|  |
| --- |
|  |
| **2016年同济大学数学建模竞赛** |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **选题：A**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 参赛组信息 | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  |   日期：2016年5月3日 |

**库存问题的优化讨论**

**摘要**

仓储调控是物流的一个重要环节，宏观定量的物流管理可以大大提高运送效率，具有重大的实际意义。针对以供应库存为背景提出的三个问题，本文运用了概率统计、线性优化，存储论等相关知识，明晰了进货下界，订单量，物流网，到货率等因素之间的潜在联系。

针对问题一，根据供应关系，结合对附件所提供的网状关系参考图，通过筛选数据分析得到符合实际情况的网状关系图。从而弄清供应商、分销商以及末端收货商之间的供应关系。基于分析得到的供应关系，对网状关系中的单元体进行剖析，宏观上利用正态分布估计分销商的进货概率，得出分销商的订单量以及实际收到的货物量，进而得到到货率与库存的关系，由这些规律在Matlab上开发出到货率的计算器（见附录第1项）。

针对问题二，根据实际数据，对问题一的模型进行简化，将原有复杂关系网简化为1:2的展开，无跨级进货关系的简单关系网；将原有销售模式简化为末端紧销持续进货的连续进货模式。最终导出一个相对简洁，需求变量少的近似公式，即在已知结构与拖延比例下可以求得整个系统的最优库存与各级别公司的订货点、订货量，并制作了相应的随机分布下的到货率计算程序（见附录第3项）。

针对问题三，在原有问题的解决下，提出了到货率变动对订货量，订货点以及最优库存的影响，并分析了更适用与较广的整体系统，而订货分支较复杂的末端，相比在灵敏度上较敏感同时准确度较低。

在解决三个问题后，综合模型整体进行讨论，得出了模型的适用性评价与准确性分析，并由已有发现提出了模型的改进空间与发展可能。为不同需求对象提供了围绕到货率的分级评价机制。

**关键字**：网状关系；元胞；订货点；订货量；首尾端；

**一、问题重述**

某行业货物供应商通过各公司，下级子公司，直至零售商发行某种专业商品，。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，因此多方的交叉使他们通常会形成一个网状结构，即多对多型。

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某几个上级下订单补货，订单量为常数。而一个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%。

在尚未提高生产能力之前，供应商希望提升到货率，降低流通库存。由此分析：

（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。

**二、模型假设**

1. 订货的网状结构为：对于某一级别的某一个公司，有若干个上级与若干个下级，公司的级别由订货量决定；同级之间不相互供货与要货，下级也不向上级供货；同一个公司一般具有若干个较为稳定的供货商和收货商。

2. 同一级别的公司具有相同的订货量（题中q值）与订货下界（题中r值），一旦公司库存低于订货下界r值便向所有的上级发送相同份的订单，订单量为q。

3. 上级供货商对下级收货商供货，上级在供货时尽量满足所有订单，即上级供货商的库存小于订单需求数时，按订单比重将库存分给所有的下级。

4. 零售末端在每一个操作周期内，都能将所有收到的货物投入实际使用，即消耗；生产企业在每一个操作周期内，都生产固定数量的某种类商品。

5. 下级收货商在长期的交易变动中，逐渐形成稳定于订货下界附近的库存波动，从而进货的概率满足正态分布。

**三、符号说明**

|  |  |
| --- | --- |
| **符号** | **意义** |
|  | 某收货商第i级上级发货商的个数 |
|  | 某发货商第i级下级收货商的个数 |
|  | 网络特征关系，即与某一商家相联系的上级与下级网络 |
|  | 第i级，网络关系为S的收货商的到货率 |
|  | 第i级，网络关系为S的供货商的库存数量 |
|  | 第i级，收货商订单需求的货品数量 |
|  | 第i级，网络关系为S的收货商的订货概率 |
|  | 第i级，订货下界 |

**四、问题分析**

**4.1 问题一分析**

由于模型求解的是供应商关心末端收货商的到货率和减少总流通库存，那么我们可以不去关心具体的网状关系图，而是根据网状关系的几个特征：级数、收货商数量、收货商继续发货的比例等来推导出首末端的关系。

**4.2 问题二分析**

流通库存总和即是求中间公司（既收货又发货的公司）的总的平均库存总和，可以把模型根据时间特征来确定周期与拖后时间的关系。在一个周期内的库存量-时间模型的梯形面积即为中间公司的流通库存总和。

**4.3 问题三分析**

供应商的最优库存问题，对于不同货物，由于存在不同等级，网络模型不同。在长时间范围来看，末端收货率由90%提高到95%，需求不变，而生产能力提高，那么生产能力就变为原来的95/90，相当于供应公司的实际得货均值变为原来的95/90。

**五、模型建立和解决**

**5.1 第一种思路：网络节点公司的通用公式**

**5.1.1问题一模型的建立和解决**

5.1.1.1符号说明

|  |  |
| --- | --- |
|  | 某收货商第i级上级发货商的个数 |
|  | 某发货商第i级下级收货商的个数 |
|  | 网络特征关系，即与某一商家相联系的上级与下级网络 |
|  | 第i级，网络关系为S的收货商的到货率 |
|  | 第i级，网络关系为S的供货商的库存数量 |
|  | 第i级，收货商订单需求的货品数量 |
|  | 第i级，网络关系为S的收货商的订货概率 |
|  | 第i级，订货下界 |

