ARIMA扰动下基于季节模型的自动定价与补货决策

摘要

由于蔬菜各品类、产地不一且其进货交易时间通常在凌晨，则商家须在不确切知道具体单品和进货价格的情况下，做出当日各蔬菜单品的补货决策。因此，为商超提供一套基于时间序列的商品自动定价与补货策略就显得尤为重要。**基于带有ARIMA扰动项的季节回归模型和遗传算法优化模型，为该商超制定收益最大化的蔬菜类商品的自动定价与补货策略。**

针对问题一，对本地菠菜、藕、芜湖青椒(2)、芜湖青椒(份)和虫草花(盒)(1)这五种无销售记录的商品进行剔除，2021年2月11日和2021年2月12日该商超无销售记录是因为该商超在这两天处于暂停营业状态。接着，对各品类及各单品销量刻画波动图，发现其**分布特征均呈现较强的季节性**。然后从**相关性、互补性和替代性**三方面对各品类和各单品进行**相关分析，得出各品类间的关系及同品类中各单品的关系均呈现较强的正相关性的结论。**

针对问题二，引入销量占比作为权重，通过构建成本加成定价模型及**带有ARIMA扰动项的季节回归模型**，利用回归分析、销量残差预测及引入虚拟季节变量，得出销售总量与成本加成定价的关系，**即成本加成定价的升高对销量具有一定抑制作用**。接着，利用折扣率将定价策略分为正常定价和打折定价两部分。通过构建**0-1整数规划模型**，利用优化算法，得出未来一周各类别蔬菜的日补货总量和定价策略以及对应的收益。

针对问题三，考虑到市场对各品类蔬菜的需求和近期单品蔬菜供应，通过销量占比加权处理，构建**带ARIMA扰动的0-1规划模型**，并通过**遗传算法**求解单品蔬菜共需订购29种，同时得到7月1日的单品补货量和定价策略。

针对问题四，为了更好的构建补货与定价决策模型，需为商超提供**相关决策数据**帮助其更好的进行补货与定价决策。基于**可靠的市场需求分析，**考虑到产量、运输与存储成本、客流量和突发状况，分别针对**天气、节假日、重大事件及购物福利**四个方面为商超提供相关数据。通过此类数据，商超能更准确地做出补货与定价决策，从而及时调整补货与定价策略，使得收益尽可能最大化。同时通过娃娃菜销量进行**数值模拟**，验证了所提供的相关数据的正确性。

最后，对带ARIMA扰动的0-1规划模型进行**灵敏度分析，**并对带有ARIMA扰动项的季节回归模型进行**残差分析，**结果均通过检验，**说明了模型的准确性和稳定性较好。**

关键词：ARIMA模型、回归分析、遗传算法、补货与定价决策

一、问题重述

1.1问题背景

蔬菜是生鲜商超的勤销菜品，但蔬菜类商品的保鲜期一般情况下比较短暂，且售卖时间与品相呈现负相关关系，因此当日未售出的蔬菜隔日就无法再次售卖。为了满足市场需求，商超通常会基于各商品历史销售和需求情况每天进行补货。

由于蔬菜品种、产地不一此外，而蔬菜商品的进货交易需在商超开始销售之前进行，因此商家需要在不确切知道具体单品和进货价格的情况下进行进货，可能造成商超收益不稳定与市场供求不当。

1.2问题重述

"成本加成定价"是商超通常用于蔬菜定价的方法，但在实际生活中常受到温度、季节、运输等因素的影响，导致蔬菜损耗和品相变差，通常需要进行折扣销售。附件1为6个蔬菜品类的信息，附件2为销售流水明细，附件3为单品批发价，附件4为各单品近期损耗率。基于附件1-4，解决以下四个问题：

问题一：通过分析附件1和附件2的相关信息，研究各蔬菜品类和单品的销售量分布规律以及它们之间的相互关系。

问题二：基于蔬菜品类，分析销售总量与成本加成定价之间的关系。在确保商超最大化收益的前提下，规划2023年7月1日至7月7日一周内各蔬菜品类的日补货总量和定价策略。

问题三：考虑商超销售空间有限，单品总数需保持在27-33个，每个单品的最小陈列量为2.5千克。基于前一周的可售品种信息，制定7月1日的单品补货计划和定价策略，以最大化商超收益。

问题四：市场需求分析对补货和定价决策至关重要。提出商超需要收集的相关数据，并分析这些数据如何帮助制定蔬菜商品的补货和定价策略。给出建议和理由。

二、问题分析

本文主要依据附件1-4的相关信息，分析蔬菜各单品、各品类的统计特征，为商超构建起利益最大化的补货和定价策略。

针对问题一，本文基于描述统计分析，给出蔬菜各品类、各单品的销售量分布规律，并分析各品类和典型单品的销售旺季和淡季及成因。通过正态检验和相关分析，给出不同品类及同一品类下的不同单品的相关系数热图，并分析不同单品的互补性和替代性。

针对问题二，构建“成本加成定价”模型和带有ARIMA扰动项的季节回归模型，给出销量与成本加成定价之间的关系。通过构建收益模型，基于优化算法给出未来一周各品类的补货总量和定价策略。

针对问题三，基于问题二的模型，构建问题三的收益方程优化模型，依据2023年6月24-30日的销售数据，得出7月1日的订购单品补货量和定价策略。

针对问题四，给出商超需要的相关数据类型，根据文献和数值模拟给出理由。

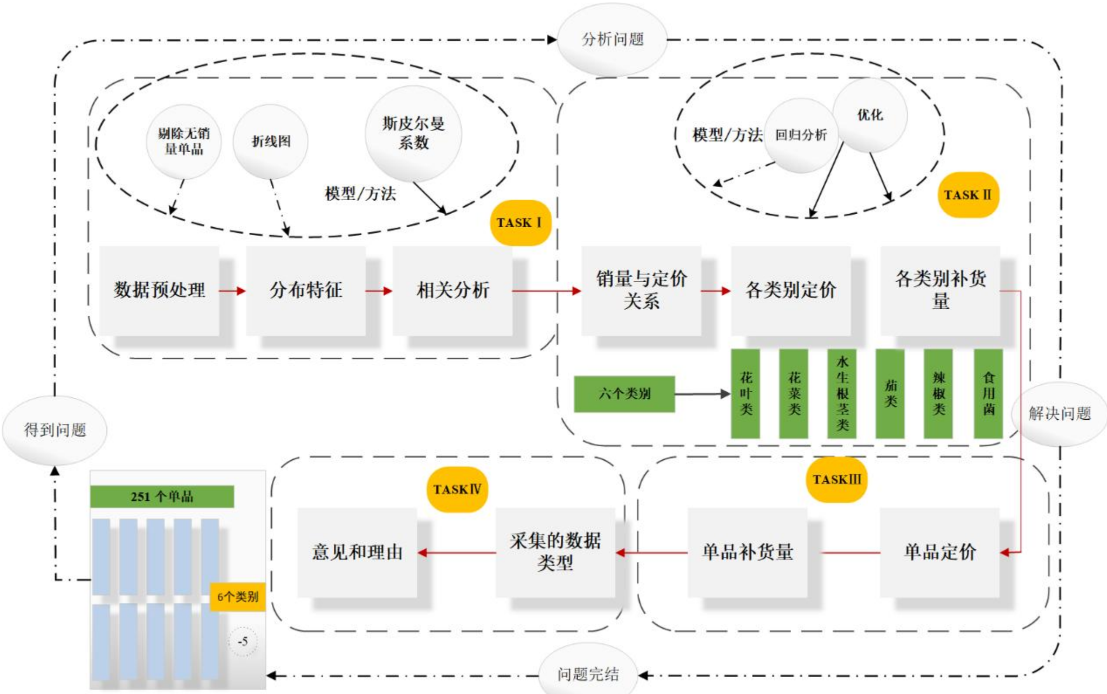


图2.1.1 解题流程图

三、模型假设

(1) 假定当日未售出的菜品，隔日无法再售；
(2) 假定问题三中的损耗率包含自然损耗率和非自然损耗率；
(3) 假定附件4中的损耗率不随时间而变化；
(4) 假定该商超在2020年7月1日-2023年6月30日间不存在倒闭甩卖行为。

四、符号说明

表4.1.1 符号说明表

序号	符号	含义
1	$s2_l$	第 l 类单品日销售总量
2	$P1_i$	单品 i 每日售价
3	$P1_i^{'}$	单品 i 每日打折销售价
4	$C1_i$	单品 i 的每日成本价
5	$\rho1_i$	第 i 个单品的损耗率
6	$P3_l$	第 l 类菜品的销售定价
7	$C3_l$	第 l 类菜品成本
8	d_k	第 k 月的季节虚拟变量
9	γ_l	第 l 类菜品的扰动因子
10	Y_l	第 l 类菜品的进货总量
11	$\hat{P}$	成本加成定价
12	ε_l	第 l 类菜品的残差

五、模型的建立与求解

5.1问题一的模型建立与求解

5.1.1数据预处理

根据附件1、2，运用MATLAB得到五种菜品无销售记录，分别为本地菠菜、藕、芜湖青椒(2)、芜湖青椒(份)、虫草花(盒)(1)。在后续分析过程中无需对这五种菜品进行分析。此外，发现在2021年2月11日和2021年2月12日，该商超未销售出任何蔬菜，因此，本文认为该商超在这两天处于暂停营业状态。

