基于网络流的材料采购运输策划模型

摘要

本文旨在提供一种基于网络流算法的企业材料采购运输策划模型。

在第一问中，首先对供应商数据进行处理，分析其订单完成率，根据**3σ法则**以及**数据可视化**分析，排除不良供应商，在精确选择范围的同时排除极端数据对于后续评价模型参数的影响；从供应商数据**提取出 9 种特征，**确定对应的评价指标。使用**熵权法**确定上述每个指标对于供应商评价的**影响权重。**最后基于**灰色关联分析法**，结合得出的指标权重建立**多指标评价模型，**计算出供应商的评分，并选出 50 家评价最优的供应商，具体结果置于附录。

在第二问中，首先使用**聚类算法**对选出的 50 家供应商进行二分类,选出其中大型供应商，给出一种表示供应商生产能力的方法，以及一种综合供应商评分、订单数据浮动等特征的，对实际供应情况的模拟方法。其次，分析第二问情形使用网络流图描述的契合性，建立**网络流模型**，即构造一个**有源汇有上下界最小费用可行流**问题，将其转化为**最小费用最大流**的经典问题，使用**最短路费用流算法**求解结果，并分析其相比使用传统非线性规划方法算法**时间复杂度低**，可以求得**全局最优解**的优势。在确定购买方案与供应商后，通过贪心算法得出运输方案。最后，确定对于**方案的测评方法：**使用上述模拟方法生成多组数据，对方案进行可行度测试，统计符合企业生产要求的测试比例。求得第二问的结果为：**企业第一周最少供货商 47 个，其余 23 周最少供货商 19 个，24 周平均总花费 646277 元，测试通过率92.2%。**

在第三问中，首先给出基于第一问中全部供应商**评分极差，**降低 C 类材料供应商评分的供应商评价参数调整方案，同时限制每周购买 C 类材料的总量，并且对本题目中尽量排除 C 类供应商造成的备选供应商质量下降情形做出分析。在此基础上继承使用第二问的网络流模型进行求解，并使用模拟方法进行测试，得到结果为：**第一周供应商数量 47，后续 23 周供应商数量 30，24 周平均总费用 638473 元，测试通过率69.4%，相比第二问的方案平均节省成本1.208%。**

在第四问中，首先分析生产材料需求增加带来的风险，使用统计平均值计算供应商违约、转运损失的风险参数，使用这些参数构建对最劣情况利润率进行计算的**供应商选择方案评价模型，**并通过该模型评价增加产量后需要的供应商选择方案的可行性与稳定性，利用评价模型给出的供应商评分排名，确定可选的供应商组。其次继承问题二中构建的网络流模型，求解可行的最大增产方案。得到结果为：**每周生产量提升 1.423 倍。**

关键词：网络流模型 熵权法 灰色关联分析法 聚类分析

一、 问题重述

1.1 问题背景

1、某建筑和装饰板材的生产企业所用的原材料分为A、B、C三种，该企业每年生产48周，需要提前制定24周的材料订购和转运的计划，根据产能需求来确定供应商和材料订购数量的数量，并确定第三方转运公司的转运方案。

2、该企业每周的产能为2.82万立方米，每立方米产品消耗A类原材料 $0.6m^3$，或B类原材料 $0.66m^3$，或 C 类原材料 $0.72m^3$。由于原材料的特殊性，供应商不能保证严格按订货量供货，实际供货量可能多于或少于订货量。为了保证正常生产的需要，该企业要保持不少于满足两周生产需求的原材料库存量，为此该企业对供应商实际提供的原材料总是全部收购。

3、实际转运过程中，原材料会有一定程度的损耗,损耗量占供货量的百分比称为损耗率，转运商实际运送到企业仓库的原材料数量称为接收量。每家转运商的运输能力为 $6000m^3$/周。通常情况下，一家供应商每周供应的原材料应尽量由一家转运商运输。

4、原材料的采购成本影响到企业的生产效益，实际中 A 类和 B 类原材料的采购单价分别比 C 类原材料高 20%和 10%。三类原材料运输和储存的单位费用相同。

1.2 问题重述

问题一：对附件一中402家供应商的供货特征进行量化分析，建立反映保障企业生产重要性的数学模型，在此基础上确定 50 家最重要的供应商。

问题二：参考问题 1，确定该企业应至少选择多少家供应商供应原材料才可能满足生产需求。针对这些供应商，为该企业制定未来 24 周每周最经济的原材料订购方案，并据此制定损耗最少的转运方案，对订购方案和转运方案的实施效果进行分析。

问题三：企业为了压缩生产成本，计划尽量多地采购A类和尽量少地采购C类原材料，以减少转运及仓储的成本，同时希望转运商的转运损耗率尽量少。制定新的订购方案及转运方案，并分析方案的实施效果。

问题四：该企业通过技术改造已具备了提高产能的潜力，根据现有原材料的供应商和转运 商的实际情况，确定该企业每周的产能可以提高多少，并给出未来24周的订购和转运方案。

二、 问题分析

2.1 问题一分析

在第一题中，首先需要基于附件中给出的 240 周的订单与交付数据提取供应商的供货特征，才能进行进一步分析数据，在对数据做了初步处理，观察供应商订单与供应商供货量的关系后，我们发现不同供应商完成订单的情况差距极大，因而首先通过计算供应商违约期望的分布来剔除不良供应商及其不具有代表性的数据。步骤如下：

1、剔除供应商过小的供应商。

2、通过将供应商的周平均供应量分段，将供应商初步分成大小两类。

3、对于分类为小的供应商，依据3σ原则排除供货满足订单情况不佳的供应商。

3、对于分类为大的供应商，排除违约订单的材料总额过大的供应商。

计算供应商供货总量与订单总量的比值，计为：

$$\mu_0 = \frac{Delivery}{Order}$$

并分别计算每一个供应商的供货订单比μ_i，并计算该样本分布的总体标准差σ：

$$\delta = \sqrt{\frac{\sum_{i=1}^{n}(\mu_i - \mu_0)^2}{n}} \tag{1}$$

依据3σ原则，剔除供货比小于$\mu_0 - \sigma$的供应商，即该类供应商对订单的供货情况差于 68%的供应商，认为此类供应商违约率高，对于安排的合理计划有极大的风险，故不做考虑。

进而，只保留剩余的供应商数据，首先选出下述可以从数据中提取出可能影响公司评价的参数：

1、企业 240 周的订单总额O_Z

2、供应商 240 周的交付总额D_Z

3、供应商单周的最大交付量D_{max}

4、企业单周的最大订单量O_{max}

5、240 周内订单订货量的平均值$\bar{O}$

6、240 周内供应商交货量的平均值$\bar{D}$

7、交易周中供应商交货量未达到订单量的周数比例B

8、交易周中供应商交货量超过订单量的周数比例m

9、供应商违约总额L

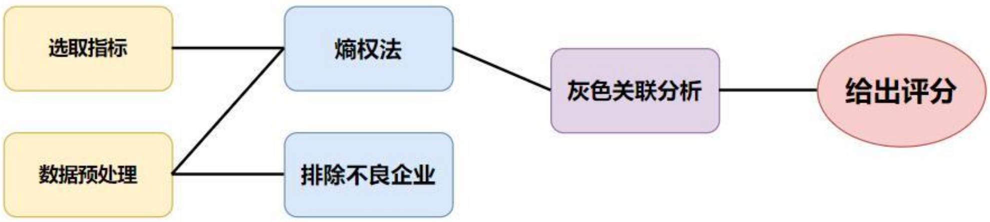
我们对附件给出的数据进行缺值判断剔除坏值后提取上述特征，进而计算它们对供应商评价的影响权重，最后建立评价模型

熵权法是一种根据总数据中指标变异性的大小来确定客观权重的多指标评价算法，由于很难利用现有知识直接分析上述六个指标对于供应商评价的影响，故使用熵权法确定每一指标的权重。

这样即可构造第一问要求的**评价模型**，将供应商的各个指标值归一化处理后，使用灰色关联分析法构建多指标评价模型，代入各指标权值，即可使用每个供应商的 9 种指标计算其评分r_i，评分高者更优，据此选择出前 50 家优秀供应商。

第一问求解思路如下：

图 1 第一问求解流程图



2.2 问题二分析

本题目中公司购买材料的策略有两点主要考虑，首先做出分析：

1、公司要求保障生产，购买材料时大体策略都有很大冗余，要求预留出两周的生产材料，并且题目设定公司对于供应商产出的所有材料会全部收购，可以看出，**为了保障生产，公司希望计划供应的材料尽量充足。**

2、企业想要压缩生产成本，本模型也致力于给出最经济的购买材料方案，因而公司的本质要求是**整个供应流程的花费尽量少。**

同时，本题目公司供应链存在如下特点：

1、存在多个中间**节点**，包括不同种类材料的供应商、以及不同的转运商。

2、每个节点在供应链中可以承载的货物量都有**流量**限制，例如生产企业有产量限制，运输企业有运量限制，同时企业对于供应链的总流量也有可满足本周生产并冗余两周材料的限制。

3、最终每一条供应链的**终点**都为企业仓库。

综合上述分析，该问题非常适合使用**网络流模型**描述，因而以此为模型基础开展后续工作。

2.2.1 网络流模型

网络流模型用于研究一类网络图上的最优化问题，给定一个有向图$G(V,E)$

并且该图满足：

1、有唯一一个源点S，入度为 0，作为出发点

2、有唯一的一个汇点T，作为终点

3、图中的每条边(u,v)都有一非负容量$C(u,v)$, 即为该边的边权。

这样一个图即为网络流图，记为$G(V,E,c)$。

对于该图，若给出一组流量满足：

1、源点S的流量=汇点T的流量=整个网络的流量

2、中间节点的总流量=总流出量

3、每一条边的流量都不超过边权且非负

那么此时网络的流量称为可行流，最大的可行流称为最大流。

依据 2.2 中的初步分析，将生产材料视为流量，节点首先包括所有可选择的供应商与材料种类，与其连接的边的权重即为供应商的生产能力，而企业生产则可以视为汇点T，所有的材料最终都进入生产环节。

构建一个网络流图，在本题求解最经济策略的情形下，给每一条边添加数值为单位流量费用的参数，这样就可以模拟实际问题中各环节的费用消耗。

具体分析企业的要求，发现每一个环节的材料通过量有上限和下限两种限制，例如每周材料流量至少不能低于一周的生产需求，而每个供应商提供的材料量不能超过其生产能力，那么该网络流图中的每一条边需要表示为(u,v,w,min,max)，l表示边的费用，(min,max)代表该条边(u,v)上的流量上限与下限，这是一个**有源汇有上下界最小费用可行流**问题，可以转化为最大流最小费问题求解。

