**组长：**

**基于EOQ模型的去库存问题研究**

**摘要**

库存管理是企业运转不可或缺的重要部分，更是实现供应链增值的重要环节。在供应链管理模式下，库存量的高低不仅影响着单一企业的综合成本，而且制约着整条供应链的性能。

本文立足于树状供应链结构，在第一题中针对库存与到货率的关系，编写C++程序，在系统的初始库存一定的情况下，按等差数列从高到低依次分布到各级公司，商家订单量和库存量作为自变量，到货率作为因变量，得到足够的数据，并利用MATLAB二元非线性回归分析得到到货率与商家库存的指数次方成正比，与商家订单量的指数次方成反比的结果。

针对第二题，将供应链抽象为数学模型，研究在源头供应商供货量稳定的基础上，通过调整各级分销商和末端收货商的补货时机和补货批量，使得在到货率有所保障的前提下，整个供应链系统的库存的总成本尽可能降低。

首先根据经典模型：经济订购批量—EOQ模型，得出在理想条件下库存管理的方法，之后基于EOQ模型，考虑上初始成本以及缺货的情况等因素作出更适用于实际供应链的改进模型，利用微分方程知识寻找使库存总成本最小的库存批量q，并得到此时的补货点r。

第三题基于第二题建立的库存管理模型，求解到货率为95%时的最优库存。

**关键词：**供应链管理 EOQ模型 多级库存

**1.问题重述：**

某行业货物供应商（Supplier）通过各公司（Fac），进而向下级子公司（Fac）直至零售商发行某种专业商品，货物的流通过程如下图。一般地，某个发货商有可能同时在其它订单中也作为收货商，所以该图只是显示了某批货物可能的运输销售流程，但不足以表示发货商与收获商的上下级关系，通常它们会形成一个网状结构，如附件。

另一附件数据集“产品流转数据”是一份销售链行为记录表。数据集的每条观测记录了一次订单情况，即某个上级发货商到下级收货商的某货物批次的情况，包括某上级供应商的某批次产品在某天流通到某个下级收货商的具体信息。在平日里，各公司都有一个初始库存，假设公司的库存量一旦小于某个值就会立即向其某个上级下订单补货，订单量为常数。而上级供应商要向多个下级供货，因此下级发来的订单请求未必能得到满足，记下级收货商实际收到的货量占其需求量的百分比的值为到货率。目前该商品较为紧俏，末端收货商（实际使用部门）需求旺盛，到货率也仅有90%。

该行业供应商关心如下问题，邀请你们小组在尚未提高生产能力之前提升到货率，降低流通库存。

（1）库存与到货率之间究竟有什么关系？

（2）求若要满足目前到货率90%不变，并且使所有分销商的库存量总和最小，和的值应该为多少？库存总和需要你自己定义。

（3）若生产能力提高，估算能使末端收货商的到货率提高至95%，请重新估算供应商的最优库存。

**2.问题分析**

第一题要研究库存与到货率的关系。到货率即供应链中所有公司实际收到的商品/系统中所有公司发出的总订单数量，由所给数据分析可发现，共有七个等级的商家，且各商家之间的关系复杂。同一等级的供货商，面对的收货商数目不尽相同，同一商家由于需求共同的多种商品，因而会处在不同的等级上。同时，通过查阅资料及对数据进一步分析可知，到货率与订单量q，供货商库存A之间存在着直接的联系，与补货点r存在间接关系：订单量的大小是由末端市场需求决定的，而库存的大小除了公司管理层的人为主观因素，在一定程度上简间接的也是由市场决定的，并且，对于库存确定的上一级，随着订单量的增大，收货商的到货率会相应的降低；同样的对于订单量固定的情况，随着库存的增大，在一定范围内到货率也将随之增大，但对于补货点r而言，由于补货点主要是由订购前置时间，使用率决定，因而，在此忽略其对到货率的影响。