5.1.2模型Ⅰ的建立

Step1.每个单品的每日销售量的求解

根据数据预处理可以得出每个单品每日的销售情况。需要将同一个时间的销售量相加得出当日的销售量。设单品的销售情况为A_i，$i=1,2,\cdots,246$，

$$A_i=\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ \vdots & \vdots \\ a_{n,1} & a_{n,2} \end{pmatrix} \tag{5.1.1}$$

其中第一列表示单品售卖时的日期，第二列表示单品售卖出的重量。

将相同日期的销量相加，即得到各单品在不同日期的销量：

$$s1_{i,j}=\sum_{k=m}^{n} a_{k,2}, i=1,2,\cdots,246 \tag{5.1.2}$$

其中$s1_{i,j}$表示第i个单品在第j天的销售量。

Step2.品类的每日销售量求解

我们可以通过附件1中每个品类所包含的单品，利用step1中各单品的每日销量情况得出每类的每日销售情况，即

$$s2_{lj}=\sum_{q=1}^{n_l} s1_{iq,j}, l=1,2,\cdots,6, i=1,2,...,246, j=1,...,n1 \tag{5.1.3}$$

其中n_l为第l类商品内的单品个数，n_1为销售总天数。

Step3.对蔬菜各品类和各单品分别做折线图，分析其分布特征。

针对上文已经处理好的各类销量数据和各个单品销量数据，本文将给出其折线图和瀑布图，对蔬菜各品类和各单品的分布特征做出统计分析。

Step4.对蔬菜各品类和各单品分别做正态性检验，计算相关系数并分析相关性。

针对各品类和各单品，本文使用SPSS软件对其进行正态性检验，若通过正态性检验则使用皮尔逊相关系数，若无法通过则使用斯皮尔曼相关系数。

5.1.3模型Ⅰ的求解

5.1.3.1分布特征分析

首先，本文利用Origin软件对各品类的销量做出折线图，然后针对各品类的峰点，我们进一步给出局部放大结果，如图5.1.3.1所示。

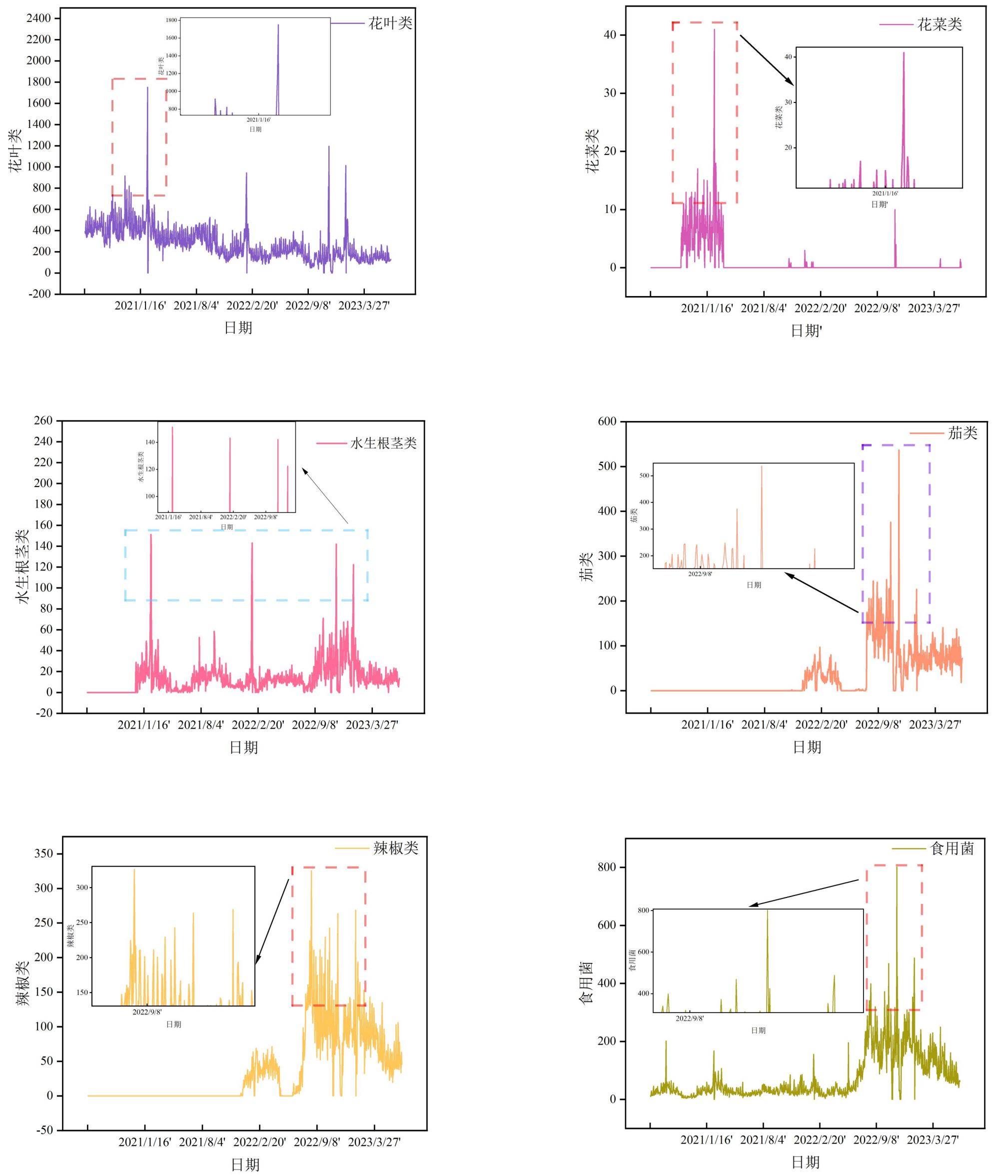


图5.1.3.1 各品类销量局部放大图

从图中可以清晰的看出所有蔬菜品类的销售量基本处于平稳销售状态，且都在部分时间有较高的峰值。其中花叶类销售量总体最多，花菜类销售量总体最少。值得关注的是，所有种类的销售高峰都呈现出了一定的季节性，说明季节对蔬菜类商品的销量有着十分显著的影响。根据图5.1.3.1的峰值分布，本文将这六种蔬菜类别分为两类，即冬季蔬菜和夏季蔬菜。其中，冬季蔬菜为花叶类、花菜类、水生根茎类，其销售旺季主要集中在11-3月，夏季蔬菜为茄类、辣椒类、食用菌，其销售旺季主要集中在4-10月。其中花叶类在南方常在春夏季较为常见，而北方则在秋冬季较为常见，根据花叶类蔬菜的季节性我们也可以判断出本题所研究的商超位于北方。

基于对蔬菜各单品的销量统计，本文使用Origin软件，对销量数据进行处理，绘制了各单品的瀑布图，如图5.1.3.2所示：

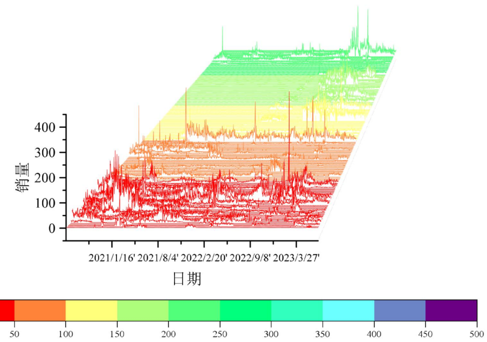


图5.1.3.2 蔬菜各单品销量的瀑布图

注：从垂直于屏幕的方向看，其分布依次为附件1中陈列的各单品名称。

总体趋势分析：从图中，发现蔬菜各单品销量基本上呈现平稳分布状态，销售量最大值接近450kg，销售量在最小值为0。在春夏季和秋冬季，不同的单品呈现出季节性峰值，这主要是由于各蔬菜的生产季节不一所导致，但同时也受到当地居民喜好、习俗、价格等诸多因素的影响。

条块高度分析：每种单品的销售量随时间的分布呈现出间断性峰值行为。在销售旺季的菜品销售量对该菜品总销售量的贡献较大，而在销售淡季的菜品销售量对该菜品总销售量的贡献较小。

贡献比例分析：通过对瀑布图的观察，本文发现对商超总销售量影响较大的是芜湖青椒（1）、西兰花等。

下面给出销售总量前10排行图，如图5.1.3.3所示：

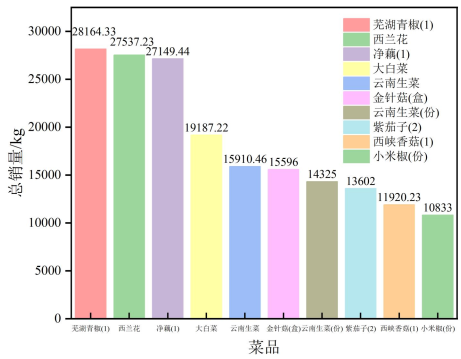


图5.1.3.3 总销量前11的单品排行图

从图中发现研究销量靠前的单品的分布特征十分重要，其销售量的数量巨大不仅对商超收益做出了巨大贡献，而且也一定程度的降低了该菜品的进货成本，对其单独研究具有较大意义，其销售行为的变化可能对该商超的蔬菜利润率产生重要影响。因此本文给出芜湖青椒(1)和西兰花的销量分布图如图5.1.3.4所示：