相对于使用传统的规划方法求解本题，网络流模型有两大优势：

1、算法时间复杂度小，速度快

2、不会出现求出局部最优的情形，结果更优

2.2.2 有源汇有上下界最小费用可行流问题

本段将简述上述最小费用可行流问题的求解方案，即如何将其转变为传统的最大费用最小流问题编程求解。

求解方法：

$\boldsymbol{step1}$ 在原图中建立附加源点S_f，附加汇点T_f

$\boldsymbol{step2}$ 连接边(S,T)，流量限制为$[0,inf]$, 使得源点汇点也满足流量平衡条件，即流入等于流出。

$\boldsymbol{step3}$ 对于原图中的边(u,v)，若其流量限制为$[a,b]$，那么新图中(u,v)的流量为$b-a$

$\boldsymbol{step4}$ 对于原图中的节点i，计算其所有流入边流量下界的总和减去所有流出边流量下界的和，记为$d(i)$：

如果$d(i)>0$，则连接边(S_f,i)，流量为$d(i)$

如果$d(i)<0$，则连接边$\left(i,T_f\right)$，流量为$-d(i)$

$\boldsymbol{step5}$ 在新图上求解从附加源点到汇点最小费用最大流问题，实际答案即为求出的费用+原图每条边下限×每条边费用，这个答案是在满足流量限制条件以及流量平衡时费用最小的方案。

引例：

对于一个简单的网络流图，每条边都设定了流量的上下限$[a,b]$，对于某一条边(u,v)，$a(u,v)$代表其流量下限，$f(u,v)$代表其实际流量，$b(u,v)$为流量上限，

显然有：$a(u,v) \le f(u,v) \ \le b(u,v)$，设$g(u,v) = f(u,v) - \ a(u,v)$，则显然：

$$0 \le g(u,v) \le b(u,v) - a(u,v)$$

$\boldsymbol{step3}$的操作即将原图改为边上限为$b(u,v) - \ a(u,v)$的普通网络流图，$g(u,v)$是新图中每一边的流量，而$g(u,v) - \ a(u,v)$为实际求解的边流量，它的最大流是 2，按照该流量，由于每条边下限不同，可以满足上下限限制，但无法满足流量平衡条件, 对于节点v, 需满足流量平衡条件：

$$\sum f(u,v) = \sum f(v,j)$$

将$f(u,v)$代换：

$$\sum g(u,v) + \sum a(u,v) = \sum g(v,j) + \sum a(v,j)$$

$\boldsymbol{step4}$规定的节点流入流出流量下限差$d(v)$即为：

$$d(v) = \sum a(u,v) - \sum a(v,j)$$

进而可以得出新图中两边流量需满足的关系：

$$\begin{cases} \sum g(v,j) = \sum g(u,v) + d(v) \quad d(v) > 0 \\ \\ \sum g(u,v) = \sum g(v,j) + (-d(v)) \quad d(v) < 0 \end{cases}$$

因而，对于$d(v) > 0$，构造流量为$d(v)$的边从虚拟源点指向该节点，为该节点补充不足的流量; 对于$d(v) < 0$，构造流量为$d(v)$的边从该节点指向虚拟汇点，分走该点多余的流量。由上述分析，**当且仅当与附加源点、附加汇点相连的所有边都满流，才可能让所有节点满足流量平衡条件**，因而问题可以转变为求该新图源点、汇点间的最小费用最大流问题。

2.2.3 供应商分类与供货预测

最后，考虑供应商选择的具体问题，观察处理后的供应商的数据，发现不同的供应商获得的订单、以及完成的订单数量相差巨大，不同类的供应商在建立图模型时赋予边权上限，即其生产能力的方案应该有所分异；同时，数据中的订单交付数额分布也有一定特点，可以大体区分出大额订单和其他订单，我们对这些情形都应当具体分析。

首先通过聚类算法在 50 个供应商中再次区分出大型供应商与小型供应商。

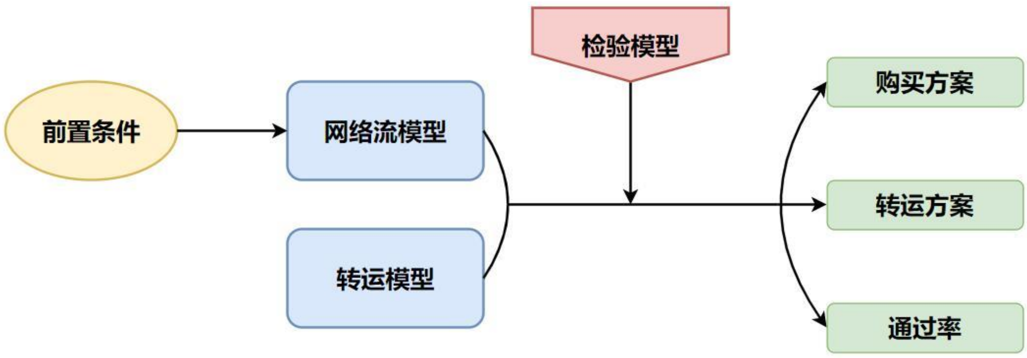
对于大型供应商，其实际生产交付能力的浮动对计划与企业生产有较大影响，因而将这些供应商在实际生产中的波动纳入考虑。

通过这些大型供应商的数额较大的订单计算均值，估算这些供应商大体的生产能力；同时根据供应商评分、订单数据浮动等特征，计算出该供应商的供应能力预期会产生的正、负幅度，利用随机数生成供货序列。

2.2.4 转运商的分析

在确定了购买方案后，进一步考虑 8 家转运商的运输方案，该问题是一个典型的动态规划背包问题，但是考虑到仅有 8 家转运商，并且 8 家转运商的承运能力相对于我们的计划已经没有多少余量，因而采用贪心算法，可以求得较优解。第二问以及后续问题的总体求解思路如下：

图 2 第二、三、四问求解流程图



2.3 问题三分析

2.3.1 供应商评分调整策略

问题三与问题二具有很高的继承性，相比问题二，首先需要调整的参数是对于 C 类供应商的评分进行调整，由于题目规定“尽量多的购买 A，尽量少的购买 C”，因此减少 C 类材料的供应商的评分，使得 C 类供应商的优先级大幅降低，并限制购买 C 材料的数量。

2.3.2 总流量设定调整策略

同时，企业想通过这种方式进一步优化成本，而在尽量不购买 C 材料的情况下对 C 类中优秀供应商的选择倾向会减少，多余的生产材料需求会被分给更差的A,B类型供应商，会导致稳定性的下降。继承第二问的网络流模型，并按照其测评方法衡量稳定性的变化。

2.4 问题四分析

在本问增加企业生产量的过程中，需要考虑如何平衡增加产量与该计划增加供应商供应压力而相比现设定增加的风险。给出几个指标参数来评价模型得出的方案：违约扰动率w，转运扰动率z，最低利率预测值σ。

2.4.1 违约扰动率

方案的违约扰动率由每一个方案选择的所有供应商决定，其值为选出的所有供应商在 240 周的附件数据中交付量较订单数量的差值（即所有供应商违约总额）与这些供应商的总订单量的比值，对于方案选择的t个供应商，记为：

$$w = \frac{\sum_{i=1}^{t}(O_i - D_i)}{\sum_{i=1}^{t} O_i} \tag{2}$$

2.4.2 转运扰动率

方案的转运扰动率由该方案选择的t个转运商的平均损失率S_i决定：

$$z = \frac{\sum_{i=1}^{t} S_i}{t} \tag{3}$$

运商数量t由方案选择的供应商的供应能力决定：
1、计算供应商供应能力总和Sum。
2、$t = [\frac{Sum}{6000}] + 1$，即供应能力处以单个转运商的转运能力并向上取整。
3、转运商的选择顺序由平均损失率由低到高来排列。

2.4.3 最低利率预测值

依据模型假设中的基准利率值，给出最低利率预测值：

$$\sigma = (1 - \omega - z) \times (10.85\% - z) \tag{4}$$

该值反应了最差情况时企业的剩余利率。

2.4.4 求解方法

在本问中以供应商为对象考虑问题，在一个方案中，选择的供应商数量越多，企业每周可以获得的材料上限也就更多，可以提高产量；同时，供应商越多，供应商平均质量下降，违约等情形增加会导致方案稳定性变差。

首先使用第一问中得出的所有供应商评分r_i，将供应商以评分从高到低排列，不断将新的供应商按顺序纳入评分，此时企业生产上限增加，但稳定性随着供应商质量变差下降。使用之前分析的三个参数不断判定企业选择的可行性，直到指标超出阈值。

在确定了可选供应商组后，通过不断增加产能对最优的预期周生产量Fw求解获得第四问的答案。

三、模型假设

1、二、三问进行预测时，加工厂处于刚刚开工的阶段，即库存只能提供刚开工这一周的原材料。

2、经查阅论文[1]，得到单位原料储存费用 0.056、运输费用 0.2；查阅期刊[2]，确认利润率 10.85%。

3、附件中的全部供应商以及转运公司不会因为特殊情况倒闭或者停运。

4、为方便计算，设以 C 种材料的单价为 1 元。

5、由于现实生活中存在的不确定性，只要连续两周内库存都高于两周生产需求的 90%，就不会对企业生产产生严重影响。

四、符号说明

符号	说明
O_Z	企业 240 周订单总额$/m^3$
D_Z	供应商 240 周的交付总额$/m^3$
D_{max}	供应商一周交付量峰值$/m^3$
O_{max}	企业一周订单峰值$/m^3$
$\bar{O}$	订单订货量平均值$/m^3$
$\bar{D}$	供应商交货量的平均值$/m^3$
B	交货量未达订单量周数比
m	交货量超标指数
$V_{A/B/C}$	A,B,C材料的生产率
r_i	供应商评分
d_i	大订单平均数额$/m^3$
Fw	企业每周生产量$/m^3$
ω	供应商违约扰动率
z	转运损失率
σ	预估最劣利润率

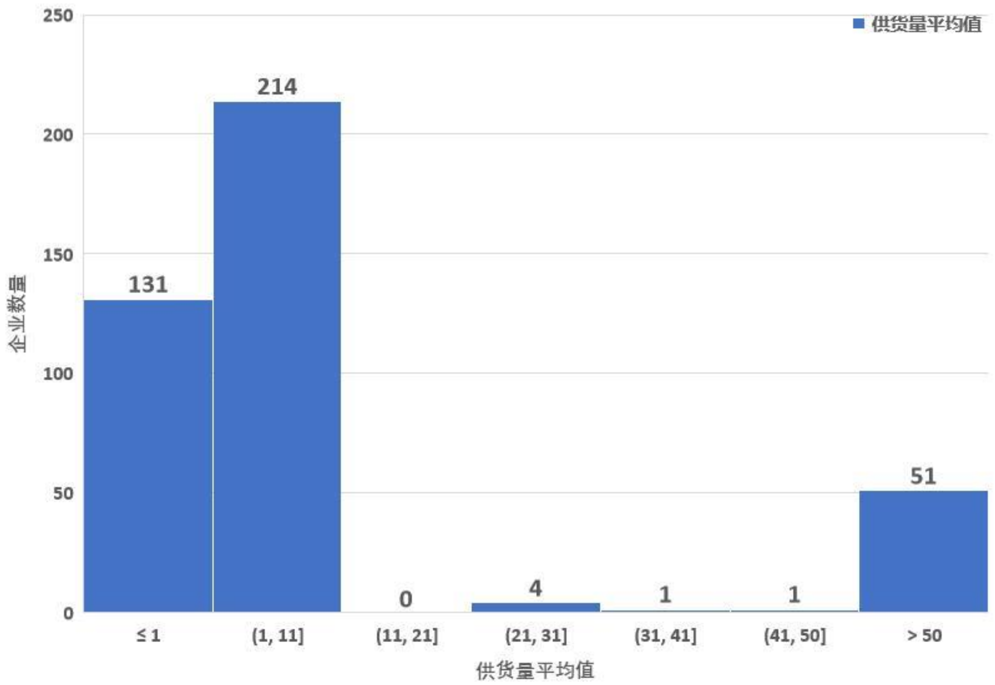
五、模型的建立与求解

5.1 问题一建模求解

5.1.1 不良供应商排除

依据问题分析，首先对附件给出的数据进行缺值判断，剔除坏值，进而依照问题分析中的方式，首先区分供应商的供应能力，计算每个供应商周供应量在分段后的分布并绘制柱状图：

图 2 供应商周平均供应量分布图



这样我们发现本题中的供应商有很显著的大小分异，大部分供应商的周平均供应量小于 11，可以界定其为小型供应商，剩余的供应商周平均供应量在 20-50 之间，并且大部分集中在 50 以上，界定为大型供应商。

