

基于蒙特卡洛的农作物种植策略选择

摘 要

首先，基于附件中的数据，进行整合、分析与清洗。针对不同的变量，如作物类型、地块类型、种植季次、特殊作物（豆类作物、萝卜、食用菌等）进行**数值编码与聚合**。对数值数据使用**箱型图**进行异常值检验，结合农作物实际情况，发现无异常数据。

针对问题一，对 2023 年农作物的销售量、种植成本、亩产量、销售价格、种植亩数分别按照种植季次建立透视表，根据**亩产量、种植亩数和价格均值**三者之积计算出预期销售量。由于不同年份之间的种植策略相互依赖，故以未来 7 年总收益最大为**目标函数**，以不同年份、不同季度、不同地块、不同作物的种植面积**为决策变量**，以旱地、水浇地、人工地三种类型地块上的农作物品类限制、面积限制、农作物重岔限制、豆类作物三年轮作、种植地不分散、种植面积不过小为**约束条件**，建立两种不同情况的多阶段滚动单目标规划模型。其中，种植不分散、种植面积不易过小分别量化为选择相邻地块种植、种植面积占耕地面积的一半或者全部。由于数据维数过高、约束条件过于复杂，使用传统规划算法与启发式算法难以求解，故本文使用**蒙特卡罗算法**生成大量符合约束条件的候选解，分别计算两种情况的收益情况寻找策略，进行**逆维度**编程求解。过期作废与降价出售两种情况的 7 年**总收益**分别是 41662119.8 元与 41441844 元，具体的种植策略见附件 result1_1 和 result1_2。对结果检验分析后发现，两种情况的收益情况均符合正态分布，且情况二总收益相对前者波动性更小。

针对问题二，考虑到问题一中的情况二的收益更为稳健，以该情况下的 7 年的总收益最大为目标函数，在问题一的基础上建立**多阶段滚动随机规划模型**。针对不同作物预期销售量、亩产量、种植成本、销售价格的波动性与潜在风险等不确定因素，引入均匀分布、正态分布、三角分布等**概率密度分布函数**进行量化，融入滚动模型，使用**蒙特卡洛算法**进行求解。七年的总收益为 40904868.8 元，相较于问题一收益相对下降，收益逐年呈现波动式起伏，具体的种植策略见附件 result2。选取经济效益为特征进行灵敏度分析，对销售单价进行扰动后，计算出变异程度均值为 0.03169，模型相对稳健。

针对问题三，在互补性与可替代性方面，使用 **Kmeans 聚类与肘部图**将农作物分为两类后，对使用主**成分分析降维**后的农作物经济特征数据，进行**恒定替代弹性函数**计算，发现豆类植物与其他作物互补性较高，黄瓜和空心菜、青椒与包菜的替代性较高。使用**斯皮尔曼相关系数**对销售量、价格区间、种植成本进行相关性分析，发现价格区间与种植成本存在正相关性，区间大小与销售量存在负相关性，且均通过 **P 值**检验。基于此，建立**线性回归模型**拟合出对用关系式，在问题二的基础上建立**多因素不确定性多阶段随机规划模型**，其中根据豆类植物的互补性，将豆类作物三年轮作约束修改为两年，进一步使用蒙特卡洛算法求解后，最优种植策略的 7 年的总收益为 40786522.3 元。与问题二的结果进行比较后发现，问题三的收益普遍低于问题二，且收益差距之间的差距随着时间的推移不断增大，绘制箱型图比较后发现，问题三的总收益的离群值较多，整体波动性更大。

关键词：多阶段滚动随机规划模型 蒙特卡洛 概率密度分布 农作物种植策略

一、问题重述

1.1 问题背景

为实现乡村经济的可持续性发展，应结合实际情况，充分利用耕地资源，选择适宜农作物并制定合理策略，以达到提升生产效益和便于管理的效果，另外也可减少种植风险。本文以华北某一乡村为例，基于所给附件信息，制定了不同条件下的最优种植方案。

1.2 问题数据

该题的数据包括两个表单，具体介绍如下。

附件 1:

- (1) 乡村现有耕地的基本信息，包含名称、类型、面积以及必要说明。
- (2) 乡村种植农作物的基本信息，包含名称、类型、适种植耕地及必要说明。

附件 2:

(1) 2023 年农作物种植情况，包含编号、名称、类型、种植地块、种植面积以及种植季次。

(2) 2023 年统计的相关数据，除上述数据外还包括具体的亩产量、种植成本以及销售单价。

1.3 问题描述

问题 1: 在假定各农作物未来的预期销售量、种植成本、亩产量和销售价格相对于 2023 年保持稳定，且每季种植的农作物在当季销售的条件下，分别讨论两种非正常销售情况，该乡村 2024~2030 年农作物的最优种植方案。

具体情况如下：

- (1) 过量作废处理
- (2) 过量半价处理

问题 2: 已知小麦和玉米未来年增长率介于 5%~10%，其余农作物未来每年预期销售量相对于 2023 年有 5% 的上下浮动。农作物亩产量受天气影响有 $\pm 10\%$ 的变化，而种植成本受市场影响平均每年增长 5% 左右。蔬菜类作物销售价格每年增长 5% 左右，食用菌销售价格每年下降 1%~5%，其中羊肚菌每年下降幅度固定为 5%。

综合考虑各农作物的预期销售量、亩产量、种植成本和销售价格的不确定性及潜在种植风险，给出该乡村 2024~2030 年农作物的最优种植方案。

问题 3: 考虑各农作物之间存在可替代性和互补性，并探究预期销售量与销售价格、种植成本之间的相关性。然后在问题 2 的基础上综合考虑这些因素，给出该乡村 2024~2030 年农作物的最优种植策略，模拟数据进行求解，最后与问题 2 的结果作比较。