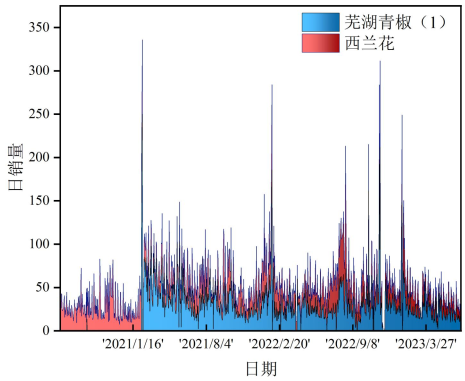


图5.1.3.4芜湖青椒（1）和西兰花销量面积图

从图中可以看出芜湖青椒（1）和西兰花的销售量分布同样具有很强的季节性，且其分布规律与其各自品类的分布规律几乎保持一致，这同样也验证了上文的各类别分布规律的正确性。

5.1.3.2相关分析

(1) 相关关系

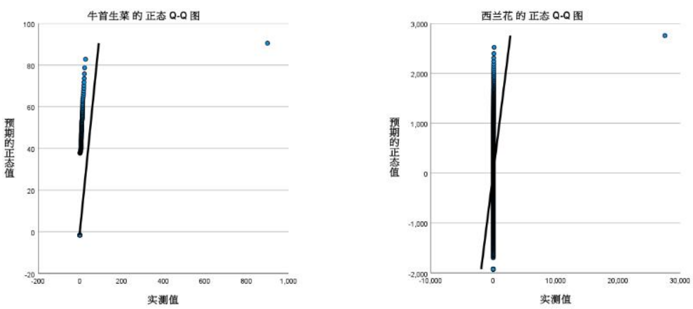
首先，本文利用SPSS软件对蔬菜各类菜品之间进行正态性检验，在0.05的显著性水平下，通过K-S检验法对各品类的销量分布进行正态性检验，检验结果如下表所示：

表5.1.3.1 蔬菜各品类销量分布的蒙特卡洛正态性检验表

	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌类
检验统计	0.072	0.089	0.119	0.088	0.134	0.113
渐近显著性（双尾）	0.000	0.000	0.000	0.000	0.000	0.000

从表中我们得到蔬菜各品类销量的p值均小于0.05，即拒绝原假设 H_0：销量分布服从正态分布，接受备择假设 H_1：销量分布不服从正态分布。即在0.05的显著性水平下，各品类的销售分布不服从正态分布。因此对于各品类之间的相关性分析，我们采用斯皮尔曼相关系数。

然后，本文利用Q-Q图对蔬菜各单品的销量分布进行正态性检验，从每个类中任选一个作为代表，检验结果如图5.1.3.5所示：



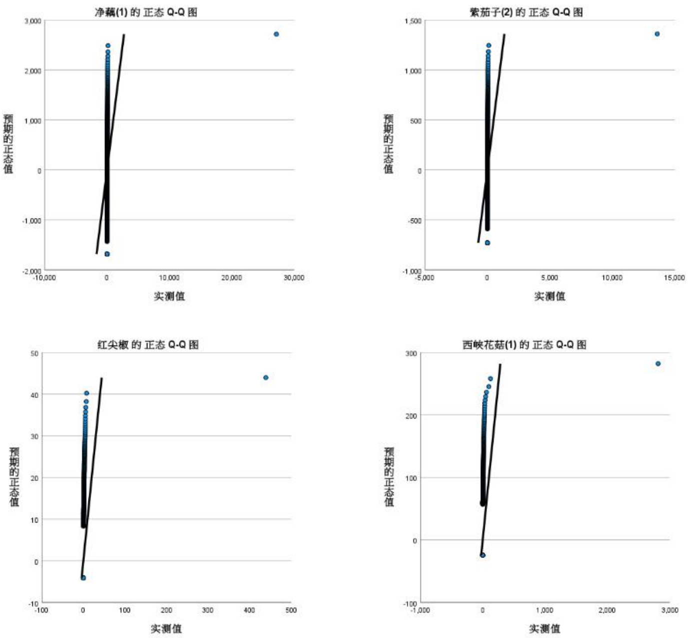


图5.1.3.5 各单品的销量正态分布的q-q图

从图中我们可以直观看出，所有单品的销量与正态分布的偏差都较大，因此认为这些菜品的销量分布均不服从正态性分布。由于所有菜品的Q-Q图与图5.1.3.5的分布状态基本类似，因此有理由认为所有菜品的销量分布均不服从正态性分布。对于下文各单品之间的相关性分析，我们同样采用斯皮尔曼相关系数。

最后，我们利用SPSS软件，通过斯皮尔曼相关系数,对蔬菜各品类以及各单品进行相关性分析。基于各品类的日总销量和各单品总销量，本文利用Python软件，可以得到各品类之间的相关性热力图和同类不同单品之间的相关性热力图，分别如图5.1.3.6-图5.1.3.12所示：

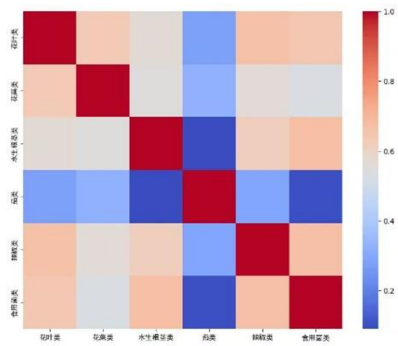


图5.1.3.6 蔬菜各品类销量间的相关性热力图

从上图可以清晰看到各类蔬菜之间整体相关系数较高，相对而言，茄类与其他类菜品的关系较弱，但呈现正向相关关系。说明各类别蔬菜之间的相关性很大，具有强关联度。

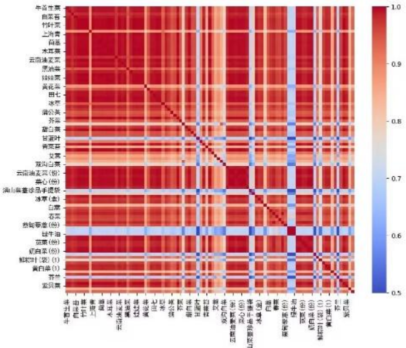


图5.1.3.7 花叶类相关性热力图

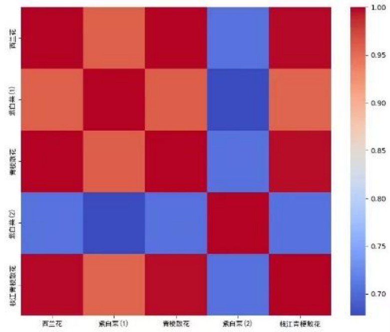


图5.1.3.8 花菜类相关性热力图

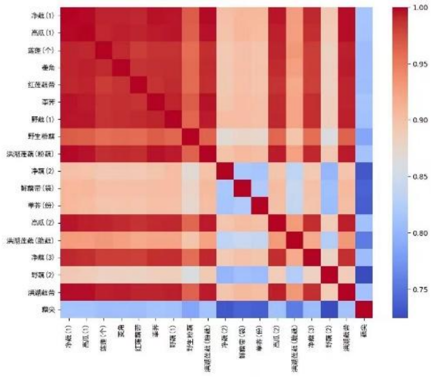


图5.1.3.9 水生根茎类相关性热力图

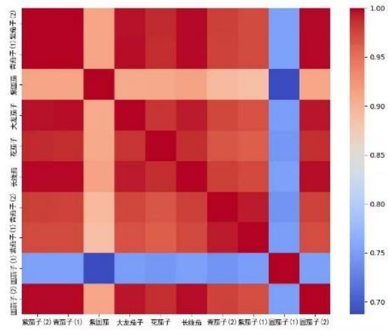


图5.1.3.10 茄类相关性热力图

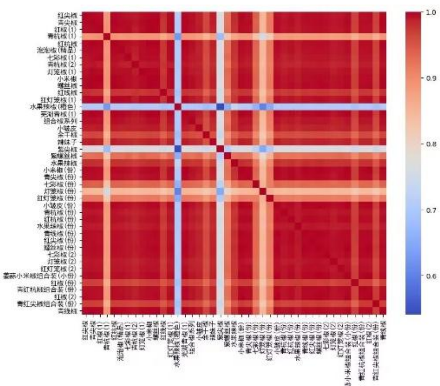


图5.1.3.11 辣椒类相关性热力图

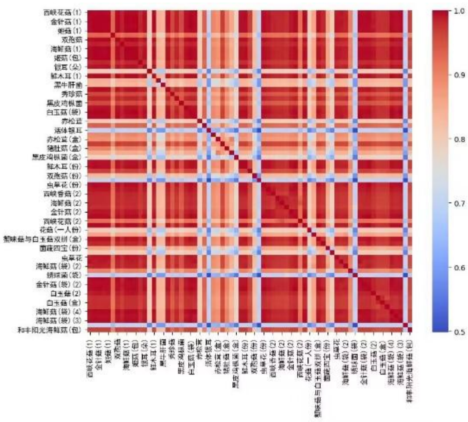


图5.1.3.12 食用菌相关性热力图

从上图可知，花叶类相关度均在0.5以上，花菜类相关度均在0.7以上，水生根茎类相关度均在0.75以上，茄类相关度均在0.7以上，辣椒类相关度均在0.5以上，食用菌相关度均在0.5以上，仅有部分单品与其他单品关系较弱，说明各类别的相关性均很大，具有强关联度。

总体来说本文认为针对各个类别的蔬菜，其各单品的相关程度均很强。其中花菜类、水生根茎类、茄类、辣椒类蔬菜的各单品之间的相关程度很强，而花叶类、食用菌的各单品间的相关程度较强。

（2）互补关系

下面，本文将针对菜品的互补性和替代性，举出具体案例来分析它们之间的关系。通过对附件2的观察，发现有某些商品几乎在同一时间被购入，因此认为它们是互补商品。下面我们例举西峡香菇（1）和泡泡椒(精品)，其某次购入时