5.1.1.2 网状关系确定

根据附件中提供的网状关系参考图，在题目提供的不完整关系图的基础上，分析出符合实际的供应商、分销商以及末端收货商之间的关系建立原则：

1. 网状关系中供应商、分销商以及末端收货商存在明显的分级现象，其中承担生产任务的供应商（Supplier）作为第一级，也兼具有向下一级的分销商发货的能力；仓库容量，发货能力，发货数量，拥有的下级订货商数量相差不大的分销商视为同一级；末端收货商作为最后一级；（以级数较小的为上级）
2. 网状关系的主体为分销商，实际情况中，同一等级的分销商之间罕有互相订发货的情况，在建立网状关系图时可以忽略。所以规定网状关系中同一等级分销商之间视作不存在订货发货关系；
3. 下级分销商可以拥有多个不同等级的上级分销商（即子公司可以向不同等级的大公司订货）。同样一个分销商可以拥有多个不同等级的向其订货的分销商；进而得到较为符合实际的网状关系图：

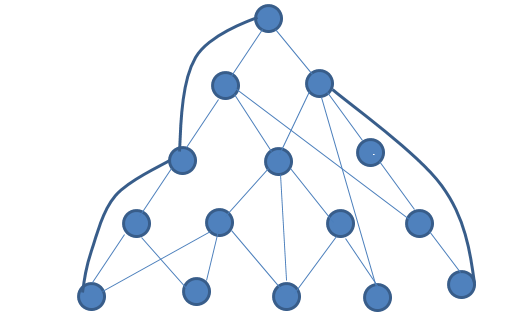


图5-1 实际网状关系

5.1.1.3 网状关系“元胞”的分析

分析上述建立的网状关系图，可将其视作由若干下图所示的单元体构成：

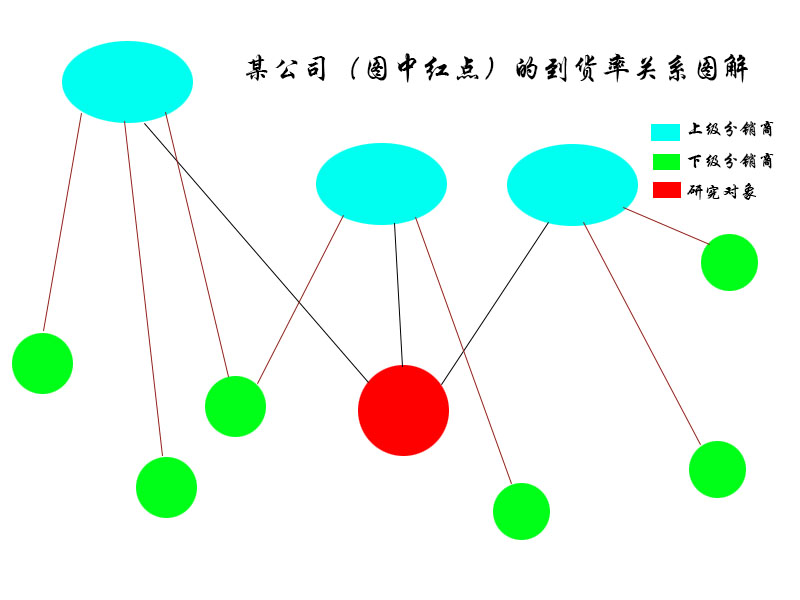


图5-2 网状关系“元胞”

在这个单元体中，由某一固定分销商，向该分销商供货的上级分销商或供货商，以及相应的下级分销商（这些下级分销商向图中浅蓝色表示的分销商订货）组成。该单元体是整个网状关系的构成单元，我们将它称之为网络关系的“元胞”。

在讨论到货率与库存之间的关系时，我们以“元胞”中的固定分销商（即图中红色圆代表的公司）为研究对象，分析它的到货率与为其发货的上级分销商的库存量之间的关系。

已知该“元胞”来自于固定关系网S，“元胞”中的固定分销商处于S中的第级，该分销商的发货公司（即图示浅蓝色的圆）可能分布于向S中的第1到第i-1级中任意级；假设每一级分销商向上一级单个分销商提交的订单量不一样，用表示第i级分销商向上一级提交的订单量。分以下几步计算“元胞”中的固定分销商的到货率：

（1）确定该分销商的需求量

表示向该分销商供货的上级分销商个数（即该公司向几个上级公司订货）。则该分销商在每个订货时间段内，向上级提供的总订单数量为\*，也就是该分销商的需求量。

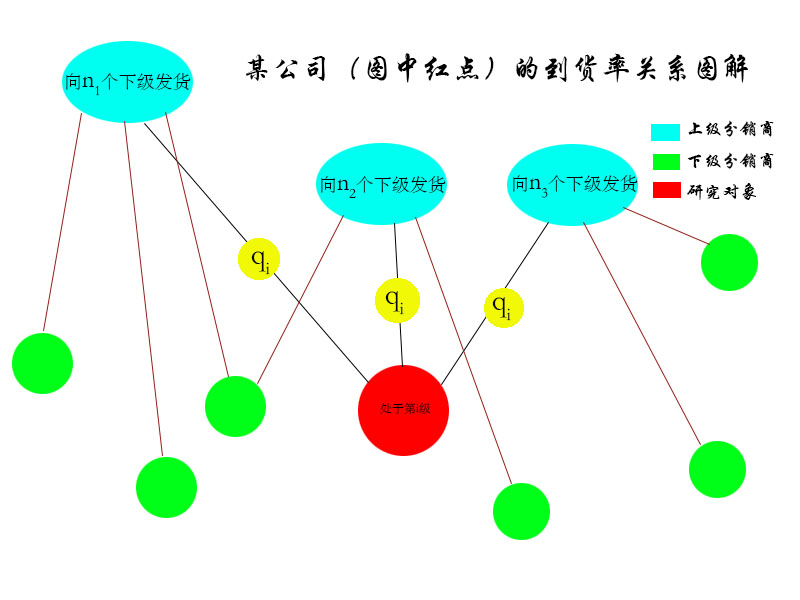


图5-3 网状关系“元胞”