在小型供应商中剔除不良供应商，计算供应商供货订单比：

$$\mu_0 = \frac{\sum delivery}{\sum order} = 0.6344 \tag{5}$$

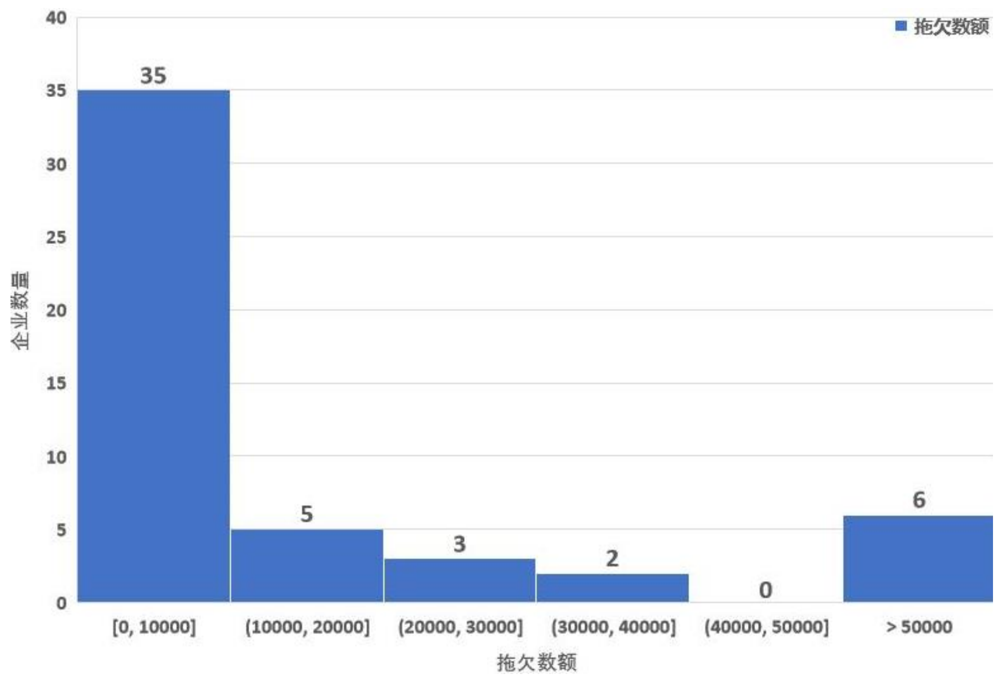
以及样本分布的标准差：

$$\delta = \sqrt{\frac{\sum_{i=1}^{n}(\mu_i - \mu_0)^2}{n}} = 0.2612 \tag{6}$$

以1σ为边界剔除了 94 个小型不良供应商。

对于剩余的大型供应商，观察其数据，制作出了 240 周内其未达到订单要求的差值总量，同样发现很显著的分布规律：

图 3 大型供应商未达到订单要求拖欠情况图



差值总量大于 50000 的六个大型供应商我们将其界定为无信用供应商，其信用率过于低下，而其产量在订单中占比较大，会对企业生产造成威胁，故剔除了 6 个不良供应商。

剩余 302 家供应商编号于附件中给出。

5.1.2 供应商评价指标

接下来计算问题分析中给出的 9 个指标：企业 240 周的订单总额O_Z、供应商 240 周的交付总额D_Z、供应商单周的最大交付量D_{max}、企业单周的最大订单量O_{max}、240 周内订单订货量的平均值$\bar{O}$、240 周内供应商交货量的平均值$\bar{D}$、交易周中供应商交货量未达到订单量的周数比例B、交易周中供应商交货量超过订单量的周数比例m，供应商的违约总额L。

其中需要特别说明的是比例B与比例m，设交易周数α，其定义为生产企业给供应商订单量不为 0 的周数，而交付量未达订单量的周数为β，则：

$$B = \frac{\beta}{\alpha}$$

显然该指标对供应商的影响是负向的。而对于比例m，设交付量大于订单量的周数为α，每一周交付量为D_i, 订单量为O_i，则：

$$m = \frac{\sum_{i=1}^{\alpha} \frac{D_i}{O_i}}{\alpha} \tag{7}$$

具体分析指标m，对于一个供应商，其供货量较订单量大一方面说明其供货

能力强，另一方面如果供货量超出过多会对计划与成本产生不利影响，该指标很难简单界定为正向或负向影响指标，对其进行归一化处理：

$$m=\begin{cases} 1 & 1 \le m \le m_0 \\ \\ 1-\frac{m_0-m}{m_0-m_{max}} & m>m_0 \end{cases} \tag{8}$$

其中$m_0 = 1.04$，是所有供应商该指标的平均值。

5.1.3 熵权法求影响权重

使用上述给出的每个供应商对应的参数组，使用熵权法确定其影响权重，在选出的众多指标中，既有对供应商的评价为正面影响的正向指标，也有为负面作用的负向指标，因而首先要对供应商指标进行正项化归一化处理，对于第i个供应商的指标组中的第j项X_{ij},其归一化的结果为：

$$Y_{ij} = \frac{X_{ij} - min(X_j)}{max(X_j) - min(X_j)} \tag{9}$$

当为X_{ij}负向指标：

$$Y_{ij} = \frac{min(X_j) - X_{ij}}{max(X_j) - min(X_j)} \tag{10}$$

其中$\max(X_j)$、$\min(X_j)$分别为第i家供应商的第j项指标的最大值与最小值。

进而计算第j个指标中该第i个供应商的数据在所有n个供应商的数据占的权重，将其作为计算信息熵时的概率P_{ij}：

$$P_{ij} = \frac{Y_{ij}}{\sum_{k=1}^{n} Y_{kj}} \tag{11}$$

利用上述结果计算第 j 项指标的信息熵：

$$E_j = -\frac{1}{\ln n}\sum_{i=1}^{n}(P_{ij} \times \ln P_{ij}) \tag{12}$$

最后计算每一个指标的权值,其中m为所选指标的个数，在本题中为 9 个。

$$W_j = \frac{1 - E_j}{m - \sum_{j=1}^{m} E_j} \tag{13}$$

计算得到上述 9 个参数的权值如下：

表 1 熵权法求出特征指标权指表

指标	权值	指标	权值
O_Z	0.1533	$\bar{O}$	0.1341
D_Z	0.1684	$\bar{D}$	0.1587
O_{max}	0.1282	B	0.0295
D_{max}	0.1838	M	0.0198
		L	0.0241

5.1.4 灰色关联分析法建立评价模型

获得了上述权值，可以进一步利用灰色关联分析法建立多指标评价模型。首先对所有公司的各个指标值做归一化处理，对于正向指标X_i，所有公司中该指标的最大值为X_{max}，最小值为X_{min}，处理后的指标Y_i表示为：

$$Y_i = 1 - \frac{X_{max} - X_i}{X_{max} - X_{min}} \tag{14}$$

对于负向指标X_i，所有公司中该指标的最大值为X_{max}，最小值为X_{min}，处理后的指标Y_i表示为：

$$Y_i = \frac{X_{max} - X_i}{X_{max} - X_{min}} \tag{15}$$

这样，就可以生成每一个指标k的含n个供应商数据的比较序列$\{Y_i(k)|i = 1,2..n\}$，同时确定参考序列，即所有指标都取最佳的参考序列$\{Y_0(k)|k = 1,2,3..8, Y_0(k) = 1\}$。

接下来利用参考序列和比较序列计算两级最小差MIN、两极最大差MAX二参数：

$$\begin{cases} MIN = \min_s \min_t |Y_0(t) - Y_s(t)| \\ \\ MAX = \max_s \max_t |Y_0(t) - Y_s(t)| \end{cases} \tag{16}$$

以及每一个指标k的灰色相关系数$\xi_i(k)$：

$$\xi_i(k) = \frac{MIN - \rho \cdot MAX}{|Y_0(k) - Y_i(k)| + \rho \cdot MAX} \tag{17}$$

最后，利用 4.1.2 中求得的 9 个指标的权值ω，计算各个供应商的评价系数r_i：

$$r_i = \sum_{k=1}^{9} \omega_k \cdot \xi_i(k) \tag{18}$$

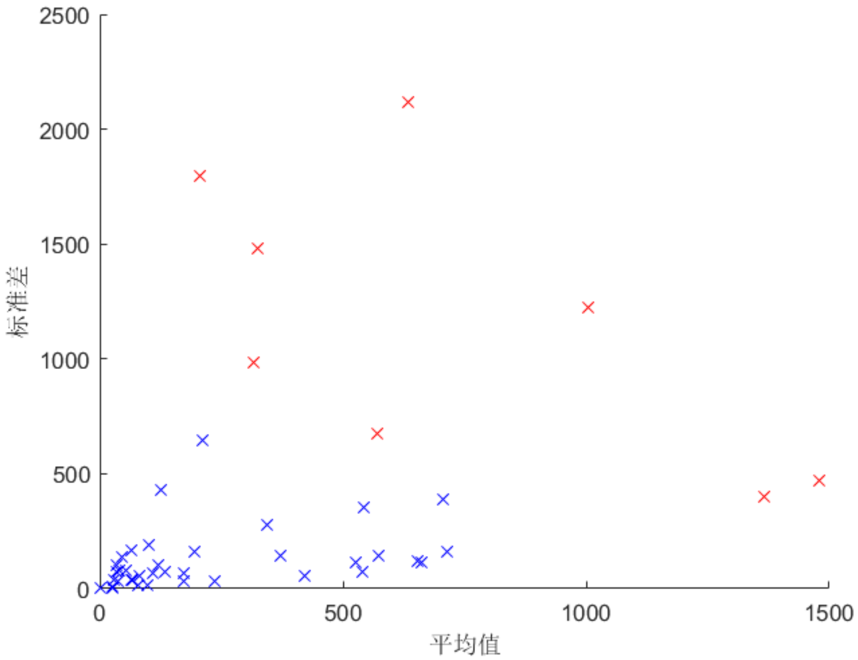
这样，就建立了对供应商进行量化评价的方法,将 302 家供应商的信息代入模型，获得了所有供应商的分数评级，越接近 1，该供应商的评价越高。所选 50 家供应商具体数据于附录中可见。