间分别是2020年07月02日12时09分11.596秒和2020年07月02日12时09分11.676秒，即为同一消费者同一时间购入，其分布关系图如图5.1.3.14所示：

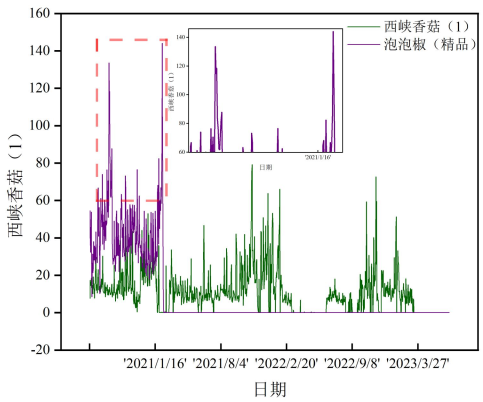


图5.1.3.13 西峡香菇（1）和泡泡椒(精品)的销量局部放大图

从图中，可以发现其销售高峰基本保持一致，即认为同一时间购入的菜品(如西峡香菇（1）和泡泡椒(精品))几乎为互补关系。

（3）替代关系

通过对附件2的观察，发现有某些商品存在诸如盒、袋、份这些分量关系，在充分考虑实际情况下，认为它们是互补商品。下面例举金针菇袋、份、盒，其分布关系图如图5.1.3.14所示：

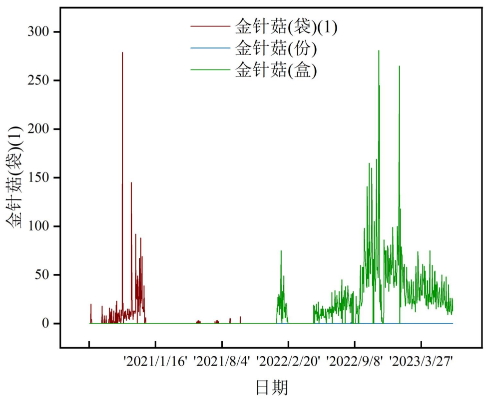


图5.1.3.14 金针菇袋、份、盒的销量分布图

从图中，可以发现其销售高峰完全不一致，且金针菇按份销量相对于盒和袋销量来说非常小，可能是考虑到实惠因素，因此本文认为同一商品的不同分量销售单元为替代关系。

5.1.5模型Ⅰ的结论分析

针对分布规律，本文认为蔬菜各类别菜品销量分布分别呈现出春夏和秋冬季节性，且除了销售旺季外，其余时间销量基本稳定；蔬菜各单品的销量分布也与其对应类别基本保持一致，同样具有季节性。

针对相关关系，在上文中针对相关性、互补性和替代性进行分析，通过结果对比分析，认为各类别之间的相关性与同类别内不同单品的相关性均较强,并发现同一时间购买的几乎均为互补商品，而不同分量的销售单元几乎均为替代商品。

5.2问题二的模型建立与求解

5.2.1数据预处理

本文根据附件2、3进行数据处理，得出单品i每日的售价$P1_i$、打折售价$P1'_i$、成本价$C1_i$，分别为$P1_i=\begin{pmatrix} p1_{i,1} & p1_{i,2} & \cdots & p1_{i,n1}\end{pmatrix}$、$P1'_i=\begin{pmatrix} p1'_{i,1} & p1'_{i,2} & \cdots & p1'_{i,n1}\end{pmatrix}$、$C1_i=\begin{pmatrix} c1_{i,1} & c1_{i,2} & \cdots & c1_{i,n1}\end{pmatrix}$，其中$n_1$为销售总天数，$i=1,2,...,246.$

5.2.2模型Ⅱ的建立

(1)针对问题二的第一小问

Step1.引入自定义权重

本文引入权重定义如下，即利用单品的每日销售量在该品类的每日销售量的占比作为该单品的权重。

$$\omega_i=\frac{s1_{ij}}{s2_l} \tag{5.2.2.1}$$

记$\rho1_i,i=1,2,...,246.$为第i个单品的损耗率，用该权重可以得出每类菜品的每日加权售价$P2_l$、加权成本$C2_l$和加权损耗量$\rho2_l$如下：

$$P2_l=\sum_{i=1}^{n_1}\omega_i P1_i\ ；\ C2_l=\sum_{i=1}^{n_1}\omega_i C1_i\ ；\ \rho2_l=\sum_{i=1}^{n_1}\omega_i\rho1_i \tag{5.2.2.2}$$

Step2.构建成本加成模型

建立各品类的销售定价与成本价之间的关系，根据“成本加成定价”法[1],构建第l类菜品定价与该类菜品成本之间的关系，得到如下模型：

$$P3_l=C3_l(1+\alpha_l),l=1,2,...,6 \tag{5.2.2.3}$$

其中α_l为加成率，通过成本加成模型，可以根据每类菜品的成本得到该类菜品的成本加成定价。

Step3.构建销量与价格、季节因素、扰动因子的关系模型

本文引入季节虚拟变量$d_k,k=1,2,...,11.$和扰动因子$\gamma_l,l=1,2,...6.$其中d_k表示第k月的季节虚拟变量。

$$d_k=\begin{cases}1,\text{月份为第}k\text{月}\\0,\qquad\text{其他}\end{cases},k=1,...,11 \tag{5.2.2.4}$$

通过经济学关系[2]构建销量与价格模型如下：

$$s2_l=\beta_l P3_l+\gamma_l+\sum_{k=1}^{11}c_{kl}d_{kl}+c \tag{5.2.2.5}$$

其中c为常数项,每次出现常数项c都可能代表不同的数值。

Step4:ARIMA模型

作为统计模型中常见的用来进行时间序列预测的模型[3]，其基本步骤如图5.2.2.1所示：

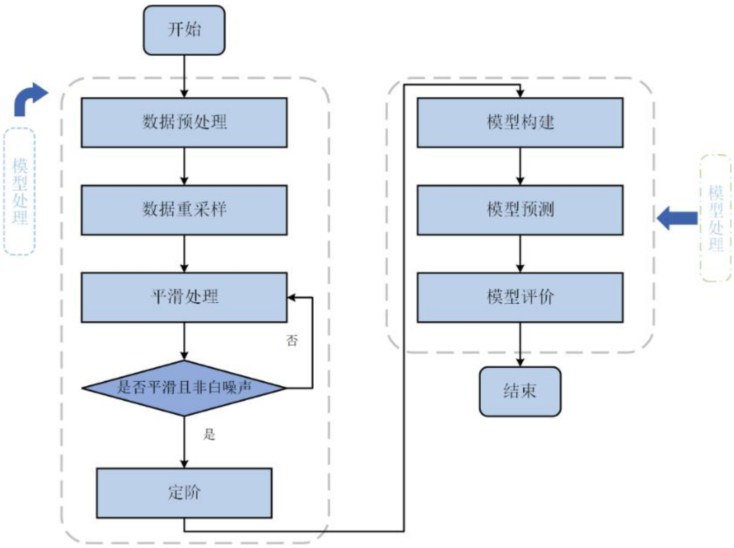


图5.2.2.1 ARIMA模型流程图

（2）针对问题二的第二小问

Step1.构建收益方程模型

本文通过经济学知识分析，得到目标函数收益Q与其相关影响因素的模型：

$$Q=\sum_{l=1}^{6}x_l\left[s4_l(P3_l-C3_l)+s4_l^{'}P3_l^{'}-s4_l^{''}C3_l-\rho2_lY_lC3_l\right] \tag{5.2.2.6}$$

其中$s4_l$表示第l类菜品的日正常价格的销量，$s4_l^{'}$表示第l类菜品的打折销量，$P3_l^{'}$表示第l类菜品的日打折价格，$s4_l^{''}$表示第l类菜品的正常价格的滞销量，$\rho2_l$表示第l类菜品的损耗率，Y_l表示第l类菜品的进货总量。

$x_l=\begin{cases}1，订购第l类菜品\\0，不订购l类菜品\end{cases}, l=1,2,..,6$ Y_l和$P3_l^{'}$可根据如下关系得出：

$$Y_l=\frac{s4_l+s4_l^{''}}{1-\rho2_l} \tag{5.2.2.7}$$

$$P3_l^{'}=\eta_lP3_l \tag{5.2.2.8}$$

Step2.构建优化模型

为了使得商超的收益最大化，本文使用MATLAB构建优化模型，由此得出收益Q的最大值及各类蔬菜的日补货总量、正常定价和打折价格。

5.2.3模型Ⅱ的求解

(1)针对问题二的第一小问

根据本文定义的权重公式，通过eviews软件进行回归分析可以得出价格与成本之间的关系模型，即公式(5.2.2.3)中的α_l的值如表5.2.3.1所示：

表5.2.3.1 各蔬菜品类加成率取值表

	α_1	α_2	α_3	α_4	α_5	α_6
值	0．666	0.518	0.491	0.551	0.513	0.606

将加权后的成本代入公式（5.2.2.3）得到成本加成定价 $\hat{p}$，再将 $\hat{p}$ 与总销量代入公式(5.2.2.5)，通过EVIEWS软件可以得到 β_l, c_{kl}, c 的取值如表5.2.3.2所示：

