基于遗传算法与非线性规划的企业订购和转运决策模型

摘要

本文针对供应商对企业重要性问题，基于**主成分分析法**对402家供应商进行综合评价；之后本文针对企业生产的订货、转运方案决策问题，基于**非线性规划**对企业订货、转运方案构建**决策模型**；并且本文以**启发式算法**代替非线性规划模型进行求解，基于“**逐步最优，得到整体最优解**”的思想，最后得到企业的订货和转运方案。

**针对问题一：**首先，我们建立了三级七类的供货商**供货特征指标体系**，用于后续综合评价模型的输入参数；随后，我们为了探究七个特征指标是否适用于主成分分析，运用SPSS软件进行了数据先验：**KMO检验和Bartlett's检验**，其中KMO检验系数为**0.883**和Bartlett's检验的P值小于**0.01**，证明了七个特征指标适用于主成分分析；最后我们运用Matlab软件编程构建基于**主成分分析法**的供应商重要性综合评价模型，计算出每家供应商的综合评分，并列出50家最重要的供应商。

**针对问题二：**首先，我们运用Python软件编程对附件1中数据进行深度挖掘，对企业**原始库存的存在性**进行先验，并且通过**数值模拟**企业240周的生产历程，溯源求得企业的原始库存最小数值为**11.28万立方米（四个标准周产能）**；随后，我们通过**非线性规划**来建立最优企业生产决策模型，分别确定目标函数和约束条件，并且通过**遗传算法**和**罚函数法**进行求解，计算所得在满足企业正常生产的情况下，所需供应商的最小数量为**32**，展示32家供应商结果；最后，我们分别建立了最经济的订购决策模型和最优转运损耗决策模型，求解所得未来24周最经济的订购方案和损耗最小的转运方案，并且通过未来每周的转换产能值和转运损耗产能总值来对订购方案和转运方案进行分析。

**针对问题三：**首先，我们在通过非线性规划建立最经济的订购方案过程中，将购买A类原材料和C类原材料的次数作为**决策变量**加入**目标函数**中；同时，由于目标函数的趋向是极小值，而购买A类原材料次数趋向是极大值，我们对购买A类原材料次数进行**化处理**，即先取负数，再取指数；最后，我们重新构建最经济的订购决策模型和最优转运损耗决策模型，求解所得未来24周最经济的订购方案和损耗最小的转运方案，并且通过未来每周的转换产能值和转运损耗产能总值来对订购方案和转运方案进行分析。

**针对问题四：**首先，我们利用**蒙特卡洛模拟**系统随机生成对24周，每周402个**仿真变量**，限定仿真变量的取值为0和1，利用Python软件编程，按照蒙特卡洛模拟进行1000次**仿真**，求解出企业的最优产能值为**7.24万立方米，**产能提高了**4.42万立方米**。最后，我们将企业产能值更换为最优产能值，并且重新构建最经济的订购决策模型和最优转运损耗决策模型，求解所得未来24周最经济的订购方案和损耗最小的转运方案。

我们对决策变量的对应权重和对应幂次进行调参，分析模型的**灵敏度**；并且分析了本文模型的优缺点，提出了模型的推广和改善方向。

关键词： **主成分分析 数值模拟 遗传算法 罚函数法 蒙特卡洛仿真模拟**

一、问题重述

# 1.1问题背景

在当今社会的发展进程中，我国的经济发展务必坚持以供给侧结构改革为发展主线，把发展经济的着力点放在实体经济上。然而对于企业而言，供给侧结构性改革包括两个方面，一方面是企业原材料的订购方案改革；一方面是企业原材料的转运方案改革[1]。

企业的原材料订购方案和转运方案的改革能够助力企业生产能力的提升。其中，企业的订货量、供应商的供货量、转运商的转运损耗、原材料转换产能的比例系数、企业每周的生产产能等因素都影响了企业的生产情况。因此，如果通过上述多种因素，为企业找寻最重要的供应商、制定订购方案和转运方案对企业的生产具有重要意义。

# 1.2问题提出

现根据上述题目背景以及附件所提供的相关数据，建立数学模型解决以下问题：

**问题一：**附件1中已有402家供应商在240周内的供货量和企业的订货量，需要提取数据中的特征，构建一个能够评价402家供应商对企业生产重要性的模型，并且检索重要性排名前50家的供应商，给出结果。

**问题二：**依据问题一的所得结果，求解在满足企业正常生产的条件下，企业需要至少要多少家供应商提供原材料；随后，根据所得供应商，替企业制定订购方案；紧接着，根据所得订购方案，替企业制定转运方案；最后，对所得的两个方案进行分析。

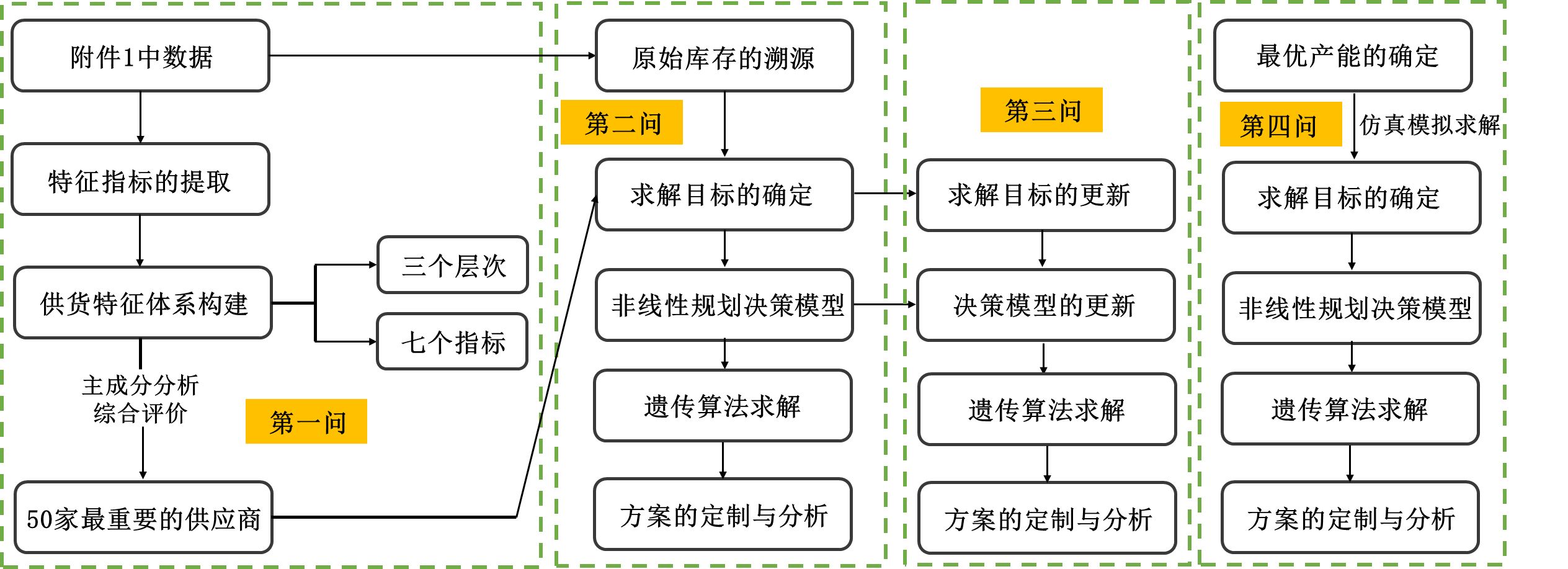
**问题三：**企业为了降低生产成本，现打算尽可能多地购买A类原材料，尽可能少地购买C类原材料。为此，需要重新替企业制定订购方案和转运方案。同时，还要对所得的两个方案进行分析。

**问题四：**由于企业经过了技术的改造，企业的产能上限得到了提升。需要根据原有的供货、转运情况重新对企业每周的产能上限进行求解，并计算产能提高了多少。同时，还需要重新替企业制定订购方案和转运方案。

二、问题分析

# 2.1总体分析

本文主要从供应商、转运商两个维度来解决企业生产原材料的订购、转运决策问题。问题一是针对402家供应商在240周内的订货量和供货量进行特征提取，搭建特征指标体系，构建供应商重要性的评价模型，选出对企业最重要的50家供应商；问题二是针对前一问中求得的50家供应商进行搜索，求解至少需要多少家供应商能够保证企业的正常生产，同时针对所得供应商，替企业制定未来的订购方案和转运方案，并对方案进行分析；问题三是针对企业降低生产成本的需求，保障尽可能多地购买A类原材料和尽可能少地购买C类原材料，替企业制定未来的订购方案和转运方案，并对方案进行分析；问题四是针对企业的产能潜力提高，需要重新替企业制定未来的订购方案和转运方案。



**图1 全文总体思路**

# 2.2具体分析

**针对问题一：**

为了建立能够分析402家供应商重要性的综合评价类模型，首先我们需要对附件1中402家供应商的订货量数据和供货量数据进行量化分析和数据挖掘，搭建供应商供货特征指标体系，用于后续代入综合评价类模型，求解402家供应商的重要性综合评分。

通过对题干的分析和附件1中数据的结合，我们将供应商的重要性评价影响因素主要分为供应商的供货稳定性、供应商的生产规模实力（供货能力）以及企业的订购情况（需求意愿）。并且给出供应商的供应量的标准差、极差、供需满足率、供应商有效供货总次数、供货总量、企业有效订货总次数、订货总量七个特征指标，组成供应商供货特征指标体系。

为了探究七个特征指标数据是否适用于主成分分析，我们对七个特征指标数据进行了数据先验步骤，其中包含：KMO检验和Bartlett's检验；并且我们用SPSS软件对七个特征指标进行检验。

随后我们以七个特征指标作为原始数据，构建基于主成分分析法的供应商重要性综合评价模型。

计算出每个主成分的信息贡献率和累积贡献率，得出每家供应商的重要性中和评分，并对评分结果进行归一化处理，降序得到对企业最重要的50家供应商。

**针对问题二：**

为了探究企业的生产需求问题、制定未来24周的订购方案和运转方案；我们首先对附件1中数据进行了深度挖掘，对原始库存的存在性进行先验。

根据题干，我们得到两个约束条件分别为：；。因此，我们以每一周为单位，根据三种原材料的转换产能比例系数计算出每一周原材料转换的产能值。通过结果先验每一周原材料转换的产能值是否同时满足以上两个约束条件。同时，我们以48周为一个周期，绘制每个周期内每一周企业的订货量与供应商的供货量的对比柱状图和每个周期内累积的企业的订货量与供应商的供货量的对比柱状图，通过柱状图来分析企业生产是处于“供不应求”或是“供过于求”或是“供求平衡”的阶段来进一步验证原始库存的存在性。

随后，我们利用Python软件编程对企业在240周内的生产历程进行数值模拟，给出目标函数和约束条件，找寻企业在240周生产过程中产能不足的极大点，以此来反推原始库存的最小数值。

最后，我们通过非线性规划来建立最优企业生产决策模型，分别确定目标函数和约束条件，并用遗传算法和罚函数法进行求解；并且对所得的订购方案和转运方案进行分析。

**针对问题三：**

为了降低企业的生产成本，我们需要更改决策模型为企业生产成本最优决策模型，保障企业尽可能多地购买A类原材料和尽可能少地购买C类原材料。

由于单位体积的A、B、C三种原材料运输成本和储存成本一致；因此，我们在该问题中无需考虑运输和储存费用。

在确定目标函数的过程中，我们加入了购买A类原材料的次数和购买C类原材料的次数作为决策变量。但是，由于目标函数的趋向是越小越好，而根据题干要求，购买A类原材料的次数趋向是越大越好；因此，我们对购买A类原材料的次数先取负数，再取对数，以此来满足目标函数的趋向要求。

随后，我们更改约束条件，重新构建原材料最优配比决策模型，并且继续用Python软件编程求解最优订购方案；利用罚函数法求解最优转运方案，并且对所得的订购方案和转运方案进行分析。

**针对问题四：**

由于企业经过了技术的改造，企业的产能潜力得到了一定的提升；因此，我们需要根据现有的供应商和转运商实际的供货情况和转运情况来求解出企业改造后的最优产能。

我们利用蒙特卡洛模拟系统随机生成402个仿真变量，且使得仿真变量的值只能为0或1。我们利用Python软件编程，按照蒙特卡洛模拟的流程进行1000次仿真，求解企业的最优产能值。