5.2 问题二求解

5.2.1 使用聚类算法分类优秀供应商

利用第一问中的权值，选择出了前 50 项分数最高的供应商，但这些供应商具有不同的特点，有些供应商长期稳定的完成数量较少的订单，这类供应商稳定性高，而有的供应商在一两次数量巨大的订单中表现出色，这类供应商拥有较强的生产能力。两类供应商在图模型中的边权与在不同情形下的策略有所不同，因而首先使用$Kmeans$聚类算法区分出 50 个供应商中的大型与小型供应商，聚类结果如下：

图 4 50 家供应商根据供货量平均值和标准差的聚类分类结果

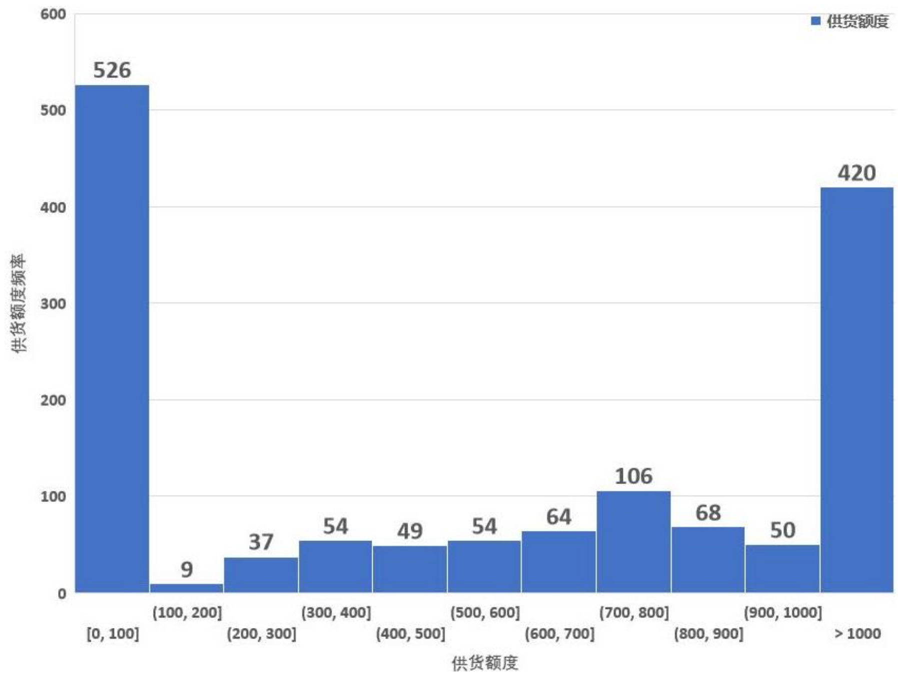


其中红色节点代表的供应商或是平均值较高，或是平均值较低而方差高，产量峰值高，认为其为大型供应商，而蓝色节点则认为是较小的供应商。

5.2.2 大型供应商供货浮动预测

首先对订单的具体情况进行分析，同样通过数据可视化的方式，将所有的订单按照其额度区间画出柱状图，并结合分析一周内生产需求的材料，取1000 以上的订单为大订单。

图 5 所有订单单次供货量柱状图



这些大订单多数由聚类分出的大型供应商生产交付，而这些订单是否被完成对计划是否能满足企业生产需求有较大的影响，因而需要将大型供应商的供货浮动做出模拟和预测：

对于一个大型供应商，取其所有供货量大于 1000 的n个订单d_i，计算供货量平均值d_0，定为基准值：

$$d_0 = \frac{\sum_{i=1}^{n} d_i}{n} \tag{19}$$

进而求该供应商大订单序列$\{d_1, d_2 ... d_n\}$关于基准值d_0的浮动序列：

$$F = \{f_i | i = 1, 2...n, f_i = |d_i - d_0|\}$$

同时统计这些浮动中正浮动、负浮动的占比，记正浮动占比为ε：

$$\varepsilon = \frac{card(\{f_i | d_i - d_0 > 0\})}{n} \tag{20}$$

取每一个供应商交付量浮动序列的最大最小绝对值作为该供应商的浮动上下限，即该供应商的浮动数值范围为：$[min(|F|), max(|F|)]$。考虑到第一问中的供应商评分影响，该评分是一个正向指标，评分越高的供应商，对其供应稳定性期望越高，因而纳入评价系数r_i，最终的供应商交付浮动值范围定为：

$$e = [min(|F|), max(|F|)] \times (1 - r_i) \tag{21}$$

最后，确定模拟一个大型供应商在实际中供应能力浮动的方法：

$\boldsymbol{step1}$：统计该供应商所有大订单交付情况的平均值d_0作为基准值

$\boldsymbol{step2}$：计算上述数值浮动范围与正浮动占比ε

$\boldsymbol{step3}$：设定正负浮动参数k，我们在 0-1 间随机生成一个数，若该数大于正浮动占比ε，则$k = -1$，反之若该数小于ε，$k = -1$

$\boldsymbol{step4}$：确定下一周的供应能力为：$d_0 + k \cdot e$

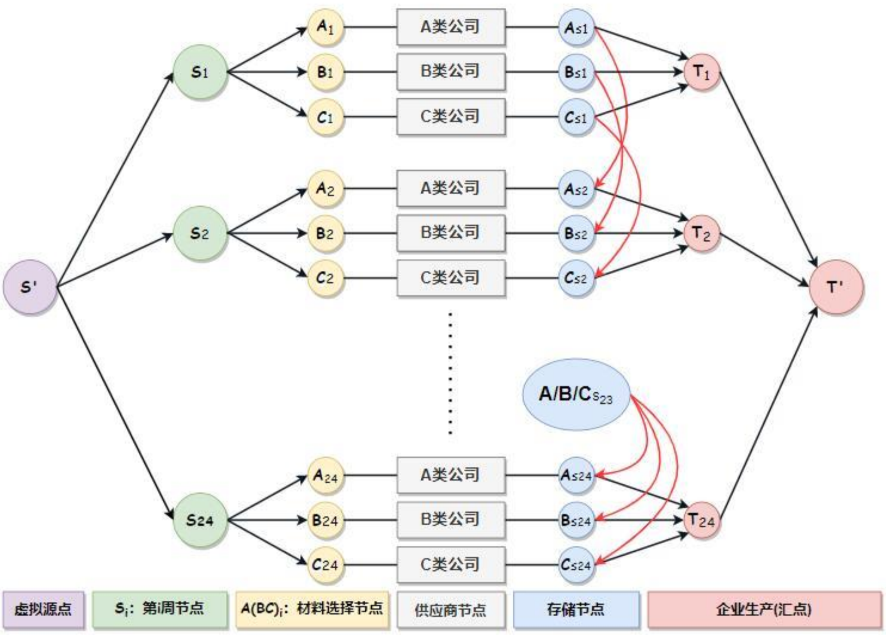
$\boldsymbol{step5}$：重复$\boldsymbol{step3}$、$\boldsymbol{step4}$

这样，就可以使用该基于数据的方法更合理的模拟每一个大型供应商在之后 24 周中不断变化的供应能力V_i(以和产品 1:1 计算)，最后将该等效供应能力结合企业生产的A, B, C类型材料的生产率$V_{A/B/C}$，即题目中给出的材料量与生产的产品量的关系。

5.2.3 建立网络流图

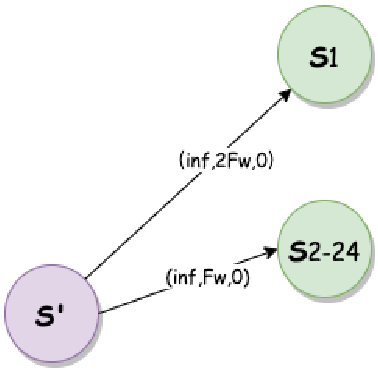
基于上面的一切准备，建立一个网络流模型，总体示意图如下：

图 6 网络流模型结构示意图



本段将分块解释模型设置，**每条边权的表示方式为(流量上限,流量下限,费用)**：

周节点：



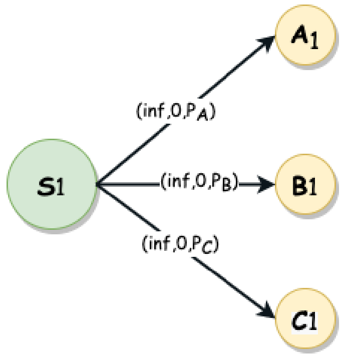
使用拆点法拆除虚拟源点S'以便在网络中体现每周材料的限制条件，其中F_w为每周生产所需的材料量，$F_w = 28200$，以企业的生产量为基准，并且认为该企业在做计划时已经有第一周生产的材料。因而，对于第一周(S', S_1)，其流量下限为两周生产材料量 $2F_w$，在其他周，(S', S)的流量下限为该周的生产量需求F_w。

在此基础上，提前考虑运输带来的损耗在计入影响后可能导致计划不达要求的情形。我们将运输损失纳考虑，设计保障系数$a\%$，规定企业的每周流量为：

$$Fw = (1 + a\%)Fw \tag{22}$$

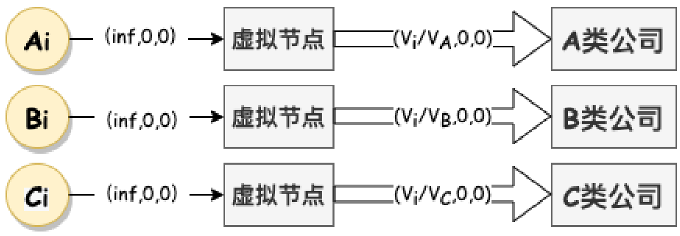
观察所有供应商数据，发现单周转运商的最高损失率为百分之 5，因而以此作为参考，令$a = 5$，则每周流入节点的流量应置为$1.05Fw$。

材料节点：



此节点仅用于分流三种材料选择，在这一节点引入材料的费用P_A, P_B, P_C作为边的费用权重，上下限由后续的步骤限制，与此节点连接的边上下限仅保权重非负。

供应商节点：



在网络流中无法对节点a的流量进行限制，但可以将a拆为a_{in}与a_{out}，即多拆出一个虚拟节点，将所有流入的边连向a_{in}，出边从a_{out}流出，(a_{in}, a_{out})边容量为a点限制的流量。