第二题旨在研究供应链中库存管理的问题。库存管理要解决的主要问题是补货时机和补货批量。在源头供应商供货量稳定的基础上，通过调整各级分销商和末端收货商的补货时机和补货批量，使得在到货率有所保障的前提下，整个供应链系统的库存的总成本尽可能降低。根据经典模型：经济订购批量—EOQ模型，寻找合适的补货点和订单量，然而EOQ模型假设条件太强，过于理想化，所以在EOQ模型的基础上，考虑上初始成本以及缺货的情况等因素作出更适用于实际供应链的改进模型，寻找使库存总成本最小的库存批量q，并得到此时的补货点r。

第三题即利用第二题所建立模型，对到货率提升的情况进行研究。相较于到货率提升前的情况，末端销售商的到货率虽然得到提高，但对于而言，自身的销售能力在一定范围内是固定的，不会因为到货率的提高自身的需求量也随之显著提高，因而可假定末端销售商的订货量保持不变，进而可知，每一级的订货量是由下一级的订货需求决定的，由于下一级的需求不变，因而，其自身向上一级的订单量也不变。由此而带来的只是实际到货量的增加，从而影响到商家在一个周期内的的订单数量。于是用第二题的模型可以得到相应的最优库存。

**3.模型假设**

（1）该供应链系统运行无限长的时间；

（2）模型中涉及到的产品在运输、存储过程中,不会被损毁；

（3）每次订货的订货费相同,与订货批量的大小无关

（4）订货提前期固定；

（订货提前期[2],又称为前置时间,是指从提出订货申请直至收到订货的期间;或者从另一个角度看,为了自某一时刻能补充存储,必需提前订货,这段时问也称之为订货提前期。）

（5）末端收货商需求量的预测值，认为是直接建立在顾客历史上购买产品的订单数据上的。

（6）每次订货的运输量不受限制。

（7）在整条供应链物流中，只考虑单一产品

（8）同一商家只在一个级

（9）0级表示制造商，数字越大等级越低，末端为收货商

**4.变量说明**

η：到货率

q ：订货量

r ：补货点

A：库存

：常数

：常数

i：商家所处的级数

j：相应级数上的商家标识

：第i级第j个商家的周期订购成本

：第i级第j个商家的周期存储成本

：第i级第j个商家的周期缺货成本

：第i级第j个商家的周期运输成本

**5.模型建立与求解**

**5.1针对问题一：为了获取到货率、订单量、库存三者的近似关系，做以下假设：**

（1）每一级的初始库存相同，上一级的初始库存总是大于下一级的初始库存

（2）除了0级经销商之外，每一个经销商都只有一个上级

（3）在一个周期以内，整个总的销售系统并不会有货物的输入，只有输出

（4）最底层每一个经销商的销售量（即输出）满足均值为100的泊松分布

（5）假设从最上级经销商到最下层经销商数量分别为1、5、10、20、30、50、100

（6）补货点r=200

**5.1.1获取数据：**

假设最低库存为r，每次的订货量为q，所有经销商的库存总和为total，根据经验，一级经销商的总库存为0.1\*total，二级所有经销商的总库存为0.2\*total，三级:0.15\*total,四级:0.15\*total，五级:0.15\*total,六级:0.1\*total，七级:0.15\*total。

首先用系统函数随机给每一个下级商家分配一个上级商家，生成一个随机销售网络。然后利用泊松分布给每一个最下级商家生成一个销售量，用其初始库存减去销售量所得即为其目前实际库存。

利用for循环，对每一个七级经销商进行判断，如果实际库存小于r，便向六级申请调货，如果六级此时的库存大于q便满足下级的订货，如果六级库存小于q便向下级输送剩余的库存量，然后向五级经销商订货，五级经销商再次重复这个过程。如果六级经销商调货失败，

那么他的下级剩余的订单都将不会被满足，此七级经销商的调货过程全部结束，进入下一个经销商的调货过程。然后在改变参数观察到货率的变化，由此得出各个参数与到货率的关系。q 由200—40 变化幅度为20，total 由100000-10000变化幅度为2000。