随后，在求解新的订购方案和转运方案过程中，只需要将企业原先每周的产能更改为企业目前最优产能即可。更新目标函数和约束条件即可求解出更新后的订购方案和转运方案。

三、模型假设

（1）假设企业具有原始库存，并且原始库存的最小数值为4周的产能。

（2）假设供应商对企业的重要性仅由附件1中数据所反映的特征指标所决定，不考虑企业自身的其他主观因素。

（3）假设在溯源企业原始库存的时候，不考虑原材料转运带来的损耗。

四、符号说明

|  |  |
| --- | --- |
| **符号** | **说明** |
|  | 第家供货商的有效供货总次数 |
|  | 第家供货商的供货总数量 |
|  | 第家供货商的有效订货总次数 |
|  | 第家供货商的订货总数量 |
|  | 供应商240周内供货量的标准差 |
|  | 供应商240周内供货量的极差 |
|  | 供需满足率 |
|  | 供应商的综合得分 |
|  | 归一化处理后的综合得分 |
|  | 每周的转换产能值 |
|  | A、B、C三种原材料转换成产能的比例系数 |
|  | 原始库存最小数值 |
|  | 目标函数 |
|  | 决策变量的权重 |
|  | 所需供应商的数量 |
|  | 第周企业完成生产后更新的库存容量 |
|  | 供货量所需要花费的金额 |
|  | A、B、C三种原材料的购买金额 |
|  | 8家转运商的转运损耗值 |
|  | 第家转运商负责转运第家供应商的原材料数量 |
|  | 第家转运商负责转运的原材料总数量 |
|  | 类原材料的购买次数 |
|  | 蒙特卡洛仿真次数 |
|  | 仿真变量 |
|  | 企业的最优产能值 |

五、问题一的模型建立与求解

# 5.1供应商供货特征指标体系的构建

为了建立能够分析402家供应商重要性的评价模型，我们首先对附件1中的数据进行量化分析和数据挖掘。

结合题目大意以及附件1中所知数据，供应商的重要性主要由供应商的供货稳定性、供应商的生产规模实力（供货能力）以及企业的订购情况（需求意愿）三个主要方面的影响因素决定。

其中，供应商的供货稳定性与供应商的供应量的标准差和极差以及供需满足率有关联；供应商的生产规模实力与供应商有效供货总次数和供应商240周内供货总量有关联；企业的订购情况与企业有效订货总次数和企业240周内订货总量有关联。

首先，我们对衡量供应商重要性的七个特征指标给出明确定义；随后构建供应商供货特征指标体系。

**（1）有效供货总次数**

定义：有效供货总次数即每家供应商240周内的有效供货总次数。

|  |  |  |
| --- | --- | --- |
|  |  | （1） |

|  |  |  |
| --- | --- | --- |
|  |  | （2） |

式中: 代表第家供应商在第周的供货量；代表第家供应商在第周是否为有效供货，当时，，即有效供货；反之，则为无效供货；代表第家供应商的有效供货总次数。

每家供应商的有效供货总次数能够体现该供应商的生产规模实力；当有效供货总次数越大时，该供应商的供货能力越强，对企业也越重要；反之，重要性越低。

**（2）供货总量**

定义：供货总量即每家供应商在240周内的供货总数量。

|  |  |  |
| --- | --- | --- |
|  |  | （3） |

式中:代表第家供应商第周的供货量。

每家供应商的供货总量能够体现该供应商的生产规模实力；当供货总量越大时，该供应商的供货能力越强，对企业也越重要；反之，重要性越低。

**（3）有效订货总次数**

定义：有效订货总次数即企业在240周内对每家供应商的有效订货的总次数。

|  |  |  |
| --- | --- | --- |
|  |  | （4） |
|  |  | （5） |

式中: 代表第家供应商在第周的被企业要求的订货量；代表第家供应商在第周是否为有效订货，当时，，即有效订货；反之，则为无效订货；代表第家供应商被企业要求的有效订货总次数。

每家供应商的有效订货总次数能够体现企业对该供应商的依赖程度；当有效订货总次数越大时，企业对该供应商的依赖程度越高，该供应商对企业也越重要；反之，重要性越低。

**（4）订货总量**

定义：订货总量即企业在每家供应商在240周内的订货总数量。

|  |  |  |
| --- | --- | --- |
|  |  | （6） |

式中:代表第家供应商第周的供货量。

企业对每家供应商的订货总量能够体现企业对该供应商的需求程度；当供货总量越大时，企业对该供应商的需求程度越高，该供应商对企业也越重要；反之，重要性越低。

**（5）供货量的标准差**

定义：供货量的标准差即每家供应商240周内供货量的标准差。

|  |  |  |
| --- | --- | --- |
|  |  | （7） |

式中: 代表第家供应商第周的供货量；代表该家供应商240周内供货量的均值，代表供货总周次。

在评价402家供应商的重要性的过程中，每家供应商供货量的标准差可作为度量供应商供应原材料稳定性的指标。当标准差的数值越大，代表稳定性远离均值，因此供应商供应原材料的稳定性越低。反之，标准差的数值越小，供应商供应原材料的稳定性越高，对企业也越重要。

由于下文采用主成分分析法进行求解，我们采用校正值 作为一个参数输入。

**（6）供货量的极差**

定义：供货量的极差即每家供应商240周内供货量的最大值最小值

|  |  |  |
| --- | --- | --- |
|  |  | （8） |
|  |  |  |

式中:代表第家供应商第周的供货量；代表该家供应商的最大供货量；代表该家供应商的最小供货量。

每家供应商供货量的极差也可作为度量供应商供应原材料稳定性的指标。当极差的数值越大，代表供应商的供应能力越强，稳定性较高，对企业也越重要。

**（7）供需满足率**

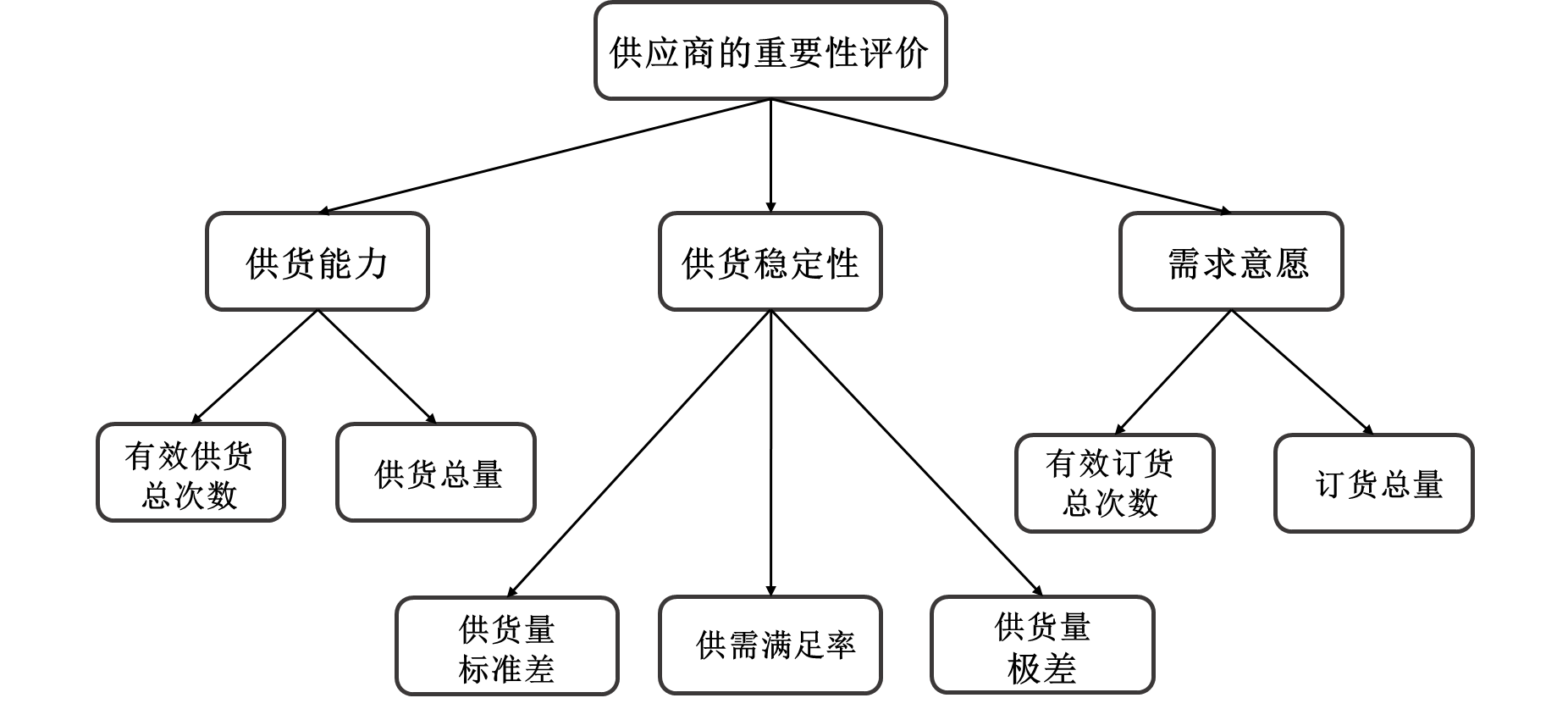
定义：供需满足率即供应商240周内有效供货量与有效订货量的比值的均值。

我们通过Python软件编程在附件1中进行数据搜索，通过搜索结果，可以发现，当企业对某一个供应商的订货量为0时，该供应商的供货量也一定为0，因此我们可以利用前文的有效供货订货总次数和供货订货总量来定义供需满足率。

|  |  |  |
| --- | --- | --- |
|  |  | （9） |
|  |  |  |

每家供应商的供需满足率也可作为度量供应商供应原材料稳定性的指标。当企业对某家供应商给出订货量，倘若该供应商的供需满足率越接近于1，则代表该供应商持续稳定供货的可能性更高，对企业也更为重要；反之，则对企业的重要性越低。

**（8）供应商供货特征指标体系**



**图2 供应商供货特征指标体系**

由此可知，供应商的有效供货总次数越大，供货总量越大，该供应商的生产规模实力越强，极差越大，供货能力越强；企业对某家供应商的有效订货总次数越大，企业制定的订货总量越大，表明企业对该供应商的需求程度越高；供应商供货量的标准差越小，供需满足率越接近于1，则表明该供应商的供货稳定性越高；当某家供应商满足上述7种情况，则该供应商对企业的重要性越高。

# 5.2基于主成分分析法的供应商重要性评价模型

通过附件1中的所有数据和已经建立的供应商供货特征指标体系，可以发现供应商的重要性评价与七个特征指标有着必然联系；并且众多指标之间存在复杂的相关性，增加了量化分析的复杂度。

单独对每个指标进行分析，往往会将指标孤立化，不能够完全利用数据中的信息，盲目导致指标损失许多有用的信息，从而产生错误的结论。因此我们采用主成分分析法来建立供应商重要性评价模型。

**数据先验：**

为了更好地应用主成分分析, 我们需要对主成分分析结果进行统计检验并建立统计检验体系。其中不可或缺的一步便是主成分适用性检验，即该七个特征指标数据是否适合使用主成分方法进行分析[2]。

KMO检验是从比较原始变量之间的简单相关系数和偏相关系数的相对大小出发来进行的检验。当所有变量之间的偏相关系数的平方和, 远远小于所有变量之间的简单相关系数的平方和时, 变量之间的偏相关系数很小, KMO 值接近1, 变量适合进行主成分分析。

Bartlett's检验的原假设是研究数据之间的相关矩阵是一个完美矩阵，即所有对角线上的系数为1，非对角线上的系数均为0。在这种完美矩阵的情况下，各变量之间没有相关关系，即不能将多个变量简化为少数的成分，没有进行主成分提取的必要。因此，我们希望拒绝Bartlett's检验的零假设，即Bartlett's检验的P值小于0.01。

我们利用SPSS软件，对七个特征指标进行KMO检验和Bartlett's检验，结果表如下：

**表1 数据先验结果**

|  |  |  |
| --- | --- | --- |
| KMO检验 | 检验系数 | 0.883 |
| Bartlett’s检验 | P值 | 0.0001 |