表5.2.3.2 销量与价格、季节因素、扰动因子的关系模型参数表

l	β	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c
1	-15.988	32.203	0	0	0	0	0	48.957	123.320	75.731	70.221	0	239.018
2	-2.396	13.560	8．108	0	0	0	0	17.021	22.527	12.998	16.360	9.485	52.689
3	10.387	11.694	0	-27.211	-39.756	-47.387	-45.893	-26.748	0	-12.236	0	-12.969	32.915
4	0.799	8.381	9.952	5.417	10.599	18.241	21.269	23.916	20.997	9.234	6.826	0	0
5	-3.722	56.910	56.405	29.675	14.409	0	0	0	30.846	0	15.051	0	98.966
6	-1.746	11.922	0	-34.365	-42.986	-48.617	-52.035	-41.960	-32.755	-33.953	0	0	107.669

注：其中 l=3时，在公式（5.2.2.5）中对应的是 $\sqrt[3]{P3_l}$ 与 $s2_l$ 的关系，即第三类的销量与价格、季节因素、扰动因子的关系模型为非线性模型。

由于温度、时间、误差等随机因子的影响，因此本文将对（5.2.2.5）加入 γ_l 作为扰动因子。通过上文EVIEWS对（5.2.2.5）的回归分析，本文将销量 $s2_l$ 的残差 ε_l 利用SPSSPRO软件，基于ARIMA模型预测扰动因子 γ_l 未来七天（即2023年7月1日-2023年7月7日）的值如表5.2.3.3所示：

表5.2.3.3 各蔬菜品类扰动因子未来七天取值表

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌
2023/7/1	55.077	5.585	-18.297	19.095	0.681	48.937
2023/7/2	51.591	5.437	-18.297	15.408	0.841	47.168
2023/7/3	51.697	5.716	-18.297	15.058	0.923	45.481
2023/7/4	52.537	5.716	-18.297	14.731	0.965	43.873
2023/7/5	52.939	5.716	-18.297	14.427	0.987	42.340
2023/7/6	53.030	5.716	-18.297	14.143	0.998	40.877
2023/7/7	53.090	5.716	-18.297	13.879	1.004	39.483

（2）针对问题二的第二小问

本文通过eviews软件回归分析，给出（5.2.2.8）中 η_l 的值如表5.2.3.4所示：

表5.2.3.4 各蔬菜品类折扣率取值表

	η_1	η_2	η_3	η_4	η_5	η_6

值	0.756	0.726	0.913	0.806	0.753	0.84

然后根据（5.2.2.2）的加权损耗率公式得到各类菜品的损耗率 $\rho 2_i$。

表5.2.3.5 各蔬菜品类损耗率取值表

	$\rho 2_1$	$\rho 2_2$	$\rho 2_3$	$\rho 2_4$	$\rho 2_5$	$\rho 2_6$
值	0.125	0.109	0.082	0.064	0.081	0.070

5.2.4模型 II 的结论分析

(1) 针对问题二第一问

本文得到各品类菜品的总销量与其成本加成定价的关系如图5.2.4.1所示：

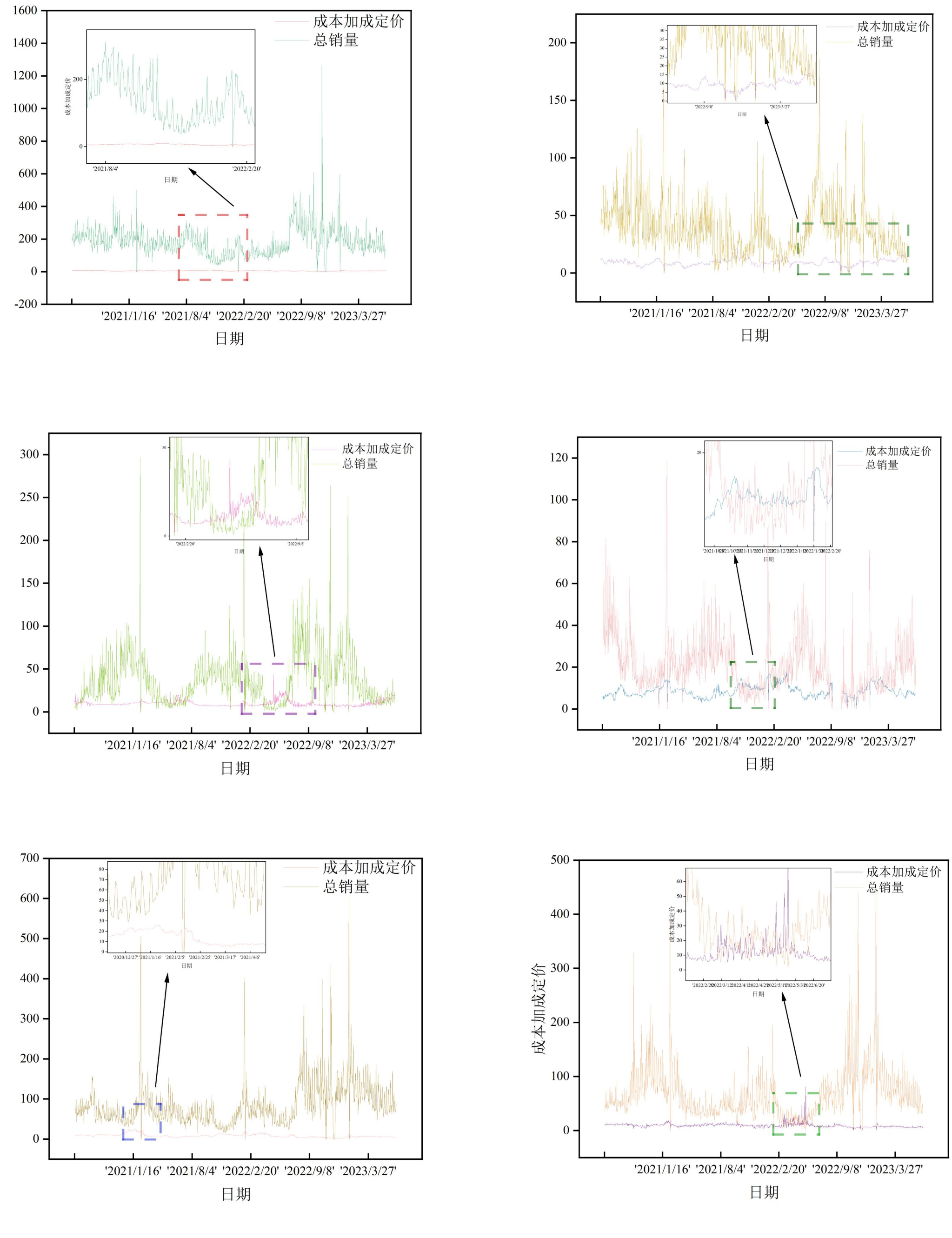


图5.2.4.1 各品类菜品的总销量与成本加成定价的局部放大图

从整体上看，根据上图可以发现各品类的销量与成本加成定价之间存在一定的关联性，从放大部分可以看出，当成本加成定价升高时，各类的销售量都呈现出显著降低趋势，反之，当成本加成定价降低时，各类的销售量都呈现出显著升高趋势，即从整体上来看，即成本加成定价的升高对销量具有一定抑制作用。

(2) 针对问题二第二问

本文利用matlab求解收益方程模型，使得收益最大化，得出未来一周每日每类菜品的补货量、定价策略如表5.2.4.1-5.4.2.4所示：

表5.2.4.1 各类蔬菜产品在未来七天的进货量

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌类
2023/7/1	125.5	32.5	21.13	25.6	75.2	26
2023/7/2	125.5	32.5	21.13	25.6	75.2	26
2023/7/3	125.4	32.5	21.13	25.6	75.1	26
2023/7/4	125.5	32.5	21.13	25.6	75.2	26
2023/7/5	125.5	32.4	21.13	25.6	75.2	26
2023/7/6	125.5	32.5	21.13	25.6	75.2	26
2023/7/7	125.5	32.5	21.13	25.6	75.2	26

从上表中，发现花叶类的进货量相对较大，这说明了花叶类的销售受众广、销售市场大，相反，辣椒类和食用菌的进货量相对较小，这说明辣椒类和食用菌的受众小、销售市场小。

表5.2.4.2 各类蔬菜产品的正常价格

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌类
2023/7/1	8.8	11.7	10.2	12.5	10.95	4.5
2023/7/2	8.7	11.65	10.2	12.5	10.95	4.5
2023/7/3	8.7	11.7	10.2	12.5	11	4.5
2023/7/4	8.75	11.7	10.2	12.5	11	4.5
2023/7/5	8.75	11.7	10.2	12.5	11	4.5
2023/7/6	8.75	11.7	10.2	12.5	11	4.5
2023/7/7	8.75	11.7	10.2	12.5	11	4.5

从上表中，发现茄类的定价相对较高，而食用菌的定价相对较低，这说明茄类的产量大，而食用菌的产量小。

表5.2.4.3 各类蔬菜产品的打折价格

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌类
2023/7/1	6.6528	8.4942	9.3126	10.075	8.24535	3.78
2023/7/2	6.5772	8.4579	9.3126	10.075	8.24535	3.78
2023/7/3	6.5772	8.4942	9.3126	10.075	8.283	3.78
2023/7/4	6.615	8.4942	9.3126	10.075	8.283	3.78
2023/7/5	6.615	8.4942	9.3126	10.075	8.283	3.78
2023/7/6	6.615	8.4942	9.3126	10.075	8.283	3.78
2023/7/7	6.615	8.4942	9.3126	10.075	8.283	3.78