运用C++编写程序得到数据4005条数据(附录1)，同时，由问题分析可设：

其中，表示到货率，表示库存，表示订单量，为未知参数，对上式进行线性处理得到：

由此，用MATLAB软件将获得的数据带入做二元非线性回归分析，得如下图：

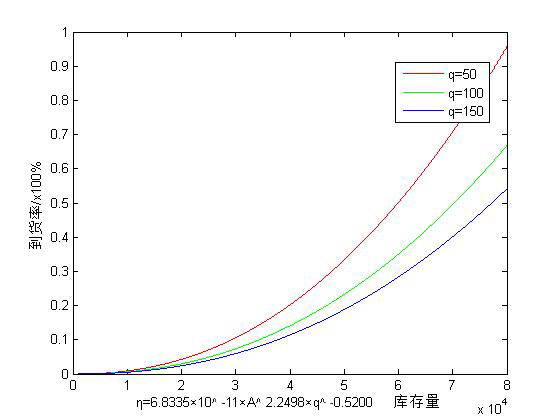


图1

由此可得到货率η与库存和订单量之间的关系：

**5.2针对问题二：整条供应链为供货商—分销商—末端收货商**

我们可以将整条供应链有n个销售级，每个销售级有个

**5.2.1经济补货批量（EOQ）模型** [3]

（1）每当库存余额小于订货点时,就发出固定批量的订货

（2）库存余额=现有库存量十己订未到量一欠付量

（3）计算经济订货批量的目的是为了平衡订货成本和持有成本之间的关系,使得库存总成本最小。使库存总成本最小的订货批量称为经济订货量。

**5.2.2在EOQ模型中我们假设：**

（1）单位时间内的需求量不变;

（2）物品的购入单价为常数，到货率为百分之90%，且不允许断货情况。

订货点为订货提前期内的需求量。刚开始时，库存数量为q单位。由于需求速率固定,随后库存数量以固定的速率降低。当库存量降低到订货点时,发出一批订货量为q单位的订单。经过一个固定的提前期后,有0.9q单位的物品到达并入库,物品即将入库时的库存数量为零。（虽然到货率不是百分百，但依然能保证其不断货）物品即将入库时，库存数量为零，该模型库存量与时间的关系如下图所示（cd为订货间隔期，ca为订货提前期）：

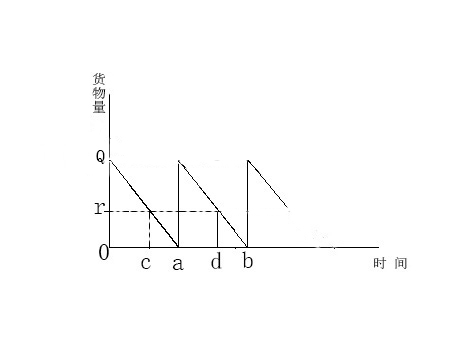


图2

在一个周期内的总需求量一定的情况下,订货批量越小,平均库存量就越小,但发生的订货次数越多。如果能寻找到合适的订货批量，使库存总成本最小，就能加快库存资金周转,有利于提高企业效益。

在不允许缺货的情况下：其周期库存总成本可分析如下:

周期总库存成本=周期购入成本+周期定购成本+周期存储成本，即：

式中：

D—周期需求量；

Q—订货批量

为了确定经济订货批量，求周期总成本对订购批量的一阶导数并令其为零，得：

从而求得最佳经济订货批量为：

则最佳年订货次数n和两次订货间的平均时间T可如下确定：

平均订货间隔期为：

补货点由下式确定：

r=D\*L—订货提前期内的平均需求

r—补货点；

l—订货提前期，

将经济订货量代入年总成本的表达式，得到最小年库存总成本的公式：

**5.3 EOQ改进模型**

针对一在第级的末端收货商：每天的需求量随机分布，均值为，服从泊松分布,,，。假设在一个周期内的需求量为常数

则订单量:

**5.3.1费用函数建立：**

1. 第级供应商向第级的供应商的周期订购成本:

—第级供应商每次向第级供应商订货时发生的订购费；

—订货批次；

—在一个周期内的需求量;