检验标准，表如下所示：

**表2 KMO检验标准**

|  |  |
| --- | --- |
| KMO检验系数 | 主成分分析适用性 |
| 0.90-1.00 | 适用性极好 |
| 0.80-0.89 | 适用性较好 |
| 0.70-0.79 | 适用性一般 |
| 0.60-0.69 | 适用性差 |
| 0.50-0.59 | 适用性极差 |
| 0.00-0.49 | 完全不适用 |

根据检验结果表和检验标准表可知，KMO检验系数为0.883，代表将七个特征指标应用于主成分分析的适用性较好；并且Bartlett's检验的P值小于0.01，拒绝了零假设；因此，七个特征指标之间具有相关关系，能够进行主成分分析。

**算法概述：**

主成分分析方法，是一种使用最广泛的数据降维兼综合评价算法。能够将关系紧密的指标降维至尽可能少的新指标，用其分别代表存在于七个特征指标中的信息[2]。具体流程如下所示：

（1）对原始数据进行标准化处理。将七个特征指标值转换成标准化指标，有

|  |  |  |
| --- | --- | --- |
|  |  | （10） |

式中：；，即，为第个指标的样本均值和样本标准差。

|  |  |  |
| --- | --- | --- |
|  |  | （11） |

为标准化指标变量。

（2）计算相关系数矩阵。相关系数矩阵，有

|  |  |  |
| --- | --- | --- |
|  |  | （12） |

式中：为第个特征指标与第个特征指标的相关系数。

（3）计算特征值和特征向量，公式如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （13） |

式中：为第1主成分；为第2主成分；……为第7主成分。

（4）计算特征值的信息贡献率和累计贡献率，公式如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （14） |
|  |  | （15） |

式中：为主成分的信息贡献率；为累计贡献率。

（5）选择个主成分，并且计算综合得分，公式如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （16） |

（6）对计算所得的综合得分进行归一化处理，将所得综合得分取值限制在之间

|  |  |  |
| --- | --- | --- |
|  |  | （17） |

通过主成分分析法的计算结果，可以得知402家供应商重要性的最后综合评分；当供应商的综合评分越高，其对企业越重要。因此，通过主成分分析法所得结果与其供应商的重要性呈正相关的关系。

**求解：**

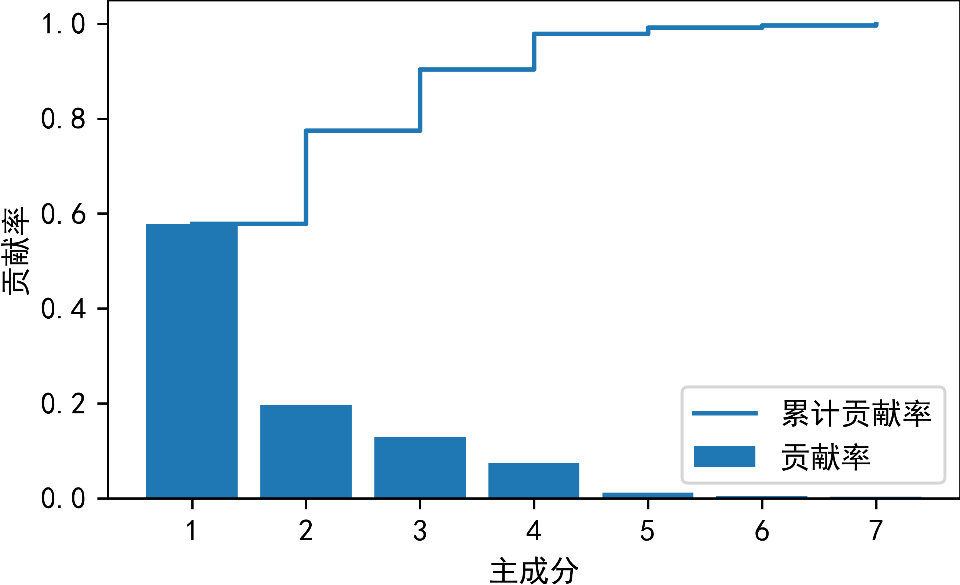
我们首先运用Python软件对附件1中数据进行指标构建，得到7个特征指标；随后我们依据主成分分析法的算法流程，运用Matlab软件进行编程，对7个特征指标进行主成分分析；求得相关系数矩阵的7个特征根、贡献率、累计贡献率，如下表所示：

**表3 相关系数矩阵的特征根、贡献率、累积贡献率统计表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 特征根 | 贡献率 | 累计贡献率 | 序号 | 特征根 | 贡献率 | 累计贡献率 |
| 1 | 4.0487 | 57.8387 | 57.8387 | 5 | 0.0875 | 1.249859 | 99.1735 |
| 2 | 1.3771 | 19.6732 | 77.5119 | 6 | 0.0322 | 0.459995 | 99.6335 |
| 3 | 0.9034 | 12.9062 | 90.4180 | 7 | 0.0257 | 0.366541 | 100 |
| 4 | 0.5254 | 7.5056 | 97.9236 |  |  |  |  |

根据上表可知，前4个特征根的累计贡献率就达到了97.9236%以上，主成分分析效果极佳。

同时根据上表种的贡献率和累计贡献率，我们作图如下所示：



**图3 主成分的信息贡献率和累积贡献率**

由上图可以明显发现前4个主成分的贡献率较高；因此，我们选取前4个主成分进行综合评价。求得特征根对应的特征向量，如下表所示：

**表4 特征根对应的特征向量统计表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 |  |  |  |  |  |  |  |
| 1 | -0.3997 | 0.4425 | 0.0812 | 0.2245 | 0.7611 | 0.0633 | 0.0635 |
| 2 | 0.4389 | 0.0106 | 0.1048 | 0.6141 | 0.0973 | -0.1678 | -0.6175 |
| 3 | 0.4581 | -0.1534 | 0.0948 | 0.4227 | 0.1152 | 0.2443 | 0.7112 |
| 4 | 0.3672 | 0.5142 | -0.2291 | -0.2244 | -0.0564 | 0.6783 | -0.1855 |
| 5 | 0.3694 | 0.4883 | -0.3121 | -0.1703 | 0.0294 | -0.6654 | 0.2341 |
| 6 | 0.3637 | -0.4695 | -0.0676 | -0.4814 | 0.6257 | -0.0089 | -0.1387 |
| 7 | 0.1846 | 0.2386 | 0.9049 | -0.2872 | -0.0491 | -0.0702 | 0.0147 |

由此可得4主成分分别为:

我们分别以4主成分的贡献率为权重，构建主成分综合评价模型，如下所示:

|  |  |  |
| --- | --- | --- |
|  |  | （18） |

|  |  |  |
| --- | --- | --- |
|  |  | （19） |

最后，我们将每个供应商的4主成分值代入上式，求解得到附件1中402家供应商的重要性综合评分和排名结果。表格如下所示：

**表5 重要性综合评分统计表**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 供应商 | S140 | S229 | S361 | S151 | S108 | S348 | S139 | S330 | S308 | S282 |
| 排名 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 评分 | 1.000 | 0.854 | 0.814 | 0.759 | 0.745 | 0.632 | 0.591 | 0.589 | 0.588 | 0.585 |
| 供应商 | S340 | S201 | S275 | S329 | S131 | S356 | S268 | S306 | S194 | S143 |
| 排名 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 评分 | 0.582 | 0.577 | 0.560 | 0.558 | 0.534 | 0.533 | 0.518 | 0.514 | 0.478 | 0.471 |
| 供应商 | S352 | S374 | S307 | S247 | S284 | S365 | S031 | S126 | S040 | S055 |
| 排名 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 评分 | 0.464 | 0.455 | 0.418 | 0.411 | 0.401 | 0.389 | 0.388 | 0.387 | 0.378 | 0.372 |
| 供应商 | S364 | S367 | S395 | S346 | S294 | S080 | S037 | S244 | S218 | S338 |
| 排名 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 评分 | 0.370 | 0.368 | 0.366 | 0.364 | 0.351 | 0.344 | 0.343 | 0.343 | 0.335 | 0.313 |
| 供应商 | S123 | S114 | S086 | S150 | S074 | S291 | S266 | S003 | S314 | S007 |
| 排名 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 评分 | 0.313 | 0.310 | 0.304 | 0.302 | 0.302 | 0.301 | 0.298 | 0.296 | 0.293 | 0.282 |

六、问题二的模型建立与求解

# 6.1数据挖掘溯源企业原始库存

为了探究企业的生产需求问题、制定未来24周的订购方案、制定损耗最少的转运方案；我们首先对附件1中的数据进行了深度挖掘，以此溯源企业的原始库存，为后续问题的求解提供数据支撑。

**原始库存的存在性先验：**

附件1中数据包含了企业在240周内的订货量和402家供应商在240周内的供货量。

分析题干能够得到以下两个约束条件：

我们对附件1中402家供应商每周的供货量进行数据转换，将供货量转换成每周的产能，判断是否能满足两个约束条件。

根据附件1中的供货量，我们以每一周为单位，分别按照式（20、21）计算每一周A、B、C三种原材料数量转换成的产能值。

|  |  |  |
| --- | --- | --- |
|  |  | （20） |
|  |  | （21） |

式中：为A、B、C三种原材料转换成产能的比例系数；为第个供应商第周种类为的原材料供货量。

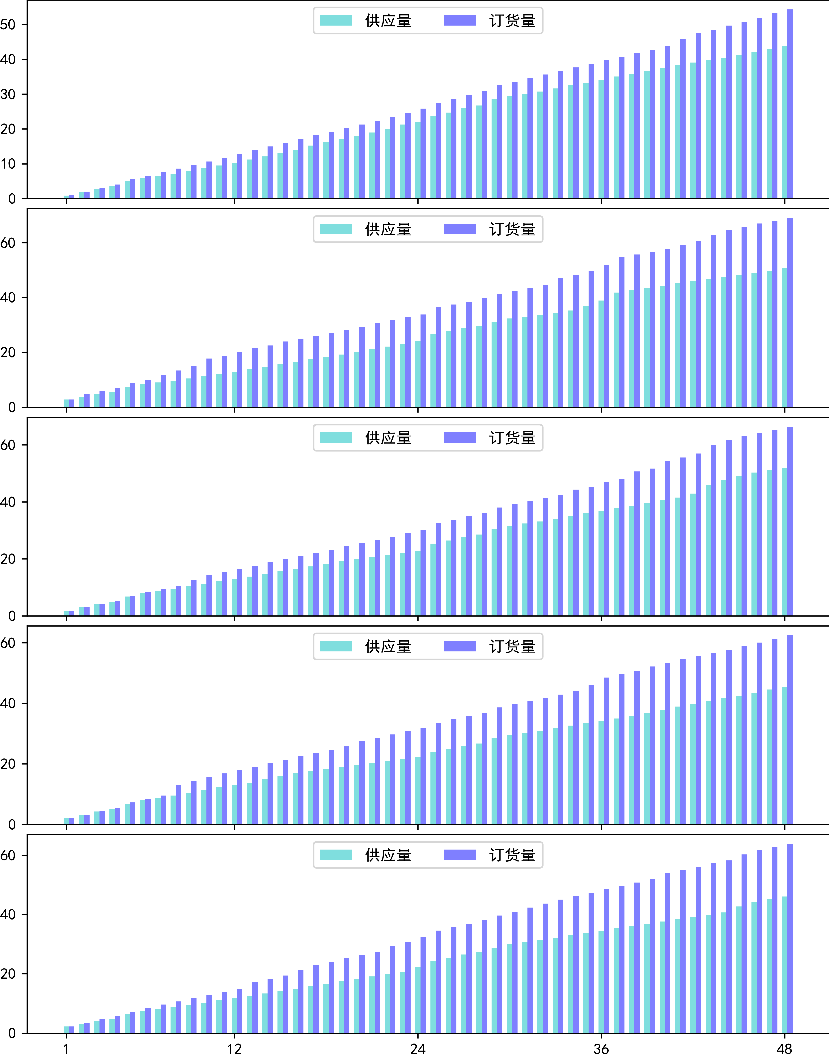
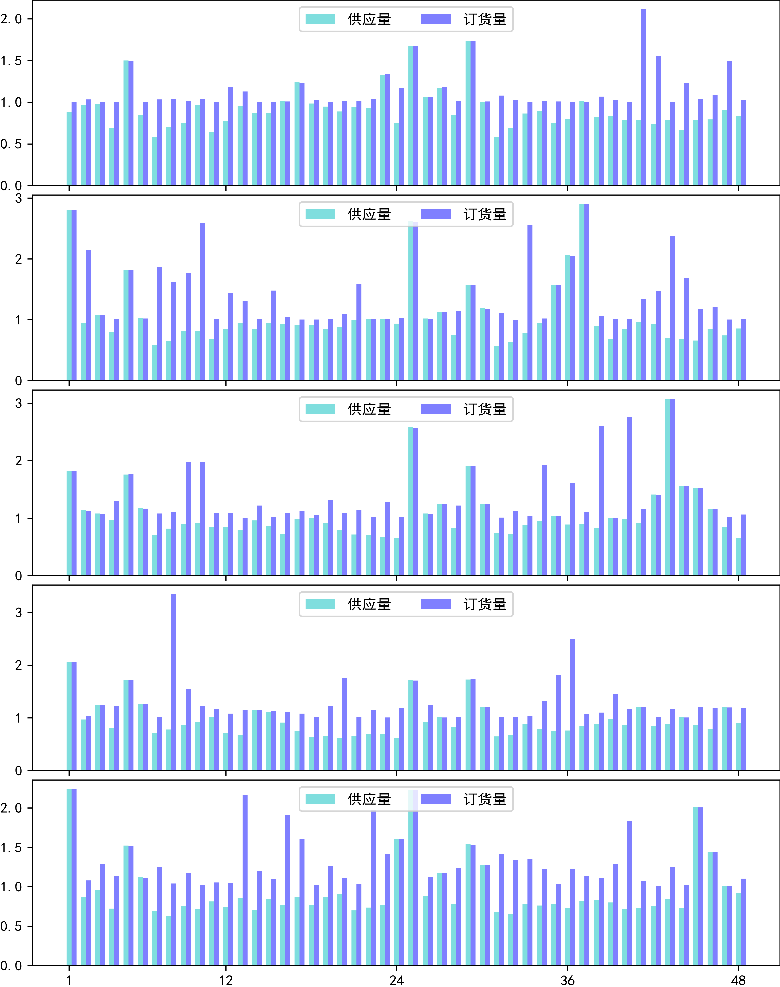
计算每一周A、B、C三种原材料数量转换成的产能值，下表仅展示前八周的转换产能值。

