**基于动态模拟规划与映射匹配模型的材料订购与运输方案设计**

## 摘 要

原材料的订购与运输方案对企业的生产运营起着重要作用。本文从近五年企业订购与供应商供货关系入手，构建供货稳定性等多项评价指标，从而对供应商建立了基于熵权法的重要性综合评价模型，选出最重要的供应商；基于供应特征分别建立供应商供货预测模型，在此基础上进一步建立订购、转运方案的单/多目标规划模型，制定出合理的订购转运方案。

针对问题一，本文对于不同供应商的供货差异性过大的问题，首先设置了供应规模第一优先、供应期望第二优先的初步考察筛选机制，筛去了供应规模极小和远低于供应期望的供应商。对于筛选后的58家供应商进一步建立了基于熵权法的重要性综合评价模型，从供应市场份额占比，有效周的供应能力，供货稳定性三个方面对供应商进行量化评价，最后得出重要程度最高的前50家供应商。

针对问题二，首先，本文对问题一中50家供应商的有效周产能进行降序选择，发现前19家恰好能满足生产需求，但是存在供应稳定性较低的供应商，所以最终需要至少25家供应商。其次对这25家的供应特征进行分析，将其分为稳定型、周期型、离散型，分别以正态分布、三次指数平滑、一定概率出现的正态分布来模拟其未来24周的供应量。进一步建立了以花费最少为目标，具有两周产能的库存为约束，考虑供订比的供货量为订货量上限的单目标动态模拟规划模型。对于求解出的最经济24周订购方案，本文以损耗成本最少化为第一优先、供应商与转运商的映射匹配为第二优先准则，建立了订购计划的**最优匹配转运机制**，制定了损耗最少的转运方案。

针对问题三，本文在第二问所建立的单目标规划模型的基础上，加入了最少转运损失成本和最大仓储产能成本这两个目标，构建了**多目标动态模拟规划模型**，以尽可能多采购A材料，少采购C材料。根据未来24周预测的供货情况，模拟出总成本最低的订购计划，出现了大型供应商持续供应、小型C材料供应商向小型A材料供应商转移的现象。并根据优先准则匹配机制安排转运方案，平均每周损耗产能为106.56立方米，占每周产能0.38%，其转运损耗率较低，效果理想。

针对问题四，本文首先根据第二问模拟得到的供应商的供货量建立了动态产能潜力模型，通过动态模拟每周的最小产能需求，得到企业在满足每周生产需求后产能可以提升到30380，提高了近7.7%。以此制定订购计划，因为提高了产能，于是本文解开经济限制，以周产能最大化为目标，并将每周转运的损耗会减少周供量加入到约束中。发现订购方案中A类订购总量为200441立方米、B类150238立方米、C类131939立方米，分析认为A材料性价比最高，同时制定的转运方案效果较为理想。

**关键词：熵权法 重要性评价模型 映射匹配 单/多目标规划 供应分类预测模型**

# 问题重述

## 问题背景

对于生产制造类型的企业，其企业的运作发展、产品的生产周期、成本的利润盈亏计算等，不仅与其产品的制造方法、生产过程、定价销售等因素有关，也与产品原材料的不同种类、对于不同的供应商的不同的订购量的订购方案以及从原料产地运输到企业仓库的运输成本息息相关。生产企业原材料的订购与运输往往决定着该企业的输出产能、消耗成本、未来运作计划，从而直接影响到企业的长远发展。因此，识别判断出与企业联系紧密的、供货稳定供货商，对原材料制定合理的运输方案和采购方案对企业的长远发展有重要意义。

## 待解决问题

（1）已知某企业5年来分别在402家供货商的订货量和各供货商的实际供货量，根据以上数据建立判断供货商重要性的数学模型，确定50家对该企业最重要的供货商。

（2）已知8种转运商240周的运输损耗率，根据问题一所得结果，选择尽可能少的供货商以尽可能满足该企业每周2.82万立方米的产能的两周原材料库存量，并制定未来24周利润最大的订购方案和原材料损耗最小的运输方案，分析其实施效果。

（3）企业可通过尽量多采购A类和尽量少采购C类原材料以减少成本，在保持运输损耗率尽量低的情况下，定制新的订购方案和运输方案，并分析其效果。

（4）当该企业可提高每周输出产能时，根据已知条件和上述问题结果，分析出供应商的最大供应量，以确定该企业可提高的产能，并给出未来24周的订购和转运方案。

# 问题分析

## 问题一的分析

针对问题一，由于附件一提供了各个供货商每周收到的该企业的订货量和对该企业的实际供货量，对数据初步观察，存在部分订货量和供货量都很小的数据或者供货量难以满足订货量的现象，以此确定考察筛选机制对数据进行剔除和保留。同时订货量和供货量的数值可以反映出不同种类的供货市场份额和不同厂商的供应期望、供应稳定性等因素，针对以上因素建立评价分析模型即可判断出经过筛选后的各个厂商对于该企业的重要程度，确定50家最重要的供货商。

## 问题二的分析

针对问题二，题目首先要求选择尽量少的供货商以满足生产和库存的需求，可以在问题一的基础上，从对企业重要程度高的50家供货商中选取供货量多的供货商。对于制定最经济的订购方案，订货量的采购价格可作为目标函数，生产、库存的原料需要以及供货商的可能供货量可作为约束，构建目标归化模型对每周订购方案进行求解，其中供货量可由附件一中的数据通过时间序列预测的方法得到。对于制定损耗最少的转运方案，可先由附件二的数据求出各个运输商的损耗率，从损耗的经济角度出发，由于原材料不同，应尽量安排损耗率低的运输商运输价格高的原材料，实现损耗最小。

## 问题三的分析

针对问题三，制定新的订购与转运方案从问题二确定的供应商出发，一方面减少仓储的成本需要多采购A类原材料，少采购C类原材料，另一方面需要减少转运的成本，可以在问题二的目标规划模型的基础上进行改进，增加相应的目标函数。转运方案的制定要求同问题二，同样需要转运损耗率尽量少，则其转运的原则不变，同样以损耗率低的运输商运输价格高的原材料。其方案的实施效果可与问题二进行对比分析。

## 问题四的分析

针对问题四，在企业提高每周产能的条件下，不需要考虑在订购原材料时的经济限制，一定程度上增多了供应商的数量。根据供应商和转运商的实际情况，供应量在转运中实际产生了损耗，会对产品的产量和库存量以及订购量的制定产生影响。企业每周提高的产能则是依据实际情况下的供货量进行计算。转运方案也随着企业产能提高提高了转运量，转运时损耗率低的转运商则优先运输产能转化率高的原材料。

## 总体分析

根据上述的研究思路，作出本文的整体思路框图，如图1所示。

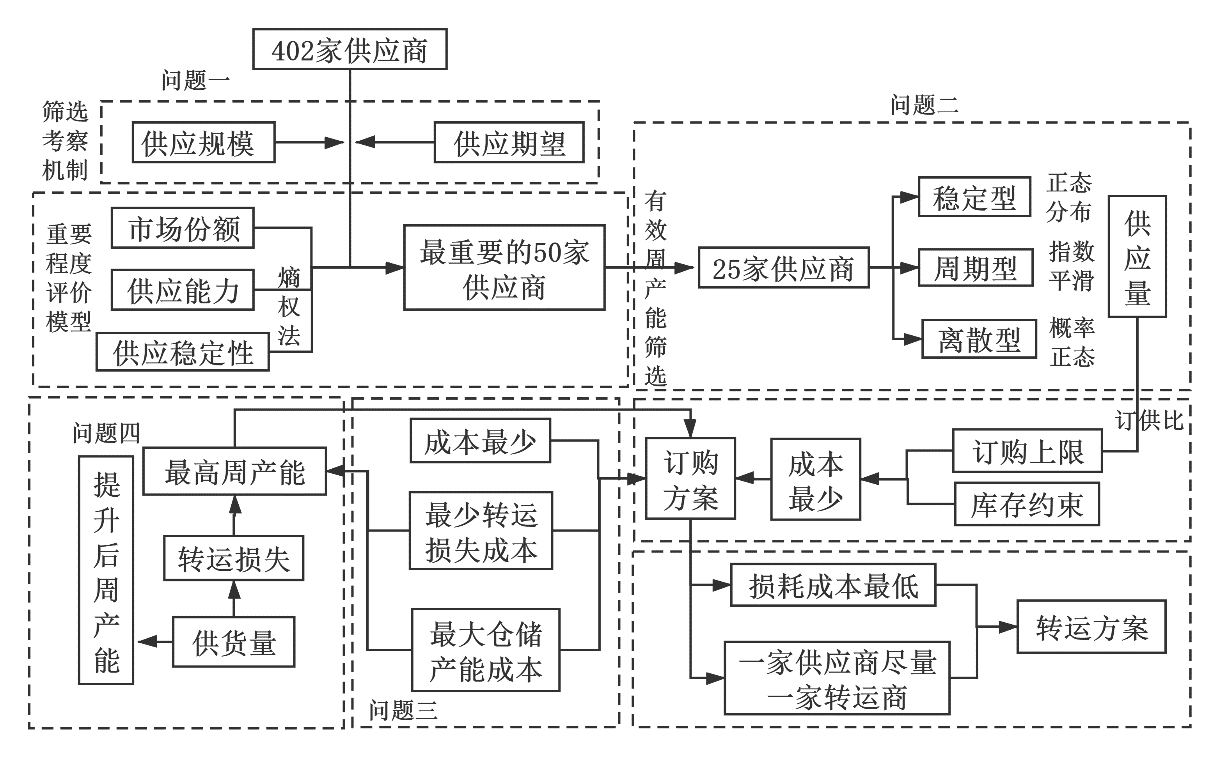


图1 整体思路框图

# 模型假设

1．假设每周转运商转运货物的时间上不会影响企业生产安排。

2．假设每家转运商可以给多家供应商提供服务。

3．假设企业在利用原材料生产时不会出现材料浪费等情况。

# 符号说明

根据本文我们建立的模型，确定以下变量，并给予相应的符号说明，具体如下表1所示。

表1 符号说明表

|  |  |
| --- | --- |
| 符号 | 说明 |
|  | 供应商在第周的供货量 |
|  | 企业在供应商的订货量 |
|  | 供应商的供应期望 |
|  | 供应商的有效周供应期望 |
|  | 供应商的供货稳定性 |
|  | 供应商的市场份额 |
|  | 各指标信息熵 |
|  | 各指标所占权重 |
|  | 企业第周在供应商订货方案 |
|  | 企业第周采购原材料总花费 |
|  | 企业第周材料供货量的产出量 |
|  | 企业第周材料订货量的产出量 |
|  | 企业第周生产后的库存量 |
|  | 转运商第周在供应商的损耗率 |
|  | 某种原材料的总转运损耗量 |
|  | 提高后的每周产能产能 |