—购入产品的单价；

1. 第级供应商的周期存储成本

—的初始库存

—单位产品的周期存储费

收货商库存量与时间的关系如图所示：

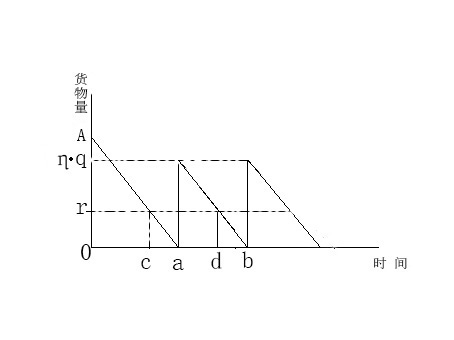


图3

1. 周期缺货成本

设第级供应商的周期缺货成本, 提前期服从泊松分布：，，，则在提前期t内的平均消耗量为：，第天的实际消耗量，服从泊松分布：，提前期内的总消耗量：

所以，实际缺货为个的概率为：

进而，第级供应商的周期缺货成本：

1. 运输成本

第级供应商向第级收货商的周期运输成本

—第级供应商的运输单位产品发生的运输费

1. 的总成本:

*s.t*

供应链有级,且第级供应链有个供应商,则处在末端第级的所有收货商的总成本：

针对在第级的公司，假设它要向个处于第级的公司供货，处于第级的公司，满足上述讨论的模型，那么的库存量与时间的关系如图所示：

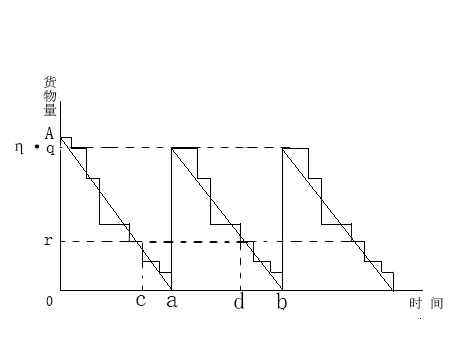


图4

的初始库存为，个跟它进货的公司中有公司向其下订单的时候，库存量垂直下降减少订单量个数量，因为第级个公司的订货周期是可以根据历史数据确定的，所以公司可以预测出补货点，在库存量降到时又向上一级的供货商进货。

故公司的库存量与时间的关系如梯形折线图所示，为了考虑计算的方便，将其拟合为类似于公司模型的直线图。

由此，整个供货链的周期总成本可表示为：

*s.t*

由，，，，，式有,周期成本变化率与订货量之间的关系为：

由于每个订单量最小，则总的订货量最小，而第级，第个的商家的周期最小订单量为：

所以，最小的周期订单总量为：

*s.t*

由问题一可知，周期到货率与库存、订货量之间的关系为：

所以，当到货率为90%时，周期库存可表示为：

所以，整条供应链的周期最小库存和为：

此时，

**5.4针对问题三**

对于问题三，假定第0级为生产商，因其每个周期能产生一定量固定的商品，同样可类似于问题二中的公司模型。设整条供应链有级，每一级拥有个商家。从而可知整条供应链的周期总成本仍为：

*s.t*

从而其周期最小订货总量保持不变，为：

又由问题一可知：

所以，当到货率为95%时，周期库存可表示为：

所以，整条供应链的周期最小库存和为：

**6.模型评价：**

**优点：**

电脑所产生的随机数据具有参考价值，符合客观规律，运用Matlab软件对库存与到货率进行了非线性二次回归拟合，具有良好的拟合效果。

结合概率分布、多元线性规划和微分方程等数学方法,以整个供应链库存总成本最小为目标,建立了多级供应链库存控制模型，具有实际意义和可操作性。

在经典的库存管理模型的基础上，考虑初始库存，断货成本等因素，使模型更加贴近实际。

**不足：**

供应链中的每个企业的需求预测,我们认为是建立在直接顾客历史上购买产品的订单数据上的,需求预测是安排生产调度,协调生产能力,控制库存和生产资源的基础。采用这样简单的预测方法来预测向供应商订货的数量,其预测的结果往往与实际有所出入。