**表6 前八周的转换产能值统计表（单位：万立方米）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 周数 | W001 | W002 | W003 | W004 |
| 转换产能值 | 2.4788 | 2.7145 | 2.7553 | 1.9578 |
| 周数 | W005 | W006 | W007 | W008 |
| 转换产能值 | 4.2313 | 2.3818 | 1.6280 | 1.9691 |

分析上表可以发现：首先，第一周的转化产能值仅2.4788万立方米，无法满足约束条件Ⅰ；第二、三、四、六、七、八周的转化产能值也无法满足约束条件Ⅰ；其次，针对约束条件Ⅱ，前八周中仅第七周的转化产能值满足了约束条件I，即前八周中仅有第七周企业是在满足了该周产能的情况下，补充了库存；其余周次都需要库存的填补才能满足约束条件Ⅰ和Ⅱ。

由此，我们确定了原始库存的存在性；该企业具有一定的原始库存，才能在满足约束条件的情况下，保持企业的正常生产。

为了进一步验证原始库存的存在性，我们利用附件1中的订货量数据和供货量数据，以48周为一个周期，我们作图如下所示：

**图4 订货量与供应商的供货量的对比柱状图**

**图解：**

左图为5个周期内每一周企业的订货量与供应商的供货量的对比柱状图；其中，紫色柱代表企业的订货量；蓝色柱代表供应商的供应量。（纵坐标单位：一周的产能）

右图为5个周期内累积的企业的订货量与供应商的供货量的对比柱状图；其中，紫色柱代表企业的订货量；蓝色柱代表供应商的供应量。（纵坐标单位：一周的产能）

**分析：**

通过左图，我们可以发现，240周内，有超过78.75%的周次呈现“供不应求”的情况，导致企业需要不断挪用库存来填补每周的产能不足。

通过右图，我们可以发现，在5个周期内，累积的供货量随着时间的推移愈发难以满足累积的订货量需求。

由此可知，长期的“供不应求”无法满足题干中的两个约束条件，无法保障企业的正常生产。据此，原始库存的存在性是必然的。

**原始库存的求解：**

我们利用附件1中402家供应商在240周内的供货量，对企业进行数值模拟；由于8家转运商在240周内的最大损耗量仅为5%；因此，在原始库存的求解过程中，不考虑转运带来的损耗。

设置原始库存为（单位：一周的产能）；我们利用Python软件编程模拟企业在240周内的生产历程，逐步求解每周A、B、C三种原材料转换的产能值，并将其与企业每周的产能做差比较，求得二者之间的差值。

|  |  |  |
| --- | --- | --- |
|  |  | （22） |
|  |  | （23） |
|  |  | （24） |

式中：为A、B、C三种原材料转换成产能的比例系数；为第个供应商第周种类为的原材料供货量；企业每周的产能，其值为2.82万立方米

当差值为负数且最小时，即为企业在240周生产过程中产能不足的极大点。根据该企业产能不足的极大点，我们能够得知企业在生产过程中亏损库存的基线情况，以此来反推原始库存的最小数值。

|  |  |  |
| --- | --- | --- |
|  |  | （25） |

式中：为企业产能不足极大点对应的亏损产能值的绝对值。

经过240周的仿真模拟，我们计算出企业产能不足极大点对应的亏损产能的绝对值万立方米，原始库存最小数值万立方米。因此，我们成功溯源企业原始库存至少为万立方米，即4周的产能。

# 6.2 基于非线性规划的最优企业生产决策模型

**6.2.1最少供应商决策模型**

1. **目标函数的确定**

为了求解在满足企业正常生产需求的情况下，所需供应商的最小数量；我们将其定义为多目标规划，并按照一定的权重确定目标函数。

|  |  |  |
| --- | --- | --- |
|  |  | （26） |
|  |  | （27） |

式中：、分别为决策变量和的权重；为所需供应商的数量；为第周企业完成生产后更新的库存容量。

1. **约束条件的确定**

该问题中的约束条件有：企业在保证每周产能的情况下，还能保证有2周的库存容量。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （28） |
|  |  |  |

1. **基于非线性规划的最少供应商决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （29） |

|  |  |  |
| --- | --- | --- |
|  |  | （30） |

模型描述：为目标函数，和均为决策变量。

**④ 基于遗传算法的求解**

为了提高求解的速度，我们把问题一中求解所得50家供应商的供货量作为求解数据，先将该规划问题转变成一个组合优化问题；随后我们选择启发式算法进行求解，极大地提高了求解的效率。

遗传算法是一种基于生物进化启发的优化算法，并不是简单的搜索假设，而是利用变异和重组当前已知的最优假设来衍生后续的假设；逐步优化以达到逐步最优的效果，从而组合成总体最优的目的。

遗传算法的执行过程：1.初始化种群；2.计算适应度值；3.选择；4.交叉；5变异；6.重复执行步骤2和5,直到适应度达到期望值或进化代数达到最大值。

我们利用Python软件编程，按照上述流程进行求解，通过300次的迭代，所得供应商的最小数量为32。供应商结果如下表所示：

**表7 最少数量供应商统计表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 供应商 | S140 | S229 | S361 | S108 |
| 供应商 | S139 | S330 | S308 | S282 |
| 供应商 | S340 | S201 | S275 | S329 |
| 供应商 | S131 | S356 | S268 | S306 |
| 供应商 | S194 | S352 | S307 | S365 |
| 供应商 | S126 | S040 | S055 | S367 |
| 供应商 | S395 | S294 | S338 | S114 |
| 供应商 | S086 | S150 | S074 | S291 |

**6.2.2最经济的订购决策模型**

**① 目标函数的确定**

为了针对已求出的供应商来为企业安排未来24周最经济的订购方案；我们将其定义为多目标规划，并按照一定的权重确定目标函数。

|  |  |  |
| --- | --- | --- |
|  |  | （31） |
|  |  | （32） |
|  |  | （33） |
|  |  | （34） |

式中：、分别为决策变量和的权重；为供货量所需要花费的金额；为第周企业完成生产后更新的库存容量。

**② 约束条件的确定**

该问题中的约束条件有：首先，每周企业购买原材料的金额为3种原材料的金额总和；其次，企业在保证每周产能的情况下，还能保证有2周的库存容量。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （35） |

1. **基于非线性规划的最经济的订购决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （36） |

|  |  |  |
| --- | --- | --- |
|  |  | （37） |

模型描述：为目标函数，和均为决策变量。

1. **基于遗传算法的求解**

我们利用Python软件编程，按照遗传算法流程进行求解，通过300次的迭代，计算得出最经济的订购方案。具体结果见支撑材料附件A中。

**6.2.3最优转运损耗决策模型**

为了确定8家转运商的转运损耗值，我们对附件2中每家公司240周的转运损耗求解有效均值，并将其作为8家转运商的转运损耗值。

1. **目标函数的确定**

针对前面所得的订购方案，需要定制损耗最小的转运方案。由于A、B、C三种原材料的转运成本相同，则目标函数如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （38） |

|  |  |  |
| --- | --- | --- |
|  |  | （39） |

式中：为第i家转运商负责转运第j家供应商的原材料数量；为第家转运商负责转运的原材料总数量；为第家转运商的转运损耗值。

1. **约束条件的确定**

该问题中的约束条件有：1.每周一家转运商最多运输6000立方米的原材料；2.每周每家转运商运输原材料的数量必须大于等于0；3. 每家供应商的供货量都要被转运商运完。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （40） |

式中：为第家供应商在该周的供货量；为第家供应商被第家转运商运送原材料的数量。

1. **基于非线性规划的最优转运损耗决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （41） |

|  |  |  |
| --- | --- | --- |
|  |  | （42） |

模型描述：为目标函数，和均为决策变量。

1. **基于罚函数法求解约束极值问题**

利用罚函数法，能够将非线性规划问题的求解，转化为求解一系列无约束极值问题。

罚函数法的主要流程：1.把问题中的约束函数作出适当的罚函数；2.构造带参数的增广目标函数；3.将问题转化为无约束非线性规划问题[2]。

我们利用Python软件编程，按照罚函数法流程进行求解，计算得出损耗最小的转运方案。具体结果见支撑材料附件B中。

**6.2.4订购、转运方案分析**

通过前面的模型建立和算法求解，我们得到了最经济的订购方案和损耗最少的转运方案，并且对两个方案进行分析。

**对于最经济的订购方案分析：**

我们以48周为一个周期，利用前五个周期的第周的产能与未来24周的第周进行数据同期对比。数据表展示未来24周的前3周数值，如下所示（单位：万立方米）：

**表8 前3周转换产能值统计表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W001 | W049 | W097 | W145 | W193 | 未来第1周 |
| 产能 | 2.48 | 3.89 | 3.14 | 3.82 | 3.33 | 3.09 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W002 | W050 | W098 | W146 | W194 | 未来第2周 |
| 产能 | 2.71 | 2.68 | 3.2 | 2.74 | 2.47 | 3.61 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W003 | W051 | W099 | W147 | W195 | 未来第3周 |
| 产能 | 2.76 | 3.04 | 3.05 | 3.52 | 2.70 | 2.93 |

通过上表可以发现，未来24周的前3周产能数值与前5个周期的同期水平相近，上下波动的干扰较小，代表求解的订购方案精准科学。

**对于损耗最小的转运方案分析：**

我们根据求解所得的转运方案，求解出每一周的损耗产能总值，结果如下所示（单位：立方米）：

**表9 未来24周损耗产能总值统计表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 损耗产能 | 301.39 | 309.44 | 291.11 | 491.01 | 359.09 | 348.08 |
| 周次 | 7 | 8 | 9 | 10 | 11 | 12 |
| 损耗产能 | 357.39 | 269.38 | 250.17 | 247.62 | 447.64 | 331.72 |
| 周次 | 13 | 14 | 15 | 16 | 17 | 18 |
| 损耗产能 | 372.46 | 304.23 | 288.18 | 334.65 | 462.93 | 338.56 |
| 周次 | 19 | 20 | 21 | 22 | 23 | 24 |
| 损耗产能 | 264.18 | 251.34 | 353.81 | 273.02 | 407.42 | 286.71 |