（2）确定该分销商实际收到的货物数量

向该分销商供货的个上级分销商分布在第1级到第i-1级，它们除了收到该固定分销商的订单外，还会收到来自于其他级的分销商的订单，n表示这些上级分销商中的一个向n个分销商供货（即与单个浅蓝色圆链接的绿色圆的个数）。处于第a(a≤i)级的供销商收到的总的订单量为：

其中表示分销商的下级在固定时间段内向其发订单的概率（后文会作描述）。

而第a级的分销商向下级发货的发货量须取决于它该时间段内的库存量。第i级的分销商向此供销商发出的订单在这个供销商收到的的总订单量中所占比重为：

根据模型假设：

①当分销商现有库存量小于其收到的订单量时，需要按照订单比重将其库存全部发出；

②分销商现有库存量大于等于收到的订单量时，则能满足每一笔订单需求，即发货量等于订单量。

所以第a级供销商向研究对象（即第i级分销商，红色圆）发出的货物量为：

因为在第i级的分销商的供货方（即浅蓝色圆）中，从第1级到第i-1级每一级的分销商供货时都遵循这个假设，所以只需要累加即可得到第i级收获方实际收到的货物量：

（3）计算到货率

根据（1）（2）步中得到的结论，根据第i级分销商实际收到的货量占其需求量的百分比的值为到货率，计算得到：

从而得到到货率与库存量的关系。

(4)补充说明

：表示的是分销商的下级在固定时间段内向其发订单的概率

由于当每一级分销商的库存量低于时该分销商就得向上级分销商提交订单进货；当库存量大于等于时不进行补货。这可以称作分销商的进货原则。

所以相当于一个进货点，或者将它称之为安全库存。在实际情况下，分销商的库存量一直在值附近波动，波动偏差维持在它每次的最大进货数量附近。在仓库容量允许的情况下每一级的库存量最大将达到+,最小可以为0。所以每个分销商的库存量可以用均值为,方差为的正态分布，且库存量取值限定在0至+之间。

设某级分销商的库存量为X，X服从的分布为:

X～N（, ）（0≤X≤+）

则根据分销商的进货原则，估计进货概率即为

其中：

**5.1.2 问题二模型的建立和解决**

5.1.2.1 模型准备

在模型近似方面，问题一中的综合模型可以准确求解该问题，但计算量和需求量较多，需要假定多个可能不同的已有变量，因此为方便求解，找出可接受误差范围内的模型最优值，综合4.2中的问题分析，将原有复杂关系网简化为末端为1:2的下分模式，无跨级进货关系的简单关系网；将原有销售模式简化为末端紧销持续进货的连续进货模式。

5.1.2.2 模型建立

对一个n级分销系统，针对已有的模型，通过对原有数据的分析，我们发现跨越2级或以上的供货模式在末端与中后端较少，同时所占比重远远小于上下两级的直接交易，因此去除小量的跨界进货，在符合紧销限制下，可以将原有公式简化为：

上式中去除了有关跨级进货的求和计算，但求解仍需大量参数。在此考虑到，模型末端的数据统计简化，可以令上式中的m为1，n为某定值。则单一供货可以确定内的网状特征关系为S(1,n)，则不存在多项不同的。由此，可再此简化模型：

式中为某一级分销商的到货率模型，将所有分销商库存累加，即求解各i级别库存和，可得到简化模型：

在此处,概率不能简单沿用模型一中的分布，而是关于进货点r的函数，因为在各级在供发稳定后，一般的，订货等待时间小于销售时间并保持一定比例,同时订货等待时间大于缺货时间，可知，这个比例系数k的范围为:

在确定的比例下，r值的变动将直接影响进货可能，改变整个级别进货的分布，即：

同时，式子满足=0.9(i=1,2,…,n)，则以递推需求，则带入简化模型，最终可以得到：

由此，等式左侧为模型的初始设定值：公司的下级分级数n，初始需求，左侧为本问题所需求解的式，对于整体库存最小时，即本体所需最优解为：

**5.1.3 问题三模型的建立和解决**

问题三的模型即为问题二的最终简化模型，但在此基础上，模型三将到货率=0.95(i=1,2,…,n)，提升，由此带入最终的简化模型，则相比于问题二，最优解的系数发生了变化，同时由于生产能力的提高，初始需求能得以更好满足，因此此时：

在此时最优解为：

我们可以发现，在此模型中，订货点没有直接影响库存量而影响库存的反而是需求，这可能与末端现实有差异，因为现实中的需求供给没有将时间简化至极小值，但对于长期的大范围需求，该模型具有一定的参考意义。

**5.2第二种思路：通过网络关系得到供应商—末端收货商的关系**

第二种思路和第一种思路是建立在和第一种思路相同的假设基础之上的，而关键不同之处在于第二种思路不考虑具体的网络模型，而是建立在足量的数据分析时可以得到的需求量波动情况、不同级公司数量关系等参数的基础上仅研究供应商和末端收货商的属性关系，对于供应商来说有更具实用意义的关系模式。