# 模型建立与求解

## 问题一

### 模型建立：基于熵权法的供应商重要性模型

首先本文根据附件一数据特点，以总供货量和供应期望建立考察筛选机制，对供应商进行初步筛选。同时，为避免部分供应商的个别周较小或0的订货量与供货量影响评价结果产生误差，本文剔除掉以上数据称剩下的供货周为“有效周”，有效周的数据对于判断供应商的重要性有较强的说服力。本文在有效周的基础上建立了供应商的供货稳定性、有效周的供应期望和三种原材料所占不同的市场份额三个评价指标，最终以熵权法[1]对供应商的重要程度进行评价。

#### （1）数据的初步筛选：考察筛选机制

本文建立考察筛选机制从供应商的总供货量和供应期望出发，在保证总供货量的前提下对其供应期望进行考察，假设供应商的总供货量为第一优先度，供应商的供应期望为第二优先度，供货商的筛选结果记为：



**a.第一优先度**

由于该企业收购的原材料需要保证两周的生产量，因此总供货量大的供应商对企业的重要性较高，虽然此类供货商常出现供货量不满足于订货量的情况，如供货商S201，对其进行保留。同理，对总供供货量较少的供应商筛选剔除。设编号供货商的第周的供货量为，总供货量为，制定第一优先度筛选规则如下：



**b.第二优先度**

对于的总供货量，则从供应期望的角度对其筛选，假设编号供货商的供应期望为，总订货量为，则有：



式中以供货量与订货量之比表示供应期望，比值的大小直接反映出供应商的供应期望，供应期望强的供应商能够更好的满足企业的原材料需求，如供货商S097供应期望低应当被剔除。制定第二优先度筛选规则如下：



综上，本文建立初步供货商考察筛选模型为：



#### （3）订货量与供货量确定的有效周

通过供货商考察筛选模型从总供货量和供应期望对供货商进行第一步筛选后，得到的数据中由于企业的进货量需求不同，同一供货商在不同周的供货量之间有较大的差距，部分结果如表2所示。

表2 部分筛选后供应表（单位：立方米）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 供应商ID | 材料种类 | W001 | W002 | W003 | W004 | W005 | W006 | W007 |
| S003 | C | 8 | 1 | 0 | 0 | 0 | 1 | 5 |
| S005 | A | 37 | 62 | 60 | 65 | 76 | 76 | 65 |
| S007 | A | 9 | 94 | 37 | 102 | 25 | 21 | 22 |
| S031 | B | 5 | 84 | 69 | 79 | 97 | 103 | 84 |
| S037 | C | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

由表2可知，如供应商S003和S037，其供应总量分别为13138和50686立方米，但是其供应数据中出现有大量0或1等此类数值较小的供货周，当对供应商的重要程度做出综合评价时，为避免此类数据的影响，本文筛选出供货量高于一定数值的供货周作为“有效周”用于对各个供应商做进一步的比较分析。经过分析，不同材料种类最低有效周供应量不同，如表3所示。

表3 有效周最低供应量（单位：立方米）

|  |  |
| --- | --- |
| 材料种类 | 有效周的最低供应量 |
| A | 3 |
| B | 4 |
| C | 5 |

#### （2）供应商重要程度评价指标：

在判别有效周的基础上，本文根据企业与供应商之间的运作方式特点建立出以下三个指标用于衡量供应商对于该企业的重要程度。

**a.****供货稳定性**

当某一供货商供货的有效周数量足够多时，即使在供应量小于订货量的情况下，也可以认为该供货商在一定程度上可满足该企业的订单需求，且供货源货足够稳定。供货商的有效周数越多，则说明该供货商的原料储配多，提供原料更加稳定。记编号的供应商的供货稳定性为，得：



其中为有效周个数。

**b. 有效周的平均供应期望**

设供应商的有效周的供应期望为，由式可从总供货量的角度对供应商的供应期望作出筛选，但由于其中存在着大量的由0与1构成的无效周，使得，因此可从有效周供货量的角度对各供应商作出评价。供应商的有效周平均供应期望为：



其中为周编号，为有效周集合。

**c.****供货商的供货市场份额**

由于供应商提供的原材料种类不同，直接对所有供应商的供应量采取同种方法分析可能出现偏差，因此本文针对供应商的原材料种类设置了供货商的供货市场份额这一指标减轻其影响。

依据题意，该企业产品的生产原材料可分为A、B、C三个种类，同一供应商只提供一种类型的原材料。经过统计，提供给该企业的不同种类的原材料的总供货量的及其占比分布如图2所示。

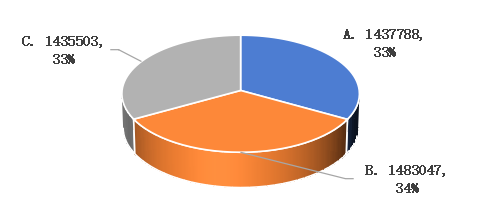


图2 三种原材料供货量比例图

设各原材料在市场中的占比为，则有：



式可从一定程度上反映A、B、C三种原材料的产量，从数值大小中可看出原材料获取的难度为。

同时，供应商在提供同一种类原材料的供应商之间的占的比例也不同，设供应商占同种材料供货比例为，列举部分供应商占比数据如表4所示。

表4 部分供应商占同种材料供应量比例（单位：立方米）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 材料种类 | 总供货量 | 供应商ID | 供应商供货量 | 占同种材料供货比例 |
| A | 1437788 | S005 | 6912 | 0.0048 |
| S007 | 6948 | 0.0048 |
|  |  |  |  |  |
| B | 1483047 | S031 | 41207 | 0.0277 |
| S038 | 31905 | 0.0215 |
|  |  |  |  |  |
| C | 1435503 | S003 | 13138 | 0.0092 |
| S037 | 50686 | 0.0353 |

综上供货商的供货市场份额指标同时受不同原材料种类的比例与占同种材料供应量比例影响。设供货商的供货市场份额为，则有：



#### （3）基于熵权法的重要程度评价模型

上文通过对供应商经过考察筛选机制筛选后，建立了供货稳定性、平均有效周的供应期望、供货市场份额三个指标对供货商的重要程度进行考量，由于各指标的制定方法、代表意义各不同，本文选择熵权法分别对各指标进行赋权，便于同时从这三个指标对供货商的重要性进行评价。以供货市场份额为例，使用熵权法对其赋权的步骤如下：

**a.数据预处理**

首先需对供货市场份额去量纲化处理，消除不同指标之间量纲的影响。设供货市场份额有个待处理数据：



得供货市场份额经标准化处理后的值为：



**b.计算各指标占总指标比例**

计算在中所占的比重：



**c.计算各指标信息熵**

根据信息熵的定义，计算供货市场份额的信息熵：



其中当时，。

**d.计算各指标权重**

通过计算信息冗余度计算供货市场份额所占权重。供货市场份额信息冗余度为：



根据式-，可分别计算出供货稳定性的冗余度和平均有效周的供应期望的冗余度，求得供货市场份额的指标权值为：



同理可计算出效周供货稳定性的权值和平均有效周的供应期望的权值

综上，设供货商的重要性评价模型为：



其中、、都经过了标准化处理，供应商的重要程度值大小反映了该供应商的重要性。

### 模型求解

#### （1）各指标权重值

经过计算得各指标权重如表5所示。

表5 各指标权重表

|  |  |
| --- | --- |
| 指标类型 | 权重 |
| 市场份额 | 0.6570 |
| 供货稳定性 | 0.1349 |
| 有效周供应期望 | 0.2081 |

#### （2）模型结果

对该企业50家最重要的供应商见附录附表2。选取重要性前10位的供应商列出相关指标如表6所示。

表6 前10家最重要的供应商信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 供应商ID | 材料种类 | 市场份额 | 供货稳定性 | 有效周供应期望 | 重要程度 |
| S229 | A | 0.247 | 1 | 0.994 | 0.8108 |
| S140 | B | 0.21 | 0.783 | 1.149 | 0.7808 |
| S361 | C | 0.229 | 1 | 0.992 | 0.7585 |
| S108 | B | 0.168 | 1 | 0.984 | 0.5891 |
| S139 | B | 0.106 | 0.9 | 1.253 | 0.5775 |
| S151 | C | 0.135 | 1 | 0.983 | 0.5011 |
| S282 | A | 0.118 | 0.996 | 1.009 | 0.4695 |
| S340 | B | 0.119 | 1 | 0.999 | 0.4676 |
| S275 | A | 0.11 | 1 | 1.003 | 0.4461 |
| S329 | A | 0.109 | 1 | 1.002 | 0.4413 |

## 问题二

### 模型建立

本文以平均有效周的供应量作为供应商的选择标准，选择平均有效周供应量多的供应商使选择的个数尽量减少，考虑到实际条件下存在供货无效周的情况，对此进行了补偿处理。订购方案的制定，本文基于经济要求、生产要求、供货量的限制建立单目标规划模型，其中本文给出了供货量随时间变化的三种时间序列模型：稳定型、周期型、离散型，以约束制定的订货量。转运方案的制定，在求出各转运商的损耗率后，从原材料经济损耗的角度出发建立了原材料转运的动态规划算法。

