由于变异系数是标准差与平均值的比值，所以当平均值接近于零或非常小的时候，计算得到的变异系数可能会失真。这样 TOPSIS 模型在评价那些供货量有较多 0 的供应商。
- 在很多时间供货商接到的没有接到订货，这可能会影响 ARIMA 模型的预测性能，模型的修正只能在一定程度上减小这种影响而不能消除。
- 分支定界法的性能受到问题规模的影响，对于大规模问题可能会面临搜索空间过大的挑战。因此，实际生产中供货商过多会导致求解速度太慢。

参考文献

[1] 王海宇. 基于 TOPSIS 方法的自适应 EWMA 图多目标优化设计 [J]. 工业工程与管理,2022,27(05):45-52.DOI:10.19495/j.cnki.1007-5429.2022.05.006.

[2] 丁宇超, 苏宇, 姜波等. 基于自回归差分滑动平均的云南高速公路冬季路温短临预测模型研究 [J]. 价值工程,2023,42(10):118-120.

[3] 杜彦武. 整数线性规划中分支定界法及其应用 [C]//Singapore Management and Sports Science Institute,Singapore.Proceedings of 2015 3rd Asian Conference on the Social Sciences（Advances in Social and Behavioral Sciences Volume 15）上.Singapore Management and Sports Science Institute,2015:105-111.

[4] 代祖华, 周斌, 龙玉晶等. 折扣 0-1 背包问题粒子群算法的贪婪修复策略探究 [J]. 计算机应用研究,2022,39(08):2363-2368.

附录 A　代码

```
clear fore;
```

```
forel=[];
for i=1:1:8
    clear forecast
    y = lo(i,:);
    y = reshape(y, [], 1);
    data=[];
    for j=1:1:240
        if y(j)~=0
            data=[data;y(j)];
        end
    end
    d=1;
    while adftest(data)~=1 && kpsstest(data)~=0
        d=d+1;
        data=diff(data);
    end
    % 拟合 ARIMA 模型
    p=4;q=4;
    model = arima(p,d,q); % ARIMA(p, d, q) 中的参数根据具体情况设定
    %model.Variance = 'Nonnegative'; % 设置方差为非负
    fitModel = estimate(model, data);
    [res,~,logL] = infer(fitModel,data); %res即残差
    stdr = res/sqrt(EstMdl.Variance);%标准差
    % 进行预测
    numSteps = 24; % 预测的步数
    [forecast, ~] = forecast(fitModel, numSteps,'Y0',data);
    forecast(forecast < 0) = 0;
    lower=forecast-1.96*sqrt(YMSE);
    upper=forecast+1.96*sqrt(YMSE);
    % forecast=round((upper+forecast)/2);

    % figure()
    % plot(data,'Color',[.7,.7,.7]);
    % hold on
    % h1 = plot(length(data):length(data)+step,[data(end);lower],'r:','LineWidth',2);
    % plot(length(data):length(data)+step,[data(end);upper],'r:','LineWidth',2)
    % h2 = plot(length(data):length(data)+step,[data(end);forecast],'k','LineWidth',2);
    % legend([h1 h2],'95% 置信区间','预测值',...
    %        'Location','NorthWest')
    % title('Forecast')
    % hold off

    forel=[forel;forecast'];

end
```

```
b=[];
f=[];
ic=[];
for i=1:24
    b=[b;28200];
end
for j=1:50
    f=[f,1];
    ic=[ic,j];
end
[x,y]=intlinprog(f,ic,-data,-b,[],[],zeros(50,1),ones(50,1));
```

```
function BPSO_MultipleKnapsack(weights,va )
    capacities=[];

    %% Problem Definition
    num_bags = 8;
    num_items = 50;

    for i=1:8
        capacities = [capacities,6000];
    end

    %% Parameters of PSO
    MaxIt = 200; % Maximum Number of Iterations
    nPop = 50;  % Population Size (Swarm Size)
    w = 1;      % Intertia Coefficient
    wdamp = 0.99; % Damping Ratio of Inertia Coefficient
    c1 = 2;     % Personal Learning Coefficient
    c2 = 2;     % Global Learning Coefficient

    %% Initialization
    particle.Position = randi([0, 1], num_items, num_bags);
    particle.Cost = Fitness(particle.Position, weights, values, capacities);
    particle.Velocity = zeros(num_items, num_bags);
    particle.Best.Position = particle.Position;
    particle.Best.Cost = particle.Cost;

    particles = repmat(particle, nPop, 1);
    GlobalBest = particles(1).Best;

    %% Main Loop of PSO
    for it = 1:MaxIt
```

```
    for i = 1:nPop
        % Update Velocity
        particles(i).Velocity = w * particles(i).Velocity ...
            + c1 * rand(num_items, num_bags) .* (particles(i).Best.Position -
                particles(i).Position) ...
            + c2 * rand(num_items, num_bags) .* (GlobalBest.Position - particles(i).Position);

        % Update Position
        particles(i).Position = round(sigmoid(particles(i).Velocity));

        % Ensure each item is only selected once
        for item = 1:num_items
            bags_with_item = find(particles(i).Position(item, :) == 1);
            if length(bags_with_item) > 1
                bag_to_keep = bags_with_item(randi(length(bags_with_item)));
                bags_to_remove = setdiff(bags_with_item, bag_to_keep);
                particles(i).Position(item, bags_to_remove) = 0;
            end
        end

        % Evaluation
        particles(i).Cost = Fitness(particles(i).Position, weights, values, capacities);

        % Update Personal Best
        if particles(i).Cost > particles(i).Best.Cost
            particles(i).Best.Position = particles(i).Position;
            particles(i).Best.Cost = particles(i).Cost;

            % Update Global Best
            if particles(i).Best.Cost > GlobalBest.Cost
                GlobalBest = particles(i).Best;
            end
        end
    end

    w = w * wdamp;
end

%% Results
disp('Best Solution:');
disp(GlobalBest.Position);
disp('Best Fitness:');
disp(GlobalBest.Cost);
disp(weights);
disp(values);
disp(capacities);
end
```

```
function y = sigmoid(x)
    y = 1 ./ (1 + exp(-x));
end

function fit = Fitness(position, weights, values, capacities)
    total_value = 0;
    for bag = 1:size(position, 2)
        if sum(position(:, bag) .* weights') <= capacities(bag)
            for item = 1:size(position, 1)
                if position(item, bag) == 1
                    total_value = total_value + values(item);
                    position(item, bag) = 0; % Ignore this item in the current bag
                end
            end
        end
    end
    fit = total_value;
end
```

```
E=zeros(24,1);
syms x;
E(1)=R(1);
e(1)=E(1)+x;
for i=2:24
    E(i)=E(i-1)+R(i);
    e(i)=E(i)-(i-2)*x;
    X(i)=solve(e(i)-x==0,x);
end
s=min(double(X'));
```

附录 B　供应商列表

表 5　前 50 名供应商列表

1 到 10	11 到 20	21 到 30	31 到 40	41 到 50
s201	s139	s055	s126	s189
s229	s275	s346	s273	s078
s140	s340	s040	s208	s005
s348	s150	s114	s338	s300
s108	s307	s367	s268	s016
s330	s131	s374	s306	s053
s308	s143	s154	s194	s026
s282	s352	s364	s291	s030
s329	s151	s356	s123	s203
s395	s361	s031	s266	s211