二、问题分析

由于数据的复杂性，本文首先对数据进行预处理，依次通过缺失值处理、重编码、数据宏观分析以及异常值处理来对数据进行处理，以此为解决后面的问题奠定基础。

2.1 问题一的分析

首先，本文根据所给附件，分别建立种植面积、经济数据以及销售量的数据透视表。然后以最大经济效益为目标函数，综合考虑不同地块类型种植选择限制，地块面积和重茬限制，豆类轮作限制，种植面积限制以及种植地不太分散限制这 5 大约束条件，通过蒙特卡洛模拟，分两种过量处理情况讨论其 2024 年到 2030 年的最优种植策略。

2.2 问题二的分析

在问题一的基础上，引入预期销售量、亩产量、种植面积和销售价格的预期不确定性考虑，根据其波动特征，分别得到其隶属的随机分布，接着，通过构建基于 IGDT 的改进蒙特卡洛模拟模型对最优策略进行 1000 次模拟，分析得到整体最优策略。

2.3 问题三的分析

针对不同作物间的可替代性与互补性，首先对其进行 k-means 聚类得到初步的相关性，然后基于恒定替代弹性函数，划分可替代性和互补性。接着，本文在斯皮尔曼相关性分析的基础上，通过线性回归构建出销售单价、销售量以及种植成本间的关系。基于问题 2 的模型，综合考虑相关问题，给出最优种植策略和最大经济效益，并与问题 2 结果比较分析。