#### （1）供应商的选择：优先供货量多的供应商

**a.理想条件的周供货量**

由于要尽可能减少选择供货商的数量以满足每周产能2.82万立方米的需求，所以选择的供货商的供货量应尽量大。可以认为当在理想条件下，供货商的每周供货量都是有效周时，此时的供货商的供货量最大，以每一供货商的平均有效周供货量作为选择依据，若供货商的有效周集合为M，平均有效周供货量为：



不同原料产出每立方米产品消耗不同，则各种类原材料的平均有效周产能为：



**b.供货商的排序选择**

本文在问题一的基础上，对从筛选出的50家对企业最重要的供货商以平均有效周产能进行排序如图3所示。

图3 50家供货商均有效周供货量排序图

由图3依次对供货商的均有效周供货量从高到低遍历选择，选择的供货商应满足以下条件：



#### （2）供货量的预测：三种供应量模型及预测方法

供货商所能提供的供货量会对企业的订购量产生影响，本文在未来24周的订购方案制定中，需要结合供货商的供应期望对订购的方案进行约束和规划。为了预测未来24周供货商的实际供货量，本文绘制出各个供应商的供货量曲线图，并将其按照其供货量的时间序列分布规律将其分为稳定型、周期型与离散型三种类型，其分布特点及预测方法如下：

**a.稳定型：正态分布的蒙特卡洛模拟**

以供货商S275为例，其整体的供应稳定程度高，且无周期性供应规律。在剔除某些异常值后，通过统计分析可知稳定型供货量呈正态分布规律，如图4和图5所示。

图4 S275稳定型供货量曲线图

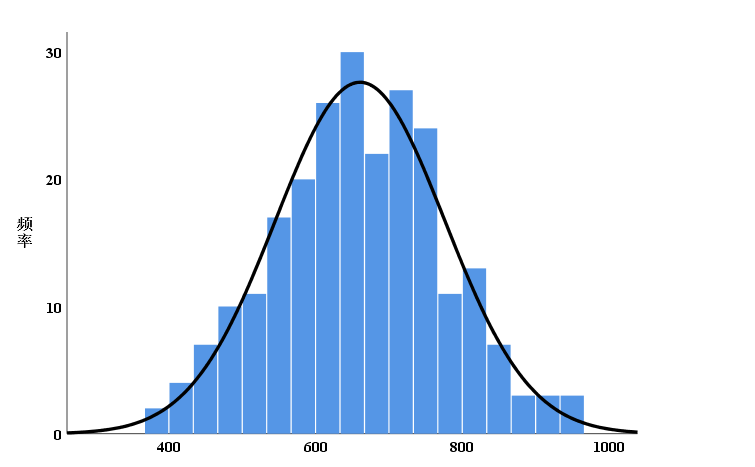


图5 S275正态分布例图

对于呈稳定性性分布的未来24周供货量的预测，统计出正态分布的均值和标准差后，使用蒙特卡洛模拟即可进行预测。

**b.周期型：三次指数平滑法**

以供货商S282为例，可以看出供货量随着时间在不断地增加和减少，具有显著的周期性供应规律。分布规律如图6所示。

图6 S282周期型供货量曲线图

对于呈周期性分布的未来24周供货量的预测，本文使用三次指数平滑法[2]进行预测，结果如图6橙色曲线所示。

**c.离散型：有效周正态分布的蒙特卡洛模拟**

以供货商S359为例，虽然供货量整体趋势也在随着时间不断地增加或减少，但其供应稳定程度低，无周期性供应规律，如图7所示。

图7 S395离散型供货量曲线图

观察图7知，离散型与稳定型分布规律相似，离散型曲线分布的绿色标记部分呈正态分布规律，橙色标记部分为无效周在供应时间中的占比，当供货稳定性为时，未来24周中供货量呈离散型正态分布规律概率为。离散型供货量的预测方法加入了有效供货周的稳定性的影响因素，使经过蒙特卡洛模拟后的供货量呈离散型分布。

#### （3）制定订购方案：单目标动态模拟规划模型

**a.决策变量**

根据题意，设为决策变量，即企业在在第周个供货商处的订货量：



其中表示供货商编号，表示订货量的周数，为至少供应商的数量。

**b.目标函数**

问题二要求设置最经济的订购方案，即当供货量满足了两周的产能需求时，尽量减少采购材料的花费。为简化模型，本文仅设置原材料的采购总价为目标函数。题目给出三类原材料采购单价之比为，则设第周的企业采购原材料的总花费为，总采购花费应尽可能的小：



其中分别表示提供不同原材料供应商编号的集合。

**c.约束条件**

该企业每周的订货量需要满足两周生产需求的原材料库存，即原材料需满足立方米的产出。设第周供货量的原材料产出为，订货量原材料产出为。根据不同原料产出每立方米产品的消耗不同有：



依据题意，需要制定紧接于附件一的240周之后的24周的订购方案，经过计算实际情况下，附件一的240周总订货量不足以满足每周的2.82万立方米的产出和两周的库存量，故本文假设第一周前一周的库存量为0。记每周生产后第周的库存量为，对第一周则有：



对第周需要有：



除了生产和库存的需求对订货量的约束外，也存在上文所得的各个供货商的供货量预测值对订货量的约束。设预测得第周编号为的供货商的供货量预测值为，将预测值乘以前240周的总的订供比所得作为未来24周的订货量的约束：



在求解过程中发现供应商S126与S201的订供比过大，考虑制造业供应链[3]的实际供应与企业需求，本文将其系数修正为1.3。

综上所述，建立未来24周的最经济订购方案的单目标动态模拟规划模型如下：



其中为选择的至少数量供应商的编号集合。

#### （4）制定转运方案

**a.转运商的损耗率**

对于转运方案的制定首先应根据附件二近五年8家转运商的损耗率求出实际平均损耗率，记为：



其中编号为的转运商的非0损耗率的个数为，排除掉转运商运货量为0时的情况。

**b.材料与转运商的排序分配**

为了使原材料转运损耗量最小，本文从三种原材料价格出发，尽量使价格高的材料让损耗率低的转运商运输。现制定材料分配与不同运输商的规则如下：

**规则1**：上一个种类的材料全部分配完成后才进行下一个种类的材料分配，顺序依次为A、B、C，优先分配给有剩余容量的损耗率最小的转运商。

**规则2**：将供货量从大至小排序，优先筛选出供货量大于6000的原材料，分配给此时剩余容量为6000且损耗率最小的转运商，多出的材料重新进入排序。

**规则3**：在分配过程中，当供货量前的两位之和大于6000时，则按多少分别分配给剩余容量充足的损耗率最小的转运商。

**规则4**：在分配过程中，若损耗率最小的转运商容量不够时，则多余的材料不进入排序，直接分配给剩下损耗率最小的转运商。

基于上述制定材料转运方案的规则，保证损耗率最小的前提下，也可使同一供应商的原材料安排尽量少的转运商转运，符合题目要求。

### 模型求解

#### （1）至少选择的供应商

在求解过程中，本文通过对附件一402家供应商的供货量数据的分析，计算实际情况下的平均每周总供货量为：



由于实际条件下供应商存在着无效周的供货量的情况，使得实际供货量小于每周28.2万立方米的产能需求。同时，如供货商S201、S395、S307的总供应量虽然较大，但其供应稳定性较小，因此本文通过式的供货稳定性，量化出无效周对总供货量的影响，并以此扩大挑选供应商的集合。

通过式，根据平均有效周产能从大到小依次累加，初步选择出19家供货商。由于这19家供货商的实际供货中会存在无效供货周，本文依据式供应商的供货稳定性，人为依次继续添加6家供货商以补偿缺失的供货量。本文选择的25家供应商ID结果如表7所示。

表7 至少选择的25家供应商ID表

|  |  |
| --- | --- |
| 19家初步选择的供应商ID | S201、S140、S229、S361、S395、S108、S307、S282、S151、S275、S329、S340、S139、S131、S308、S330、S126、S348、S356 |
| 6家补偿选择的供应商ID | S268、S306、S037、S352、S194、S143 |

#### （2）最经济的订购方案

首先对各供应商的未来供应量进行预测，本文对25家供应商在前240周的供应量规律如表8所示。

表8 供应商供应量模型分类表

|  |  |
| --- | --- |
| 供应量模型 | 供应商ID |
| 稳定型 | S361、S151、S275、S131、S308、S330、S352、S194 |
|  |  |
| 周期型 | S307、S282、S139、S229、S108、S329、S340、S356、S268、S306、S143、S140、S348、S037 |
|  |  |
| 离散型 | S201、S126、S395 |

根据表8对这25家供应商未来的供应量进行预测后，通过单目标规划模型式得出未来24周的订购方案见附件A。

在对式求解的过程中，制定订货量的顺序应优先从订供比小的供应商开始制定，使得较大货量的订单能够更能够被满足，对于供应期望较差的供应商S126与S201订单量的制定应处于最末端。同理，由于A、B、C三个种类的性价比递减，在定制中应使性价比高的原材料的订购量尽量的大。

#### （3）损耗最小的转运方案

通过计算得各转运商得平均损耗率如表9所示。

表9 转运商损耗率表

|  |  |  |  |
| --- | --- | --- | --- |
| 转运商ID | 损耗率(%) | 转运商ID | 损耗率(%) |
| T3 | 0.09 | T4 | 1.57 |
| T6 | 0.54 | T1 | 1.90 |
| T2 | 0.92 | T7 | 2.08 |
| T8 | 1.01 | T5 | 2.89 |

按照材料与转运商的排序分配规则依次制定各供应商材料的运输方案时，涉及到规则3的情况极少，但其有利于尽量使每一供应商都由同一家转运商转运。依据上文求出的订货量方案定制转运方案见附件B。