通过上表可以发现，未来24周的损耗产能总值均维持在一个相同水平，波动较小，代表求解的转运方案精准科学。

七、问题三的模型建立与求解

# 7.1基于非线性规划的企业生产成本最优决策模型

**7.1.1原材料最优配比决策模型**

单位体积的A、B、C三种原材料运输成本和储存成本一致；因此，无需考虑运输和储存费用。

该企业想要减少生产的成本，于是希望尽可能多地购买A类原材料、尽可能少地购买C类原材料。

1. **目标函数的确定**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | （43） |
|  | |  | （44） |

式中：、、分别为决策变量、和的权重；和分别为A类和C类材料的购买次数；为第周企业完成生产后更新的库存容量；由于目标函数的趋向是越小越好，而的趋向是越大越好；因此，我们对先取负数，再取指数，以此来满足目标函数的趋向。

1. **约束条件的确定**

该问题中的约束条件有：企业在保证每周产能的情况下，还能保证有2周的库存容量。如下所示（单位：万立方米）：

|  |  |  |
| --- | --- | --- |
|  |  | （45） |
|  |  |  |

1. **基于非线性规划的原材料最优配比决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （46） |

|  |  |  |
| --- | --- | --- |
|  |  | （47） |

模型描述：为目标函数，、和均为决策变量。

1. **基于遗传算法的求解**

我们利用Python软件编程，按照遗传算法流程进行求解，通过300次的迭代，计算出成本最少的订购方案。具体结果见支撑材料附件A中。给出未来前4周企业购买三种原材料的次数表，如下所示：

**表10 前4周购买三种原材料的次数统计表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 材料种类 | A | B | C | 材料种类 | A | B | C |
| 购买次数 | 82 | 72 | 16 | 购买次数 | 76 | 81 | 21 |
| 材料种类 | A | B | C | 材料种类 | A | B | C |
| 购买次数 | 73 | 70 | 9 | 购买次数 | 73 | 72 | 16 |

**7.1.2最优转运损耗决策模型**

当我们在7.1.1部分中，求解出计算出成本最少的订购方案；我们只需将所得订购方案作为求解数据，重复7.1.1的求解过程即可求解出在该订购方案下的损耗最小转运方案。

**求解：**

我们利用Python软件编程，按照罚函数法流程进行求解，计算得出该订购方案下损耗最小的转运方案。具体结果见支撑材料附件B中。

**7.1.3订购、转运方案分析**

通过前面的模型建立和算法求解，我们得到了最经济的订购方案和损耗最少的转运方案，并且对再次对两个方案进行分析。

**对于最经济的订购方案分析：**

我们以48周为一个周期，利用前五个周期的第周的产能与未来24周的第周进行数据同期对比。数据表展示未来24周的前3周数值，如下所示（单位：万立方米）：

**表11 前3周转换产能值统计表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W001 | W049 | W097 | W145 | W193 | 未来第1周 |
| 产能 | 2.48 | 3.89 | 3.14 | 3.82 | 3.33 | 2.97 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W002 | W050 | W098 | W146 | W194 | 未来第2周 |
| 产能 | 2.71 | 2.68 | 3.2 | 2.74 | 2.47 | 2.90 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | W003 | W051 | W099 | W147 | W195 | 未来第3周 |
| 产能 | 2.76 | 3.04 | 3.05 | 3.52 | 2.70 | 3.01 |

通过上表可以发现，未来24周的前3周产能数值与前5个周期的同期水平相近，上下波动的干扰还是较小，代表求解的订购方案依旧精准科学。

**对于损耗最小的转运方案分析：**

我们根据求解所得的转运方案，求解出每一周的损耗产能总值，结果如下所示（单位：立方米）：

**表12 未来24周损耗产能总值统计表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 周次 | 1 | 2 | 3 | 4 | 5 | 6 |
| 损耗产能 | 234.47 | 230.31 | 240.25 | 208.26 | 278.50 | 243.63 |
| 周次 | 7 | 8 | 9 | 10 | 11 | 12 |
| 损耗产能 | 228.80 | 244.67 | 214.74 | 220.03 | 196.95 | 233.21 |
| 周次 | 13 | 14 | 15 | 16 | 17 | 18 |
| 损耗产能 | 213.30 | 206.15 | 193.26 | 205.33 | 243.13 | 220.94 |
| 周次 | 19 | 20 | 21 | 22 | 23 | 24 |
| 损耗产能 | 205.20 | 216.59 | 187.50 | 204.78 | 247.04 | 266.16 |

通过上表可以发现，未来24周的损耗产能总值均维持在一个相同水平，波动依然较小，代表求解的转运方案依然精准科学。

八、问题四的模型建立与求解

# 8.1基于蒙特卡洛仿真模拟企业最优产能

由于企业经过了技术的改造，企业的产能潜力得到了一定的提升；因此，我们需要根据现有的供应商和转运商实际的供货情况和转运情况来求解出企业改造后的最优产能。

蒙特卡洛模拟是一种通过设定随机过程，反复生成时间序列，计算参数估计量和统计量，进而研究其分布特征的方法。其这一特性符合求解企业最优产能的需求，因此，我们采用蒙特卡洛仿真模拟来求解企业的最优产能。

蒙特卡洛模拟的主要流程有：1.构造或描述概率过程；2.实现从已知概率分布中抽样；3.建立各类估计量，求解最终目标解。

我们设定仿真次数；为仿真变量，让蒙特卡洛模拟系统随机生成24周，每周402个仿真变量，且仿真变量的值只能为0或1。

|  |  |  |
| --- | --- | --- |
|  |  | （48） |

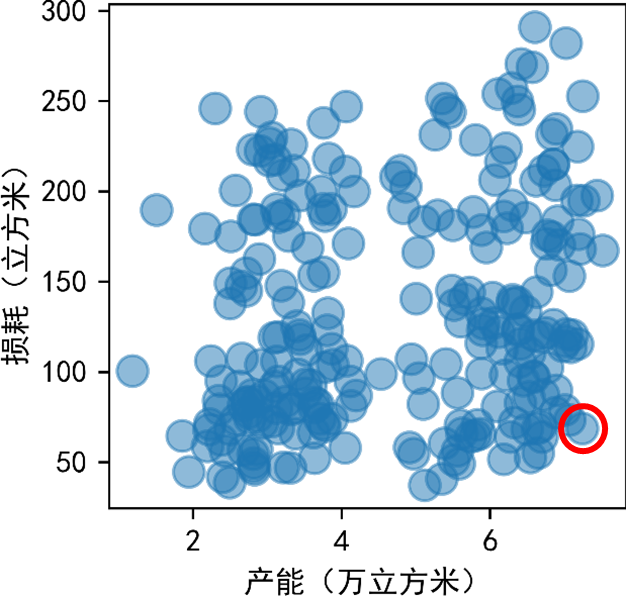
**约束条件：**

在求解企业最优产能的过程中，依旧需要满足以下约束条件：企业在保证每周产能的情况下，还能保证有2周的库存容量。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （49） |

**求解：**

我们利用Python软件编程，按照蒙特卡洛模拟的流程进行1000次仿真，求解企业的最优产能值。



**图5 蒙特卡洛仿真结果**

通过上图可以发现，当模拟结果越靠近右下角红圈处，表明模拟结果越接近于企业的最优产能值。

求解出企业的最优产能值万立方米，产能提高了4.42万立方米。

# 8.2最优产能下的订购、转运决策模型

在该问题中，由于企业的最优产能得到了更新，我们只需在问题二求解过程的基础上将企业每周的产能万立方米更改为万立方米即可。后续只需更新目标函数和约束条件即可求解出更新后的订购方案和转运方案。

**8.2.1最经济的订购决策模型**

**① 目标函数的确定**

|  |  |  |
| --- | --- | --- |
|  |  | （50） |
|  |  |  |

**② 约束条件的确定**

该问题中的约束条件有：首先，每周企业购买原材料的金额为3种原材料的金额总和；其次，企业在保证每周产能的情况下，还能保证有2周的库存容量。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （51） |
|  |  |  |

**③ 基于非线性规划的最经济的订购决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （52） |
|  |  | （53） |

模型描述：为目标函数，和均为决策变量。

**④ 基于遗传算法的求解**

我们利用Python软件编程，按照遗传算法流程进行求解，通过 次的迭代，计算得出最经济的订购方案。具体结果见支撑材料附件A中。

**8.2.2最优转运损耗决策模型**

**① 目标函数的确定**

|  |  |  |
| --- | --- | --- |
|  |  | （54） |
|  |  | （55） |

式中：为第i家转运商负责转运第j家供应商的原材料数量；为第家转运商负责转运的原材料总数量；为第家转运商的转运损耗值。

**② 约束条件的确定**

该问题中的约束条件有：1.每周一家转运商最多运输6000立方米的原材料；2.每周每家转运商运输原材料的数量必须大于等于0；3. 每家供应商的供货量都要被转运商运完。如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | （56） |

式中：为第家供应商在该周的供货量；为第家供应商被第家转运商运送原材料的数量。

**③ 基于非线性规划的最优转运损耗决策模型**

综上所述，我们给出最少供应商的决策模型;

|  |  |  |
| --- | --- | --- |
|  |  | （57） |
|  |  | （58） |

模型描述：为目标函数，和均为决策变量。

1. **基于罚函数法求解约束极值问题**

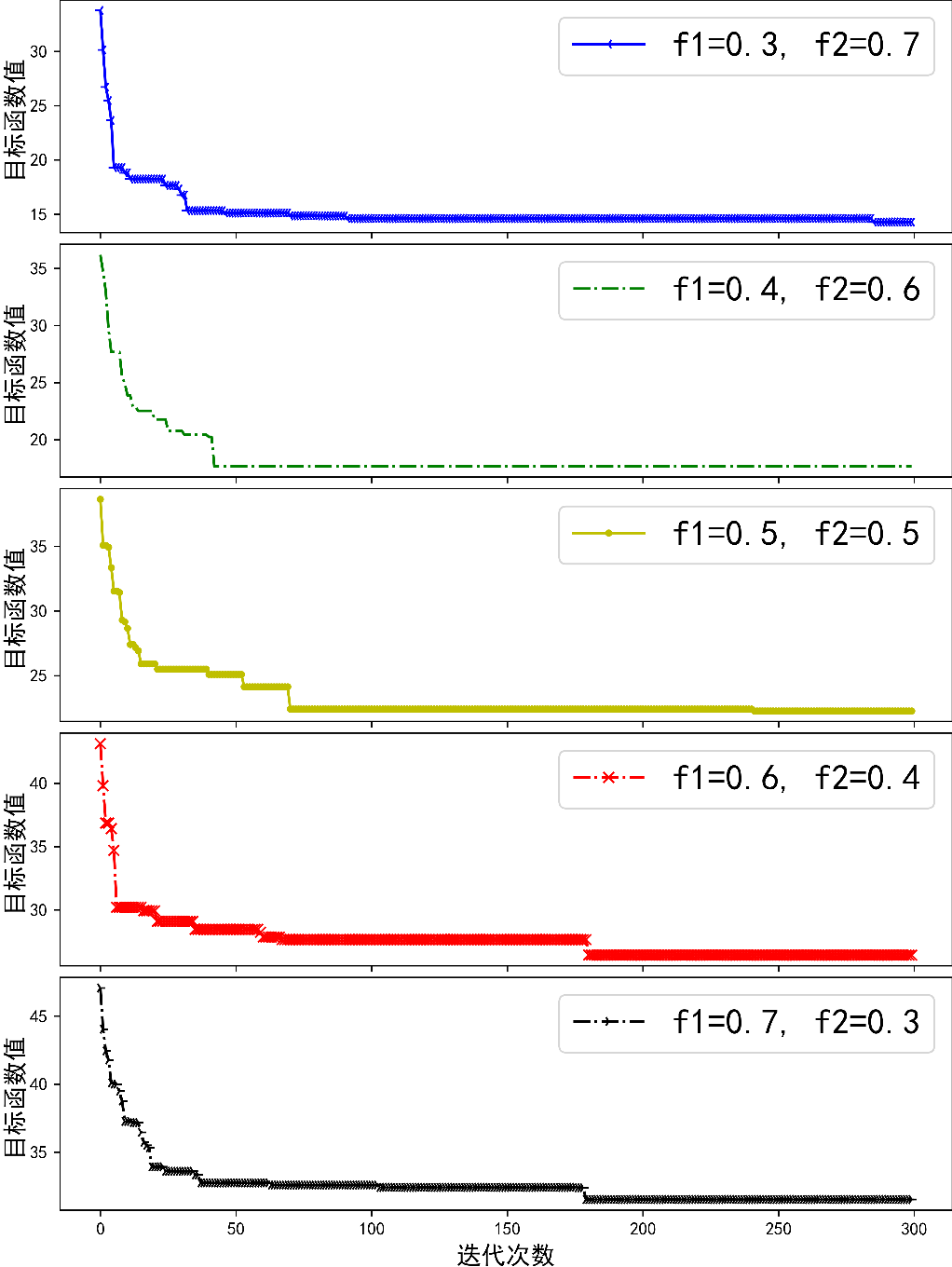
我们利用Python软件编程，按照罚函数法流程进行求解，计算得出损耗最小的转运方案。具体结果见支撑材料附件B中。