模型未考虑批量订货的价格折扣问题，也未考虑实际情况中各公司的资金限制，仓库的容量限制等问题，有待进一步完善。

**局限性： [3]**

在供应链模型中，信息的不准确性会沿着供应链逐级提高,逐级放大，达到供应链的源头—制造商时,其预测的需求信息和实际消费市场中的顾客需求信息会发生很大的偏差,这就是需求放大效应(又称牛鞭效应)。上游供应商订单的变动性明显大于下游供应商需求的变动性,为了满足这种需求需求,上游供应商往往维持比下游供应商更高的库存水平,从而产生更高的成本。

**应对措施：** [3]

在供应链上建立集中化的信息系统。通过集中信息处理,为供应链上每一级企业提供完全的及时的最终需求信息,有助于减轻需求放大效应。

设法建立更加完善的预测末端收货商消费量的模型。减小预测与实际需求的误差,有助于降低整个供应链系统的不稳定性。

改变企业经济效益测评模式。企业采用传统的单体经济效益测评方法,很容易引向局部和孤立的评价,而忽视系统的反应和结果。应使用系统分析和系统测评的方法,用于分析某项策略与措施对整个系统的作用和影响，降低整个供应链的稳定性。

**参考文献：**

[1]赵晓波，黄四民，库存管理，北京：清华大学出版社，2008.2.

[2]张玉斌，采购与仓储管理，北京：对外经济贸易大学出版社，2008

[3]秦海燕，供应链库存控制系统模型研究及仿真：硕士学位论文p15-p18,p30-p33

[4]原思聪，MATLAB语言与应用技术，北京：国防工业出版社，2011.8.

[5]郭敏，库存原理简明教程，武汉：华中科技大学出版社，2013.4

[6]王夏阳，供应链库存管理—基于提前期不确定性的研究，北京：经济管理出版社，2010.1.

**附录：**

**C++程序：**

#include <iostream>

#include <time.h>

#include <stdlib.h>

#include <math.h>

#include <fstream>

using namespace std;

#define H 100 //柏松分布均值

#define A 1

#define B 5

#define C 10

#define D 20

#define E 30

#define F 50

#define G 100

//各级供货商数量

int first ;

int second[B];

int third[C][2];

int forth[D][2];

int fifth[E][2];

int sixth[F][2];

int seventh[G][2];

double xuqiu=0;

double huode=0;

int r;

int q;

double per;

void checktwo(int a)

{

int i=a;

for(;second[i]<r;)

{

xuqiu++;

if(first>=q)

{

huode=huode+q;

second[i]=second[i]+q;

first=first-q;

}

else

{

huode=huode+first;

second[i]=second[i]+first;

first=0;

int b=(r-second[i])/q;

if(b\*q!=r-second[i])

b=b+1;

if(second[i]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

}

void checkthree(int a)

{

int i =a;

for(;third[i][0]<r;)

{

xuqiu++;

if(second[third[i][1]]>=q)

{

huode=huode+q;

third[i][0]=third[i][0]+q;

second[third[i][1]]=second[third[i][1]]-q;

}

else

{

huode=huode+second[third[i][1]];

third[i][0]=third[i][0]+second[third[i][1]];

second[third[i][1]]=0;

}

if(second[third[i][1]]==0)//二级厂家调货

{

int a=third[i][1];

checktwo(a);

}

if(second[third[i][1]]==0)//二级厂家调货失败

{

int b=(r-third[i][0])/q;

if(b\*q!=r-third[i][0])

b=b+1;

if(third[i][0]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

if(second[third[i][1]]<0)//二级厂家调货

{

int a=third[i][1];

checktwo(a);

}

}

void checkfour(int a)

{

int i=a;

for(;forth[i][0]<r;)

{

xuqiu++;

if(third[forth[i][1]][0]>=q)

{

huode=huode+q;

forth[i][0]=forth[i][0]+q;

third[forth[i][1]][0]=third[forth[i][1]][0]-q;

}

else

{

huode=huode+third[forth[i][1]][0];

forth[i][0]=forth[i][0]+third[forth[i][1]][0];

third[forth[i][1]][0]=0;

}

if(third[forth[i][1]][0]==0)//三级厂家调货

{

int a=forth[i][1];

checkthree(a);