#### （4）实施效果分析

**a.订购方案实施效果**

制定以上最经济订购方案后，企业在稳定每周产能2.82万立方米并保持两周产能的库存量的条件下，得每周剩余库存产能和订购总量产能关系如图8所示。

图8 库存产能与总订购量产能关系图

由图8可以看出，订购量与剩余库存之间存在相关关系，当剩余库存量变化增加时，定购量有一定程度的减少；当剩余库存量呈减小的趋势时，定购量则有一定程度的增加，符合生产类型企业的订购与仓储规律。

**b.转运方案实施效果**

基于订购方案，制定出转运方案的每周转运商数量变化与周订购总量关系如图9所示。

图9 转运商数量变化与总订购量关系图

由图9可以看出，转运方案的大多数情况下，转运任务多由损耗率最小的前五家转运商承担。且每周转运材料的转运商数量变化趋势服从订购量的变化，统计所求该转运方案的平均每周损耗产能为99.92立方米，占每周产能0.3%，其转运效果较好。

## 问题三

### 模型建立：新目标函数的确定

本文认为问题三中企业压缩生产成本主要通过以下两种方式：减小仓储成本和减小转运成本。仓储成本主要的影响因素是库存中原材料的总体积，一般仓储成本与库存原材料体积成正比，除了采用通过直接将原材料输出为产能降低库存体积的方式减小仓储成本，还可以通过调节企业订货量降低单位产量的储存费用的方式减小仓储成本。对于转运成本，则在制定订货量方案时可以量化出因为转运损耗产生的经济损失，企业可以通过尽量减少损耗经济损失来约束订货方案的制定。

#### （1）单位产量储存费用

同式，以原材料订货量作为储存量用于计算企业的实际产能，记第周的储存量可产能：



设该批原材料所占体积为：



则单位产量的储存费用为：



式中为单位体积的仓储费用。

依据题目要求，增加采购A类原材料，减少采购C类原材料，由于A类原材料的产能转化率最高，C类原材料的产能转化率最低，可认为尽可能减小单位产量储存费用：



#### （2）材料转运损失经济

依据问题二转运规则可计算材料转运过程中的经济损失，A、B、C三种原材料分配给各转运商的比例如图10所示。



图10 原材料分配示意图

图10中，转运商的损耗率由至递减，其中转运商承担部分材料A与部分材料B的转运任务，转运商承担部分材料B与部分材料C的转运任务。设原材料转运损耗量为，对于材料A，其转运的损耗量为：



材料B的损耗率为：



同理计算可得材料C损耗量。设转运材料引起的总转运经济损失为，需尽可能的小：



#### （3）改进后的多目标动态模拟规划模型

对式单目标函数进行改进，增加单位产量储存费用和材料转运损失经济两个目标函数，所得制定订购计划多目标动态模拟规划模型如下：



### 模型求解

#### （1）新订购方案与转运方案

基于减少生产成本与转运损耗所得订购方案见附件A，转运方案见附件B。

#### （2）实施效果分析

**a.订购方案实施效果**

企业在增多采购A类材料，减少采购B类材料的情况下制定损耗经济最低的订购方案，得每周剩余库存产能和订购总量产能关系如图11所示。

图11 库存产能与总订购量产能关系图

图11中，企业选择了更多的产能转化率高的原材料，总体市场份额占比由C类供货商向A类供货商转移，此类现象在订货量小的供应商更加明显。同时企业每周产能稳定不变，总订购量产能变化并不明显，但实际订购原材料体积有所减小。

**b.转运方案实施效果**

基于订购方案，制定出转运方案的每周转运商数量变化与周订购总量关系如图12所示。

图12 转运商数量变化与总订购量关系图

统计所求该转运方案的平均每周损耗产能为106.56立方米，占每周产能0.38%，其转运损耗率较低，效果较好。

## 问题四

### 模型的建立

#### （1）解除订购量经济限制

问题一与问题二中，由于题目要求的经济、成本限制，对于订货量的制定进行了约束，根据问题三题意，企业产能的提高改变了经济限制约束，从预测所得的每周供货量确定企业可提高的产能，以提高后的最低每周产能确定订购量。

当该企业生产技术提高后，需要尽可能多的原材料以满足增长的产量需求。显然，式与式中每周的原材料的总订购花费便不能对订购方案的制定有所约束。为尽可能提高企业产能，应扩大原材料的订购量：



#### （2）动态产能潜力

设提高后的第周的供货量产出为，库存量产能为，结合预测供货量，当库存量和供货量最低需满足每周产能需求时有：



经过分析实际求出的各个中会出现小于原每周产能2.82立方米的情况，这是由于各供应商在这周同时出现了无效供货周的情况所导致，因此在确定企业提高产能时剔除此类数据后得有效集合，余下数据的均值便为提高产能后的企业每周产能：



#### （3）实际供货量的减少

现实情况中，转运商在转运原材料时必然出现一定损耗，该损耗会影响第周的实际供货量产出，记受损耗后的周的实际供货量产出为，由式和式可知：



#### （3）实际条件下提高产能的订货量制定模型

增加总实际供货量产出为约束条件，总订货量产出为目标函数得多目标规划动态模拟规划模型如下：



### 模型的求解

#### （1）每周提高的产能

由式和式求得在供货商供货的实际条件下，企业通过技术改造提高后的每周产能立方米，较原每周产能提高了2180立方米，增长了7.7%，说明25家供应商的剩余供应潜能于此，但是在希望转运损耗成本最小和仓储产能成本最大的约束下材料购买的种类分配会偏向于A材料，C材料市场份额最少。不同材料种类定购量如表10所示。

表10 不同材料种类总定购量表（单位：立方米）

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 200441 | 150238 | 131939 |

#### （2）未来24周的订购和转运方案

基于尽量提高企业产能所制定的订购方案见附件A，转运方案见附件B。

# 模型的评价与推广

## 模型评价

### 模型优点

（1）本文首先通过考察筛选机制剔除近五年供应量极小或供应期望不足的供应商，再根据供应商的供应量占比、供应期望、供应稳定性等因素并使用熵权法建立对供应商重要性的综合评价模型，既有效简化了步骤，又能得到对供应商合理、综合的重要性评价。

（2）对于选择的25家供应商，本文通过直方图、折线图、预测图，全方位、多角度地展现出各供应商不同的供货特征，并以此将其分为稳定型、离散型、周期型三类，使用不同方法分别对其未来24周供货情况进行预测，预测数据可靠性高，切合实际。

（3）利用各供应商订供比作为订购优先级的参考量，订供比越小，说明企业越能满足订购量，优先考虑，富有实际意义。

（4）根据预测的未来24周供应商供应量与实际损耗，利用累积周均剩余产能得出符合实际的周最大产能，方法合理，数据准确。

### 模型缺点

由于本文未来24周订货考虑的是挑选出的25家供应商，可能会忽略掉近五年供应期望小，却在未来具有供应潜力的供应商。

## 模型推广

本文建立的模型可广泛应用于对合作商的评价与选择、对于未来数据的预测、对货物的订购与转运计划等问题，实用性强，可推广性高。

# 参考文献

[1]朱晓霞, 肖群. 国际物流服务供应链合作风险的熵权法评价模型研究[J]. 太原师范学院学报(自然科学版), 2021,20(02):36-40.

[2]桂德怀, 张显璇. 基于三次指数平滑法的上海港集装箱吞吐量预测分析[J]. 产业与科技论坛, 2020,19(24):59-60.

[3]陶运海. 制造业供应链中供应商评价研究及应用[D]. 重庆大学, 2013.

# 附录

## 附录1 支撑材料

附表 1支撑材料列表

|  |  |
| --- | --- |
| 材料名称 | 材料说明 |
| 1-1.xlsx | 从58家中筛选出50家最重要供应商 |
| 1-1.py | 熵权法求三类评价指标权值代码 |
| shangquan.xlsx | 熵权法所用数据表格 |
| 2-1.xlsx | 有效周产能排序挑选得25家表格 |
| 2-1.py | 稳定型、离散型供货量模拟代码 |
| 2-2.py | 问题二订购方案代码 |
| 2-3.cpp | 问题二转运方案代码 |
| 3-1.py | 问题三订购方案代码 |
| 3-2.cpp | 问题三转运方案代码 |
| 4-1.cpp | 实际供货量转运损失代码 |
| 4-2.py | 问题四订购方案代码 |
| 4-3.cpp | 问题四转运方案代码 |
| 订购方案dataset.xlsx | 问题二订购方案所用数据表格 |
| 订购方案dataset2.xlsx | 问题三订购方案所用数据表格 |
| 订购方案dataset3.xlsx | 问题四订购方案所用数据表格 |
| 23.xlsx | 二三问的库存产能，订购总量，转运数，转运损耗量 |
| 1-1.xlsx | 筛选50家最重要供应商 |
| 1-1.py | 熵权法求三类评价指标权值 |
| 2-1.xlsx | 有效周产能排序挑选得25家 |
| 2-1.py | 稳定型、离散型供货模拟 |
| 2-2.py | 问题二订购方案 |
| 2-3.py | 问题二转运方案 |
| 3-1.py | 问题三订购方案 |