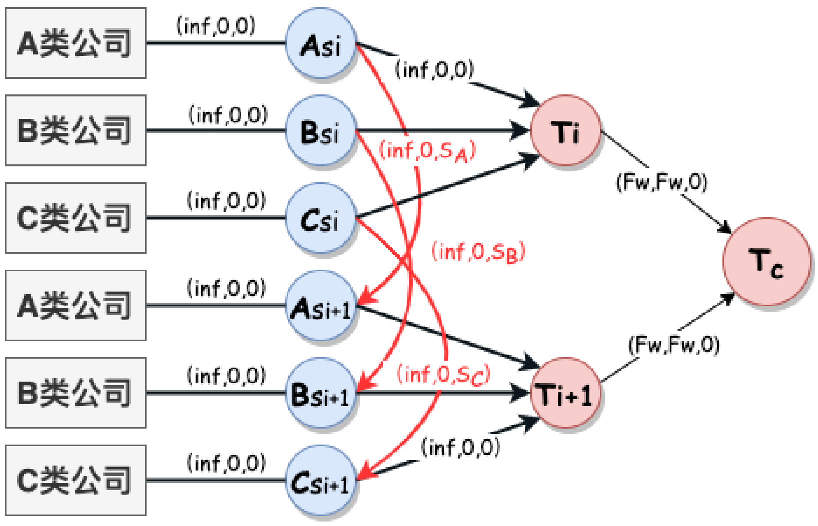
在题背景下，所有供应商按上述方式拆点，每一条边将其连接的供应商的供应能力$\frac{V_i}{V_{A/B/C}}$作为该条边的上限流量，本示意图图简化了虚拟节点A、B、C类所有供应商的节点，费用已经在材料节点被计算，故所有边费用设为 0。

存储、生产节点：

在对仓储进行分析，根据论文[1]，确定了仓储费用设定与货品价格的关系，给定仓储单周单位费用$S_{A/B/C}$：

$$S_{A/B/C} = 0.0256V_{A/B/C} \tag{23}$$

在此基础上，给出网络流模型最后一部分的整体图，该部分边权在流量平衡条件下可以一起分析。



首先可以确定的边权是(T_i,T_C)，其边流量为固定的Fw，因而流量上下限都为Fw，同时，这样设定还可以免去对红色表示的代表A、B、C材料存储边进行流量上下限赋值。

在第一周，流入节点T_1的总流量$-$流出T_1的总流量$= Fw$，流入节点S_1的流量$2Fw$较流出T_1的总流量多出Fw，由流量平衡，这$2Fw$流量即为A、B、C三个贮存节点流向下一周的A、B、C储存节点的总流量；在之后每一周，流入节点S_i的总流量$> Fw =$ 流出T_i的总流量，并且该冗余量是考虑后续最大运输损失率后的结果，这样后续每一周A、B、C储存节点流向下一周贮存节点的总流量总是大于$2Fw$，满足了企业要求的流出两周生产材料的需求。因而对于贮存节点所连的边可以不设上下限，仅设费用$S_{A/B/C}$。在建立了上述网络流模型后，我们使用问题分析 2.2.1 与 2.2.2 中的方法编程求解该模型，使用$SSP(Successive\ Shortest\ Path)$费用流算法解决转化后的最大费用最小流问题。

5.2.4 SSP费用流算法概述与复杂度分析

定义1　网络中未流满的边与其经过的点的集合构成的新图 G'为**残量网络**。

定义2　在残量网络中一条由源点到汇点的路径称为**增广路**。

该算法的具体流程如下：

***step*1**　对于一条连接u与v节点，容量为r，费用为w的边(u,v,r,w), 新连接一条反向弧$(v,u,0,-w)$

***step*2**　在残量网络中，以费用作为两点间的边权，用$SPFA$(*Shortest Path Faster Algorithm*)最短路算法求解出费用最小的增广路，记为dis_T

***step*3**　求出该增广路最大可以通过的流量f_{min}并将$f_{min} \times dis_T$更新到答案，并将路径上所有反向弧的流量增加f_{min}，将路径上所有边的流量减少f_{min}

step*4**　检查是否还有增广路，如果有返回step*2**，否则算法结束

该算法基于一种贪心思维，即先流费用和最小的路径。由于反向弧的存在，给了贪心算法可以“反悔”的机会，并可以求得**全局最优解**。设节点数为N, 边数

为E，单次求解最短路的时间复杂度为$\boldsymbol{O(kE)}$，在稀疏图中一般小于 2。对于本题的数据规模，可在**秒级时间**内出解。算法正确性以及时间复杂度分析的证明见支撑材料[5]。

5.2.5 第二问方案的确定与检验

在确定了购买方案后，进一步考虑 8 家转运商的运输方案，该问题是一个典型的动态规划背包问题，但是考虑到仅有 8 家转运商，并且 8 家转运商的承运能力相对于我们的计划已经没有多少余量，因而采用贪心算法。

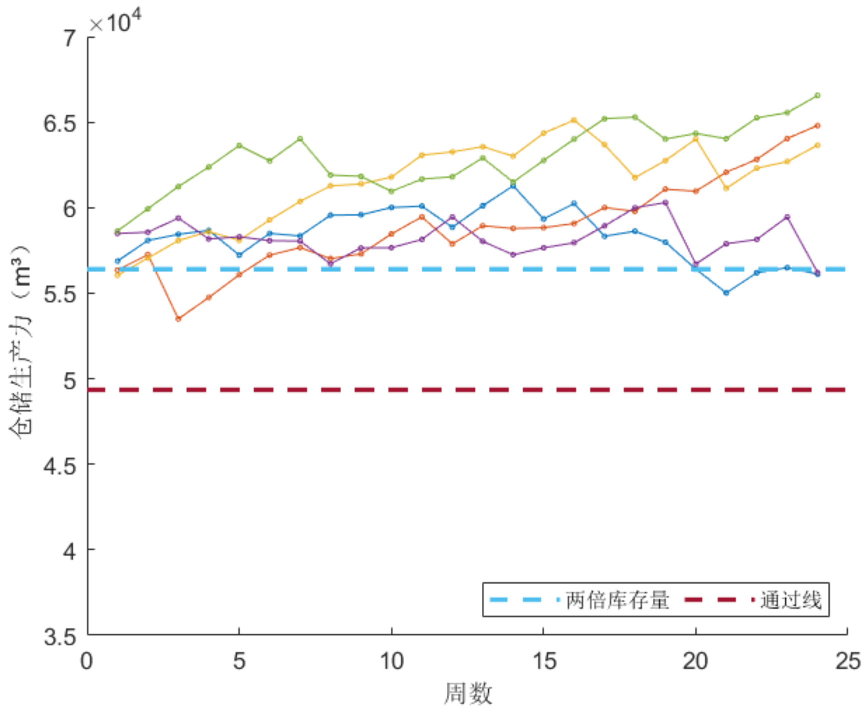
先对转运商按损失率的平均值进行优先度排序。其次对供应商按供应量与供应材料类型进行排序：材料产能的排序程度为$A > B > C$，依照供应商数量除以V_{ABC}，即优先给供应量大的A供应商选择优先度高的转运商，若超出 6000 立方米的运输上限，则依照优先度顺序选择下一个供应商。

最终，根据订货策略和运输策略得到如下结果。

对于第一小问：企业需要的供应商在第一周最少为 38 家，其余 23 周每周至少需要的供应商数量为 10。总共 24 周的所有花费为 644290.687 元。

对于第二小问：为了测试得出的方案，我们选择通过 4.2.2 中给出的大型供应商浮动预测方法，生成 10 万组已选供应商的模拟数据，对方案进行测试，最后统计出该方案的成功率为92.2%，该成功率在这样的测试数量下是满足要求的。

图 7 第二问抽取 5 次模拟结果存储量折线图



5.3 问题三求解

5.3.1 供应商评分变更

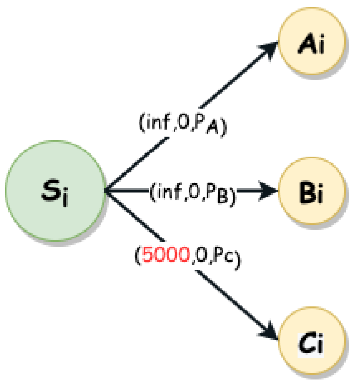
由于第三问要求尽量多买 A 材料而尽量少买 C 材料，可以认为该要求在本问中占有最高优先级，因而希望在评价系数层面让生产 C 材料的供应商评分将低到低优先级。

依据问题一中得出的所有供应商的评分，最高的为编号为 229 的供应商，评分0.793，最低的为编号 47 号的供应商，评分0.347，评分极差为0.35左右，简单将所有 C 类供应商的评分都减去接近极差的0.3，即令：

$$r_C = r_C - 0.3 \tag{24}$$

5.3.2 新增限制条件

由于题目中已经声明尽量少地购买 C 类原材料，故将每周 C 类原材料进购的总和限定上限，网络流图参数变更如下：



5.3.3 模型参数变更

与第二问相同，由于供应商质量降低，为了保持计划在模拟测试中的通过率，故一开始就将运输损失纳考虑，设计保障系数$a\%$，规定企业的每周流量为：

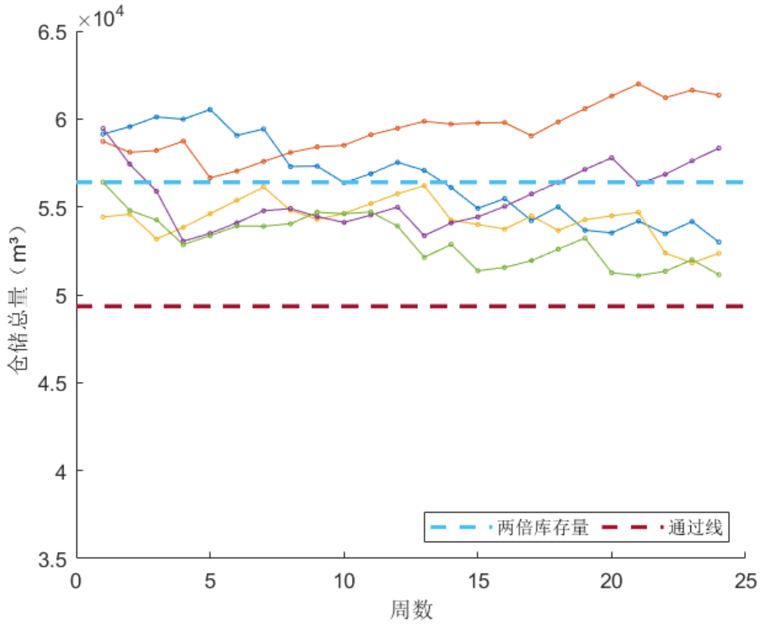
$$Fw = (1 + a\%)Fw \tag{25}$$

5.3.4 模型求解与结果分析

经由上述调整，利用第二问的模型对该问进行求解。

结果为：第一周供应商数量 47，其余 23 周供应商数量 30，平均总费用：638473元。利用模拟方法生成 10 万组数据对该方案进行测试，对于全程仓储稳定在28200×1.75以上的通过结果，通过率为69.3%，选取部分结果如下：

图 8 第三问抽取 5 次模拟结果存储量折线图



5.3.5 问题二、三结果对比分析

1、**利润分析：**问题二 24 周平均总花费 646277 元，问题三 24 周平均总费用 638473 元，两者差值为 7804 元，即第三问同比节省了 1.208%的成本费用。相比起行业平均利润率 10.85%，是很大的提升。