2.4 整体流程图

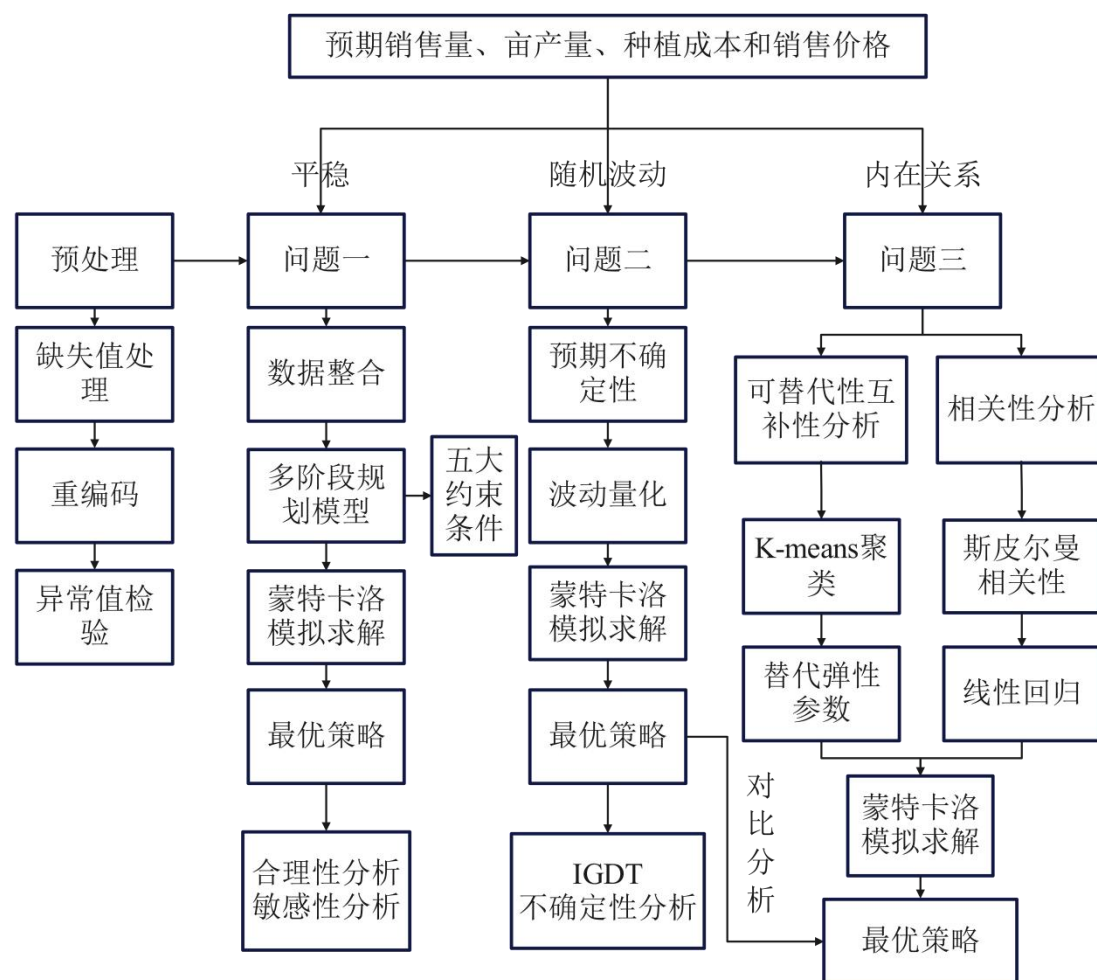


图 1 整体流程图

三、模型假设与符号说明

3.1 问题假设

为满足后续建模需求，本文做出如下假设：

假设 1: 附件 2 所给地块 ABCD 间距离不可估计，但在各类型地块内部按照数字顺序依次相邻排列；

假设 2: 不考虑各地块内部由于作物种植排列导致的影响，仅考虑各作物种植占比最后造成的影响；

假设 3: 作物只要在一个地块中占比达到 0.5 之上，就不再种植其他作物，且不考虑土地浪费。

3.2 符号说明

表 1 符号列表

符号	解释
D_{ijk}^t	2023 年 i 编号地块在第 k 季度种植 j 编号作物的销售量
x_{ijk}^t	2023 年 i 编号地块在第 k 季度种植 j 编号作物的种植面积
P_{ijk}^t	2023 年 i 编号地块在第 k 季度种植 j 编号作物的亩产量
x_{ijk}^t	第 t 年第 i 个地块(或大棚)的第 k 季度种植第 j 种作物的面积
A_i	第 i 个地块的总面积
C_{ijk}^t	第 t 年第 i 个地块在第 k 季度种植第 j 种作物的种植成本
P_{ijk}^t	第 t 年第 i 个地块在第 k 季度种植第 j 种作物的亩产量
S_{ijk}^t	第 t 年第 j 种作物在第 k 季度种植第 j 种作物的销售价格
$g(\cdot)$	超过需求部分的影响函数
y_{ij2}^t	在第 t 年的第 i 个地块的第 2 季度是否种植了第 j 种作物
A_{last}	种植完地块 i 后，剩余的所需面积。
$f(x)$	概率密度函数
$F(x)$	累计分布函数
L_d	预期销售量的不确定性波动范围
μ	销售量平均值。
a	正态分布 95%置信区间的分界点
L_c	种植成本的不确定性波动范围
L_s	销售价格的不确定性波动范围
L_p	亩产量的不确定性波动范围
X	7 年对应的种植总策略
κ	容忍的风险阈值
C_j	聚类的每个簇
$ C_j $	簇 C_j 中样本的数量
J	簇内平方和
α	要素的相对重要性
δ	替代弹性参数

四、数据预处理与宏观可视化

4.1 重编码

4.1.1 地块名称

地块名称数据为组合编码数据，相较于整数编码，组合编码在数学运算和优化问题中计算效率低，不利于计算，因此将地块名称转换成整数编码，记为 1-54。

4.1.2 种植季次

种植季次数据有单季，双季两种。因单季种植的作物每年只种一次，本文假定单季为第一季，以减少数据维度。故而可将第一季编码为 1，第二季编码为 2。

4.2 数据宏观分析

根据题目以及表格中的信息，本文分别对所给附件中的地块类型、作物类型做进一步划分，并对临近种植的作物做了可视化分析，具体步骤如下：

4.2.1 耕地细化分类

为方便后期模型的建立，本文基于各地块的特征将其进一步分为旱地、水浇地和人工地（普通大棚、智慧大棚）三类，分别整理其编号及相关种植条件如下：

表 2 细化分类表

耕地类型	地块类型	编号区间	适用作物	种植季次
旱地	平旱地、梯田、山坡地	1-26	粮食	一季
水浇地	水浇地	27-34	粮食(水稻)/蔬菜	一季/两季
人工地	普通大棚	35-50	蔬菜、食用菌	两季
	智慧大棚	51-54	蔬菜	两季

显然对于不同类型的耕地其所具有的种植特征也不尽相同，如旱地（平旱地、梯田和山坡地）只可种植一季的粮食，水浇地可以种植一季水稻或两季蔬菜，而人工地则适合种植两季的蔬菜。

4.2.2 作物细化分类

根据所给附件 1 中“乡村种植的农作物”中的信息，本文可以得到共有 41 种作物，分别归属于 4 种作物类型，即粮食、蔬菜、豆类以及食用菌。为了进一步得到不同作物的特征与适应环境，对作物分类进行分析，可以得到如下表格：

表 3 作物细化分类表

作物类型	细化类型		编号	适种条件
粮食	旱地粮食	豆类	1-5	平旱地、梯田、山坡地
		非豆类	6-15	
	水田粮食		16	水浇地
蔬菜	非季节性蔬菜	豆类	17-19	水浇地(第一季) 普通大棚(第一季) 智慧大棚(第一季、第二季)
		非豆类	20-34	
	季节性蔬菜		35-37	水浇地(第二季)
食用菌	无		39-41	普通大棚(第二季)

上表 3 为本文依据上述 1 中的适种条件及附件条件得到的结果。从中可以看出不同条件下适合种植的作物也不同。比如旱地类型（平旱地、梯田以及山坡地）下只可种植豆类和非豆类的旱作粮食，而水浇地则可以种植水田粮食水稻。

为进一步提取作物中的信息，本文聚焦特殊种类的作物，得到如下特殊作物分类表：

表 4 特殊作物列表

精确定类	作物名称	特点
根类蔬菜	大白菜，白萝卜和红萝卜	只能种在水浇地第二季
食用菌	榆黄菇，香菇，白灵菇，羊肚菌	只能种在普通大棚第二季
水田粮食	水稻	只能种在水浇地第一季
豆类	粮食: 黄豆、黑豆、红豆、绿豆、爬豆 蔬菜: 豇豆、刀豆、芸豆	三年至少种植一次豆类