## 附录2 答案数据

附表2 问题一50家最重要供应商结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 供应商ID | 材料种类 | 市场份额 | 供货稳定性 | 有效周供应期望 | 重要程度 |
| S229 | A | 0.247 | 1 | 0.994 | 0.8108 |
| S140 | B | 0.21 | 0.783 | 1.149 | 0.7808 |
| S361 | C | 0.229 | 1 | 0.992 | 0.7585 |
| S108 | B | 0.168 | 1 | 0.984 | 0.5891 |
| S139 | B | 0.106 | 0.9 | 1.253 | 0.5775 |
| S151 | C | 0.135 | 1 | 0.983 | 0.5011 |
| S282 | A | 0.118 | 0.996 | 1.009 | 0.4695 |
| S340 | B | 0.119 | 1 | 0.999 | 0.4676 |
| S275 | A | 0.11 | 1 | 1.003 | 0.4461 |
| S329 | A | 0.109 | 1 | 1.002 | 0.4413 |
| S131 | B | 0.096 | 0.996 | 0.993 | 0.3993 |
| S330 | B | 0.095 | 1 | 0.986 | 0.3934 |
| S356 | C | 0.091 | 1 | 1.003 | 0.3922 |
| S268 | C | 0.09 | 1 | 1.004 | 0.3918 |
| S308 | B | 0.095 | 1 | 0.975 | 0.3872 |
| S374 | C | 0.034 | 0.696 | 1.296 | 0.3856 |
| S306 | C | 0.088 | 1 | 1.005 | 0.3855 |
| S348 | A | 0.064 | 0.767 | 1.092 | 0.3481 |
| S307 | A | 0.054 | 0.363 | 1.207 | 0.3434 |
| S007 | A | 0.005 | 1 | 1.29 | 0.3399 |
| S194 | C | 0.071 | 1 | 1.002 | 0.3372 |
| S352 | A | 0.062 | 1 | 1.002 | 0.3137 |
| S338 | B | 0.021 | 0.679 | 1.232 | 0.3072 |
| S143 | A | 0.058 | 1 | 1.004 | 0.3033 |
| S266 | A | 0.005 | 1 | 1.205 | 0.2859 |
| S247 | C | 0.039 | 1 | 1.014 | 0.2605 |
| S284 | C | 0.032 | 1 | 1.038 | 0.2566 |
| S037 | C | 0.035 | 0.5 | 1.116 | 0.2515 |
| S114 | A | 0.008 | 0.825 | 1.168 | 0.2489 |
| S086 | C | 0.013 | 0.692 | 1.168 | 0.2457 |
| S123 | A | 0.004 | 0.996 | 1.138 | 0.2432 |
| S365 | C | 0.029 | 1 | 1.022 | 0.2369 |
| S218 | C | 0.011 | 1 | 1.098 | 0.2356 |
| S080 | C | 0.013 | 1 | 1.086 | 0.2351 |
| S031 | B | 0.029 | 1 | 1.02 | 0.2349 |
| S395 | A | 0.053 | 0.308 | 1.042 | 0.2288 |
| S314 | C | 0.001 | 0.842 | 1.155 | 0.2253 |
| S364 | B | 0.02 | 1 | 1.039 | 0.2232 |
| S244 | C | 0.011 | 1 | 1.071 | 0.2203 |
| S291 | A | 0.006 | 0.825 | 1.129 | 0.2194 |
| S040 | B | 0.022 | 1 | 1.019 | 0.2167 |
| S055 | B | 0.017 | 1 | 1.04 | 0.2151 |
| S367 | B | 0.018 | 1 | 1.031 | 0.2137 |
| S126 | C | 0.033 | 0.358 | 1.093 | 0.2135 |
| S129 | C | 0.001 | 0.454 | 1.212 | 0.2130 |
| S294 | C | 0.013 | 0.996 | 1.042 | 0.2063 |
| S346 | B | 0.016 | 1 | 1.016 | 0.1988 |
| S003 | C | 0.009 | 0.742 | 1.091 | 0.1949 |
| S074 | C | 0.009 | 0.604 | 1.07 | 0.1643 |
| S201 | A | 0.057 | 0.117 | 0.939 | 0.1518 |

## 附录3 程序代码

### 问题一

#### 1-1熵权法求权值

#熵权法求权值

import pandas as pd

import numpy as np

import math

from numpy import array

df = pd.read\_excel('shangquan.xlsx')

df.dropna()

def cal\_weight(x):

# 标准化

x = x.apply(lambda x: ((x - np.min(x)) / (np.max(x) - np.min(x))))

rows = x.index.size # 行

cols = x.columns.size # 列

ad = 1.0 / math.log(rows)

# 信息熵

x = array(x)

cd = [[None] \* cols for i in range(rows)]

cd = array(cd)

for i in range(0, rows):

for j in range(0, cols):

if x[i][j] == 0:

cdij = 0.0

else:

p = x[i][j] / x.sum(axis=0)[j]

cdij = math.log(p) \* p \* (-ad)

cd[i][j] = cdij

cd = pd.DataFrame(cd)

E = cd

# 计算冗余度

d = 1 - E.sum(axis=0)

# 计算各指标的权重

w = [[None] \* 1 for i in range(cols)]

for j in range(0, cols):

wj = d[j] / sum(d)

w[j] = wj

w = pd.DataFrame(w)

return w

if \_\_name\_\_ == '\_\_main\_\_':

w = cal\_weight(df)

w.index = df.columns

w.columns = ['weight']

print(w)

### 问题二

#### 2-1稳定型、离散型供货量模拟

import random

import numpy as np

S361=np.random.normal(1367,402.767,24)

S151=np.random.normal(620.99,143.824,24)

S275=np.random.normal(660.64,115.505,24)

S131=np.random.normal(572.97,145.886,24)

S308=np.random.normal(538.98,233.18,24)

S330=np.random.normal(530.90,314.392,24)

S352=np.random.normal(370.96,144.513,24)

S194=np.random.normal(422.35,58.326,24)

list201=[];

list395=[];

list126=[];

for i in range(0,24):

x=random.randint(0,100000);

if x>11667:

y1=random.uniform(0,10)

if y1 <=9:

list201.append(0);

else:

list201.append(1);

else:

y=-1;

while y<0:

y = np.random.normal(2928.1789, 5855.3737)

list201.append(y);

for i in range(0,24):

x=random.randint(0,100000);

if x>30833:

list395.append(0);

else:

y = -1;

while y < 0:

y = np.random.normal(1024.9054,1564.7694)

list395.append(y);

for i in range(0,24):

x=random.randint(0,100000);

if x>35833:

list126.append(0);

else:

y = -1;

while y < 0:

y = np.random.normal(609.0128,1926.86936)

list126.append(y);

print(S361);print(S151);print(S275);print(S131);

print(S308);print(S330);print(S352);print(S194);

print(list201);print(list395);print(list126);

#### 2-2问题二订购方案

#问题二订购方案

import numpy as np

import xlrd

def get\_excel\_data(path):

table = xlrd.open\_workbook(path).sheets()[0]

row = table.nrows

col = table.ncols

data = np.zeros((row, col))

for i in range(col):

cols = np.matrix(table.col\_values(i))

data[:, i] = cols

return data

data\_file2 = u'C:\\Users\\35401\\Desktop\\2021数模国赛\\订购方案dataset.xlsx'

dataset=get\_excel\_data(data\_file2)

[row,col]=dataset.shape

sum=0

lasum=0

yuliang=0

for j in range(col):

lasum=sum

sum = sum + dataset[3,j]\*dataset[2,j]

if(sum>=28200\*2):

print("第一周订购方案到第"+str(j+1)+"列结束")

#到此列满足需求 订购量不大于供应量上限

lastdinggou=28200\*2-lasum

sum=lasum+dataset[3,j]

yuliang=sum-28200

for i in range(j):

if(dataset[1,i]==1):

print(dataset[3,i]\*dataset[2,i]\*0.6)

elif (dataset[1, i] == 2):

print(dataset[3, i] \* dataset[2, i] \* 0.66)

elif (dataset[1, i] == 3):

print(dataset[3, i] \* dataset[2, i] \* 0.72)

x = dataset[1, j]

if (x == 1):

print(lastdinggou \* 0.6)

elif (x == 2):

print(lastdinggou \* 0.66)

elif (x == 3):

print(lastdinggou \* 0.72)

break

print("余量为"+str(yuliang))

sum=0

lasum=0

for i in range(row-4):

sum = yuliang

for j in range(col):

# print("余量为"+str(yuliang))

lasum=sum

sum=sum+dataset[i+4,j]\*dataset[2,j]

# print("总为"+str(sum))

if(sum>=28200\*2):

print("第"+str(i+2)+"周订购方案到第" + str(j + 1) + "列结束")

lastdinggou = 28200 \* 2 - lasum

for k in range(j):

if(dataset[1,k]==1):

print(dataset[i+4, k] \* dataset[2, k] \* 0.6)

elif (dataset[1, k] == 2):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.66)

elif (dataset[1, k] == 3):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.72)

x=dataset[1, j]

if(x==1):

print(lastdinggou\*0.6)

elif(x==2):

print(lastdinggou \* 0.66)

elif(x==3):

print(lastdinggou \* 0.72)

sum = lasum + dataset[i+4, j]

yuliang = sum-28200

print("余量充足为"+str(yuliang))

break

if(j==col-1):

yuliang=sum-28200

print("第" + str(i + 2) + "周订购方案到第" + str(j + 1) + "列结束")

for k in range(j+1):

if (dataset[1, k] == 1):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.6)

if (dataset[1, k] == 2):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.66)

if (dataset[1, k] == 3):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.72)

print("余量不充足，为"+str(yuliang))