模型图解：

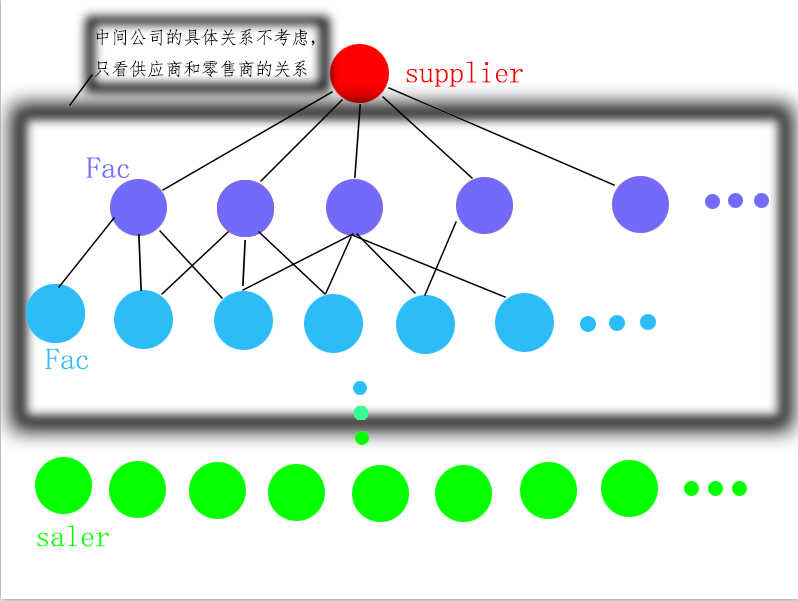


图5-4 供应分析图例

对于题目所给的数据，我们对各等级的公司数量进行了分析：

1、对上游到下游的直接发货公司数量比例分析

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 产品 | 上游 | 下游 | 上游 | 下游 | 上游 | 下游 | 上游 | 下游 | 上游 | 下游 |
| 编号 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 |
| 1 | 1 | 60 | 39 | 541 | 61 | 424 | 92 | 340 | 10 | 34 |
| 2 | 1 | 3 | 55 | 1062 | 92 | 978 | 108 | 451 | 19 | 69 |
| 3 | 1 | 89 | 47 | 989 | 88 | 893 | 94 | 396 | 9 | 26 |
| 4 | 1 | 489 | 26 | 2132 | 27 | 88 | 16 | 32 | / | / |

2、对发货公司总数量进行数量分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 产品\发货级 | 1 | 2 | 3 | 4 |
| 1 | 42 | 62 | 94 | 10 |
| **2** | **61** | **93** | **112** | **18** |
| **3** | **51** | **90** | **96** | **8** |
| 4 | 31 | 27 | 15 | / |

3、对收货公司总数量进行数量分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 产品\收货级 | 1 | 2 | 3 | 4 | 5 |
| 1 | 76 | 599 | 813 | 385 | 40 |
| 2 | 100 | 1150 | 1326 | 535 | 69 |
| 3 | 74 | 1065 | 1041 | 447 | 29 |
| 4 | 75 | 226 | 245 | 37 | / |

**5.2.1 问题一模型的建立和解决**

首先绘出实际库存波动的变化图，根据实际情况与模型假设，Q在长期变化过程中会趋向一个稳定的值，这个值是订单大小和缺货数量均值的和。图中虚线部分的意义是公司无货，但需求仍存在，实际库存量在虚线所表示的时间段内的值为0。虚线部分的存在，即无法满足的下级需求量，可以推导出下级的到货率为1-（缺货率），下面将会用公式及变量具体表达。

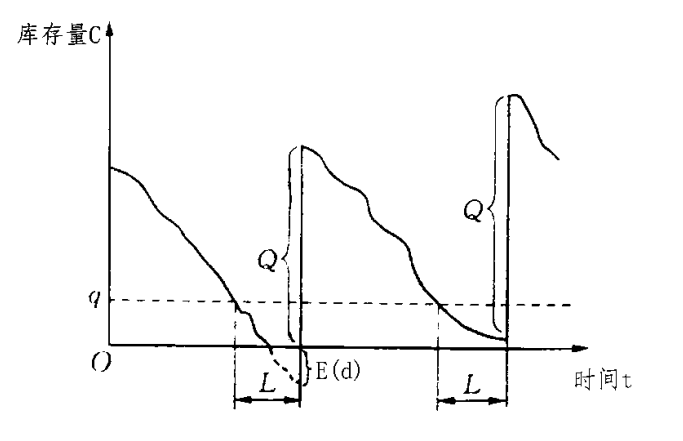


图5-5 实际库存波动变化图

5.2.1.1符号说明

|  |  |
| --- | --- |
| **符号** | **意义** |
|  | 单位时间需求量 |
|  | 拖后时间内需求量 |
|  | 拖后时间 |
|  | 订货下界 |
|  | 订单大小 |
|  | 实际得货均值 |
|  | 时间 |
|  | 需求量 |
| **C** | 库存量 |