2、**风险分析：**在通过条件相同的情况下，第二问在模拟十万次时的通过率为 **92.2%**，第三问在模拟十万次时的通过率为**69.4%**，可见有偏好地挑选原材料种类势必会导致供应商的质量下降，引起更大的不确定性。

结合以上两点分析，企业可以继续通过我们的模型，权衡利润与风险的利弊，进一步得到更优的方案。

5.4 问题四求解

5.4.1 确定最大供应商组

以供应商为对象考虑问题，在一个方案中，选择的供应商数量越多，企业每周可以获得的材料上限也就更多，可以提高产量；同时，供应商越多，供应商平均质量下降，违约等情形增加会导致方案稳定性变差。使用违约扰动率ω、运输损失率z以及预测最劣利润率三个参数来确定可选的最大供应商组。

按照供应商评分排序，由分数高到低依次向可选供应商组中加入供应商，对于每一次的供应商选择方案，计算其三个方案评价参数：

$$\begin{cases} \omega = \dfrac{\sum_{i=1}^{t}(O_i - D_i)}{\sum_{i=1}^{t} O_i} \\ \\ z = \dfrac{\sum_{i=1}^{t} S_i}{t} \\ \\ \sigma = (1 - \omega - z) \times (10.85\% - z) \end{cases} \tag{26}$$

随着加入供应商质量的下降，方案的风险也会逐渐增大，限定方案评价参数的阈值$\omega + z < 5\%, \sigma > 8\%$, 作为供应商方案可行的阈值。

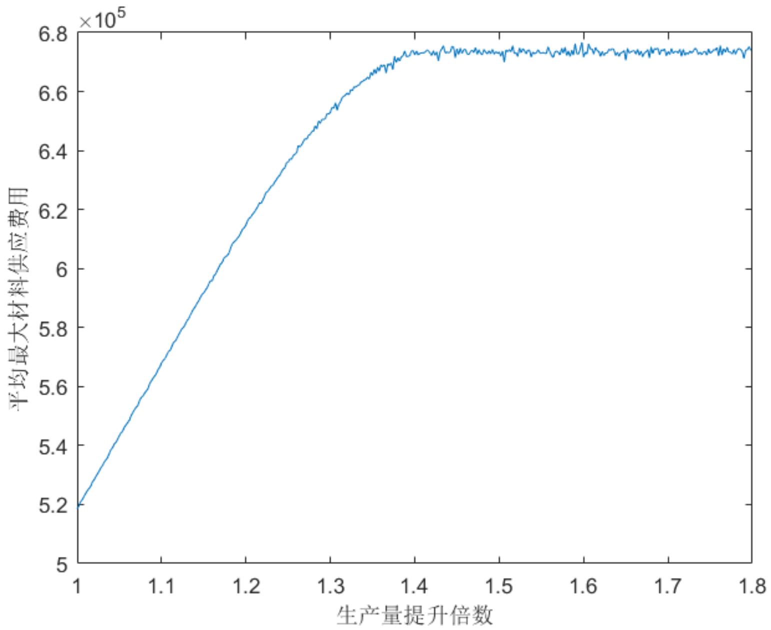
首先使用第一问中得出的所有供应商评分r_i，将供应商以评分从高到低排列，不断将新的供应商按顺序纳入评分，此时企业生产上限增加，但稳定性随着供应商质量变差下降。使用之前分析的三个参数不断判定企业选择的可行性，直到指标超出阈值。

经过处理，确定最多可以选择评分最高的 68 家供应商作为可选供应商组。

5.4.2 模型求解

确定可选供应商组后，通过增加Fw的倍率，确定企业在可选这 68 家供应商时能承担的最大订单额度，结果如图：

图 9 企业生产量提升与费用关系



在企业生产量提升倍数达到1.423时，68 家供应商的期望供应量无法再让生产量继续提高，因而确定本问中生产量的提升倍数为：1.423。

在确定了供应商组以及企业的计划生产提升量后，可以继续使用第二问的模型对 24 周内的具体计划划进行求解。

在实际处理中，通过计算，发现按照第二、三问中在第一周补全之后的 2 周冗余量的方案不可行，因为在新的生产方案下将所有供应商纳入考虑也无法满足企业需求，因而让该企业分数周来实现该目标，在未达到额定仓储量前采用买满的策略，最后得出两周可以达成目标。在做如上更改后，可以求得第四问中 24 周的具体计划。

六、模型评价与推广

6.1 模型评价

优点：

1、使用网络流模型，能够在条件确定的情况下得到最优的结果；同时由于算法复杂度低，即使每次模拟都存在生成随机序列的模型，仍然能在一秒左右给出方案；因此，能够对于方案不断优化，以得到匹配大多数实际情况的优秀方案。

2、充分分析与处理数据：首先排除不良的小供应商，避免熵权法受过大影响；其次去除“老赖”型大供应商，避免“订单总量”权值造成的“刷单”行为；再次，使用熵权法与灰色关联分析，达到有效分析多个因素与确定权值两个目的。

3、加入开工第一周这一特殊情况，不仅证明了模型能够胜任流量变化带来的影响，也使得模型进一步贴近实际，具有更佳的推广意义。

4、全部四问围绕着评价分数与网络流模型展开，具体问题求解时仅是在已有基础上做改动，模型具有统一性、整体性。

缺点：

1、对于确定转运商与供货商间的搬运关系时，采取贪心算法而非动态规划模型，有概率造成微小的损失上升

2、供应商的浮动预测模型较为简陋，以该方法生成随机序列来测试方案时无法更逼真地贴合实际情况。

6.2 模型改进

1、在建立转运模型时，对于转运商仅用平均值一个指标进行排序，可以选择方差、中位数等其他处理方式进行更科学的排序。

2、本题中转运商仅为 8 个，数量较少，在转运商较多的情况下使用动态规划而非贪心算法可以得到优势更大的结果。

6.3 模型推广

1、对于本题的应用场景，对于网络流各边对应的现实环节进行设定调整，都可以通过修改边费用以及边流量来求解新的问题，例如对第二问的模型进行一定调整就可以求解选择偏好、生产计划发生改动的第四问。

2、对于本题之外的应用场景，网络流模型可以用来求解一类可以将需要优化的目标抽象为流量的实际场景，例如交通安排，产品销售，通信网络等，具有很大的泛用性。

七、参考文献

[1] 于会强，林略，孙士涵.供应链环境下的供应商管理库存模型研究[J].2003
[2] 钟建.中国建材发布 2021 年中期业绩报告[J].2012.
[3]姜启源，谢金星，叶俊.数学模型（第四版）[M].北京：高等教育出版社，2016
[4]司守奎，孙兆亮.数学模型算法与应用[M].北京：国防工业出版社，2020
[5]SSP（Successive Shortest Path）算法正确性及复杂度分析
https://oi-wiki.org/graph/flow/min-cost/
[6]有上下界的网络流
https://blog.csdn.net/clove_unique/article/details/54884437

附录一

第一问所选 50 家供应商及其评分

评分1	供应商编号1	评分2	供应商编号2
0.793848483	229	0. 389627984	338
0.709439818	361	0.389509324	40
0.605410706	374	0.38682451	367
0.583100719	108	0.38486751	55
0.505926856	395	0.384069763	218
0.505595457	139	0.38341736	346
0.48605241	330	0.382861227	80
0.480875776	282	0.382758152	294
0.477196427	340	0.381378877	362
0.467069869	275	0.381042646	244
0.465148619	329	0.380925123	5
0.457679167	307	0.38092293	379
0.448519863	131	0.380546713	273
0.448106441	356	0.380279165	67
0.445229936	268	0.379875898	3
0.444309218	306	0.379024012	351
0.427376568	194	0.378787175	189
0.418944493	352	0.378543907	53
0.417757382	37	0.378134329	114
0.416525852	143	0.377531817	78
0.407805654	284	0.377331179	266
0.403535843	247	0.376794241	388
0.396938125	365	0.376319003	291
0.394896695	31	0.375667045	210
0.390842192	364	0.374997045	123

剔除后 302 家供应商

企业编号	企业编号2	企业编号3	企业编号4	企业编号5	企业编号6	企业编号7	企业编号8
229	105	106	147	67	392	264	380
361	85	213	42	3	225	312	159
374	401	30	91	351	155	383	48
108	398	219	310	189	157	253	230
395	363	358	305	53	254	370	28
139	269	128	217	114	259	353	311
330	117	46	195	78	384	176	79
282	326	73	319	266	10	227	122
340	98	303	158	388	54	314	99
275	50	359	220	291	129	2	116
329	36	332	110	210	76	175	354
307	17	336	133	123	149	25	24
131	109	150	111	397	288	280	369
356	72	309	376	270	75	163	102
268	357	65	171	86	202	121	345
306	142	178	255	180	274	214	205
194	27	236	226	33	66	184	185
352	59	256	223	211	233	49	182
37	167	188	313	318	323	169	18
143	252	339	377	74	341	35	61
284	196	221	315	203	141	94	120
247	237	265	115	245	172	136	200
365	56	14	216	292	146	83	399
31	92	174	207	272	173	52	41
364	337	165	360	154	26	90	389
338	299	260	378	342	324	209	82
40	187	295	286	224	88	322	118
367	97	199	191	239	13	382	127
55	104	69	333	235	138	32	124
218	304	381	250	366	197	193	156
346	81	132	23	298	263	87	57
80	145	267	89	44	349	368	335
294	206	70	248	301	186	261	134
362	64	21	316	16	232	113	391
244	20	11	152	300	325	242	101
5	39	327	166	386	258	60	347
379	279	7	170	208	271	249	107
273	296	331	96	222	47		

转运商平均损失率

转运商编号	损失率
3	0.186055556
6	0.543761111
2	0.921370417
8	1.010282759
4	1.570482353
1	1.904769167
7	2.078833333
5	2.889825301

附录二

第一问数据准备

```
clc % 在前两问的代码基础上跑
c = readmatrix('附件1 近5年402家供应商的相关数据.xlsx', 'Sheet', '企业的订货量（m³）',
'Range', 'B2:B403');
ret1 = zeros(50, 241);
ret2 = zeros(50, 241);
sel = r(1:50, 2);
for i = 1:50
	ret1(i, 1) = c(i);
	ret2(i, 1) = c(i);
	ret1(i, 2:241) = a1(r(i, 2), :);
	ret2(i, 2:241) = a2(r(i, 2), :);
end
writematrix(ret1, '50家企业的各项信息.xlsx', 'Sheet', '1')
writematrix(ret2, '50家企业的各项信息.xlsx', 'Sheet', '2')
```