4.2.3 种植地区宏观可视化

根据附件 2 所给数据，可以得到各个作物种类分别在第一与第二季时的各作物排列情况，可以分别做出栅格图如下：

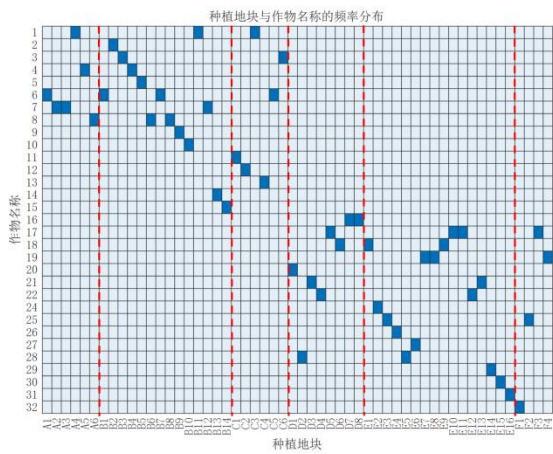


图 2 第一季各作物排列图

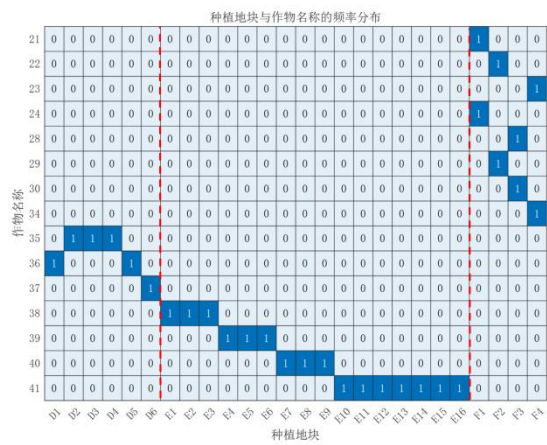


图 3 第二季各作物排列图

分析图像，由于无法考虑不同地块间的距离，仅观察同一类型地块中所种植的某一作物，可以发现除却极少量异常外该作物要么相邻地块种植要么单独种植一块。

4.3 数据预处理

4.3.1 缺失值处理

在 Excel 表中筛查所有附件和表单，未发现缺失值，故不做进一步处理。

4.3.2 异常值处理

1. 条件异常值检验

根据 4.3 可知，不同的地块类型在特定季度只能种植特定作物。将 2023 年的作物种植方案和种植要求进行一致性分析，检查是否有不符合种植要求的数据。因数据较少，采用人工校验筛选异常值，发现无异常值，故不作进一步处理。

2. 数据异常值检验

除却可能存在不满足已知条件的数据外，数据本身还可能存在离群点等数据异常问题，此处本文讨论销售单价、亩产量以及种植成本的数据是否存在异常值。

由于这里的销售单价为不确定性参数，本文提取随机范围的中间值，最小值，最大值以及区间大小作为代替，首先观察数据即可直接找到过高单价的部分异常点，即，在此基础上通过箱线图检验的结果如下：

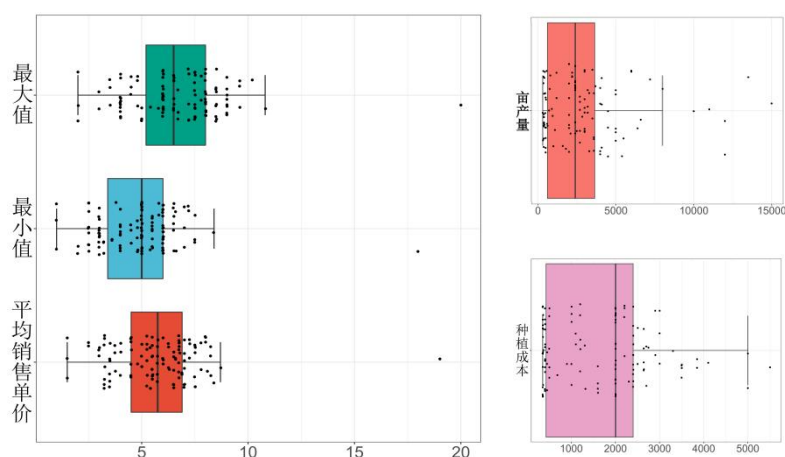


图 4 箱线图

通过整理，此处得到的所有异常值如下：

表 5 异常值列表

异常点编号	说明
11, 38, 39, 40, 41	过高单价
22, 29, 32	过高亩产量
32, 40, 41	过高种植成本

对于 11、38、39、40 和 41，主要为荞麦和食用菌等，经过资料查询发现其单价确实偏高，可以认为合理；而对于 22、29 和 32，主要为水浇地和人工地上种植的茄子、黄瓜以及空心菜，都是高产作物且均处于在较好环境中，可以不认为是异常值；同时我们发现 32、40、41 主要为在人工地和水浇地上的空心菜以及普通大棚中的白灵菇和羊肚菌，主要都是在人工地上种植的作物，其成本过高可以认为合理。

故而，本文不删除任何数据异常值。

五、问题一的模型建立与求解

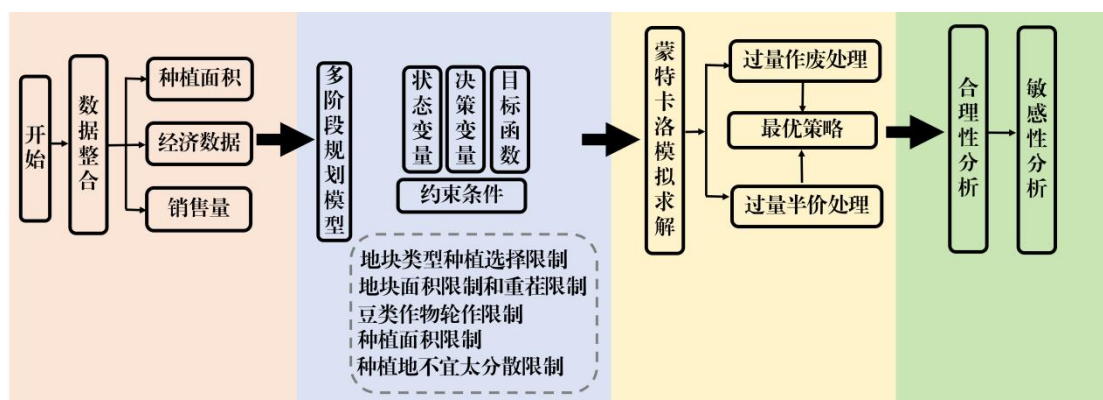


图 5 问题一流程图

5.1 多阶段规划模型

5.1.1 数据整合

为了后期的模型建立并提升数据应用的灵活性，更好地与种植方案格式匹配，本文首先将种植面积和经济数据整合为透视表，具体如下：

1.种植面积

附件 2 的表单 1 中为不同地块的作物在不同季度的种植情况，首先按照季度编码将种植面积数据分为两个季度。然后在每个季度，以所有作物编号为行，所有地块编号为列，将对应的种植面积填入透视表。透视表中没有种植数据的交叉点，视为未在该编号地块种植该种作物，故补充 0 值，得到完整的两个季度种植面积透视表。