}

if(third[forth[i][1]][0]==0)//三级厂家调货失败

{

int b=(r-forth[i][0])/q;

if(b\*q!=r-forth[i][0])

b=b+1;

if(forth[i][0]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

if(third[forth[i][1]][0]<r)//三级厂家调货

{

int a=forth[i][1];

checkthree(a);

}

}

void checkfive(int a)

{

int i=a;

for(;fifth[i][0]<r;)

{

xuqiu++;

if(forth[fifth[i][1]][0]>=q)

{

huode=huode+q;

fifth[i][0]=fifth[i][0]+q;

forth[fifth[i][1]][0]=forth[fifth[i][1]][0]-q;

}

else

{

huode=huode+forth[fifth[i][1]][0];

fifth[i][0]=fifth[i][0]+forth[fifth[i][1]][0];

forth[fifth[i][1]][0]=0;

}

if(forth[fifth[i][1]][0]==0)//四级厂家调货

{

int a=fifth[i][1];

checkfour(a);

}

if(forth[fifth[i][1]][0]==0)//四级厂家调货失败

{

int b=(r-fifth[i][0])/q;

if(b\*q!=r-fifth[i][0])

b=b+1;

if(fifth[i][0]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

if(forth[fifth[i][1]][0]<r)//四级厂家调货

{

int a=fifth[i][1];

checkfour(a);

}

}

void checksix(int a)

{

int i=a;

for(;sixth[i][0]<r;)

{

xuqiu++;

if(fifth[sixth[i][1]][0]>=q)//五级厂商库存充足

{

huode=huode+q;

sixth[i][0]=sixth[i][0]+q;

fifth[sixth[i][1]][0]=fifth[sixth[i][1]][0]-q;

}

else//五级厂商库存不足

{

huode=huode+fifth[sixth[i][1]][0];

sixth[i][0]=sixth[i][0]+fifth[sixth[i][1]][0];

fifth[sixth[i][1]][0]=0;

}

if(fifth[sixth[i][1]][0]==0)//五级厂家调货

{

int a=sixth[i][1];

checkfive(a);

}

if(fifth[sixth[i][1]][0]==0)//五级厂家调货失败

{

int b=(r-sixth[i][0])/q;

if(b\*q!=r-sixth[i][0])

b=b+1;

if(sixth[i][0]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

if(fifth[sixth[i][1]][0]<r)//五级厂家调货

{

int a=sixth[i][1];

checkfive(a);

}

}

double ber(int k)

{

double sum=0;

for(int i=0;i<k+1;i++)

{

double j = exp(-H)\*pow(H,i);

for(int k=i;k>0;k--)

j = j/k;

sum = sum + j;

}

return sum;

}

void han(int total)

{

\_sleep(1000);

srand((unsigned)time(NULL));

first = total\*0.1;

for(int i=0;i<B;i++)

second[i] = total\*0.2/B;

for(int i=0;i<C;i++)

{

third[i][0] = total\*0.15/C;

int ran=0+rand()%(B);

third[i][1] = ran;

}

for(int i=0;i<D;i++)

{

forth[i][0] = total\*0.15/D;

int ran=0+rand()%(C);

forth[i][1] = ran;

}

for(int i=0;i<E;i++)

{

fifth[i][0] = total\*0.15/E;

int ran=0+rand()%(D);

fifth[i][1] = ran;

}

for(int i=0;i<F;i++)

{

sixth[i][0] = total\*0.1/F;

int ran=0+rand()%(E);

sixth[i][1] = ran;

}

for(int i=0;i<G;i++)

{

seventh[i][0] = total\*0.15/G;

int ran=0+rand()%(F);

seventh[i][1] = ran;

}//生成随机网络

for(int i=0;i<G;i++)

{

double p=(double)rand()/RAND\_MAX;

if(p<ber(0))

seventh[i][0] = seventh[i][0];

else

{

for(int j=0;;j++)

{

if(ber(j) < p && p < ber(j+1))

{

if(seventh[i][0]>=(j+1))

seventh[i][0] = seventh[i][0] - (j+1);

else

{

seventh[i][0] = 0;

}

break;

}

}

}

}//泊松分布销售量随机化

int ran;

for(int i=0;i<G;i++)