5.2.1.2 具体计算

在拖后时间L内的缺货量的期望值：

E(d）=

单位时间内订货次数期望值为：

单位时间内缺货量为：

对实际库存-时间模型进行优化，得到下图：

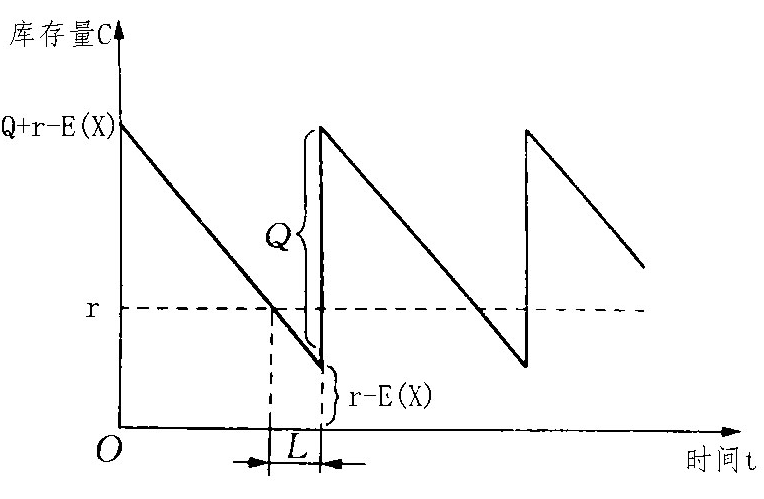


图5-6 库存-时间模型优化图

图中的库存量高点和低点为：

**5.2.2 问题二的模型建立与解决**

5.2.2.1 符号说明

|  |  |
| --- | --- |
| **符号** | **意义** |
|  | 到货率 |
| **C流** | 流通库存总和 |

5.2.2.2 具体计算

根据分析与假设，

对等式两边Q求导，得

对等式两边E（X）求导，得Q=2\*E(X)

联立求得最优情况中r=3\*E(X)

**5.2.3 问题三的模型建立与解决**

5.2.3.1 符号说明

|  |  |
| --- | --- |
| **符号** | **意义** |
|  | i+1级收货商是i级发货商倍数 |
|  | i级收货商继续向下发货的比例 |
|  | i级收货商数量 |
|  | 末端到货率 |

5.2.3.2 具体计算

对于不同货物，由于存在不同等级，故网络模型不同。

末端到货率（与各级的参数的关系）

根据已知的数据可以求得各个和的值

|  |  |  |
| --- | --- | --- |
| **等级i** |  |  |
| **1** | 20 | 50% |
| **2** | 10 | 10% |
| **3** | 4 | 10% |
| **4** | 3 | 3% |

如果数据量足够多，由稳态模型可以得知，末端到货率和供应商生产量成正比，

即：

到货率变为原来的95/90，则生产量也变为原来的95/90，需求量均值E(u)不变，供应商最优库存存在如下关系：

由上式可推导出到货率变化前供应商的最优库存r与变化后的r’的关系

即求得生产商变化后的最优库存。

**六、模型评价及改进**

**6.1 模型评价**

问题一所构造的模型为基于多对多网状供应关系下，存在跨级需求的综合通用模型。其特点是考虑了所有影响到货率的上下级因素，细致建立了各级别库存对指定网点到货率的影响的衡量方式。同时模型可以从供应链的一个公司合理延拓至该级别下的所有公司，即可反映一个级别下所有公司的综合到货率与上级有关联供货方所组成的团体库存的关系，具有较强的适用性。同时该模型可以得到在复杂的关系下到货率的准确表达，具有较高的准确度。但相对而言形式较为复杂，需求参数较多，大量运用时在算法上显得不经济。

问题二所适用的模型是在问题一模型的基础上，结合所给网状关系的统计数据，对原有复杂模型的进行的简化模型。简化后的模型具有更强的可计算性，因此可更方便求解最优值与最优解，同时需求参数较少，更贴合信息源较为缺少的实际应用。简化模型思路更加明确，避免了微小变量对到货率的影响，但该模型不适用于跨级别之间的存库、到货率分析，在复杂网络中存在较大误差。

问题三所采用的模型即为存在变动的模型二，综合程序（见附表）的运算结果，我们发现该模型在较为长宽的模型，即链条较长，时间趋于稳定，下级数多的情况下更为贴合；相对的，在链数短，时间段，下级数少的情况下，初值的小变动易引起大变动，同时相对误差较大。由此认为该模型更适合5级及以上的大规模物流网，而在这一方面，与原题的大数据较为贴合。

**6.2模型改进**

6.2.1 基于发货商的发货率评价

问题中涉及的模型是以下级收货方角度，建立的到货率评估算法。在实际情况中，对于某一供应商，其更关心自己所提供的货品是否能满足下级收货方，即更关注由自己出发而不受其他公司影响的下级到货率。

以此分析，可以认为这是一个单上级，多下级的简单树状供应链，此时的所有下级的综合到货率可以转化为上一级的给货率。由此改进公式一：

定义下方总需求为X（X为一个服从已知分布的变量），下方综合到货率（即目标公司的给货率）为α，以（Q，R）策略储存（如图6.2-1）进行简化：

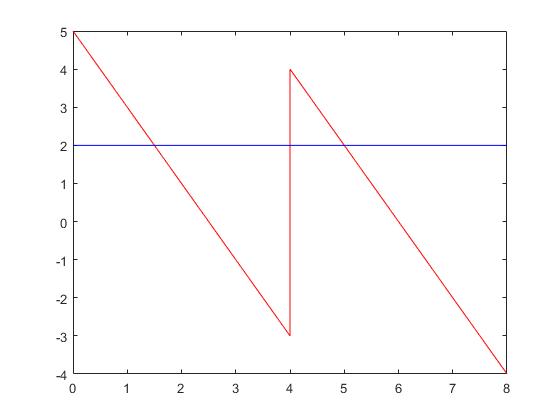


图6-1 （Q，R）策略储存

将需求高于库存时的平均供货过程看作以原需求供给的缺乏库存，由此,在X<r时，即需求小于供货点库存r时，缺货量为零；而当X>r时，即缺货量超出订购点库存，缺货量为。