2.经济数据

附件 2 表单 2 中的数据为不同地块类型上不同作物编号的经济数据，即亩产量，种植成本和销售单价。表单中隐藏了与普通大棚的第一季数据相同的智慧大棚的第一季数据。首先添加隐藏的数据，然后按照季度编码将种植面积数据分为两个季度。

在附件 1 表单 1 中有地块类型和地块名称的对应数据，且同一地块类型包括多个地块编号。按照这个对应数据，将经济数据对应的地块类型转换成多个地块编号，且同一地块类型对应的地块编号保留相同的经济数据值。以与种植面积相同的方式处理数据，获得亩产量，种植成本和销售单价在不同季度的透视表。

3.销售量

因本题缺少销售量数据，为简化计算，不妨令 2023 年的销售量等于产量。所以销售量为种植种植面积与亩产量的乘积，即：

$$D_{ijk}^{2023} = x_{ijk}^{2023} p_{ijk}^{2023} \quad (5.1)$$

其中， D_{ijk}^{2023} ， x_{ijk}^{2023} 和 p_{ijk}^{2023} 分别为 2023 年 i 编号地块在第 k 季度种植 j 编号作物的销售量，种植面积和亩产量。

对不同季度的销量分别统计透视表，行为作物编号，列为地块编号，值为销售量获得两季度销售量透视表。

5.1.2 多阶段规划模型的建立

根据题目要求，本文综合考虑不同地块类型、作物种类、种植季次以及特定种植和市场需求，构建多阶段随机规划模型。具体步骤如下：

1.决策变量

第 t 年，第 i 个地块(或大棚)的第 k 季度种植第 j 种作物的面积(亩)，即

$$x_{ijk}^t$$

其中， $i=1,2,\dots,N$:地块或大棚的索引、 $j=1,2,\dots,M$:作物的索引、 $k=1,2$:季度索引（1 表示第一季度，2 表示第二季度）、 $t=2024,2025,\dots,2030$:年份索引。

2.状态变量

在第 t 年，第 i 个地块的第 k 季度是否种植第 j 种作物，即：

$$y_{ijk}^t = \begin{cases} 0 & \text{不种植} \\ 1 & \text{种植} \end{cases} \quad (5.2)$$

第 t 年，第 i 个地块是否种植了豆类作物，即：

$$z_i^t = \begin{cases} 0 & \text{不种植} \\ 1 & \text{种植} \end{cases} \quad (5.3)$$

其中，豆类作物可分为粮食豆类(索引 1-5)和蔬菜豆类(索引 17-19)。

3.目标函数

目标函数要求 2024 年至 2030 年的种植总收益最大,同时考虑到超过市场需求部分的非正常处理的损失。具体公式如下:

$$\max \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t - g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \right) \right) \quad (5.4)$$

其中, A_i 为第 i 个地块的总面积; C_{ijk}^t 为第 t 年,第 i 个地块在第 k 季度种植第 j 种作物的种植成本; P_{ijk}^t 为第 t 年,第 i 个地块在第 k 季度种植第 j 种作物的亩产量; S_{ijk}^t 为第 t 年,第 j 种作物在第 k 季度种植第 j 种作物的销售价格; D_{ijk}^{2023} 为第 j 种作物的市场预期销售量。

情况 1: 过量作废处理

在情况 1 的条件下,影响函数 $g(\cdot)$ 为:

$$g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) = 0 \quad (5.5)$$

即情况 1 条件下过量农作物成本全部损失

此时目标函数具体为

$$\max \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t \right) \right) \quad (5.6)$$

情况 2: 过量半价处理

在情况 2 的条件下,影响函数 $g(\cdot)$ 为:

$$g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) = -\frac{1}{2} S_{ijk}^t (P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \quad (5.7)$$

即情况 2 条件下过量农作物能够弥补一定损失,但是实际获利仅有理想情况(全部原价售卖)的一半

此时目标函数具体为

$$\max \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t + \frac{1}{2} S_{ijk}^t (P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \right) \right) \quad (5.8)$$

4.约束条件

● 约束 1:不同地块类型种植选择限制

➤ 旱地(地块索引 1-26)

(1) 种植季度约束:在每一年只有一个季度种植,即:

$$x_{ij2}^t = 0, \forall i = 1, \dots, 26, \forall j = 1, \dots, 15, \forall t \quad (5.9)$$

(2) 作物种类约束:只能选择种植粮食作物(索引 1-15),而对于这些地块上所有非粮食作物(索引 16 及以上),种植面积 x_{ijk}^t 必须为 0,即:

$$x_{ijk}^t = 0, \forall i = 1, \dots, 41, \forall k, \forall t \quad (5.10)$$

➤ 水浇地(地块索引 27-34)