{

for(;seventh[i][0]<r;)

{

xuqiu++;

if(sixth[seventh[i][1]][0]>=q)//六级厂家库存充足

{

huode=huode+q;

seventh[i][0]=seventh[i][0]+q;

sixth[seventh[i][1]][0]=sixth[seventh[i][1]][0]-q;

}

else//六级厂家库存不足

{

huode=huode+sixth[seventh[i][1]][0];

seventh[i][0]=seventh[i][0]+sixth[seventh[i][1]][0];

sixth[seventh[i][1]][0]=0;

}

if(sixth[seventh[i][1]][0]==0)//六级厂家调货

{

int a=seventh[i][1];

checksix(a);

}

if(sixth[seventh[i][1]][0]==0)//六级厂家调货失败

{

int b=(r-seventh[i][0])/q;

if(b\*q!=r-seventh[i][0])

b=b+1;

if(seventh[i][0]<r)

{

xuqiu=xuqiu+b;

}

break;

}

}

if(sixth[seventh[i][1]][0]<r)//六级厂家调货

{

int a=seventh[i][1];

checksix(a);

}

}

xuqiu=xuqiu\*q;

per=huode/xuqiu;

xuqiu=0;

huode=0;

}

int main()

{

fstream fout;

fout.open("text.txt");

for(r=200;r>20;r=r-20)

{

for(q=200;q>20;q=q-20)

{

for(int k=100000;k>10000;k=k-2000){

han(k);

fout<<r<<" "<<q<<" "<<k<<" "<<per<<endl;

}

}

}

return 0;

}

**MATLAB程序：**

function [M,l,m]=finder

A=[100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000 100000 98000 96000 94000 92000 90000 88000 86000 84000 82000 80000 78000 76000 74000 72000 70000 68000 66000 64000 62000 60000 58000 56000 54000 52000 50000 48000 46000 44000 42000 40000 38000 36000 34000 32000 30000 28000 26000 24000 22000 20000 18000 16000 14000 12000]';

q=[200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 180 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 160 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40]';