则可得：

化简后为:

即可以通过此算法衡量一家公司的供给水平，为发货公司提供自我考评的依据，也能方便进货方了解该公司进货能力。

6.2.2 多类商品分析

问题中的模型以一种商品为分析目标，在考虑单体时，可以较好拟合与解决到货率，订货点，进货量等取值问题。而多商品模式下，可能存在更复杂的收货到货网络，这将会直接影响如上级数，下级数，末端需求量等多方面模型的设定参数；同时，对于多类商品综合到货率，实际情况一般对不同商品的重要程度或储存方法认定不同，加以权重分析或设定不同等比的储量。由此，可在复杂网络模型中加入重要度与等比储量的转化：

这两个式子的实际意义分别是：对于权重分析，衡量某多商品的发货公司，可能有不同重要程度，对于不同的分销商，也有不同重点的发展产品，由此不同重要度的商品可以得到衡量；对于等比储存转化分析，则是储存空间上的不同占比，如大体积商品的到货率会更多影响相应的储存空间，则对于多商品到发，合算为同一计量空间可以更好评估与计算进货点与进货量。

**七、参考文献**

[1] 傅家良、李枫、郝勇、孙焰、吴晓，《运筹学方法与模型》，出版地：复旦大学出版社，2012年

[2]同济大学概率统计教研组，《概率统计》，出版地：同济大学出版社，2009年6月

[3]姜启源、谢金星、叶俊，《数学模型》，出版地：高等教育出版社·北京，2011年

**附录**

**一、网络关系进货发货模拟程序代码（基于java语言的模型开发）:**

package modeltest;

import java.util.\*;

public class Test\_1 {

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

Supplier sp;

System.out.println("请输入生产商的产量");

int Q=sc.nextInt();

sp=new Supplier(Q);

Provider [] pr2=new Provider[2];

for(int i=0;i<2;i++){

pr2[i]=new Provider();

}

Provider [] pr3=new Provider[4];

for(int i=0;i<4;i++){

pr3[i]=new Provider();

}

Provider [] pr4=new Provider[8];

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

pr4[i]=new Provider();

}

int CSKC1,XQL1,AQKC;

for(int i=0;i<2;i++){

System.out.println("请您输入第二级第"+(i

+1)+"个供应商初始库存量");

CSKC1=sc.nextInt();

System.out.println("请您输入第二级第"+(i

+1)+"个供应商需求量");

XQL1=sc.nextInt();

System.out.println("请您输入第二级第"+(i

+1)+"个供应商安全库存量");

AQKC=sc.nextInt();

pr2[i].csh(CSKC1, XQL1,AQKC);

}

for(int i=0;i<4;i++){

System.out.println("请您输入第三级"+(i

+1)+"个供应商初始库存量");

CSKC1=sc.nextInt();

System.out.println("请您输入第三级第"+(i

+1)+"个供应商的需求量");

XQL1=sc.nextInt();

System.out.println("请您输入第三级第"+(i

+1)+"个供应商安全库存量");

AQKC=sc.nextInt();

pr3[i].csh(CSKC1, XQL1,AQKC);

}

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

System.out.println("请您输入第四级第"+(i

+1)+"个供应商初始库存量");

CSKC1=sc.nextInt();

System.out.println("请您输入第四级第"+(i

+1)+"个供应商的需求量");

XQL1=sc.nextInt();

System.out.println("请您输入第四级第"+(i

+1)+"个供应商安全库存量");

AQKC=sc.nextInt();

pr4[i].csh(CSKC1, XQL1,AQKC);

}

Store[] st=new Store[8];

System.out.println("输入客户需求量");

XQL1=sc.nextInt();

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

st[i]=new Store(XQL1);

}

System.out.println("初始化成功");

System.out.println("输入您想模拟的次数");

int t=sc.nextInt();

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

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

pr4[i].fh(st[i].xq);//第四级的供应商开

始发货;

XQL1=pr4[i].jhsq();

if(XQL1>=0)

{

if(i/2<=1){

pr3[0].fh(pr4[i].q);

XQL1=pr3[0].jhsq();

if(XQL1>0){

pr2[0].fh(pr3

[0].q);

XQL1=pr2[0].jhsq

();

if(XQL1>0){

sp.fh

(pr2[0].q);

}

}

}

if(i/2>1&&i/2<=2){

pr3[1].fh(pr4[i].q);

XQL1=pr3[1].jhsq();

if(XQL1>0){

pr2[0].fh(pr3

[1].q);

XQL1=pr2[0].jhsq

();

if(XQL1>0){

sp.fh

(pr2[0].q);

}

}

}

if(i/2>2&&i/2<=3){

pr3[2].fh(pr4[i].q);

XQL1=pr3[2].jhsq();

if(XQL1>0){

pr2[1].fh(pr3

[2].q);

XQL1=pr2[1].jhsq

();

if(XQL1>0){

sp.fh

(pr2[1].q);

}

}

}

if(i/2>3&&i/2<=4){

pr3[3].fh(pr4[i].q);