#### 2-3问题二转运方案

#include<bits/stdc++.h>//第二问

using namespace std;

int ay[24][10][8];

int by[24][7][8];

int cy[24][8][8];

int main(){

int a[24][10]={

430,0,2565,624,306,5,705,22,651,0,

45,0,1579,661,154,14,783,5,638,0,

620,0,1355,666,102,17,842,8,673,0,

64,0,1410,754,76,601,821,0,649,0,

746,0,1418,835,73,350,834,0,660,0,

101,0,1221,713,175,7109,809,0,681,0,

454,0,1455,634,294,65,766,2,749,1575,

27,0,1661,731,452,39,771,0,650,0,

595,0,1963,680,545,10,703,399,705,1122,

33,0,1783,540,603,8,705,264,673,0,

456,10610,1519,768,570,5,652,322,678,1641,

36,5092,1672,694,706,6,726,580,671,0,

404,7496,2548,674,884,4,702,116,631,734,

64,6060,1562,718,887,0,780,228,610,474,

471,2959,1337,689,826,2,839,291,631,1916,

51,1353,1393,551,840,2,818,220,664,0,

442,0,1401,640,896,0,831,309,684,0,

89,0,1263,642,988,0,806,73,671,0,

419,0,1438,597,1142,2,763,101,685,4160,

31,0,1644,853,1298,0,768,154,747,0,

395,0,1946,558,1433,0,700,76,772,0,

99,0,1766,603,1385,0,702,106,790,0,

454,11409,1502,576,1121,0,649,108,743,0,

56,6788,1656,524,690,0,723,56,762,0

};

int b[24][7]={

835,553,10273,13284,394,840,787,

0,541,0,0,322,124,811,

886,807,7271,0,0,0,870,

787,545,0,0,723,190,783,

925,423,6,12539,792,698,793,

0,0,0,0,0,0,0,

861,763,35,12,705,803,832,

781,547,32,9,1241,1103,789,

900,358,40,19,765,1072,843,

807,515,13,3,747,413,816,

871,557,14,13,501,774,837,

764,621,9,6,723,297,859,

863,27,0,0,0,0,841,

0,0,0,0,0,0,0,

0,0,0,0,0,0,0,

0,0,0,0,0,0,0,

897,441,22,9,1375,1034,927,

793,472,11,5,16,909,971,

774,506,22,6,1440,811,1001,

745,533,8,3,840,1170,997,

756,583,15,8,1050,1042,1018,

726,132,18,6,206,767,970,

855,513,9,5,546,398,866,

792,764,15,9,644,413,901,

};

int c[24][8]={

3,437,817,408,528,512,1065,1049,

1,303,1007,397,553,390,916,989,

0,82,588,456,601,451,868,915,

0,10,392,459,621,393,761,1171,

0,843,566,510,609,449,740,569,

0,0,0,531,634,390,691,1784,

0,1169,927,431,484,430,683,1602,

0,992,940,421,472,384,939,1382,

0,498,825,447,507,634,945,1375,

0,1518,887,437,495,621,960,1063,

0,1650,758,376,436,625,1051,1806,

0,171,1021,392,426,510,1003,1335,

0,0,582,392,454,545,1282,1097,

617,0,940,438,512,512,1128,1942,

3118,7,684,436,497,593,1210,1297,

1116,0,785,403,477,512,1158,1148,

20,0,959,398,472,555,1045,1270,

614,0,1055,425,467,487,1222,1042,

1983,0,630,406,480,560,1191,931,

1730,0,1099,400,469,540,1358,2423,

773,0,981,329,402,545,1498,2067,

705,0,838,408,474,489,1394,1516,

3,0,813,428,507,586,1356,875,

1,0,808,459,530,530,1490,1035

};

int yun[8]={6000,6000,6000,6000,6000,6000,6000,6000};

for(int i=0;i<24;i++){

int cnt=0;

for(int j=0;j<10;j++){

if(a[i][j]>=6000){

ay[i][j][cnt]=6000;

cnt++;

a[i][j]=a[i][j]-6000;

}

}

for(int j=0;j<10;j++){

if(j==0){

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}

if(j>0){

yun[cnt]=yun[cnt]+a[i][j];

if(yun[cnt]>6000){

ay[i][j][cnt]=6000-(yun[cnt]-a[i][j]);

a[i][j]=a[i][j]-ay[i][j][cnt];

cnt++;

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}else{

ay[i][j][cnt]=a[i][j];

}

}

}

for(int j=0;j<7;j++){

yun[cnt]=yun[cnt]+b[i][j];

if(yun[cnt]>12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

while(b[i][j]>6000){

by[i][j][cnt]=6000;

cnt++;

b[i][j]=b[i][j]-6000;

}

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else if(yun[cnt]>6000&&yun[cnt]<12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else{

by[i][j][cnt]=b[i][j];

}

}

for(int j=0;j<8;j++){

yun[cnt]=yun[cnt]+c[i][j];

if(yun[cnt]>6000){

cy[i][j][cnt]=6000-(yun[cnt]-c[i][j]);

c[i][j]=c[i][j]-cy[i][j][cnt];

cnt++;

cy[i][j][cnt]=c[i][j];

yun[cnt]=c[i][j];

}else{

cy[i][j][cnt]=c[i][j];

}

}

}

for(int i=0;i<24;i++){

//cout<<"第"<<i+1<<"周A料供应商运输分配"<<endl;

for(int j=0;j<10;j++){

cout<<ay[i][j][5]<<","<<ay[i][j][2]<<","<<ay[i][j][0]<<","<<ay[i][j][4]<<","<<ay[i][j][7]<<","<<ay[i][j][1]<<","<<ay[i][j][6]<<","<<ay[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周B料供应商运输分配"<<endl;

for(int j=0;j<7;j++){

cout<<by[i][j][5]<<","<<by[i][j][2]<<","<<by[i][j][0]<<","<<by[i][j][4]<<","<<by[i][j][7]<<","<<by[i][j][1]<<","<<by[i][j][6]<<","<<by[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周C料供应商运输分配"<<endl;

for(int j=0;j<8;j++){

cout<<cy[i][j][5]<<","<<cy[i][j][2]<<","<<cy[i][j][0]<<","<<cy[i][j][4]<<","<<cy[i][j][7]<<","<<cy[i][j][1]<<","<<cy[i][j][6]<<","<<cy[i][j][3]<<",";

cout<<endl;

}

cout<<endl;

}

}

### 问题三

#### 3-1问题三订购方案

#第三问订购方案

import numpy as np

import xlrd

def get\_excel\_data(path):

table = xlrd.open\_workbook(path).sheets()[0]

row = table.nrows

col = table.ncols

data = np.zeros((row, col))

for i in range(col):

cols = np.matrix(table.col\_values(i))

data[:, i] = cols

return data

data\_file2 = u'C:\\Users\\35401\\Desktop\\2021数模国赛\\订购方案dataset3.xlsx' # Excel文件存储位置

dataset=get\_excel\_data(data\_file2)

[row,col]=dataset.shape

sum=0

lasum=0

yuliang=0

for j in range(col):

lasum=sum

sum = sum + dataset[3,j]\*dataset[2,j]

if(sum>=28200\*2):

print("第一周订购方案到第"+str(j+1)+"列结束")

#到此列满足需求 订购量不大于供应量上限

lastdinggou=28200\*2-lasum

sum=lasum+dataset[3,j]

yuliang=sum-28200

for i in range(j):

if(dataset[1,i]==1):

print(dataset[3,i]\*dataset[2,i]\*0.6)

elif (dataset[1, i] == 2):

print(dataset[3, i] \* dataset[2, i] \* 0.66)

elif (dataset[1, i] == 3):

print(dataset[3, i] \* dataset[2, i] \* 0.72)

x = dataset[1, j]

if (x == 1):

print(lastdinggou \* 0.6)

elif (x == 2):

print(lastdinggou \* 0.66)

elif (x == 3):

print(lastdinggou \* 0.72)

break

print("余量为"+str(yuliang))

sum=0

lasum=0

for i in range(row-4):

sum = yuliang

for j in range(col):

# print("余量为"+str(yuliang))

lasum=sum

sum=sum+dataset[i+4,j]\*dataset[2,j]

# print("总为"+str(sum))

if(sum>=28200\*2):

print("第"+str(i+2)+"周订购方案到第" + str(j + 1) + "列结束")

lastdinggou = 28200 \* 2 - lasum

for k in range(j):

if(dataset[1,k]==1):

print(dataset[i+4, k] \* dataset[2, k] \* 0.6)

elif (dataset[1, k] == 2):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.66)

elif (dataset[1, k] == 3):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.72)

x=dataset[1, j]

if(x==1):

print(lastdinggou\*0.6)

elif(x==2):

print(lastdinggou \* 0.66)

elif(x==3):

print(lastdinggou \* 0.72)

sum = lasum + dataset[i+4, j]

yuliang = sum-28200

print("余量充足为"+str(yuliang))

break

if(j==col-1):

yuliang=sum-28200

print("第" + str(i + 2) + "周订购方案到第" + str(j + 1) + "列结束")

for k in range(j+1):

if (dataset[1, k] == 1):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.6)

if (dataset[1, k] == 2):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.66)

if (dataset[1, k] == 3):

print(dataset[i + 4, k] \* dataset[2, k] \* 0.72)

print("余量不充足，为"+str(yuliang))

#### 3-2问题三转运方案

#include<bits/stdc++.h>//第三问

using namespace std;

int ay[24][10][8];

int by[24][7][8];

int cy[24][8][8];

int main(){

int a[24][10]={

430,0,2565,624,306,5,705,22,651,0,

45,0,1579,661,154,14,783,5,638,0 ,

620,0,1355,666,102,17,842,8,673,0,

64,0,1410,754,76,601,821,0,649,0,

746,0,1418,835,73,350,834,0,660,0,

101,0,1279,713,175,11615,809,0,681,2678,

454,0,1455,634,294,65,766,2,749,1575,

27,0,1661,731,452,39,771,0,650,0,

595,0,1963,680,545,10,703,399,705,1122,

33,0,1783,540,603,8,705,264,673,0,

456,10610,1519,768,570,5,652,322,678,1641,

36,5092,1672,694,706,6,726,580,671,0,

404,7496,2548,674,884,4,702,116,631,734,

64,8429,1562,718,887,0,780,228,610,474,

471,9653,1337,689,826,2,839,291,631,1916,

51,2048,1393,551,840,2,818,220,664,0,

442,0,1401,640,896,0,831,309,684,0,

89,0,1263,642,988,0,806,73,671,0,

419,0,1438,597,1142,2,763,101,685,4160,

31,0,1644,853,1298,0,768,154,747,0,

395,0,1946,558,1433,0,700,76,772,0,

99,0,1766,603,1385,0,702,106,790,0,

454,11409,1502,576,1121,0,649,108,743,0,

56,6788,1656,524,690,0,723,56,762,0

};

int b[24][7]={

835,553,10273,17702,394,840,787,

6803,541,0,0,1029,124,811,

886,807,8957,2,844,860,870,

787,545,0,0,723,190,783,

925,423,6,17604,792,698,793,

0,36,0,0,0,0,820,

861,763,35,12,705,803,832,

781,547,32,9,1241,1103,789,

900,358,40,19,765,1072,843,

807,515,13,3,747,413,816,

871,557,14,13,501,774,837,

764,621,9,6,723,297,859,

863,543,20,12,806,145,841,

829,647,23,0,0,530,854,

0,0,0,0,0,0,0,

764,396,9,12,658,651,974,

897,441,22,9,1375,1034,927,

793,472,11,5,16,909,971,

774,506,22,6,1440,811,1001,

745,533,8,3,840,1170,997,

756,583,15,8,1050,1042,1018,

726,132,18,6,206,767,970,

855,513,9,5,546,398,866,

792,764,15,9,644,413,901

};

int c[24][8]={

0,0,0,0,0,0,0,0,

1,303,1007,397,553,390,916,989,

0,82,588,456,601,451,868,915,

0,10,392,459,621,393,761,1171,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,1169,927,431,484,430,683,1602,