上表的数据可以根据折扣率η_i和表5.2.4.2得出，其一定程度上也反映出了正常价格。通过对打折价格的计算，本文的定价策略分为正常定价和打折定价。

表5.2.4.4 各类蔬菜产品的收益

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌类
2023/7/1	3403.6000	947.5742	2081.7000	1315.4000	1234.5000	1016.6000
2023/7/2	3332.7000	943.4692	2081.7000	1212.1000	1238.5000	999.1599
2023/7/3	3334.8000	951.2180	2081.7000	1202.3000	1240.6000	982.5309
2023/7/4	3351.8000	951.2180	2081.7000	1193.2000	1241.7000	966.6806
2023/7/5	3360.0000	951.2180	2081.7000	1184.6000	1242.3000	951.5696
2023/7/6	3361.9000	951.2180	2081.7000	1176.7000	1242.6000	937.1486
2023/7/7	3363.1000	951.2180	2081.7000	1169.3000	1242.7000	923.4077

根据上表可知，花叶类对收益的贡献度较大，因而可以相对多进货，而花菜类和食用菌类对收益的贡献度较小，因而可以适当减少进货。

通过上文各表之间的关系可知，在某品类的盛产季节，定价较低，人们的需求也比较高，从而商超在此时间内应适当多进货该产品，从而使得收益最大化。此外，当菜品的需求量较大时，也可适当增加产品的定价来增加收益。

5.3问题三的模型建立与求解

5.3.1数据预处理

根据题目要求需从2023年6月24日-30日的可售品种中挑选出7月1日的订购单品，可找出该段时间内有销售记录的单品，共有49个单品。

5.3.2模型Ⅲ的建立

Step1.根据(5.2.2.1)中权重的定义，可求出每个单品的成本价、打折价、打折销售量的值。

Step2.构建收益$Q2$的函数模型

(1) 目标函数的建立

本文通过经济学知识分析，得到目标函数收益$Q2$与单品的选择、单品的定价、单品补货量的关系模型：

$$Q2 = F\left\{\sum_{j=1}^{49} x_j[p_j y_j(1-s_j)+q_j b_j-c_j y_j]\right\} \tag{5.3.2.1}$$

其中$x_j=\begin{cases}1,\ \text{订购第}j\text{个单品}\\0,\ \text{不订购j个单品}\end{cases}, j=1,2,..,49$，$y_j$表示第$j$个单品的进货量，$p_j$表示第$j$个单品的定价，$s_j$表示第$j$个单品的损耗率，$q_j$表示第$j$个单品的打折价，$b_j$表示第$j$个单品的打折销量，$c_j$表示第$j$个单品的成本价，$F$为随机扰动项。

(2) 约束条件的给定

考虑到7月1日的可售单品总数需控制在27~33个，即

$$27 \le \sum_{i=1}^{49} x_j \le 33 \tag{5.3.2.2}$$

并且各单品订购量满足最小陈列量2.5千克，即

$$y_j \ge 2.5 \, j = 1, 2, \ldots 49 \tag{5.3.2.3}$$

(3) 模型的构建

综合公式(5.3.2.1)、(5.3.2.2)、(5.3.2.3)可得出模型：

$$\max \quad Q2 = \sum_{j=1}^{49} x_j [p_j y_j (1 - s_j) + q_j b_j - c_j y_j]$$

$$s.t.\begin{cases} x_j = 0 \text{ 或 } 1 \\ 27 \le \sum_{i=1}^{49} x_j \le 33 \\ y_j \ge 2.5, j = 1, 2, \ldots 49 \end{cases} \tag{5.3.2.4}$$

5.3.3模型III的求解

遗传算法流程图如图5.3.4.1所示：

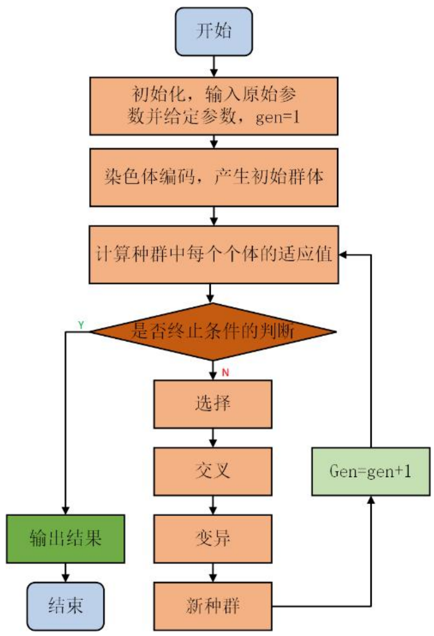


图5.3.4.1 遗传算法流程图

本文基于遗传算法，利用MATLAB得出订购单品的补货量与定价，如表5.3.4.1所示:

表5.3.4.1 订购单品的补货量与定价表

单品名称	苋菜	竹叶菜	菜心	云南油麦菜	菠菜	娃娃菜	外地茼蒿
补货量	208.952	221.915	200.761	200.157	145.958	198.930	191.859
定价	19.251	17.436	19.971	14.749	15.924	20.162	20.221

单品名称	奶白菜	小青菜(1)	云南油麦菜(份)	木耳菜(份)	枝江青梗散花
补货量	159.856	121.548	97.197	200.500	179.571
定价	18.865	18.220	17.834	18.074	18.284

单品名称	高瓜(1)	红莲藕带	高瓜(2)	紫茄子(2)	长线茄	圆茄子(2)
补货量	214.545	199.926	202.202	187.408	178.847	192.794
定价	19.974	19.909	18.227	13.991	20.724	18.326

单品名称	小皱皮(份)	螺丝椒(份)	姜蒜小米椒组合装(小份)	白玉菇(袋)	红椒
补货量	220.392	216.925	228.224	169.638	169.1
定价	18.513	15.390	15.488	18.975	14.182

单品名称	青红杭椒组合装(份)	虫草花(份)	蟹味菇与白玉菇双拼(盒)	金针菇(盒)	海鲜菇(包)
补货量	73.672	163.055	205.072	93.328	212.834
定价	7.146	17.787	17.874	8.570	16.764

从上述表中可以看出人们对于不同蔬菜的需求程度。

5.3.5模型Ⅰ的结论分析

从表5.3.4.1中可清晰得到姜蒜小米辣组合装这类的调味品需求量较高，这种结果与实际情况也较为相符。同时，日常生活中常吃的蔬菜单品对整体收益所做的贡献较大，因此对于销量较好的产品可以适当增加其定价，以此获得较大收益。

5.4问题四的分析与求解

5.4.1采集数据意见

(1) 商超可以采集在天气的影响下各菜品的蔬菜成本情况与供货情况，根据此类数据，及时对某些产品的进货量与进价进行修改，确保收益不受影响。

(2) 商超可以采集各个节假日期间对菜品的销售量的情况及特殊节假日的特殊物品需求的情况，针对性的提高某些菜品的进化量与定价，适当提高收益。

(3) 商超可以关注重大事件的发生与趋势,提高敏锐度。

(4) 商超可以关注购物福利（如：购物狂欢节），推出折扣优惠，促进销量的提高。

5.4.2采集数据理由

(1) 天气的情况比较差会影响蔬菜的产量，从而会导致部分菜品供货不足，通过该信息及时调整补货策略，使整体收益情况尽量不受供货不足的影响。通过调整对不同产地菜品的进货量，降低成本，并适当提高定价，确保收益正常。[4]

(2) 在节假日期间，无论是家庭还是餐馆对蔬菜的需求都会显著增加，可针对相应节假日对某些需求量较大的菜品增加补货，并对部分产品适当的提高定价，使得收益增大。

(3) 商超应多关注重大事件，如新冠疫情的爆发初期消费者大量囤积蔬菜，导致蔬菜的需求量增加。针对这种情况，可以提前补充当季食品，并适当的提高定价，从而可以获取最大收益。[5]

(4) 在当下网络流行的时代，一些商家会举办购物狂欢节，商超可在这期间推出折扣优惠，吸引顾客，从而促进销量增长，使得收益增大。

5.4.3数值模拟

通过查找数据[6],得到某公司娃娃菜在2021年12月12日-2021年12月31日的真实销量数据，数值模拟结果如图5.4.3.1所示：

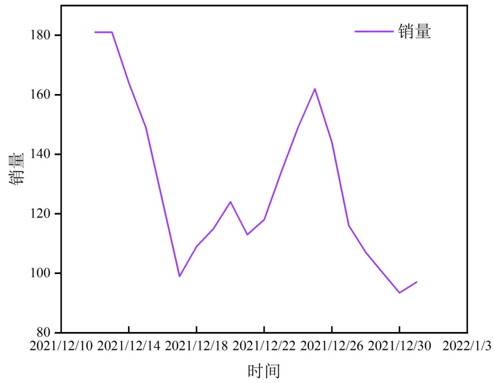


图5.4.3.1 娃娃菜销量与时间关系图

从图中可以看出，娃娃菜的销售峰值有两个，分别在12月12日与12月25日，其中12月12日为购物狂欢节，12月25日为圣诞节，通过数值模拟，本文认为购物福利与节假日是影响蔬菜销量的重要因素。

六、模型的检验与分析

6.1灵敏度检验

对于上文建立的遗传算法模型，本文将通过调整第 j 个单品成本价 c_j 的计算方式为平均成本价，代入该模型得平均成本价对应的补货量与加权成本价对应补货量的关系如图6.1.1所示：