yita=[1 0.997543 0.995214 0.991588 0.988462 0.986969 0.985347 0.984068 0.986566 0.983972 0.937007 0.648929 0.857321 0.469453 0.685032 0.574247 0.618033 0.279887 0.45885 0.561179 0.523033 0.353555 0.484298 0.170593 0.185111 0.165419 0.127315 0.129862 0.213466 0.0940911 0.0937079 0.0706904 0.0880662 0.0468394 0.0518998 0.0486124 0.0401071 0.0411591 0.0362385 0.0307122 0.0303748 0.0226557 0.0208649 0.0183308 0.0139161 1 0.988556 0.984447 0.986842 0.975446 0.978476 0.966109 0.958516 0.47304 0.729579 0.546137 0.305173 0.38941 0.296973 0.170032 0.151201 0.132606 0.147734 0.144817 0.113078 0.144211 0.102524 0.0962423 0.117316 0.0803674 0.0808462 0.0769717 0.0657177 0.0766566 0.0616385 0.0588928 0.0472677 0.0365945 0.0408001 0.040729 0.0429742 0.0338567 0.026291 0.0272413 0.020892 0.0204117 0.0201908 0.0154288 0.0149407 0.0108422 0.978117 0.974949 0.96521 0.955834 0.936209 0.960471 0.954499 0.749903 0.613851 0.691907 0.306167 0.455598 0.609285 0.274142 0.272664 0.211721 0.183135 0.266766 0.0812548 0.156751 0.162967 0.166764 0.146167 0.153155 0.131056 0.111477 0.0959076 0.0887643 0.087179 0.0653408 0.0546081 0.0613073 0.0567271 0.0500901 0.038941 0.0431077 0.0448866 0.0392923 0.0314125 0.0309202 0.0261031 0.0214833 0.0182416 0.014447 0.0115387 0.947169 0.940419 0.968247 0.964268 0.959408 0.953387 0.956209 0.952719 0.950058 0.956571 0.473901 0.652716 0.432177 0.408707 0.929283 0.514918 0.374926 0.414752 0.166188 0.230559 0.2151 0.179422 0.121891 0.142848 0.120918 0.108112 0.12876 0.120114 0.1232 0.0855583 0.0754684 0.0675619 0.0630242 0.071844 0.0451508 0.0452561 0.0363913 0.0330359 0.0328592 0.0306312 0.0245723 0.0238936 0.0172732 0.0176333 0.0136358 0.978805 0.978717 0.974449 0.961651 0.950099 0.960232 0.973436 0.968603 0.973557 0.971316 0.492554 0.921914 0.91984 0.671552 0.97018 0.906558 0.879599 0.593846 0.274697 0.453855 0.268967 0.287169 0.18646 0.333106 0.173478 0.22878 0.126213 0.160702 0.231148 0.114428 0.0959512 0.0674917 0.0723836 0.0638795 0.0606247 0.0468836 0.0516554 0.0475093 0.0319325 0.03505 0.0333761 0.025991 0.0202827 0.0215893 0.016905 1 0.997388 0.994303 0.992105 0.99019 0.988016 0.986078 0.984266 0.982878 0.983479 0.984031 0.98125 0.980738 0.97796 0.975688 0.855915 0.971092 0.971528 0.826741 0.965605 0.809164 0.348951 0.544634 0.31069 0.309672 0.322914 0.263513 0.225606 0.234028 0.107907 0.131668 0.0994235 0.0746097 0.081504 0.0913503 0.0614315 0.0548638 0.0602942 0.0571401 0.0427029 0.0403591 0.0274819 0.0317867 0.0220335 0.0225212 0.99187 0.993978 0.989052 0.979034 0.970043 0.96793 0.97596 0.970317 0.972649 0.964223 0.999657 0.992631 0.990986 0.98392 0.875567 0.987919 0.989655 0.811874 0.414772 0.44652 0.610678 0.271245 0.212949 0.193797 0.17691 0.184995 0.240182 0.183897 0.161238 0.117265 0.126446 0.0897196 0.0977529 0.0720481 0.0662652 0.0601453 0.0553589 0.0585691 0.0431263 0.0403657 0.0395164 0.0280209 0.0247814 0.0213528 0.0175767 1 0.999529 0.999694 0.996303 0.98968 0.991698 0.99739 0.990355 0.991008 0.987923 0.982222 0.976016 0.983306 0.98326 0.985536 0.964553 0.96785 0.962406 0.575543 0.851647 0.456359 0.281983 0.444779 0.261075 0.260443 0.206589 0.19004 0.155419 0.108274 0.129813 0.143539 0.0999474 0.101766 0.0745129 0.0773203 0.0717185 0.0560003 0.0463553 0.0474803 0.0471472 0.039721 0.0314159 0.028745 0.0250866 0.0230021 1 0.999161 0.996543 0.995693 0.993309 0.99045 0.98774 0.987259 0.981326 0.979514 1 0.997819 0.995735 0.994134 0.993176 0.990562 0.989944 0.987826 0.986536 0.903498 1 0.969891 0.946419 0.535618 0.511241 0.266753 0.239664 0.216095 0.286174 0.203164 0.19283 0.149448 0.131256 0.105977 0.103554 0.102263 0.0902712 0.0756815 0.0673582 0.0601195 0.056852 0.0524852 0.0375942 0.0330309 0.0282679]';

A=log(A);

q=log(q);

yita=log(yita);

G=[ones(size(A)),A,q];

ans1=G\yita;

M=exp(ans1(1));

l=ans1(2);

m=ans1(3);

A=1000:10:80000;

q1=50;

yita1=M.\*A.^l.\*q1.^m;

plot(A,yita1,'-r');

hold on

q1=100;

yita1=M.\*A.^l.\*q1.^m;

plot(A,yita1,'-g');

hold on

q1=150;

yita1=M.\*A.^l.\*q1.^m;

plot(A,yita1,'-b');

xlabel('¦Ç=6.8335¡Á10^-11¡ÁA^2.2498¡Áq^-0.5200 ¿â´æÁ¿');

ylabel('µ½»õÂÊ/x100%');

end