0,992,940,421,472,384,939,1382,

0,498,825,447,507,634,945,1375,

0,1518,887,437,495,621,960,1063,

0,1650,758,376,436,625,1051,1806,

0,171,1021,392,426,510,1003,1335,

0,0,0,392,454,545,603,1097,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

1116,0,785,403,477,512,1158,1148,

20,0,959,398,472,555,1045,1270,

614,0,1055,425,467,487,1222,1042,

1983,0,630,406,480,560,1191,931,

1730,0,1099,400,469,540,1358,2423,

773,0,981,329,402,545,1498,2067,

705,0,838,408,474,489,1394,1516,

3,0,813,428,507,586,1356,875,

1,0,808,459,530,530,1490,1035

};

int yun[8]={6000,6000,6000,6000,6000,6000,6000,6000};

for(int i=0;i<24;i++){

int cnt=0;

for(int j=0;j<10;j++){

if(a[i][j]>=6000){

ay[i][j][cnt]=6000;

cnt++;

a[i][j]=a[i][j]-6000;

}

}

for(int j=0;j<10;j++){

if(j==0){

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}

if(j>0){

yun[cnt]=yun[cnt]+a[i][j];

if(yun[cnt]>6000){

ay[i][j][cnt]=6000-(yun[cnt]-a[i][j]);

a[i][j]=a[i][j]-ay[i][j][cnt];

cnt++;

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}else{

ay[i][j][cnt]=a[i][j];

}

}

}

for(int j=0;j<7;j++){

yun[cnt]=yun[cnt]+b[i][j];

if(yun[cnt]>12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

while(b[i][j]>6000){

by[i][j][cnt]=6000;

cnt++;

b[i][j]=b[i][j]-6000;

}

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else if(yun[cnt]>6000&&yun[cnt]<12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else{

by[i][j][cnt]=b[i][j];

}

}

for(int j=0;j<8;j++){

yun[cnt]=yun[cnt]+c[i][j];

if(yun[cnt]>6000){

cy[i][j][cnt]=6000-(yun[cnt]-c[i][j]);

c[i][j]=c[i][j]-cy[i][j][cnt];

cnt++;

cy[i][j][cnt]=c[i][j];

yun[cnt]=c[i][j];

}else{

cy[i][j][cnt]=c[i][j];

}

}

}

for(int i=0;i<24;i++){

//cout<<"第"<<i+1<<"周A料供应商运输分配"<<endl;

for(int j=0;j<10;j++){

cout<<ay[i][j][5]<<","<<ay[i][j][2]<<","<<ay[i][j][0]<<","<<ay[i][j][4]<<","<<ay[i][j][7]<<","<<ay[i][j][1]<<","<<ay[i][j][6]<<","<<ay[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周B料供应商运输分配"<<endl;

for(int j=0;j<7;j++){

cout<<by[i][j][5]<<","<<by[i][j][2]<<","<<by[i][j][0]<<","<<by[i][j][4]<<","<<by[i][j][7]<<","<<by[i][j][1]<<","<<by[i][j][6]<<","<<by[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周C料供应商运输分配"<<endl;

for(int j=0;j<8;j++){

cout<<cy[i][j][5]<<","<<cy[i][j][2]<<","<<cy[i][j][0]<<","<<cy[i][j][4]<<","<<cy[i][j][7]<<","<<cy[i][j][1]<<","<<cy[i][j][6]<<","<<cy[i][j][3]<<",";

cout<<endl;

}

cout<<endl;

}

}

### 问题四

#### 4-1实际供货量转运损失

#include<bits/stdc++.h>//第四问 求实际供货量的转运损耗

using namespace std;

int ay[24][10][8];

int by[24][7][8];

int cy[24][8][8];

int main(){

int a[24][10]={

377,0,2529,626,307,4,706,12,649,0,

39,0,1557,663,155,11,784,3,636,0,

543,0,1336,667,102,13,843,4,671,0,

56,0,1391,756,76,461,822,0,647,0,

653,0,1398,837,74,268,835,0,658,0,

89,0,1261,715,176,8906,810,0,679,1926,

398,0,1435,636,296,50,767,1,747,1133,

24,0,1638,733,454,30,772,0,648,0,

521,0,1936,682,548,8,704,221,703,807,

29,0,1758,542,606,6,706,146,671,0,

399,8162,1498,770,573,4,653,178,676,1180,

32,3917,1649,696,709,5,727,321,669,0,

354,5766,2512,676,888,3,703,64,629,528,

56,6484,1541,720,891,0,781,126,608,341,

412,9320,1319,691,830,1,840,161,629,1378,

45,1575,1374,552,844,1,819,122,662,0,

387,0,1382,642,900,0,832,171,682,0,

78,0,1245,644,993,0,807,40,669,0,

367,0,1418,599,1148,1,764,56,683,2992,

27,0,1621,855,1304,0,769,85,745,0,

346,0,1919,559,1440,0,701,42,770,0,

87,0,1741,605,1392,0,703,59,787,0,

398,8776,1481,578,1126,0,650,60,741,0,

49,5222,1633,525,693,0,724,31,760,0,

};

int b[24][7]={

741,545,8777,14852,290,664,785,

6039,533,0,0,757,98,808,

787,796,7653,1,621,679,867,

698,537,0,0,532,150,781,

821,418,5,15503,582,552,790,

828,807,29,12,174,736,818,

764,753,30,7,519,634,829,

693,539,28,6,913,871,787,

799,353,34,12,562,847,840,

717,508,11,2,549,326,813,

773,550,12,8,368,612,834,

678,613,8,4,532,234,856,

766,535,17,7,593,115,838,

736,638,20,5,638,822,851,

789,630,22,10,452,531,828,

678,391,8,7,484,514,971,

796,435,19,6,1011,817,924,

704,465,9,3,12,718,968,

687,499,19,4,1059,641,997,

661,526,7,2,618,924,994,

671,575,13,5,772,823,1015,

644,131,15,4,151,606,967,

759,506,8,3,401,314,863,

703,754,13,6,474,326,898

};

int c[24][8]={

2,336,596,408,530,514,1047,1032,

1,233,735,397,555,391,900,973,

0,63,429,456,603,453,853,901,

0,8,286,459,623,394,747,1152,

0,649,413,510,611,450,727,560,

2,274,669,531,636,391,679,1755,

0,899,677,431,485,431,671,1576,

0,763,686,421,474,385,923,1359,

0,383,602,447,508,636,929,1353,

0,1168,647,437,497,623,943,1046,

0,1269,553,376,437,627,1033,1777,

0,132,745,392,428,512,986,1313,

0,0,425,392,455,547,1259,1079,

462,0,686,438,514,514,1109,1911,

2335,6,499,436,499,595,1189,1276,

836,0,573,403,479,513,1138,1130,

15,0,700,398,474,557,1027,1249,

460,0,770,425,469,489,1201,1025,

1485,0,460,406,482,562,1170,916,

1296,0,802,400,471,542,1335,2384,

579,0,716,329,403,547,1472,2034,

528,0,611,408,476,491,1370,1491,

2,0,593,428,508,588,1333,861,

1,0,590,459,531,532,1464,1018

};

int yun[8]={6000,6000,6000,6000,6000,6000,6000,6000};

for(int i=0;i<24;i++){

int cnt=0;

for(int j=0;j<10;j++){

if(a[i][j]>=6000){

ay[i][j][cnt]=6000;

cnt++;

a[i][j]=a[i][j]-6000;

}

}

for(int j=0;j<10;j++){

if(j==0){

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}

if(j>0){

yun[cnt]=yun[cnt]+a[i][j];

if(yun[cnt]>6000){

ay[i][j][cnt]=6000-(yun[cnt]-a[i][j]);

a[i][j]=a[i][j]-ay[i][j][cnt];

cnt++;

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}else{

ay[i][j][cnt]=a[i][j];

}

}

}

for(int j=0;j<7;j++){

yun[cnt]=yun[cnt]+b[i][j];

if(yun[cnt]>12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

while(b[i][j]>6000){

by[i][j][cnt]=6000;

cnt++;

b[i][j]=b[i][j]-6000;

}

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else if(yun[cnt]>6000&&yun[cnt]<12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else{

by[i][j][cnt]=b[i][j];

}

}

for(int j=0;j<8;j++){

yun[cnt]=yun[cnt]+c[i][j];

if(yun[cnt]>6000){

cy[i][j][cnt]=6000-(yun[cnt]-c[i][j]);

c[i][j]=c[i][j]-cy[i][j][cnt];

cnt++;

cy[i][j][cnt]=c[i][j];

yun[cnt]=c[i][j];

}else{

cy[i][j][cnt]=c[i][j];