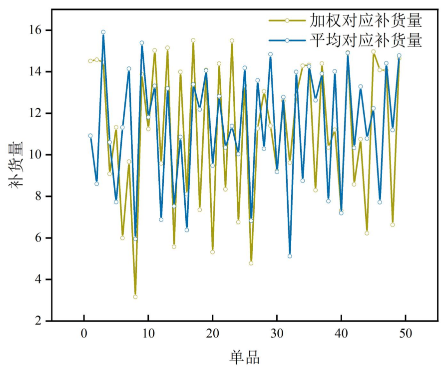


图6.1.1 补货量差异图

从图中发现加权对应补货量与平均对应补货量误差较小，即认为该模型稳定性较好。

6.2残差分析

对于上文建立的ARIMA模型，本文将通过模型残差自相关图对问题二中建立的ARIMA模型做残差分析。

图6.2.1 残差自相关图

上图中横轴代表延迟数目，纵轴代表自相关系数。从图中可以得到模型的自相关系数，95%置信上限和置信下限。

不难发现，相关系数均在置信区间内，即认为自回归模型残差为白噪声序列，符合时间序列的标准，因此验证了问题二中ARIMA模型的正确性。

七、模型的评价与推广

7.1模型的评价

(1) 模型的优点

对于问题二中的带ARIMA扰动项的季节回归模型，它在预测精度方面表现出色，拟合效果良好，准确地捕捉了销量与价格之间的关系。此外，从统计学角度来看，模型的各参数均呈现显著性关系。

对于问题三中的遗传算法优化模型，我们通过微调模型参数，例如迭代次数和初始种群，确保在有限的时间和计算资源内实现了最佳的预测效果。在模型的灵敏度分析中，我们通过将加权向量替换为平均值进行灵敏度分析，结果显示出本遗传算法模型在变化时的稳健性

(2) 模型的缺点

问题二中的带ARIMA扰动项的季节回归模型，其拟合优度还有待提高。可以通过继续加入更主要的影响因素来提高拟合优度。

而对于问题三中的遗传算法优化模型来说，模型初期使用的数据量有限，导致了预测精度上的一定误差。

(3) 模型的创新点

本文问题二中的带ARIMA扰动项的季节回归模型，其创新点在于引入季节虚拟变量和扰动因子来分离模型变量，使得模型的实用性更强，解释性更广、预测性更高。

问题三的遗传算法优化模型的创新在于：我们采用了遗传算法对7月1日的单品补货量和定价策略进行建模。通过将原始的0-1规划问题引入遗传算法的迭代框架中，我们成功地解决了高维变量的整体优化问题。

7.2模型的推广

针对问题二中的带ARIMA扰动项的季节回归模型，其不仅可以给出未来一周的预测值，甚至可以是未来几个月份、季度的预测值。此外，该模型对所有时间序列数据都适用，且未来可用于金融、医药、经济等相关领域。

问题三中的遗传算法模型可以广泛应用于不同的销售策略，促销、季节性定价等。其应用范围不仅限于特定行业，而是可以扩展至零售、制造、金融等多个行业，为各种业务优化问题提供解决方案。

参考文献

[1]韩俊华,干胜道.成本加成定价法评介[J].财会月刊,2012(22):74-75.
[2]许岚.图书销量与定价的制约关系[J].新闻传播,2021(04):68-69.
[3]陈华友,周礼刚,刘金培,陶志富,2023. 统计预测与决策[M]. 北京：科学出版社.
[4]封碧红,顾杨.产地天气因素对蔬菜价格的影响研究[J].农业开发与装备,2023(01):89-90.
[5]谭雅蓉,王一罡,于金莹.新冠肺炎疫情对北京市蔬菜价格影响实证研究[J].北方园艺,2020(24):153-161.
[6]崔云浩. 基于CNN-PSO-LSTM组合模型的生鲜蔬菜销量预测研究[D].安徽农业大学,2023.

附录

<table>
<tr><th colspan="4">附录 1：支撑材料的列表</th></tr>
<tr><td rowspan="14">支撑材料</td><td rowspan="7">问题一</td><td colspan="2">data_1.m</td></tr>
<tr><td colspan="2">data_2.m</td></tr>
<tr><td colspan="2">relitu.py</td></tr>
<tr><td colspan="2">各单品的日销售总量(新).xlsx</td></tr>
<tr><td colspan="2">各品类的销售情况(新).xlsx</td></tr>
<tr><td colspan="2">框架.xlsx</td></tr>
<tr><td colspan="2">无售出记录数据.xlsx</td></tr>
<tr><td rowspan="7">问题二</td><td rowspan="6">分类整理</td><td>花菜类.xlsx</td></tr>
<tr><td>花叶类.xlsx</td></tr>
<tr><td>辣椒类.xlsx</td></tr>
<tr><td>茄类.xlsx</td></tr>
<tr><td>食用菌.xlsx</td></tr>
<tr><td>水生根茎类.xlsx</td></tr>
<tr><td colspan="2">better_max.m</td></tr>
</table>

		data_4.m
		data_5.m
		data_6.m
		打折价格.xlsx
		打折销量.xlsx
		各类商品进价.xlsx
		三维.bmp
		正常价格.xlsx
		正常销量.xlsx
	问题三	49 个数据的损、折、销.xlsx
		2023 年 6 月 24-30 日销量排行.xlsx
		better_max_2.m
		my_simple_fitness.m

附录 2：分离各单品数据代码(MATLAB)

```
clear
clc
close all

%读取文件
[num1, txt1, raw1] = xlsread('附件 1.xlsx');
[num2, txt2, raw2] = xlsread('附件 2.xlsx');
% [num3, txt3, raw3] = xlsread('附件 3.xlsx');
% [num4, txt4, raw4] = xlsread('附件 4.xlsx');

raw1=raw1(2:end,:);
raw2=raw2(2:end,:);
[n1,m1]=size(raw1);
[n2,m2]=size(raw2);

data_None=[];

for i=21:251
    x=raw1(i,1);                %取出第一列方便后面匹配
    data=[];
       for j=1:n2
           if strcmp(x, raw2(j,3))
               temp=[raw1(i,1:2) raw2(j,1) raw2(j,4:7)];
               data=[data;temp]; %往后扩展，对于每一类累计数据
```

```
            end
        end

     if isempty(data)
         temp_2=raw1(i,1:2);
         data_None=[data_None;temp_2];

    else
     tbl = array2table(data,'VariableNames', {'单品编码','单品名称', '销售日期', '销量(千克)', '销售单价(元/千克)', '销售类型', '是否打折销售'});     %转化成表格
     outputFolder = 'D:\数学建模 C\数据处理';
     filename = char(fullfile(outputFolder, strcat(raw1(i,1), '.xlsx')));
     writetable(tbl, filename);
     end

end

tbl = array2table(data_None,'VariableNames', {'单品编码','单品名称'});     %转化成表格
     outputFolder = 'D:\数学建模 C\数据处理';
     filename = char(fullfile(outputFolder, strcat('无售出记录数据', '.xlsx')));
     writetable(tbl, filename);
```

附录 3：每个单品每日销售量代码(MATLAB)

```
clear
clc
close all

%获取最终框架矩阵
[num0, txt0, raw0] = xlsread('D:\数学建模 C\数据处理_2\框架.xlsx');
raw0=raw0(1:1096,:);

%指定要处理的文件夹路径
folderPath = 'D:\数学建模 C\数据处理';

%获取文件夹中的所有文件
fileList = dir(fullfile(folderPath, '*.xlsx'));  % 筛选出扩展名为 .xlsx 的文件
```

```
%for i = 1:numel(fileList)
% 遍历 Excel 文件列表并处理每个文件
for j = 1:1:numel(fileList)
    % 获取当前文件的文件名
    fileName = fullfile(folderPath, fileList(j).name);
    [num, txt, raw] = xlsread(fileName);
    [m,n]=size(raw);
    sum0=num(1,1);
    for i=2:m
        x=raw(i-1,3);%获取相应日期
    % for i=3:m
    %     x=raw(i-1,3);%获取相应日期
    %     if strcmp(raw(i,3),raw(i-1,3))==0 || i==m
    %         if i==m
    %             sum0 = sum0 + num(i-1,1);
    %         end
    %          z=raw0(:,1);
    %         for k=1:length(z)
    %             if strcmp(z{k},x{1})==1
    %                 break
    %             end
    %         end
    %         raw0(k, j+1) = num2cell(sum0); % 将 double 转换为 cell 并赋值给
raw0
    %         sum0=num(i-1,1);
    %     else
    %        sum0 = sum0 + num(i-1,1); % 将 cell 转换为数字并相加
    %     end

    end

end

% 指定要保存的完整文件路径，包括文件夹路径和文件名
outputFilePath = 'D:\数学建模 C\详细单类数据.xlsx';

% 使用 xlswrite 函数将矩阵写入 Excel 文件
xlswrite(outputFilePath, raw0, 'Sheet1');
```

附录 4：热力图(Python)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties

# 读取 Excel 文件数据
data = pd.read_excel(&apos;f:\\食用菌.xlsx&apos;, header=0)

# 计算相关系数矩阵
corr_matrix = data.corr()

# 设置中文字体
font_path = &apos;C:\\Windows\\Fonts\\SimHei.ttf&apos;
font_prop = FontProperties(fname=font_path)

# 绘制热力图
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=False, cmap=&apos;coolwarm&apos;, cbar=True, cb
ar_kws={&apos;label&apos;: &apos;&apos;})

# 设置中文列名
plt.xticks(fontproperties=font_prop)
plt.yticks(fontproperties=font_prop)

plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties

# 读取 Excel 文件数据
data = pd.read_excel(&apos;f:\\食用菌.xlsx&apos;, header=0)

# 计算相关系数矩阵
corr_matrix = data.corr()

# 设置中文字体
font_path = &apos;C:\\Windows\\Fonts\\SimHei.ttf&apos;
font_prop = FontProperties(fname=font_path)

# 绘制热力图
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=False, cmap=&apos;coolwarm&apos;, cbar=True, cb
```

```
ar_kws={&apos;label&apos;: &apos;&apos;})