九、灵敏度分析

由于在遗传算法的构建过程中，目标函数的确立涉及了决策变量对应的权重与决策变量对应的幂次。我们基于问题二里求最经济的订购方案，分别从调参决策变量对应的权重与决策变量对应的次方数两个维度进行灵敏度分析。

**调参决策变量对应的权重：**

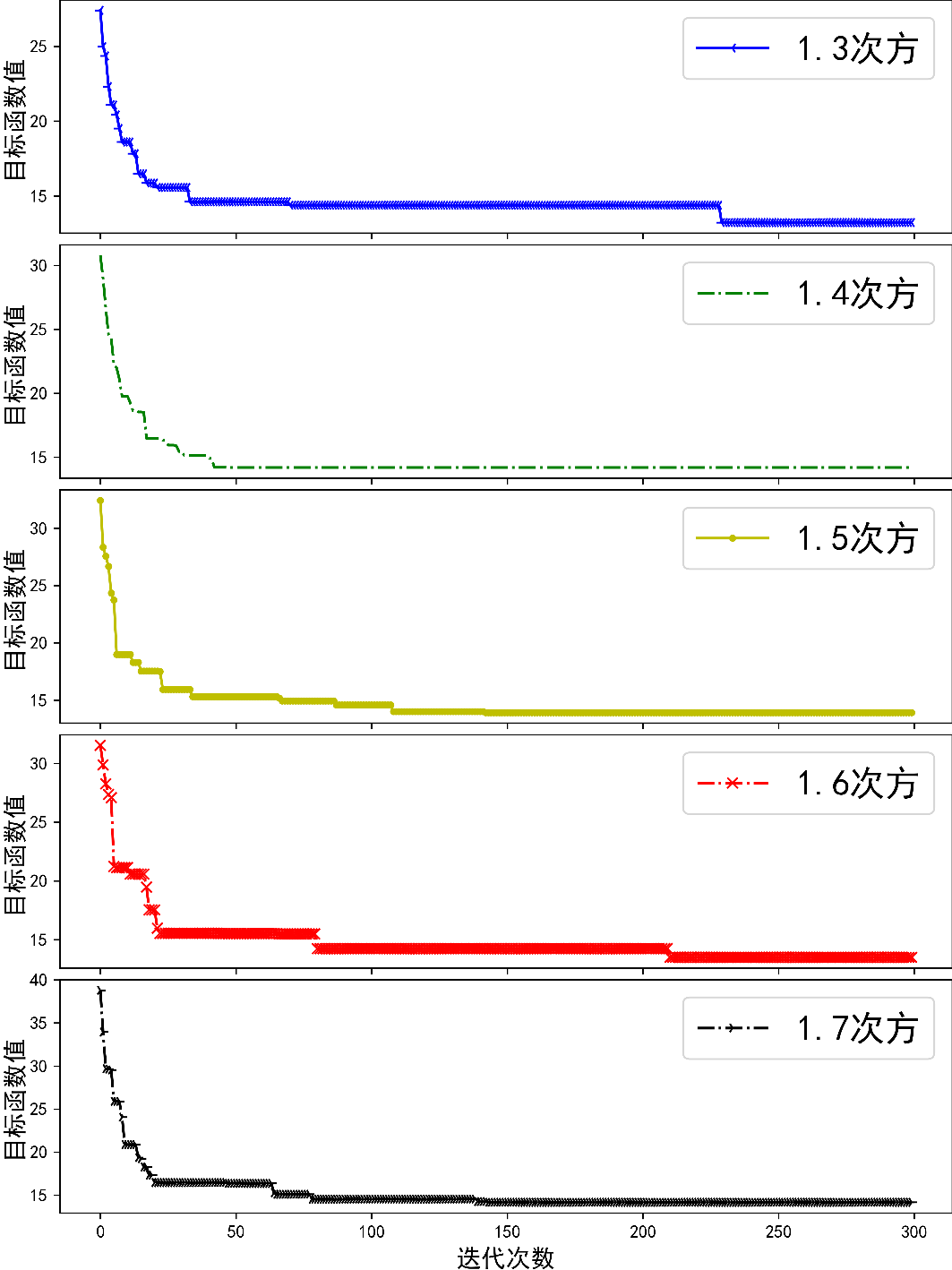
我们在目标函数中分别更改决策变量对应的权重，进行遗传算法300次迭代，绘制目标函数的迭代图，如下所示：



**图6 更改权重的目标函数迭代图**

**调参决策变量对应的幂次：**

我们在目标函数中分别更改决策变量对应的幂次，进行遗传算法300次迭代，绘制目标函数的迭代图，如下所示：



**图7 更改次方数的目标函数迭代图**

综合上述，我们可以发现，遗传算法在企业最优订货方案、转运方案的制定过程中对决策变量的对应权重和对应的次方数的灵敏度较高。

因此，若是要提高模型的准确性，我们应当对真实数据进行统计分析其特征规律，从中提取出决策变量的权重和对应幂次的取值。

十、模型的评价与推广

# 10.1模型的评价

**模型的优点：**

**（1）**构建供应商重要性评价模型的过程中，充分考虑了供应商的供货稳定性、供应商的生产规模实力以及企业的订购情况三个维度，具有较强的实际意义。

**（2）**在进行主成分分析之前，对七个特征指标进行了数据先验，考量了主成分分析的适用性。

**（3）**对企业的原始库存存在性进行先验，并且求解出企业原始库存的最小数值，保障了后续方案求解的科学性。

**（4）**我们通过启发式算法—遗传算法来取代了非线性规划求解，在提高求解效率的同时也保障了结果的准确性。

**模型的缺点：**

**（1）**主成分分析法的主成分具有正负相关性，在综合评价的应用之前，需要明确传入参数的正负相关性。

**（2）**遗传算法对初始种群的选择具有一定的依赖性，可能影响结果的走向。

# 10.2模型的推广与改进

**（1）**供应商提供的供货量和企业制定的订货量之间存在着运筹关系，可以从运筹学的角度出发，对评价模型进行改善。

**（2）**在供应商供货特征指标体系的建立过程中，应该找寻一些建筑企业作为参照，根据实际情况来确定特征指标。

参考文献

[1]李鑫. S公司原材料供给体系质量分析与改进策略研究[D].河北科技大学,2019.

[2]司守奎，孙兆亮. 数学建模算法与应用[M]. 北京：国防工业出版社，2015—4.

附录

|  |  |
| --- | --- |
| Matlab代码 | 主成分分析 |
| clc,clear  gj=load('pca.txt');  gj=zscore(gj);  r=corrcoef(gj);  [x,y,z]=pcacov(r)  f=repmat(sign(sum(x)),size(x,1),1);  x = x.\*f  num=7;  df=gj \* x(:,[1:num]);  tf=df \* z(1:num)/100;  [stf,ind]=sort(tf,'descend');  stf=stf',ind=ind' | |

|  |  |
| --- | --- |
| Python代码 | 数据清洗与提取特征 |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from collections import Counter  data\_dinghuo\_df = pd.read\_excel(r"E:\桌面\2021国赛\CUMCM2021Probelms\C\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\_df = pd.read\_excel(r"E:\桌面\2021国赛\CUMCM2021Probelms\C\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）')  data\_dinghuo\_use\_array = data\_dinghuo\_df.values[:, 2:]  data\_gonghuo\_use\_array = data\_gonghuo\_df.values[:, 2:]  gongxv\_manzu\_radio = []  for i in range(len(data\_dinghuo\_use\_array)):  temp = []  for j in range(240):  if data\_dinghuo\_use\_array[i][j]!=0 and data\_gonghuo\_use\_array[i][j]!=0:  temp.append(data\_gonghuo\_use\_array[i][j]/data\_dinghuo\_use\_array[i][j])  gongxv\_manzu\_radio.append(np.array(temp))  gonghuo\_std = [-np.std(i) for i in data\_gonghuo\_use\_array]  gonghuo\_sum = [np.sum(i) for i in data\_gonghuo\_use\_array]  dinghuo\_sum = [np.sum(i) for i in data\_dinghuo\_use\_array]  dinghuo\_cishu\_sum = [len(i[i!=0]) for i in data\_dinghuo\_use\_array]  gonghuo\_cishu\_sum = [len(i[i!=0]) for i in data\_gonghuo\_use\_array]  jicha = [i.max()-i.min() for i in data\_gonghuo\_use\_array]  gongxv\_manzu\_radio = [-abs(i.mean()-1) for i in gongxv\_manzu\_radio]  pca\_array = np.vstack((gonghuo\_std, gonghuo\_sum, dinghuo\_sum, dinghuo\_cishu\_sum, gonghuo\_cishu\_sum, jicha, gongxv\_manzu\_radio)).T  np.savetxt('E:/桌面/2021国赛/pca.txt', pca\_array) | |

|  |  |
| --- | --- |
| Python代码 | 求解供应商 |
| from scipy.io import loadmat  from collections import Counter  pca = list(loadmat(r"E:\桌面\2021国赛\pca供应商主成分.mat")['tf'].flatten())  pca\_dict = {}  for i in range(len(pca)):  pca\_dict[pca[i]] = i  pca\_dict\_sorted = sorted(pca\_dict.items(), key=lambda d:d[0], reverse = True)  pca\_dict\_sorted\_top\_50 = pca\_dict\_sorted[:50]  pca\_sorted\_top\_50\_index = np.asarray(pca\_dict\_sorted\_top\_50)[:,1].astype('int')  for i in pca\_sorted\_top\_50\_index:  print('S'+'%03d'%(i+1))  bar = [57.8387, 19.6732, 12.9062, 7.5056, 1.249859, 0.459995, 0.366541]  plot = [57.8387, 77.5119, 90.4180, 97.9236, 99.1735, 99.6335, 100]  plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签  plt.rcParams['axes.unicode\_minus']=False  plt.figure(figsize=(5, 3))  plt.step(np.arange(1, len(plot)+1), np.array(plot)/100, where="post", label=u'累计贡献率')  plt.bar(np.arange(1, len(bar)+1), np.array(bar)/100,label=u'贡献率')  plt.ylabel(u'贡献率')  plt.xlabel(u'主成分')  plt.legend(loc='lower right')  plt.savefig(r'E:\桌面\2021国赛/pca.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0) | |