第一问数据处理

```
clc, clear

a1 = readmatrix('附件1 近5年402家供应商的相关数据.xlsx', 'Sheet', '企业的订货量（m³）
', 'Range', 'C2:IH403');
a2 = readmatrix('附件1 近5年402家供应商的相关数据.xlsx', 'Sheet', '供应商的供货量
（m³）', 'Range', 'C2:IH403');
% 缺值判断
if isempty(find(isnan(a1), 1)) && isempty(find(isnan(a2), 1))
	disp('没有缺失值')
end
%% 确定小型供应商
clc
n = 402; m = 240;
m2 = zeros(n, 1);
for i = 1:n   % 计算订单和供货的平均值
	sum2 = 0;
	num = 0;
	for j = 1:m
		if a1(i, j) ~= 0   % 有订货才算和
			sum2 = sum2 + a2(i, j);
			num = num + 1;
		end
	end
	m2(i, 1) = sum2 / num; %供货平均值
```

```
end
% writematrix(m2(:, 1), '企业单次供应量平均值分布图.xlsx')   %  绘制供货量直方图
%%  确定违约
clc
cs = 0;
tmp = [];
for i = 1:n
    for j = 1:m
        if a1(i, j) ~= 0 && a2(i, j) ~= 0 && a1(i, j) > a2(i, j)   %  供货不足
            cs = cs + 1;
            tmp(cs) = a2(i, j) / a1(i, j);
        end
    end
end
sig = std(tmp);
me = mean(tmp);
me, sig
lim = me - sig;   %  算出阈值

tot = 0;   %  小型供应商剔除
del = []; %  剔除企业编号

for i = 1:n
    if m2(i, 1) <= 11 %  属于小型企业
        s = 0; cnt = 0;
        for j = 1:m
            if a1(i, j) ~= 0
                s = s + a2(i, j) / a1(i, j);
                cnt = cnt + 1;
            end
        end
        if s / cnt <= lim %  将该企业剔除
            tot = tot + 1;
            del(tot) = i;
        end
    end
end

%%  确定大型供应商
cs = 0;
qian = [];
for i = 1:n
    if m2(i, 1) >= 50 %  属于大型企业
        s = 0;
```

```
        for j = 1:m
            if a1(i, j) ~= 0 && a1(i, j) > a2(i, j)
                s = s + a1(i, j) - a2(i, j);
            end
        end
        cs = cs + 1;
        qian(cs) = s;
        if (s >= 50000)
            tot = tot + 1;
            del(tot) = i;
        end
    end
end
qian = qian';
% writematrix(qian, '企业拖欠订单总额分布图.xlsx')
%%
del = sort(del);
d = zeros(402, 1);
for i = 1:tot
    d(del(i)) = 1;
end
% writematrix(del, '第一步剔除企业编号.xlsx')
```

第一问灰色关联分析

```
clc % 继承上一步变量
nn = n - 100; m = 240;
k = 9; % 9个指标
x = zeros(nn, k);

x(:, 1) = sum(a1(d(:)==0, :), 2); % 总数
x(:, 2) = sum(a2(d(:)==0, :), 2);
x(:, 3) = max(a1(d(:)==0, :), [],   2); % 最大值
x(:, 4) = max(a2(d(:)==0, :), [], 2);

cs = 0;
for i = 1:402   % 平均值
    if d(i) == 1
        continue
    end
    cs = cs + 1;
    s1 = 0; s2 = 0;
    num = 0;
    for j = 1:m
        if a1(i, j) ~= 0
```

```
                s1 = s1 + a1(i, j);
                s2 = s2 + a2(i, j);
                num = num + 1;
            end
        end
        x(cs, 5) = s1 / num;
        x(cs, 6) = s2 / num;
end
cs = 0;
for i = 1:n   %  交货小于订货占总交易周数数的比
        if d(i) == 1
            continue
        end
        cs = cs + 1;
        s = 0; num = 0;
        for j = 1:m
            if a1(i, j) ~= 0    %  发生交易
                num = num + 1;
                if a2(i, j) < a1(i, j)
                    s = s + 1;
                    x(cs, 8) = x(cs, 8) + a1(i, j) - a2(i, j); %违约总金额
                end
            end
        end
        x(cs ,7) = s / num; %  计算比例
end
cs = 0;
for i = 1:n
        if d(i) == 1
            continue
        end
        cs = cs + 1;
        num = 0;
        for j = 1:m
            if a1(i, j) ~= 0 && a2(i, j) >= a1(i, j)    %  发生交易且供货量多了
                num = num + 1;
                x(cs, 9) = x(cs, 9) + a2(i, j)/a1(i, j);    %  累加比率
            end
        end
        x(cs, 9) = x(cs, 9) / num;    %  最后除以周数
end
m0 = mean(x(:,9));
%%
mx = max(x);
```

```
mi = min(x);
for j = 1 : k
    if j <= 6
        x(:, j) = (x(:, j) - mi(j)) / (mx(j) - mi(j));
    elseif j <= 8
        x(:, j) = (mx(j) - x(:, j)) / (mx(j) - mi(j));
    else
        for i = 1:nn
            if x(i, j) <= lim
                x(i, j) = 1;
            else
                x(i, j) = 1 - (m0 - x(i, j))/(m0 - mx(j));
            end
        end
    end
end
%%
x(x <= 0.0001) = 0.0001;
x(x >= 0.9999) = 0.9999;
s = sum(x);
p = x ./ s;
e = p .* log(p);
E = - sum(e) / log(n);
w = (1 - E) ./ (k - sum(E))

%% 灰度模型
mx = 0; mi = inf;
for i = 1:nn
    for j = 1:k
        mx = max(mx, 1 - x(i, j));
        mi = min(mi, 1 - x(i, j));
    end
end
eta = zeros(nn, k);
for i = 1:nn
    for j = 1:k
        eta(i, j) = (mi + 0.5*mx)/(1 - x(i,j) + 0.5*mx);
    end
end
r = zeros(nn, 2);
cs = 0;
for i = 1:n
    if d(i) == 1
        continue
```

```
    end
    cs = cs + 1;
    r(cs, 1) = sum(w .* eta(cs,:));
    r(cs, 2) = i;
end
r = sortrows(r, 'descend');
writematrix(r, '第一问企业编号和分数.xlsx')
```

第二问模型测试

```cpp
#include <bits/stdc++.h>
using namespace std;

const int n = 50;   // 50家企业
int abc[n + 5], ed[n + 50], st[n + 50];
double fr[n + 50], r[n + 50];
const double V[] = {0, 0.6, 0.66, 0.72};

double rnd(int val, int i)
{
    double bia = rand() < 26000 ? ed[i] * (1 + r[i]): -1 * st[i] * (1 - r[i]) * (1 - r[i]);
    bia *= rand() / 32767.0;
    if (val + bia > 5990.0) return 5990.0;
    return max(val + bia, 0.0);
}
int c[55][55], lo[55][55];
double lv[10][250];
double lost(int i)
{
    int x = rand()%240 + 1;
    while (lv[i][x] < 1e-8)
        x = rand()%240 + 1;
    return lv[i][x];
}

int work(int id = 0)
{
    double s = 28200 / 0.97;
    int cnt, tmp[30], tot = 0;;
    for (int w = 1; w <= 24; w ++)
    {
        for (int i = 1; i <= 50; i++)
        {
            if (c[i][w] == 0) continue;
```

```
            double x = rnd(c[i][w], i), cur;

            x -= x * lost(lo[i][w]);
            x /= V[abc[i]];
            s += x;
        }
        s -= 28200;
        tmp[++tot] = s;
        if (s >= 28200 * 1.9) cnt = 0;
        if (s < 28200 * 1.9 && w > 1) ++cnt;
        if (cnt >= 2) return 0;
    }
    if (id == 1)
    {
        for (int i = 1; i <= tot; i++)
            cout<<tmp[i]<<" ";
        puts("");
    }
    return 1;
}

int main()
{
    freopen("data.txt", "r", stdin);
    srand(time(0));
    for (int i = 1; i <= 50; i++)
        scanf("%lf%d%d%d%lf", fr+i, abc+i, ed+i, st+i, r+i);

    freopen("ans2.txt", "r", stdin);
    for (int i = 1; i <= 50; i++)
        for (int j = 1; j <= 24; j++)
            scanf("%d", &c[i][j]);
    freopen("tsel.txt", "r", stdin);
    for (int i = 1; i <= 50; i++)
        for (int j = 1; j <= 24; j++)
            scanf("%d", &lo[i][j]);

    freopen("lv.txt", "r", stdin);
    for (int i = 1; i <= 8; i++)
        for (int j = 1; j <= 240; j++)
        {
            scanf("%lf", &lv[i][j]);
            lv[i][j] *= 0.01;
        }
```

```
    int C = 100000, cnt = 0;
    for (int cs = 1; cs <= C; cs++) // 模拟十万次
        cnt += work();
    printf("%.5lf\n", 1.0 * cnt / C);
    freopen("5line.txt", "w", stdout);
    for (int cs = 1; cs <= 5; )
        cs += work(1);
    return 0;
}
```

第二问聚类算法

```
clc, clear

a1 = readmatrix('50家企业的各项信息.xlsx', 'Sheet', '1', 'Range', 'B1:IG50');
a2 = readmatrix('50家企业的各项信息.xlsx', 'Sheet', '2', 'Range', 'B1:IG50');
%%
clc
x = mean(a2, 2);
y = std(a2, 0, 2);
a = zeros(50, 2);
a(:, 1) = x;
a(:, 2) = y;
C = kmeans(a, 2);

%%
clc
r = zeros(5, 50);
for i = 1:50
    if C(i) == 1   % 小企业直接取平均值
        r(1, i) = mean(a2(i, :));
    else
        cnt = 0; % 大订单总数
        suc = 0; % 大订单成功数
        s = 0; % 大订单平均值
        for j = 1:240
            if a2(i, j) >= 1000 && a1(i, j) >= 500
                cnt = cnt + 1;
                s = s + a2(i, j);
                if a2(i, j) / a1(i, j) >= 0.88
                    suc = suc + 1;
                end
            end
        end
```

```
        r(1, i) = s / cnt * suc / cnt;
    end
end
c = readmatrix('50家企业的各项信息.xlsx', 'Sheet', '1', 'Range', 'A1:A50');
r(2, :) = c';

for i = 1:50 %  最大上差值和下差值
    for j = 1:240
        if (a2(i, j) > a1(i, j))
            r(3, i) = max(r(3, i), a2(i, j) - a1(i, j));
        else
            r(4, i) = max(r(4, i), a1(i, j) - a2(i, j));
        end
    end
end
c = readmatrix('第一问企业编号和分数.xlsx', 'Sheet', '1', 'Range', 'A1:A50');
r(5, :) = c'; r = r';
writematrix(r, 'data.txt', 'Delimiter', ' ')
%%
hold on
scatter(x(C == 1), y(C == 1), 50, 'bx');
scatter(x(C == 2), y(C == 2), 50, 'rx');
hold off
xlabel('平均值')
ylabel('标准差')

%%  大订单指标
clc
cs = 0; y = [];
for i = 1:12
    if C(i) == 1
        continue
    end
    for j = 1:204
        if a1(i, j) ~= 0
            cs = cs + 1;
            y(cs) = a2(i, j);
        end
    end
end
y = y';
writematrix(y, '大企业供货额度分布图.xlsx')
```