XQL1=pr3[3].jhsq();

if(XQL1>0){

pr2[1].fh(pr3

[3].q);

XQL1=pr2[1].jhsq

();

if(XQL1>0){

sp.fh

(pr2[1].q);

}

}

}

}

else{

System.out.println("错误");

}

}

}

sc.close();

System.out.print("运行结束");

}

}

class Provider{

int cskc;//初始库存

int dhl;//到货量

int q;//订单量

int kc;//库存

int r;//安全库存

int js;

Provider(){

}//空的构造方法

public void csh(int CSKC,int q1,int R){

q=q1;

cskc=CSKC;

kc=CSKC;

r=R;

}//初始化

public int jhsq(){

if(kc<r)

{

return q;

}

else

return 0;

}//发出进货申请

public void sh(int n){

dhl=n;

}//实际到货量为n

public int fh(int m){

int fhl;

if(kc>=m){

kc-=m;

fhl=m;

return fhl;

}

else{

kc=0;

fhl=kc;

return fhl;

}

}

}

class Supplier{

int Q;//生产量

int kc;

Supplier(int q){

Q=q;

kc=q;

}

public int fh(int m){

int fhl;

if(m<=kc){

kc-=m;

fhl=m;

return fhl;

}

else

{

kc=0;

fhl=kc;

return fhl;

}

}

}

class Store{

int xq;//客户的需求量

Store(int l){

xq=l;

}

public int shs(int t){

int Shs=t;

return Shs;

}

}

**二、问题一到货率计算m文件（基于matlab的模型开发）**

function y=Modeltest(s,i,x,m,n,q,t,q1)

%y表示到货率,s表示总共的级数,i表示要货的分销商的级数，X表示发货的上级分销商的库存量，m表示分销商向固定级的几个分销商要货，

%n表示某级有n个分销商向a级分销商要货，q表示某级分销商向上一级提交的订单量,t表示某级分销商向上级要货的概率,q1表示第i级分销商向上一级提交的订单量

%-计算第i级分销商向其上一级分销商提交的订单量-%

Order=0;

s0=0;

for j=1:i-1

Order=Order+(m(j)\*q1);

fprintf('第%d级向第%d级中提交的订单总量为%f\n',i,j,Order)

s0=s0+Order;

end

fprintf('第%d级的分销商下的订单总量为%d\n',i,s0);

%-计算第i级分销商的实际收货量-%

s2=0;

s3=0;

sum(i-1)=0;

Ssum=0;

temp11(100)=0;

d=1;

for j=i-1:-1:1

for k=1:size(x,2)

for c=j+1:s

temp=n(j,c)\*q(c)\*t(c);

s3=s3+temp;

end

fprintf('第%d级的第%d级中的第%d个分销商商接到的订单数为%f\n',i,j,k,s3);

if x(j,k)<s3

temp11(d)=+(x(j,k)\*q1)/s3;

sum(j)=sum(j)+temp11(d);

fprintf('第%d级的第%d级中的第%d个分销商商的库存量不足，采取按订单比重供货%f件\n',i,j,k,temp11(d));

d=d+1;

end

if x(j,k)>=s3

temp11(d)=+q1;

sum(j)=sum(j)+temp11(d);

fprintf('第%d级的第%d级的第%d个分销商的库存量充足，满足订单需求,发货量为%f\n',i,j,k,temp11(d));

d=d+1;

end

end

fprintf('第%d级的分销商商向第%d级分销商发送的货物为%f\n',j,i,sum(j));

Ssum=Ssum+sum(j);

end

%-计算第i级分销商的到货率-%

fprintf('第%d级分销商收到的的货物的数量为%f\n',i,Ssum);

y=Ssum/s0;

**三、问题二到货率随机检验模型（基于c++的模型开发）**

#include "iostream.h"

#include "time.h"

#include "stdlib.h"

#include "math.h"

#define N 8 //分级长度

#define D 12 //变动次数

#define X 5 //下级数量

#define G 2000 //生产能力

void main()

{

//part 1 初始设定

int kc[N+1],k[N+1],r[N+1];

int p[N+1],i,t;

double a[N+1];

srand((unsigned)time(NULL));

cout<<"级别数:"<<N<<endl;

cout<<"显示天数:"<<D<<endl;

cout<<"分级数:"<<X<<endl;

cout<<endl;

for(i=1;i<=N-1;i++)

{

int c=rand()%10000+5000;

kc[i]=c;

cout<<"第"<<i<<"级"<<"初始库存："<<kc[i]<<";"<<endl;

}

cout<<endl;

kc[N]=0;

r[1]=0;

k[1]=G;

for(i=1;i<=N;i++)

{

p[i]=400+400\*i;

k[i+1]=0;

r[i+1]=4600;

}

//part 2 具体计算

for(t=1;t<=D;t++)

{

for(i=1;i<=N-1;i++)

{

if(kc[i+1]>r[i+1])

{

k[i+1]=0;

kc[i]=kc[i]+k[i];

a[i+1]=0;

cout<<"KC："<<kc[i]<<";k:"<<k[i+1]<<";a="<<a[i+1]<<endl;

continue;

}

else

{

if(kc[i]+k[i]-X\*p[i+1]<0)

{

k[i+1]=(kc[i]+k[i])/X;

kc[i]=0;

a[i+1]=(double(k[i+1])/double(p[i+1]))\*100;

cout<<"KC："<<kc[i]<<";k:"<<k[i+1]<<";a="<<a[i+1]<<endl;