(1) 种植模式约束:每个水浇地(索引 27-34)在每个年份只能选择一种种植方式一季水稻或两季蔬菜,该约束可表示为:

$$z_i^t + \sum_{j=35}^{37} y_{ij2}^t = 1, \forall i = 27, \dots, 34, \forall j \in \{35, 36, 37\}, \forall t \quad (5.11)$$

$$z_i^t = \begin{cases} 1 & \text{种植一季水稻} \\ 0 & \text{种植两季蔬菜} \end{cases}$$

其中, y_{ij2}^t 表示是否在第二季度种植作物 j (大白菜、白萝卜、红萝卜之一)。

- (2) 水稻 (编号 16) 种植选择的约束条件: 满足第一季度仅种植水稻, 第二季不种植其他任何作物:

$$\begin{cases} x_{ij1}^t = 0, \forall i = 27, \dots, 34, \forall j \neq 16, \forall t \\ x_{ij2}^t = 0, \forall i = 27, \dots, 34, \forall j, \forall t \end{cases} \quad (5.12)$$

其中, x_{ij1}^t 为第一季种植情况约束, x_{ij2}^t 为第二季种植情况约束。

- (3) 两季蔬菜种植选择的约束条件: 第一季度可种植(除大白菜、白萝卜、或红萝卜外)任何蔬菜, 但第二季度只能种植大白菜、白萝卜、或红萝卜 (编号 35-37) 中的一种, 以达到便于管理的目的。

$$\begin{cases} x_{ij1}^t = 0, \forall i = 27, \dots, 34, \forall j \in \{35, 36, 37\}, \forall t \\ x_{ij2}^t = 0, \forall i = 27, \dots, 34, \forall j \notin \{35, 36, 37\}, \forall t \\ y_{i,35,2}^t + y_{i,36,2}^t + y_{i,37,2}^t \leq 1, \forall i = 27, \dots, 34, \forall t \end{cases} \quad (5.13)$$

其中, y_{ijk}^t 表示在第 t 年的第 i 个地块的第 2 季度是否种植了第 j 种作物 (1 表示种植, 0 表示不种植)。

➤ 普通大棚 (地块索引 35-50)

- (1) 种植季度约束: 在每一年可种植两个季度。

- (2) 作物种类约束:

第一季度种植种类约束: 普通大棚第一季只能种植编号为 17-34 的蔬菜, 对于其他作物种植面积 x_{ij1}^t 必须为 0:

$$x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{17, \dots, 34\}, \forall t \quad (5.14)$$

第二季度种植种类约束: 普通大棚第二季度能种植编号为 38-41 的食用菌, 且食用菌只能在普通大棚第二季种植, 而对于其他作物种植面积 x_{ij2}^t 必须为 0:

$$\begin{cases} x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \in \{38, 39, 40, 41\}, \forall t \\ x_{ij2}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{38, 39, 40, 41\}, \forall t \end{cases} \quad (5.15)$$

➤ 智慧大棚 (地块索引 51-54)

- (1) 种植季度约束: 在每一年可种植两个季度。

- (2) 作物种类约束:

智慧大棚两个季度都只能种植除大白菜、白萝卜和红萝卜 (编号 35-37) 以外的蔬菜 (编号 17-34), 即:

$$\begin{cases} x_{ijk}^t = 0, \forall i = 51, \dots, 54, \forall j \notin \{17, \dots, 34\}, \forall k = 1, 2, \forall t \\ x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{17, \dots, 34\}, \forall t \\ x_{ij2}^t = 0, \forall i = 51, \dots, 54, \forall j \in \{35, 36, 37\}, \forall t \end{cases} \quad (5.16)$$

● 约束 2: 地块面积限制和重茬限制

➤ 单季度种植地块:

- (1) **面积限制**: 每个单季度种植地块的种植面积必须小于或等于该地块的总面积 A_i , 且只计算一个季度的种植面积 (默认是第一季度)。该约束确保了单季度种植地块在任何年份 t 的第一个季度的总种植面积不会超过其总面积。

$$\sum_{j=1}^M x_{ijk}^t \leq A_i, \forall i \in \text{单季度种植地块}, \forall t \quad (5.17)$$

- (2) **重茬限制**: 为了防止作物连续种植在同一个地块上, 必须确保第 t 年种植的作物与第 $t+1$ 年的作物不同。该约束限制了每个单季度地块 i 和作物 j 在连续的两个年份内不能种植相同的作物。

$$y_{ijk}^t + y_{ijk}^{t+1} \leq A_i, \forall i \in \text{单季度种植地块}, \forall j, \forall t \quad (5.18)$$

► 双季度种植地块

- (1) **面积限制**: 对于双季度种植地块, 每个季度的种植面积之和不需要小于地块总面积 A_i , 因为每个季度的作物种植和收割是独立的。因此, 每个季度的种植面积必须单独小于或等于地块总面积。

$$\sum_{j=1}^M x_{ijk}^t \leq A_i, \forall i \in \text{双季度种植地块}, \forall k = 1, 2, \forall t \quad (5.19)$$

- (2) **重茬限制**: 对于双季度种植地块 (可以在两个季度种植的地块), 需要设置两类重茬限制约束。

同年季度: 第 t 年第 1 季度种植的作物与第 2 季度种植的作物不同。

$$y_{ij1}^t + y_{ij2}^t \leq 1, \forall i \in \text{双季度种植地块}, \forall j, \forall t \quad (5.20)$$

跨年季度: 第 t 年第 2 季度种植的作物与第 $t+1$ 年第 1 季度种植的作物不同。

$$y_{ij1}^{t+1} + y_{ij2}^t \leq 1, \forall i \in \text{双季度种植地块}, \forall j, \forall t \quad (5.21)$$

其中, $y_{ijk}^t = \begin{cases} 0 & \text{不种植} \\ 1 & \text{种植} \end{cases}$

表示在第 t 年, 第 i 个地块的第 k 季度是否种植第 j 种作物。

● 约束 3: 豆类作物轮作限制

由于含有豆类作物根菌的土壤有利于其他作物生长, 要求每个地块三年内至少种植一次豆类作物, 为便于计算可按季度对地块分类分析, 具体如下:

► 单季度地块 (Single-Season Plots):

每年只种植一个季度的作物。轮作要求需要确保在每个三年周期内至少有一个季度种植了豆类作物, 即对于每个单季度地块 i , 从第 t 年到第 $t+2$ 年内至少有一年种植了豆类作物:

$$\sum_{j \in \text{豆类}} y_{ijk}^t + \sum_{j \in \text{豆类}} y_{ijk}^{t+1} + \sum_{j \in \text{豆类}} y_{ijk}^{t+2} \geq 1, \quad (5.22)$$

$\forall i \in \text{单季度种植地块}, \forall t = 2024, \dots, 2028$

► 双季度地块 (Double-Season Plots):

每年可以种植两个季度的作物, 因此在轮作要求上更为灵活。约束条件可以表示为对于每个双季度地块 i , 从第 t 年到第 $t+2$ 年内至少有一个季度种植

了豆类作物:

$$\sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^t + \sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^{t+1} + \sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^{t+2} \geq 1, \quad (5.23)$$

$\forall i \in \text{双季度种植地块}, \forall t = 2024, \dots, 2028$

● 约束 4:种植面积限制

根据需求每种作物在单个地块（含大棚）种植的面积不宜太小，但由于不同地块的基数不同，无法统一量化，为此，应选择最小种植面积占比为标准，即每种作物 j 在地块 i 上的种植面积必须至少达到地块总面积的 α_j 的比例。（观察可以显然得到为 1 或者 0.5）

$$x_{ijk}^t \geq \alpha_j \cdot A_i \cdot y_{ijk}^t, \forall i, j, k, t \quad (5.24)$$

其中， α_j 是作物 j 的最小种植占比要求（如 20%）； A_i 为地块 i 的总面积； y_{ijk}^t 是指示变量，表示是否种植作物 j （1 表示种植，0 表示不种植）。

● 约束 5:种植地不宜太分散限制

以 4.4 为基准，本文假定种植地不宜太分散为在种植面积不过量的情况下，仅种植在单一的地块中；如果种植所需面积过量则在种满第一个地块后，再种在后一个相邻地块。

面积过量的衡量为：

$$\sum_{i \in I_i} x_{ijk}^{2023} \geq A_i \quad (5.25)$$

其中，左边式为 2023 年第 j 种作物在第 k 季度对 i 地块所在地块类型中种植面积的总和，即为作物所需面积， I_i 为地块编号 i 对应的地块类型的地块编号范围。

➤ 面积不过量情况

当面积不过量时，作物仅种植在单一的地块中。在第一地块中的具体种植方案为：作物在地块 i 所在地块类型中的种植总面积大于地块 i 面积的一半时，种满地块 i ，否则种一半，即：

$$x_{ijk}^t = \begin{cases} A_i & \sum_{i \in I_i} x_{ijk}^{2023} > 0.5 A_i \\ 0.5 A_i & \sum_{i \in I_i} x_{ijk}^{2023} \leq 0.5 A_i \end{cases} \quad (5.26)$$

➤ 面积过量情况

如果种植所需面积过量则在种满第一个地块后，再种在后一个相邻地块。具体方案为：

$$x_{ijk}^t = A_i \quad (5.27)$$

$$A_{last} = \sum_{i \in I_i} x_{ijk}^{2023} - A_i \quad (5.28)$$

$$x_{(i+1)jk}^t = \begin{cases} A_{i+1} & A_{last} > 0.5 A_{i+1} \\ 0.5 A_{i+1} & A_{last} \leq 0.5 A_{i+1} \end{cases} \quad (5.29)$$