# 设置中文列名
plt.xticks(fontproperties=font_prop)
plt.yticks(fontproperties=font_prop)

plt.show()
```

附录 5：所有单品的日打折销售价格与非打折销售价格(MATLAB)

```
clear
clc
close all

%获取最终框架矩阵
[num0, txt0, raw0] = xlsread('D:\数学建模 C\数据处理_2\框架.xlsx');
raw0=raw0(1:1096,:);

%打折的为 A，没有打折的为 B
A=raw0;
B=raw0;


%指定要处理的文件夹路径
```

附录 6：各单品的每日进价(MATLAB)

```
clear
clc
close all

% 读取 Excel 文件
[data, text, raw] = xlsread('附件 3.xlsx');

%获取最终框架矩阵
[num0, txt0, raw0] = xlsread('D:\数学建模 C\数据处理_2\框架.xlsx');
raw0=raw0(1:1096,:);
answer=raw0;
%指定要处理的文件夹路径
folderPath = 'D:\数学建模 C\数据处理\';
```

```
%获取文件夹中的所有文件
fileList = dir(fullfile(folderPath, '*.xlsx'));  % 筛选出扩展名为 .xlsx 的文件

[m,n]=size(raw)

for i=2:m
    disp(i);
    %本质是定位行列
    z=raw0(:,1);
    x=raw(i,1);
    for k=1:length(z)%找到的 k 就是行的位置(定位时间)
        if strcmp(z{k},x{1})==1
            break
        end
    end

    %之后定位列的位置
    z=raw0(1,:);
    x=raw(i,2);
    for m=1:length(z)%找到的 k 就是行的位置（定位编号）
        if strcmp(z{m},x{1})==1
            break
        end
    end
    answer{k,m}=raw{i,3};
end


filename = 'D:\数学建模 C\各个商品进价.xlsx';
xlswrite(filename, answer);
```

附录 6：各单品日正常销量和打折销量(MATLAB)

```
clear
clc
close all

%获取最终框架矩阵
[num0, txt0, raw0] = xlsread('D:\数学建模 C\数据处理_2\框架.xlsx');
raw0=raw0(1:1096,:);
```

```
%打折的销量为 A，没有打折的销量为 B
A=raw0;
B=raw0;


%指定要处理的文件夹路径
folderPath = 'D:\数学建模 C\数据处理\';

%获取文件夹中的所有文件
fileList = dir(fullfile(folderPath, '*.xlsx'));  % 筛选出扩展名为 .xlsx 的文件

% 遍历 Excel 文件列表并处理每个文件
for j = 2:247
    output_str = ['第', num2str(j), '次'];
    disp(output_str);
    % 获取当前文件的文件名
    [num, txt, raw] = xlsread([folderPath,raw0{1,j},'.xlsx']);
    [m,n]=size(raw);
    sum_A=0;%用来计算打折的一天当中的累计销量
    sum_B=0;%用来计算正常的一天当中的累计销量
    if m==2 %如果此只有一条销售记录（特殊情况）
        x=raw(2,1);%编码
        y=raw(2,3);%销售日期
        z=raw0(:,1);
        for k=1:length(z)%找到的 k 就是位置
            if strcmp(z{k},y{1})==1
                break
            end
        end
        z=raw0(1,:);
        for s=1:length(z)%找到的 s 就是位置
            if strcmp(z{s},x{1})==1
                break
            end
        end
        if strcmp(raw(2,7),'是')
            A{k,s}=raw{2,4};
        end
        if strcmp(raw(2,7),'否')
            B{k,s}=raw{2,4};
        end
        continue;
    end
```

```
for i=2:m-1
    if strcmp(raw(i,7),'是')
        sum_A=sum_A+raw{i,4};
    end
    if strcmp(raw(i,7),'否')
        sum_B=sum_B+raw{i,4};
    end
    if strcmp(raw(i,3),raw(i+1,3))==0%不相等的时候
        x=raw(i,1);%编码
        y=raw(i,3);%销售日期

        z=raw0(:,1);
        for k=1:length(z)%找到的 k 就是位置
            if strcmp(z{k},y{1})==1
                break
            end
        end

        z=raw0(1,:);
        for s=1:length(z)%找到的 s 就是位置
            if strcmp(z{s},x{1})==1
                break
            end
        end
        A{k,s}=sum_A;
        B{k,s}=sum_B;
        sum_A=0;
        sum_B=0;
    end
    if i==m-1 && strcmp(raw(i,3),raw(i+1,3))==1
        x=raw(i,1);%编码
        y=raw(i,3);%销售日期

        z=raw0(:,1);
        for k=1:length(z)%找到的 k 就是位置
            if strcmp(z{k},y{1})==1
                break
            end
        end

        z=raw0(1,:);
        for s=1:length(z)%找到的 s 就是位置
            if strcmp(z{s},x{1})==1
```

```
                    break
                end
            end
            A{k,s}=sum_A;
            B{k,s}=sum_B;
        end
    end

    %最后一个没有处理
    i=m;
    x=raw(i,1);%编码
    y=raw(i,3);%销售日期
    z=raw0(:,1);
    for k=1:length(z)%找到的 k 就是位置
        if strcmp(z{k},y{1})==1
            break
        end
    end
    z=raw0(1,:);
    for s=1:length(z)%找到的 s 就是位置
        if strcmp(z{s},x{1})==1
            break
        end
    end
    if strcmp(raw(i,3),raw(i-1,3))==0
        if strcmp(raw(i,7),'是')
            A{k,s}=raw{i,4};
        end
        if strcmp(raw(i,7),'否')
            B{k,s}=raw{i,4};
        end
    else
        if strcmp(raw(i,7),'是')
            A{k,s}=A{k,s}+raw{i,4};
        end
        if strcmp(raw(i,7),'否')
            B{k,s}=B{k,s}+raw{i,4};
        end
    end



end
```

```
filename = 'D:\数学建模 C\打折销量.xlsx';
xlswrite(filename, A);
filename = 'D:\数学建模 C\正常销量.xlsx';
xlswrite(filename, B);
```

附录 7:问题二收益最大化的求解(MATLAB)

```
clear
clc
close all

[x, y] = meshgrid(0:0.2:50, 0:0.2:40);

% 核心公式,核心符号
Alpha=0.666;      %类变化
Beta=-15.988;      %类变化
Gamma=55.077;     %表格
Alpha_x=0.756;    %类变化
Q=48.957+239.018;  %C7+C（类变化）
loss=0.125;       %损耗率(类变化)

z = (x.*Beta+Gamma+Q).*(x-x./(1+Alpha)) +
(Beta./Alpha_x.*x+Gamma+Q).*(x./Alpha_x) -
((1-loss).*y-(x.*Beta+Gamma+Q)).*x./(1+Alpha) - (loss.*y).*x./(1+Alpha);

mesh(x, y, z);

% 搜索最大值
[maxZRow, colIndex] = max(z, [], 2);
[maxZ, Index] = max(maxZRow);

disp(maxZ);
```

附录 8：遗传算法主文件(MATLAB)

```
clear
clc
close all

% 数据加载
```

```
data = xlsread('D:\数学建模 C\49 个数据的损、折、销.xlsx');
A = data(1, :);  % 损耗率
B = data(2, :);  % 成本价
C = data(3, :);  % 打折价
D = data(4, :);  % 打折销售量

% 设定下界
LB = [zeros(1, 49), 2.5 * ones(1, 49), ones(1, 49) * 3];

% 设定上界
UB = [ones(1, 49), 16 * ones(1, 49), 12 * ones(1, 49)];

ObjectiveFunction = @my_simple_fitness;
nvars = 147;  % 变量个数

% 添加约束
A = [-ones(1, 49), zeros(1, 49), zeros(1, 49);
      ones(1, 49), zeros(1, 49), zeros(1, 49)];
b = [-27; 33];

% 指定整数约束
IntCon = 1:49;

% 自定义变异函数和参数
mutationFcn = @(parent, options, nvars, FitnessFcn, state, thisScore,
thisPopulation) ...
    mutationgaussian(parent, options, nvars, FitnessFcn, state, thisScore,
thisPopulation, 0.02);
%父代个体(parent)、算法选项(options)、变量个数(nvars)、适应度函数(FitnessFcn)、
算法状态(state)、当前个体的适应度分数(thisScore)和种群(thisPopulation)


% 设置优化选项
% PopulationSize 种群个体数目大小
% Generations 迭代 50 次
% CrossoverFraction 交叉概率
options = optimoptions('ga', 'PopulationSize', 100, 'Generations',
50,'CrossoverFraction', 0.8, 'MutationFcn', mutationFcn);

% 调用遗传算法
[x, fval] = ga(@(t) my_simple_fitness(t, A, B, C, D), nvars, A, b, [], [],
LB, UB, [], IntCon, options);

disp('最优解: ');
```

```
disp(x);
disp('最优适应度值：');
disp(fval);
```

附录 9：附录 8 代码中遗传算法优化函数(MATLAB)

```
function y = my_simple_fitness(t, A, B, C, D)
%极值（求最小）
%所以求极值最大的时候取反

x=t(:,1:49);
y=t(:,50:98);
p=t(:,99:147);

%修改成平均值：B_avg 之后进行灵敏度验证

sum_0=0;
for i=1:49
    sum_0=sum_0+x(i).*(p(i).*y(i).*(1-A(i))+C(i).*D(i)-5.7068.*y(i));
end
y=-sum_0;
```