第二问网络流算法

```
#include <bits/stdc++.h>
using namespace std;
const double inf = 1e8;

const int n = 50;   // 50家企业
int abc[n + 5], ed[n + 50], st[n + 50];
double fr[n + 50], r[n + 50];
const double V[] = {0, 0.6, 0.66, 0.72};
const int fw = 28200 * 1.05;

double rnd(int i)
{
	double bia = rand() % 2 ? ed[i] * (1 -r[i]) : -1 * st[i] * (1 - r[i]);
	bia *= rand() / 32767.0;
	if (fr[i] + bia > 5990.0) return 5990.0;
	return max(fr[i] + bia, 0.0);
}

int lim[55][55], ret[55][55], ans[55][55];
int vec[5][55], siz[5];

namespace Net
{
	const int N = 110 * 24;   //总结点数
	int S, T, etop=1, head[N], d[N];
	int pre[N], vis[N], tim;
	double dis[N], cost;
	struct Edge
	{
		int u, v, r, nxt;
		double c;
	}e[N * 50]; //双倍边
	void addedge(int u,int v,int r,double c)
	{
		++etop;
		e[etop].u = u; e[etop].v = v; e[etop].r = r; e[etop].c = c;
		e[etop].nxt = head[u];
		head[u] = etop;
	}
	void add(int u,int v,int r,double c)
	{
		addedge(u,v,r,c);
		addedge(v,u,0,-c);
	}
```

```
bool spfa() //找费用最小的一条路  并用pre记录路径
{
	queue <int> q;
	q.push(S);
	memset(pre,0,sizeof(pre));
	for (int i = 0; i < N; i++)
		dis[i] = inf;
	dis[S]=0; vis[S]=++tim;
	while (!q.empty())
	{
		int u=q.front(); q.pop();
		for (int v,i=head[u];i;i=e[i].nxt)
		{
			v=e[i].v;
			if (e[i].r&&dis[v]>dis[u]+e[i].c)
			{
				dis[v]=dis[u]+e[i].c;
				pre[v]=i;
				if (vis[v]!=tim)
				{
					vis[v]=tim;
					q.push(v);
				}
			}
		}
		vis[u]=0;
	}
	return dis[T] != inf;
}
void mcmf()
{
	int flow = 0;
	while (spfa())
	{
		int minflow=inf;
		for (int i=pre[T];i;i=pre[e[i].u])
			minflow=min(minflow,e[i].r);
		for (int i=pre[T];i;i=pre[e[i].u])
		{
			e[i].r-=minflow;
			e[i^1].r+=minflow;
		}
		flow += minflow;
		cost += dis[T]*minflow;
```

```
    }
    printf("%.3lf\n", cost);
}
void build_G()
{
    int s = (1 + 3 + 100 + 3 + 1) * 24 + 1;
    int t = s + 1;
    S = t + 1; T = S + 1;

    for (int bia = 1; bia <= 2485; bia += 108) // 23 * 108 = 2484
    {
        add(bia, bia + 1, inf, 1.2 * V[1]);
        add(bia, bia + 2, inf, 1.1 * V[2]);
        add(bia, bia + 3, inf, V[3]);

        int xj = bia == 1 ? 2*fw : fw;
        add(s, bia, inf, 0);
        d[s] -= xj;
        d[bia] += xj;
    }

    for (int u, v, bia = 0, j = 1; bia <= 2484; bia += 108, j++)
    {
        for (int i = 2; i <= 108; i++)
        {
            u = bia + i;
            if (i >= 2 && i <= 4) //ABC -> 供应商
                for (int j = 1; j <= 50; j++)
                {
                    if (i - 1 != abc[j]) continue; //购买点货物类型匹配上
                    v = bia + 4 + j;
                    add(u, v, inf, 0);
                    v += 50;
                    add(v, bia + 103 + i, inf, 0); //后拆点连向储存点
                }
            if (i >= 5 && i <= 54)
            {
                lim[i - 4][j] = rnd(i - 4);
                add(u, u + 50, int(lim[i - 4][j] / V[abc[i - 4]]), 0); //企业间拆点连边
            }

            if (i >= 105 && i <= 107)
            {
                double fy = V[i - 104];
```

```
                fy *= 0.056;
                add(u, bia + 108, inf, 0); //AS -> T1

                v = bia != 2484 ? u + 108 : t; //最后一个点直接连t，不然就 +108
                add(u, v, inf, fy); //上界inf，下界0
            }
            if (i == 108) //上界减下界为0，直接不连边
            {
                d[u] -= fw;
                d[t] += fw; //流向t
            }
        }
    }

    for (int u = 1; u <= t; u++)
    {
        if (d[u] < 0)
            add(u, T, -d[u], 0);
        if (d[u] > 0)
            add(S, u, d[u], 0);
    }
    add(t, s, inf, 0);
}

int t[4];
void fill(int i, int A, int w)
{
    while (A)
    {
        while (r[vec[i][t[i]]] + 0.1 <= r[vec[2][t[2]]] && A && t[2] <= siz[2] && w != 1) //
填B企业
        {
            int del = min(A, lim[vec[2][t[2]]][w] - ret[vec[2][t[2]]][w]);
            A -= del;
            ret[vec[2][t[2]]][w] += del;

            if (lim[vec[2][t[2]]][w] == ret[vec[2][t[2]]][w]) //B企业填满了
                ++t[2];
        }
        if (A)
        {
            int del = min(A, lim[vec[i][t[i]]][w] - ret[vec[i][t[i]]][w]);
            A -= del;
            ret[vec[i][t[i]]][w] += del;
```

```
                if (lim[vec[i][t[i]]][w] == ret[vec[i][t[i]]][w])
                    ++t[i];
            }
        }
    }
    void output(int A, int B, int C, int w) //产能要乘以VA,VB,VC
    {
        t[1] = t[2] = t[3] = 1;
        A *= V[1]; B *= V[2]; C *= V[3];

        fill(1, A, w);
        fill(3, C, w);
        while (B)
        {
            int del = min(B, lim[vec[2][t[2]]][w] - ret[vec[2][t[2]]][w]);
            B -= del;
            ret[vec[2][t[2]]][w] += del;
            if (lim[vec[2][t[2]]][w] == ret[vec[2][t[2]]][w]) //B企业填满了
                ++t[2];
        }
    }
    void work()
    {
        for (int i = 2, j = 1; j <= 24; i += 8, j++)
            output(e[i + 1].r, e[i + 3].r, e[i + 5].r, j);
        for (int i = 1; i <= 50; i++)
            for (int j = 1; j <= 24; j++)
            {
                if (j == 1 && ans[i][j] == 0)
                    ans[i][j] = ret[i][j];
                if (j != 1) ans[i][j] += ret[i][j];
            }
    }
    void init()
    {
        etop=1;
        memset(head, 0, sizeof(head));
        memset(lim, 0, sizeof(lim));
        memset(ret, 0, sizeof(ret));
        memset(d, 0, sizeof(d));
        cost = 0;
    }
}
```

```
int main()
{
	freopen("data.txt", "r", stdin);
	freopen("cost2.txt", "w", stdout);
	srand(19260817);
	for (int i = 1; i <= 50; i++)
	{
		scanf("%lf%d%d%d%lf", fr+i, abc+i, ed+i, st+i, r+i);
		vec[abc[i]][++siz[abc[i]]] = i;
	}
	int cs = 100;
	while (cs--)
	{
		Net::init();
		Net::build_G();
		Net::mcmf();
		Net::work();
	}
	freopen("ans2.txt", "w", stdout);
	for (int i = 1; i <= 50; i++)
	{
		for (int j = 1; j <= 24; j++)
			printf("%d ", j != 1 ? ans[i][j] / 100 : ans[i][j]);
		puts("");
	}
	return 0;
}
/*
1.建立附加源点SS，和附加汇点TT

2.对于原图中每一个点（包括源汇）u,令d[u]代表u点的所有入边的流量下界减去出边的流量下界

2.1.如果d[u]是负数，那么从u连一条边(u,TT,-d[u],0)到TT

2.2.如果d[u]是正数，那么从SS连一条边(SS,u,d[u],0)到u

3.对于原图中每一条边(u,v,w,l,r)，连边(u,v,r-l,w)

4.连边(t,s,inf,0)（注意这里是原图的源汇点！不是附加的源汇点！！）

这样以后，从SS到TT跑新图的最小费用最大流，再加上原图中每条边的下界流量乘以费用（必须跑的部分），就是最小费用可行流的费用了
*/
```

注：第三问、第四问代码仅有少量改动

附录三（支撑材料目录）

```
C:.
│  中国建材发布2021年中期业绩报告_钟建.pdf
│  供应链环境下的供应商管理库存模型研究_罗兵.pdf
│  柱状图1.jpg
│  柱状图2.jpg
│  柱状图3.jpg
│  流程图1.jpg
│  流程图2.jpg
│  网络图.jpg
│  第一问答案.png
│  附件A 订购方案数据结果.xlsx
│  附件B 转运方案数据结果.xlsx
│
├─draw
│      8家转运商平均损失率与编号.xlsx
│
├─Q1
│      302家供应商编号.png
│      50家企业的各项信息.xlsx
│      draw.xlsx
│      draw_2.xlsx
│      GRA.m
│      init.m
│      select.m
│      企业单次供应量平均值分布图.xlsx
│      企业拖欠订单总额分布图.xlsx
│      第一步剔除企业编号.xlsx
│      第一问企业编号和分数.xlsx
│      附件1 近5年402家供应商的相关数据.xlsx
│
├─Q2
│      50家企业的各项信息.xlsx
│      50家企业聚类结果.png
│      5line.txt
│      5line_2.png
│      8家转运商平均损失率与编号.xlsx
│      ans2.txt
│      cost2.txt
```

```
|       data.txt
|       draw.m
|       julei.m
|       lv.txt
|       net.cpp
|       net.exe
|       pingjia.cpp
|       pingjia.exe
|       save_ans.m
|       tsel.txt
|       大企业供货额度分布图.xlsx
|       第一问企业编号和分数.xlsx
|       附件 2 近 5 年 8 家转运商的相关数据.xlsx
|       附件 A 订购方案数据结果.xlsx
|       附件 B 转运方案数据结果.xlsx
|
├─Q3
|       3.xlsx
|       50 家企业的各项信息.xlsx
|       5line.txt
|       5line_3.png
|       ans3.txt
|       cost3.txt
|       data.txt
|       draw.m
|       lv.txt
|       net.cpp
|       net.exe
|       pingjia.cpp
|       pingjia.exe
|       save_ans.m
|       tsel.txt
|       第一问企业编号和分数.xlsx
|       附件 2 近 5 年 8 家转运商的相关数据.xlsx
|       附件 A 订购方案数据结果.xlsx
|       附件 B 转运方案数据结果.xlsx
|
└─Q4
        8 家转运商平均损失率与编号.xlsx
        ans4.txt
        choose.m
        cost.txt
        cost4.txt
        data.txt
```

final_draw.m
net.cpp
net.exe
net_draw.cpp
net_draw.exe
Q4_final(每隔 0.002 倍取点).png
save_ans.m
第一问企业编号和分数.xlsx
第四问确定 68 家企业编号.xlsx
附件 1 近 5 年 402 家供应商的相关数据.xlsx
附件 2 近 5 年 8 家转运商的相关数据.xlsx
附件 A 订购方案数据结果.xlsx
附件 B 转运方案数据结果.xlsx