}

else

{

kc[i]=kc[i]+k[i]-X\*p[i+1];

a[i+1]=100;

k[i+1]=p[i+1];

cout<<"KC："<<kc[i]<<";k:"<<k[i+1]<<";a="<<a[i+1]<<endl;

}

}

}

cout<<endl;

}

}

**四、一对一简化模型**

%定义变量

y=input('y=');

CW=ones(7,16);

q=zeros(1,7);

r=ones(1,6);

for i=1:1:6

CW(i)=100

q(i)=2.^(6-i)

r(i)=4

end

q(7)=input('q7=');

%y次循环

for i=1:1:y

for cls=6:(-1):1

if CW(cls,i)-q(cls+1)<=0

CW(cls,i)=0;

CW(cls+1,i+1)=CW(cls+1,i)+CW(cls,i);

elseif CW(cls,i)-q(cls+1)>r

CW(cls,i)=CW(cls,i)-q(i+1);

CW(cls+1,i+1)=CW(cls+1,i)+q(i+1);

break;

else

CW(cls,i)=CW(cls,i)-q(i+1);

CW(cls+1,i+1)=CW(cls+1,i)+q(i+1);

end

end

end

**五、一对二简化模型（上级对应且只对应两个下级）**

%第一步，定义变量

y=input('运输次数=');

CW=zeros(7,16);%状态矩阵，CW（1）为供应商库存量

q=zeros(1,7);%每级订单量

r=ones(1,6);%订货库存量

Q=input('供应商每次生产量=');%供应商每次生产量，设为定值

casq=ones(7,16);%进货判定，=1进货

casr=ones(7,16);%足货判定，=1足货

M=zeros(1,7);%每级总公司数量，M（cls）

for cls=1:1:6

M(cls)=2.^(cls-1)

end

q7=ones(1,M(6));

%对变量进行具体数字定义：

q(7)=input('最下端平均要货量=');%定义最下端要货量

for cls=1:1:6

for x=1:1:M(cls)

CW(cls,x)=10000;%定义初始库存量

q(cls)=q(cls)+400;%定义每级订单大小

r(cls)=1000;%定义每级订货库存

end

end

for i=1:1:M(6)

rand=rand(1,1)

q7(i)=q(7)\*gauss(rand,[0.5 5])

end

%for q7=q(7)\*正态分布概率 end %对q7进行正态分布乘法运算

%第二步，附加状态判定量casq,casr

for i=1:1:y

CW(1,1)=CW(1)+Q;

for x=1:1:M(6)

if CW(6,x)-q7(x)>r

casq(6,x)=0

end

end

%对于末端供应商casq是否需要进货进行计算

for cls=6:(-1):2

for x=1:2:M(6)

if casq(cls,x)==1&casq(cls,x+1)==0&CW(cls-1,(x+1)/2)<q(cls)

casr(cls,x)=0;

casr(cls,x+1)=0;

elseif casq(cls,x)==0&casq(cls,x+1)==1&CW(cls-1,(x+1)/2)<q(cls)

casr(cls,x)=0;

casr(cls,x+1)=0;

elseif casq(cls,x)==1&casq(cls,x+1)&CW(cls-1,(x+1)/2)<q(cls)

casr(cls,x)=0;

casr(cls,x+1)=0;

end

%对于cls级的casr，进货是否足量进行计算

for x=1:1:M(cls)

if casq(cls,x)==1&casq(cls,x+1)==0&CW(cls-1,(x+1)/2)-q(cls)>r(cls-1)

casq(cls-1,(x+1)/2)=0

elseif casq(cls,x)==0&casq(cls,x+1)==1&CW(cls-1,(x+1)/2)-q(cls)>r(cls-1)

casq(cls-1,(x+1)/2)=0

elseif casq(cls,x)==1&casq(cls,x+1)==1&CW(cls-1,(x+1)/2)-2\*q(cls)>r(cls-1)

casq(cls-1,(x+1)/2)=0

elseif casq(cls,x)==0&casq(cls,x+1)==0&CW(cls-1,(x+1)/2)>r(cls-1)

casq(cls-1,(x+1)/2)=0

else

casq(cls-1,(x+1)/2)=1

end

end

%对于cls-1级的各个casq是否需要进货进行计算

end

end

%第三步，行动，r=1进货，q=1给足货，计算下一时刻的矩阵状态

for cls=2:1:6

for x=1:2:M(cls)

if casq(cls,x)==0

CW(cls,x)=CW(cls,x);

elseif casr(cls,x)==1

CW(csl,x)=CW(cls,x)+q(cls);

elseif casr(cls,x)==0&casq(cls,x+1)==1

CW(cls,x)=CW(cls,x)+CW(cls-1,(x+1)/2)/2;

elseif casr(cls,x)==0&casq(cls,x+1)==0

CW(cls,x)=CW(cls,x)+CW(cls-1,(x+1)/2);

end

end

%该级奇数项库存状态计算

for x=2:2:M(cls)

if casq(cls,x)==0

CW(cls,x)=CW(cls,x);

elseif casr(cls,x)==1

CW(csl,x)=CW(cls,x)+q(cls);

elseif casr(cls,x)==0&casq(cls,x-1)==1

CW(cls,x)=CW(cls,x)+CW(cls-1,x/2)/2;

elseif casr(cls,x)==0&casq(cls,x-1)==0

CW(cls,x)=CW(cls,x)+CW(cls-1,x/2));

end

end

%该级偶数项库存状态计算

end

end