|  |  |
| --- | --- |
| Python代码 | 库存的数据分析 |
| def leiji\_min(data):  # leiji\_min\_data\_list = []  # leiji\_min\_data = 0  leiji\_data\_list = []  leiji\_data = 0  for i in data:  # temp = leiji\_min\_data + i  leiji\_data+=i  # if temp<leiji\_min\_data:  # leiji\_min\_data = temp  # leiji\_min\_data\_list.append(leiji\_min\_data)  leiji\_data\_list.append(leiji\_data)  # print(leiji\_min\_data\_list)  return leiji\_data\_list  data\_gonghuo\_use\_array\_A\_sum = data\_gonghuo\_df[data\_gonghuo\_df['材料分类']=='A'].values[:, 2:].sum(axis=0)  data\_gonghuo\_use\_array\_B\_sum = data\_gonghuo\_df[data\_gonghuo\_df['材料分类']=='B'].values[:, 2:].sum(axis=0)  data\_gonghuo\_use\_array\_C\_sum = data\_gonghuo\_df[data\_gonghuo\_df['材料分类']=='C'].values[:, 2:].sum(axis=0)  channeng\_week = []  for i in range(240):  channeng\_week.append(data\_gonghuo\_use\_array\_A\_sum[i]/0.6+data\_gonghuo\_use\_array\_B\_sum[i]/0.66+data\_gonghuo\_use\_array\_C\_sum[i]/0.72)    data\_dinghuo\_use\_array\_A\_sum = data\_dinghuo\_df[data\_dinghuo\_df['材料分类']=='A'].values[:, 2:].sum(axis=0)  data\_dinghuo\_use\_array\_B\_sum = data\_dinghuo\_df[data\_dinghuo\_df['材料分类']=='B'].values[:, 2:].sum(axis=0)  data\_dinghuo\_use\_array\_C\_sum = data\_dinghuo\_df[data\_dinghuo\_df['材料分类']=='C'].values[:, 2:].sum(axis=0)  channeng\_week\_dinghuo = []  for i in range(240):  channeng\_week\_dinghuo.append(data\_dinghuo\_use\_array\_A\_sum[i]/0.6+data\_dinghuo\_use\_array\_B\_sum[i]/0.66+data\_dinghuo\_use\_array\_C\_sum[i]/0.72)  plt.figure(figsize=(10,25))  for i in range(1, 6):  year = ((np.array(channeng\_week)/10000)/2.82)[(i-1)\*48:i\*48]  leiji\_data\_list = leiji\_min(year)  year\_dinghuo = ((np.array(channeng\_week\_dinghuo)/10000)/2.82)[(i-1)\*48:i\*48]  leiji\_data\_list\_dinghuo = leiji\_min(year\_dinghuo)  # plt.subplot(221)  # plt.plot(leiji\_min\_data\_list)  plt.subplot(5, 1, i)  plt.bar(np.arange(1, 49, 1), year, 0.35, align="center", color="c", label="供应量", alpha=0.5)  plt.bar(np.arange(1, 49, 1)+0.35, year\_dinghuo, 0.35, color="b", align="center", label="订货量", alpha=0.5)  plt.xticks([1, 12, 24, 36, 48])  plt.legend(loc=9, ncol=2)  plt.subplots\_adjust(left=0.1, bottom=0.5, right=0.8, wspace=0.01, hspace=0.05)  plt.savefig(r'E:\桌面\2021国赛/当周是否满足一个标准周产能.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0)  plt.figure(figsize=(10,25))  for i in range(1, 6):  year = ((np.array(channeng\_week)/10000)/2.82)[(i-1)\*48:i\*48]  leiji\_data\_list = leiji\_min(year)  year\_dinghuo = ((np.array(channeng\_week\_dinghuo)/10000)/2.82)[(i-1)\*48:i\*48]  leiji\_data\_list\_dinghuo = leiji\_min(year\_dinghuo)  # plt.subplot(221)  # plt.plot(leiji\_min\_data\_list)  plt.subplot(5, 1, i)  plt.bar(np.arange(1, 49, 1), leiji\_data\_list, 0.35, align="center", color="c", label="供应量", alpha=0.5)  plt.bar(np.arange(1, 49, 1)+0.35, leiji\_data\_list\_dinghuo, 0.35, color="b", align="center", label="订货量", alpha=0.5)  plt.xticks([1, 12, 24, 36, 48])  plt.legend(loc=9, ncol=2)  plt.subplots\_adjust(left=0.1, bottom=0.5, right=0.8, wspace=0.01, hspace=0.05)  plt.savefig(r'E:\桌面\2021国赛/累计.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0)  plt.legend(loc='lower right')  plt.savefig(r'E:\桌面\2021国赛/pca.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0) | |

|  |  |
| --- | --- |
| Python代码 | 遗传算法求解至少选择多少家供应商 |
| from sko.GA import GA  import torch  device = torch.device("cuda:0" if torch.cuda.is\_available() else "cpu")  first\_week = data\_dinghuo\_df.values[pca\_sorted\_top\_50\_index][:, 1:]  material\_dict = {'A':0.6, 'B':0.66, 'C':0.72}  def schaffer(p):  # global W2  W2 = 4\*2.82  W1 = np.sum(p)  for i in range(1, 241):  for j in range(50):  if p[j]==1:  # print(first\_week[i][j])  W2 += first\_week[j][i]/10000/material\_dict[first\_week[j][0]]  # print(W2)  W2 = W2-2.82  W = 0.8\*W1/50 + 0.2\*pow(W2/2.82, 2)  return W  for i in range(1):  ga = GA(func=schaffer, n\_dim=50, size\_pop=50, max\_iter=200, prob\_mut=0.001, lb=np.zeros(50), ub=np.ones(50), precision=1)  ga.to(device=device)  best\_x, best\_y = ga.run()  print('best\_x:', best\_x, '\n',  'num\_x:', np.sum(best\_x), '\n',  'best\_y:', best\_y) | |

|  |  |
| --- | --- |
| Python代码 | 评价供货公司能力 |
| gonghuo\_ability = []  for i in range(len(data\_dinghuo\_use\_array)):  max\_ = 0  for j in range(240):  if data\_dinghuo\_use\_array[i][j]!=0 and data\_gonghuo\_use\_array[i][j]!=0:  if (data\_gonghuo\_use\_array[i][j]/data\_dinghuo\_use\_array[i][j]) <1:  # print(i, j)  if data\_gonghuo\_use\_array[i][j]>max\_:  # print(i, j)  max\_=data\_gonghuo\_use\_array[i][j]  # print(max\_)  gonghuo\_ability.append(max\_) | |

|  |  |
| --- | --- |
| Python代码 | 遗传算法求问题二最经济方案 |
| from sko.GA import GA  import torch  from tqdm import tqdm  device = torch.device("cuda:0" if torch.cuda.is\_available() else "cpu")  types = data\_gonghuo\_df.values[pca\_sorted\_top\_50\_index][best\_x==1][:, 1]  material\_dict = {'A':0.6, 'B':0.66, 'C':0.72}  cost\_dict = {'A':1.2, 'B':1.1, 'C':1.0}  def schaffer(ps):  # global W2  W2 = 4\*2.82  W1 = 0  ps = np.array(ps).reshape(-1, 24)  # print(ps.shape)  for i in range(24):  p = ps[i]  for j in range(len(p)):  W1 += ps[j][i]/10000\*cost\_dict[types[j][0]]  W2 += ps[j][i]/10000/material\_dict[types[j][0]]  # print(W2)  W2 = W2-2.82  W = 0.7\*W1 + 0.3\*pow(abs(W2/2.82), 1.5)  # print(W1, pow(W2/2.82, 2))  return W  ga\_ = GA(func=schaffer, n\_dim=np.sum(best\_x==1)\*24, size\_pop=100, max\_iter=300, prob\_mut=0.05, lb=list(np.zeros(np.sum(best\_x==1)\*24)),  ub=list(np.array(gonghuo\_ability)[pca\_sorted\_top\_50\_index][best\_x==1])\*24, precision=1)  # ga\_.to(device=device)  best\_x\_, best\_y\_ = ga\_.run()  print('best\_x:', best\_x\_, '\n',  'sum:', np.sum(best\_x\_.reshape(-1, 24), axis=0), '\n',  'best\_y:', best\_y\_) | |

|  |  |
| --- | --- |
| Python代码 | 导出问题2的订购数据 |
| t = use\_gongsi\_2  for i in range(len(t)):  t[i] = t[i].replace('S', '')  t = t.astype('int').reshape(-1, 1)  df = pd.DataFrame(best\_x\_.reshape(-1, 24))  df['E'] = t  df = df.sort\_values('E',inplace=False)  indexs = ['S%03d'%i for i in df.values[:, -1]]  df['E'] = indexs  df\_T = pd.DataFrame(df.values.T[:-1, :], columns=df.values.T[-1, :])  tem = pd.DataFrame(np.zeros((24, 402)), columns=['S%03d'%i for i in range(1, 403)])  for i in indexs:  tem[i] = df\_T[i]  # pd.DataFrame(tem.values.T).to\_csv(r'E:\桌面\2021国赛\CUMCM2021Probelms\C/订购2.csv')  计算平均损耗  sunhao\_df = pd.read\_excel(r"E:\桌面\2021国赛\CUMCM2021Probelms\C\附件2 近5年8家转运商的相关数据.xlsx")  sunhao\_list = []  for i in sunhao\_df.values[:, 1:]:  temp = []  for j in i:  if j!=0:  temp.append(j)  sunhao\_list.append([np.array(temp).mean(), np.array(temp).std(),len(temp)]) | |

|  |  |
| --- | --- |
| Python代码 | 规划问题二转运方案 |
| from pulp import \*  for o in tqdm(range(24)):  gongsi\_material = best\_x\_.reshape(-1, 24)[:, o]  yunshu\_cost = np.array(sunhao\_list)[:, 0]/100  num\_gongying = best\_x\_.reshape(-1, 24).shape[0]  # 1. 建立问题  m = LpProblem("Bleding Problem", LpMinimize)  # 2. 建立变量  var = [[LpVariable(f'x{i}{j}', lowBound=0, cat=LpInteger) for j in range(1, num\_gongying+1)] for i in range(1, 9)]  # 3. 设置目标函数  for i in range(1, 9):  m += (lpSum(var[i-1][j-1] for j in range(1, num\_gongying+1))\*yunshu\_cost[i-1])  # 4. 施加约束  for i in range(1, num\_gongying+1):  m += (lpSum(var[j-1][i-1] for j in range(1, 9))==gongsi\_material[i-1])  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))<=6000)  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))>=0)  # 5. 求解  m.solve()  # 6. 打印求解状态  # print("Status:", LpStatus[m.status])  # 7. 打印出每个变量的最优值  # for v in m.variables():  # print(v.name, "=", v.varValue)  # 8. 打印最优解的目标函数值  # print("Total Cost of Ingredients per can = ", value(m.objective))  m\_str = []  for i in m.variables():  m\_str.append(str(i))  for i in range(len(m\_str)):  locals()[m\_str[i]] = m.variables()[i].varValue  q\_2\_3 = []  for i in range(1, 9):  for j in range(1, num\_gongying+1):  q\_2\_3.append(locals()["x{}{}".format(str(i), str(j))])  q\_2\_3 = np.array(q\_2\_3).reshape(8, -1)  if o==0:  q\_2 = q\_2\_3.T  else:  q\_2 = np.hstack([q\_2, q\_2\_3.T]) | |

|  |  |
| --- | --- |
| Python代码 | 灵敏度分析1 |
| plt.figure(figsize=(10,25))  color = ['b3-', 'g-.', 'y.-', 'rx-.', 'k4-.']  for i in range(5):  plt.subplot(5, 1, i+1)  plt.plot(five\_fig[i], color[i], label='f1={}, f2={}'.format(weights[we], round(1-weights[we], 1)))  plt.legend()  plt.subplots\_adjust(left=0.1, bottom=0.5, right=0.8, wspace=0.01, hspace=0.05)  plt.ylabel('目标函数值')  plt.xlabel('迭代次数')  plt.savefig(r'E:\桌面\2021国赛\遗传算法选最经济1.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0)  W = 0.7\*W1 + 0.3\*pow(abs(W2/2.82), 1.5)  # print(W1, pow(W2/2.82, 2))  return W  ga\_ = GA(func=schaffer, n\_dim=np.sum(best\_x==1)\*24, size\_pop=100, max\_iter=300, prob\_mut=0.05, lb=list(np.zeros(np.sum(best\_x==1)\*24)),  ub=list(np.array(gonghuo\_ability)[pca\_sorted\_top\_50\_index][best\_x==1])\*24, precision=1)  # ga\_.to(device=device)  best\_x\_, best\_y\_ = ga\_.run()  print('best\_x:', best\_x\_, '\n',  'sum:', np.sum(best\_x\_.reshape(-1, 24), axis=0), '\n',  'best\_y:', best\_y\_) | |