}

}

}

for(int i=0;i<24;i++){

//cout<<"第"<<i+1<<"周A料供应商运输分配"<<endl;

for(int j=0;j<10;j++){

cout<<ay[i][j][5]<<","<<ay[i][j][2]<<","<<ay[i][j][0]<<","<<ay[i][j][4]<<","<<ay[i][j][7]<<","<<ay[i][j][1]<<","<<ay[i][j][6]<<","<<ay[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周B料供应商运输分配"<<endl;

for(int j=0;j<7;j++){

cout<<by[i][j][5]<<","<<by[i][j][2]<<","<<by[i][j][0]<<","<<by[i][j][4]<<","<<by[i][j][7]<<","<<by[i][j][1]<<","<<by[i][j][6]<<","<<by[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周C料供应商运输分配"<<endl;

for(int j=0;j<8;j++){

cout<<cy[i][j][5]<<","<<cy[i][j][2]<<","<<cy[i][j][0]<<","<<cy[i][j][4]<<","<<cy[i][j][7]<<","<<cy[i][j][1]<<","<<cy[i][j][6]<<","<<cy[i][j][3]<<",";

cout<<endl;

}

cout<<endl;

}

}

#### 4-2问题四订购方案

#第四问订购方案

import numpy as np

import xlrd

def get\_excel\_data(path):

table = xlrd.open\_workbook(path).sheets()[0]

row = table.nrows

col = table.ncols

data = np.zeros((row, col))

for i in range(col):

cols = np.matrix(table.col\_values(i))

data[:, i] = cols

return data

data\_file2 = u'C:\\Users\\35401\\Desktop\\2021数模国赛\\订购方案dataset3.xlsx' # Excel文件存储位置

dataset=get\_excel\_data(data\_file2)

[row,col]=dataset.shape

sum=0

lasum=0

yuliang=0

for j in range(col):

lasum=sum

if (dataset[1, j] == 1):

sum = sum + ((dataset[3, j] \* dataset[2, j]+dataset[27,j] )/ 0.6)

elif (dataset[1, j] == 2):

sum = sum + ((dataset[3, j] \* dataset[2, j]+dataset[27,j] ) / 0.66)

elif (dataset[1, j] == 3):

sum = sum + ((dataset[3, j] \* dataset[2, j]+dataset[27,j] )/ 0.72)

if (j == col - 1):

yuliang = sum - 30380

print("第" + str(1) + "周订购方案到第" + str(j + 1) + "列结束")

for k in range(j + 1):

print(dataset[3, k] \* dataset[2, k]+dataset[27,j])

print("余量不充足，为" + str(yuliang))

print("余量为"+str(yuliang))

sum=0

lasum=0

for i in range(23):

sum = yuliang

for j in range(col):

# print("余量为"+str(yuliang))

lasum=sum

if(dataset[1,i]==1):

sum = sum + ((dataset[i + 4, j] \* dataset[2, j]+dataset[i+28,j])/0.6)

elif (dataset[1, i] == 2):

sum = sum + ((dataset[i + 4, j] \* dataset[2, j]+dataset[i+28,j]) / 0.66)

elif (dataset[1, i] == 3):

sum = sum + ((dataset[i + 4, j] \* dataset[2, j]+dataset[i+28,j]) / 0.72)

if(j==col-1):

yuliang=sum-30380

print("第" + str(i + 2) + "周订购方案到第" + str(j + 1) + "列结束")

for k in range(j+1):

print(dataset[i + 4, k] \* dataset[2, k]+dataset[i+28,j] )

print("余量不充足，为"+str(yuliang))

#### 4-3问题四转运方案

#include<bits/stdc++.h>//第四问

using namespace std;

int ay[24][10][8];

int by[24][7][8];

int cy[24][8][8];

int main(){

int a[24][10]={

451,20,2585,645,326,25,725,42,671,20,

54,9,1588,670,163,23,792,14,647,9 ,

629,9,1364,674,111,26,851,16,682,9 ,

70,6,1417,760,82,607,827,6,655,6,

755,9,1426,844,82,358,843,9,669,9,

119,18,1296,731,193,11633,827,18,699,2695,

469,14,1470,649,309,80,781,16,764,1590,

39,12,1673,743,464,51,783,12,662,12,

608,12,1976,693,558,23,716,412,718,1134,

42,9,1791,549,612,16,714,272,682,9,

474,8996,1537,786,588,23,670,340,696,1658,

49,4321,1684,706,718,19,738,592,683,12,

415,6353,2558,685,895,15,713,127,642,745,

83,7152,1582,737,906,19,799,247,629,493,

491,10272,1358,709,846,21,859,311,651,1936,

62,1743,1404,561,850,12,829,231,674,10,

454,11,1413,652,907,11,843,321,696,11,

95,6,1269,649,995,6,812,79,677,6,

428,8,1446,606,1151,10,772,110,693,4168,

53,22,1666,875,1320,22,790,176,769,22,

414,19,1965,576,1452,19,719,95,791,19,

112,13,1778,616,1398,13,715,120,802,13,

463,9662,1511,585,1129,9,658,117,752,9,

65,5754,1665,533,699,9,733,65,772,9

};

int b[24][7]={

855,573,10293,16357,415,861,808

6812,549,9,9,1038,133,820

896,816,8967,10,853,869,879

793,551,6,6,730,196,790

934,432,15,17062,800,708,801

951,836,52,31,254,949,838

875,778,50,22,720,817,846

792,558,44,18,1253,1114,801

913,370,52,26,777,1085,855

816,523,21,11,755,421,824

889,575,32,27,518,793,855

776,633,21,16,735,308,871

874,553,31,19,817,156,852

848,666,43,25,887,1060,873

909,659,46,31,635,692,851

774,407,20,18,668,661,985

908,452,34,18,1386,1046,939

799,478,17,10,23,915,978

782,514,31,13,1448,820,1009

767,555,30,24,862,1192,1019

775,602,34,24,1068,1061,1037

738,146,30,17,218,780,983

864,522,18,12,554,406,875

801,774,25,16,654,422,910

};

int c[24][8]={

23,390,837,428,549,533,1086,1069,

10,265,1016,406,562,399,925,998,

9,78,597,465,610,461,877,925,

6,15,398,465,627,399,766,1177,

9,723,575,519,618,457,749,578,

20,319,934,549,652,407,709,1801,

14,1003,942,446,498,444,697,1616,

12,851,952,433,484,395,951,1393,

12,434,837,460,519,646,958,1388,

9,1293,895,446,504,629,968,1072,

18,1414,776,394,454,643,1069,1824,

12,157,1033,404,439,522,1016,1347,

11,11,593,403,464,556,1292,1108,

636,19,959,457,532,532,1148,1962,

3138,27,704,456,517,613,1230,1317,

1127,10,796,413,488,522,1169,1159,

32,11,971,410,484,567,1057,1281,

620,6,1061,431,474,494,1229,1048,

1991,8,639,414,489,569,1199,939,

1752,22,1121,422,491,562,1381,2445,

792,19,1000,348,420,564,1517,2086,

718,13,850,421,487,502,1407,1528,

11,9,821,437,515,595,1365,884,

11,9,818,468,539,540,1499,1044

};

int yun[8]={6000,6000,6000,6000,6000,6000,6000,6000};

for(int i=0;i<24;i++){

int cnt=0;

for(int j=0;j<10;j++){

if(a[i][j]>=6000){

ay[i][j][cnt]=6000;

cnt++;

a[i][j]=a[i][j]-6000;

}

}

for(int j=0;j<10;j++){

if(j==0){

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}

if(j>0){

yun[cnt]=yun[cnt]+a[i][j];

if(yun[cnt]>6000){

ay[i][j][cnt]=6000-(yun[cnt]-a[i][j]);

a[i][j]=a[i][j]-ay[i][j][cnt];

cnt++;

ay[i][j][cnt]=a[i][j];

yun[cnt]=a[i][j];

}else{

ay[i][j][cnt]=a[i][j];

}

}

}

for(int j=0;j<7;j++){

yun[cnt]=yun[cnt]+b[i][j];

if(yun[cnt]>12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

while(b[i][j]>6000){

by[i][j][cnt]=6000;

cnt++;

b[i][j]=b[i][j]-6000;

}

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else if(yun[cnt]>6000&&yun[cnt]<12000){

by[i][j][cnt]=6000-(yun[cnt]-b[i][j]);

b[i][j]=b[i][j]-by[i][j][cnt];

cnt++;

by[i][j][cnt]=b[i][j];

yun[cnt]=b[i][j];

}else{

by[i][j][cnt]=b[i][j];

}

}

for(int j=0;j<8;j++){

yun[cnt]=yun[cnt]+c[i][j];

if(yun[cnt]>6000){

cy[i][j][cnt]=6000-(yun[cnt]-c[i][j]);

c[i][j]=c[i][j]-cy[i][j][cnt];

cnt++;

cy[i][j][cnt]=c[i][j];

yun[cnt]=c[i][j];

}else{

cy[i][j][cnt]=c[i][j];

}

}

}

for(int i=0;i<24;i++){

//cout<<"第"<<i+1<<"周A料供应商运输分配"<<endl;

for(int j=0;j<10;j++){

cout<<ay[i][j][5]<<","<<ay[i][j][2]<<","<<ay[i][j][0]<<","<<ay[i][j][4]<<","<<ay[i][j][7]<<","<<ay[i][j][1]<<","<<ay[i][j][6]<<","<<ay[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周B料供应商运输分配"<<endl;

for(int j=0;j<7;j++){

cout<<by[i][j][5]<<","<<by[i][j][2]<<","<<by[i][j][0]<<","<<by[i][j][4]<<","<<by[i][j][7]<<","<<by[i][j][1]<<","<<by[i][j][6]<<","<<by[i][j][3]<<",";

cout<<endl;

}

//cout<<"第"<<i+1<<"周C料供应商运输分配"<<endl;

for(int j=0;j<8;j++){

cout<<cy[i][j][5]<<","<<cy[i][j][2]<<","<<cy[i][j][0]<<","<<cy[i][j][4]<<","<<cy[i][j][7]<<","<<cy[i][j][1]<<","<<cy[i][j][6]<<","<<cy[i][j][3]<<",";

cout<<endl;

}

cout<<endl;

}

}