|  |  |
| --- | --- |
| Python代码 | 灵敏度分析2 |
| plt.figure(figsize=(10,25))  color = ['b3-', 'g-.', 'y.-', 'rx-.', 'k4-.']  for i in range(5):  plt.subplot(5, 1, i+1)  plt.plot(five\_fig[i], color[i], label='{}次方'.format(weights[i]))  plt.legend(fontsize=20)  plt.ylabel('目标函数值', fontsize=14)  plt.subplots\_adjust(left=0.1, bottom=0.5, right=0.8, wspace=0.01, hspace=0.05)  plt.xlabel('迭代次数', fontsize=14)  plt.savefig(r'E:\桌面\2021国赛\遗传算法选最经济2.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0) | |

|  |  |
| --- | --- |
| Python代码 | 导出问题二转运方案 |
| t = use\_gongsi\_2  for i in range(len(t)):  t[i] = t[i].replace('S', '')  t = t.astype('int').reshape(-1, 1)  df = pd.DataFrame(q\_2)  df['E'] = t  df = df.sort\_values('E',inplace=False)  indexs = ['S%03d'%i for i in df.values[:, -1]]  df['E'] = indexs  df\_T = pd.DataFrame(df.values.T[:-1, :], columns=df.values.T[-1, :])  tem = pd.DataFrame(np.zeros((192, 402)), columns=['S%03d'%i for i in range(1, 403)])  for i in indexs:  tem[i] = df\_T[i]  # pd.DataFrame(tem.values.T).to\_csv(r'E:\桌面\2021国赛\CUMCM2021Probelms\C/转运2.csv') | |

|  |  |
| --- | --- |
| Python代码 | 材料分类排序 |
| sorted\_dinghuo = data\_dinghuo\_df.sort\_values("材料分类",inplace=False).values  sorted\_types = list(sorted\_dinghuo[:, 1].flatten())  print(sorted\_types.count('A'), sorted\_types.count('B'), sorted\_types.count('C')) # A 0:146 B 146:280 C 280:402 | |

|  |  |
| --- | --- |
| Python代码 | 遗传算法求问题三的订购方案 |
| from sko.GA import GA  import torch  from tqdm import tqdm  device = torch.device("cuda:0" if torch.cuda.is\_available() else "cpu")  weeks\_24 = []  for k in tqdm(range(2, 26)):  first\_week\_402 = sorted\_dinghuo[:, k]  material\_dict = {'A':0.6, 'B':0.66, 'C':0.72}  def schaffer(ps):  # global W2  wa = sum(ps[0:146])  wc = sum(ps[280:402])  w = 0  for i in range(len(ps)):  if ps[i]==1:  w += first\_week\_402[i]/10000/material\_dict[sorted\_types[i]]  # print(w)  W = 0.3\*np.exp(-wa)+0.2\*wc+0.5\*pow(abs(w/2.82-1), 1)  return W  ga\_\_\_ = GA(func=schaffer, n\_dim=402, size\_pop=50, max\_iter=50, prob\_mut=0.001, lb=np.zeros(402),  ub=np.ones(402), precision=1)  ga\_\_\_.to(device=device)  best\_x\_\_\_, best\_y\_\_\_ = ga\_\_\_.run()  print(  # 'best\_x:', best\_x\_\_\_, '\n',  'num\_A:', np.sum(best\_x\_\_\_[0:146]), '\n',  'num\_B:', np.sum(best\_x\_\_\_[146:280]), '\n',  'num\_C:', np.sum(best\_x\_\_\_[280:402]), '\n',  'best\_y:', best\_y\_\_\_)  weeks\_24.append(best\_x\_\_\_) | |

|  |  |
| --- | --- |
| Python代码 | 导出问题三订购方案 |
| gongsi\_material\_weeks\_24 = np.zeros\_like(np.array(weeks\_24).T)  temp = np.array(weeks\_24).T  # print(weeks\_24.shape)  for i in range(402):  for j in range(24):  if temp[i][j]==1:  gongsi\_material\_weeks\_24[i][j] = sorted\_dinghuo[:, 2:26][i][j]  # gongsi\_material\_weeks\_24  t = sorted\_dinghuo[:,0]  for i in range(len(t)):  t[i] = t[i].replace('S', '')  t = t.astype('int').reshape(-1, 1)  df = pd.DataFrame(gongsi\_material\_weeks\_24)  df['E'] = t  df = df.sort\_values('E',inplace=False)  df.to\_csv(r'E:\桌面\2021国赛\CUMCM2021Probelms\C/订购3.csv') | |

|  |  |
| --- | --- |
| Python代码 | 目标规划求问题三转运方案 |
| from pulp import \*  for n in tqdm(range(24)):  print(gongsi\_material\_weeks\_24.shape)  gongsi\_material = gongsi\_material\_weeks\_24[:, n]  yunshu\_cost = np.array(sunhao\_list)[:, 0]/100  num\_gongying = 402  # 1. 建立问题  m = LpProblem("Bleding Problem", LpMinimize)  # 2. 建立变量  var = [[LpVariable(f'x{i}{j}', lowBound=0, cat=LpInteger) for j in range(1, num\_gongying+1)] for i in range(1, 9)]  # 3. 设置目标函数  for i in range(1, 9):  m += (lpSum(var[i-1][j-1] for j in range(1, num\_gongying+1))\*yunshu\_cost[i-1])  # 4. 施加约束  for i in range(1, num\_gongying+1):  m += (lpSum(var[j-1][i-1] for j in range(1, 9))==gongsi\_material[i-1])  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))<=6000)  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))>=0)  # 5. 求解  m.solve()  # 6. 打印求解状态  # print("Status:", LpStatus[m.status])  # 7. 打印出每个变量的最优值  # for v in m.variables():  # print(v.name, "=", v.varValue)  # 8. 打印最优解的目标函数值  # print("Total Cost of Ingredients per can = ", value(m.objective))  m\_str = []  for i in m.variables():  m\_str.append(str(i))  for i in range(len(m\_str)):  locals()[m\_str[i]] = m.variables()[i].varValue  q\_3\_2 = []  for i in range(1, 9):  for j in range(1, num\_gongying+1):  q\_3\_2.append(locals()["x{}{}".format(str(i), str(j))])  q\_3\_2 = np.array(q\_3\_2).reshape(8, -1)  if n==0:  q\_3 = q\_3\_2.T  else:  q\_3 = np.hstack([q\_3, q\_3\_2.T]) | |

|  |  |
| --- | --- |
| Python代码 | 导出问题三转运方案 |
| t = sorted\_dinghuo[:,0]  for i in range(len(t)):  t[i] = t[i].replace('S', '')  t = t.astype('int').reshape(-1, 1)  df = pd.DataFrame(q\_3)  df['E'] = t  df = df.sort\_values('E',inplace=False)  df.to\_csv(r'E:\桌面\2021国赛\CUMCM2021Probelms\C/转运3.csv') | |

|  |  |
| --- | --- |
| Python代码 | 蒙特卡洛求解问题四的订购方案 |
| xx = []  yy = []  xx\_ = []  for hh in range(25):  q4\_weeks\_24s = np.array([np.random.randint(0,2) for i in range(24\*10\*402)]).reshape(24, 10, 402)  xx\_.append(q4\_weeks\_24s)  q4\_weeks\_24s\_channeng\_0 = []  for ps in q4\_weeks\_24s[0]:  w = 0  for i in range(len(ps)):  if ps[i]==1:  w += first\_week\_402[i]/10000/material\_dict[sorted\_types[i]]  q4\_weeks\_24s\_channeng\_0.append(w)  xx.append(q4\_weeks\_24s\_channeng\_0)  q4\_q4\_gongsi\_material\_weeks\_24 = []  for q4\_weeks\_ in tqdm(range(q4\_weeks\_24s.shape[1])):  q4\_weeks\_24 = q4\_weeks\_24s[:, q4\_weeks\_, :]  # print(q4\_weeks\_24.shape)  q4\_gongsi\_material\_weeks\_24 = np.zeros\_like(np.array(q4\_weeks\_24).T)  q4\_temp = np.array(q4\_weeks\_24).T  for i in range(402):  for j in range(24): #######################################################24  # print(i, j)  if q4\_temp[i][j]==1:  q4\_gongsi\_material\_weeks\_24[i][j] = sorted\_dinghuo[:, 2:26][i][j]  q4\_q4\_gongsi\_material\_weeks\_24.append(np.array(q4\_gongsi\_material\_weeks\_24))  q4\_q4\_gongsi\_material\_weeks\_24 = np.array(q4\_q4\_gongsi\_material\_weeks\_24)  from pulp import \*  q4\_cost = []  for x in tqdm(range(q4\_q4\_gongsi\_material\_weeks\_24.shape[0])):  q4\_gongsi\_material\_weeks\_24 = q4\_q4\_gongsi\_material\_weeks\_24[x]  temp\_cost = []  for n in tqdm(range(1)):#################################24  gongsi\_material = q4\_gongsi\_material\_weeks\_24.T[n, :]  yunshu\_cost = np.array(sunhao\_list)[:, 0]/100  num\_gongying = 402  # 1. 建立问题  m = LpProblem("Bleding Problem", LpMinimize)  # 2. 建立变量  var = [[LpVariable(f'x{i}{j}', lowBound=0, cat=LpInteger) for j in range(1, num\_gongying+1)] for i in range(1, 9)]  # 3. 设置目标函数  for i in range(1, 9):  m += (lpSum(var[i-1][j-1] for j in range(1, num\_gongying+1))\*yunshu\_cost[i-1])  # 4. 施加约束  for i in range(1, num\_gongying+1):  m += (lpSum(var[j-1][i-1] for j in range(1, 9))==gongsi\_material[i-1])  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))<=6000)  for j in range(1, 9):  m += (lpSum(var[j-1][i-1] for i in range(1, num\_gongying+1))>=0)  # 5. 求解  m.solve()  # 6. 打印求解状态  # print("Status:", LpStatus[m.status])  # 7. 打印出每个变量的最优值  # for v in m.variables():  # print(v.name, "=", v.varValue)  # 8. 打印最优解的目标函数值  # print("Total Cost of Ingredients per can = ", value(m.objective))  m\_str = []  for i in m.variables():  m\_str.append(str(i))  for i in range(len(m\_str)):  locals()[m\_str[i]] = m.variables()[i].varValue  q\_3\_2 = []  for i in range(1, 9):  for j in range(1, num\_gongying+1):  q\_3\_2.append(locals()["x{}{}".format(str(i), str(j))])  q\_3\_2 = np.array(q\_3\_2).reshape(8, -1)  if n==0:  q\_3 = q\_3\_2.T  else:  q\_3 = np.hstack([q\_3, q\_3\_2.T])  per\_total\_cost = 0  for i in range(len(q\_3\_2)):  per\_total\_cost += sum(q\_3\_2[i])\*yunshu\_cost[i]  temp\_cost.append(np.array(per\_total\_cost))  q4\_cost.append(temp\_cost)  yy.append(q4\_cost) | |

|  |  |
| --- | --- |
| Python代码 | 导出问题4订购方案 |
| gongsi\_material\_weeks\_24 = np.zeros((402, 24))  # print(weeks\_24.shape)  for i in range(402):  if np.array(xx\_)[3, :, 4, :][0][i]==1:  gongsi\_material\_weeks\_24[i] = sorted\_dinghuo[:, 2:26][i]  gongsi\_material\_weeks\_24 = gongsi\_material\_weeks\_24\*2  t = sorted\_dinghuo[:,0]  for i in range(len(t)):  t[i] = t[i].replace('S', '')  t = t.astype('int').reshape(-1, 1)  df = pd.DataFrame(gongsi\_material\_weeks\_24)  df['E'] = t  df = df.sort\_values('E',inplace=False)  df.to\_csv(r'E:\桌面\2021国赛\CUMCM2021Probelms\C/订购4.csv')  plt.figure(figsize=(3, 3))  plt.scatter(np.array(xx).flatten()\*2.82, np.array(yy).flatten(), s=100, alpha=0.5)  plt.xlabel('产能（万立方米）')  plt.ylabel('损耗（立方米）')  plt.savefig(r'E:\桌面\2021国赛\蒙特卡洛.png', dpi=600, bbox\_inches='tight', pad\_inches=0.0) | |