其中， A_{last} 为种植完地块 i 后，剩余的所需面积。剩余所需面积大于相邻地块 $i+1$ 面积的一半时，种满地块 $i+1$ ，否则种一半。

5.模型整合

综上，本文建立了多阶段规划模型，具体如下：

$$\begin{aligned}
& \max \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t - g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \right) \right) \\
& s.t. \left\{ \begin{array}{l}
\text{约束1} \left\{ \begin{array}{l}
\text{平旱地、梯田、山坡地} \begin{cases} x_{ij2}^t = 0, \forall i = 1, \dots, 26, \forall j = 1, \dots, 15, \forall t \\ x_{ijk}^t = 0, \forall i = 1, \dots, 41, \forall k, \forall t \end{cases} \\
\text{水浇地} \begin{cases} z_i^t + \sum_{j=35}^{37} y_{ij2}^t = 1, \forall i = 27, \dots, 34, \forall j \in \{35, 36, 37\}, \forall t \\ x_{ij1}^t = 0, \forall i = 27, \dots, 34, \forall j \neq 16, \forall t \\ x_{ij2}^t = 0, \forall i = 27, \dots, 34, \forall j, \forall t \\ x_{ij1}^t = 0, \forall i = 27, \dots, 34, \forall j \in \{35, 36, 37\}, \forall t \\ x_{ij2}^t = 0, \forall i = 27, \dots, 34, \forall j \notin \{35, 36, 37\}, \forall t \\ y_{i,35,2}^t + y_{i,36,2}^t + y_{i,37,2}^t \leq 1, \forall i = 27, \dots, 34, \forall t \end{cases} \\
\text{普通大棚} \begin{cases} x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{17, \dots, 34\}, \forall t \\ x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \in \{38, 39, 40, 41\}, \forall t \\ x_{ij2}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{38, 39, 40, 41\}, \forall t \end{cases} \\
\text{智慧大棚} \begin{cases} x_{ijk}^t = 0, \forall i = 51, \dots, 54, \forall j \notin \{17, \dots, 34\}, \forall k = 1, 2, \forall t \\ x_{ij1}^t = 0, \forall i = 35, \dots, 50, \forall j \notin \{17, \dots, 34\}, \forall t \\ x_{ij2}^t = 0, \forall i = 51, \dots, 54, \forall j \in \{35, 36, 37\}, \forall t \end{cases}
\end{array} \right. \\
\text{约束2} \left\{ \begin{array}{l}
\text{面积限制} \begin{cases} \sum_{j=1}^M x_{ijk}^t \leq A_i, \forall i \in \text{单季度种植地块}, \forall k = 1, 2, \forall t \\ \sum_{j=1}^M x_{ijk}^t \leq A_i, \forall i \in \text{双季度种植地块}, \forall k = 1, 2, \forall t \end{cases} \\
\text{重茬限制} \begin{cases} y_{ijk}^t + y_{ijk}^{t+1} \leq A_i, \forall i \in \text{单季度种植地块}, \forall j, \forall t \\ \left\{ \begin{array}{l} \text{同年季度: } y_{ij1}^t + y_{ij2}^t \leq 1, \forall i \in \text{双季度种植地块}, \forall j, \forall t \\ \text{跨年季度: } y_{ij1}^{t+1} + y_{ij2}^t \leq 1, \forall i \in \text{双季度种植地块}, \forall j, \forall t \end{array} \right. \end{cases}
\end{array} \right. \\
\text{约束3} \left\{ \begin{array}{l} \sum_{t=2024}^{t+2} \sum_{j \in \text{豆类}} y_{ij1}^t \geq 1, \forall i \in \text{单季度种植地块}, \forall t = 2024, \dots, 2028 \\ \sum_{t=2024}^{t+2} \sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^t \geq 1, \forall i \in \text{双季度种植地块}, \forall t = 2024, \dots, 2028 \end{array} \right. \\
\text{约束4} \quad x_{ijk}^t \geq \alpha_j \cdot A_i \cdot y_{ijk}^t, \forall i, j, k, t \\
\text{约束5} \left\{ \begin{array}{l} x_{ijk}^t = \begin{cases} A_i & \sum_{i \in I_i} x_{ijk}^{2023} > 0.5 A_i \\ 0.5 A_i & \sum_{i \in I_i} x_{ijk}^{2023} \leq 0.5 A_i \end{cases} \quad \sum_{i \in I_i} x_{ijk}^{2023} < A_i \\ x_{(i+1)jk}^t = \begin{cases} A_{i+1} & A_{last} > 0.5 A_{i+1} \\ 0.5 A_{i+1} & A_{last} \leq 0.5 A_{i+1} \end{cases} \quad \sum_{i \in I_i} x_{ijk}^{2023} \geq A_i \end{array} \right.
\end{array} \right.
\end{aligned}
\tag{5.30}$$

5.2 基于蒙特卡罗模拟的多阶段随机规划模型的求解

5.2.1 滚动优化模型的引入

在这里由于不仅有对于每一年相同的要求，如约束 1 中不同地块类型适应的作物限制，同时也有对于一段时间内的约束，如约束 2 中的重茬限制与约束 3 中的豆类轮作限制。为此，本文引入滚动优化模型^[2]，通过基于时间序列逐步更新，不断优化各个时间段最后更新整体的未来决策。具体过程如下图所示：

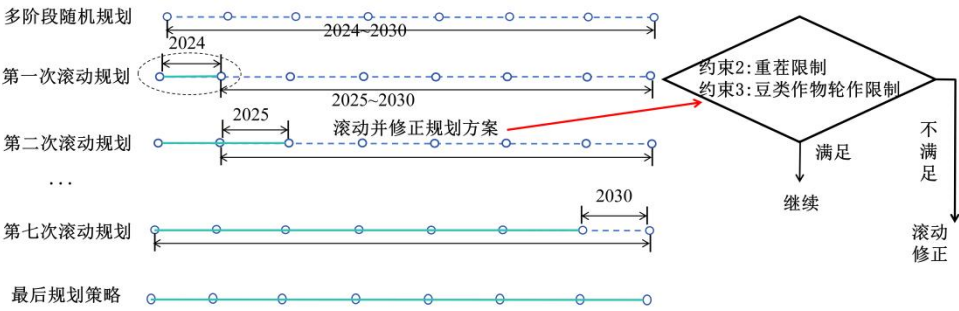


图 6 滚动优化原理图

5.2.2 蒙特卡罗模拟求解

蒙特卡罗模拟（Monte Carlo Method）也称统计模拟法、统计实验法，是一种通过随机采样来模拟系统行为并进行决策分析的数值方法，是按抽样调查法求取统计值推定未知特性量的计算方法^[1]。具体步骤如下列伪代码所示：

表 6 蒙特卡洛算法伪代码

算法 1：蒙特卡洛算法

输入：模拟次数 n 、经济效益函数、作物总数 M 、地块总数 N 、时间范围 T

输出：模拟得出的最优种植策略、最优策略下的经济效益

1:初始化输入数据：地块面积、亩产量、种植成本、销售价格等

2:For i 从 1 到模拟次数 n

3: 随机初始化种植方案，生成地块与作物的随机种植量矩阵

4: For t 从 1 到 T

5: 约束 1:不同地块类型种植选择限制

6: 约束 4:种植面积限制

7: 约束 5:种植地不宜太分散限制

8: $t=t+1$

9: end

10: 整体约束:

11: 约束 2:重茬限制

12: 约束 3:豆类作物轮作限制

13: 计算当前的经济效益

14: $i=i+1$

15:end

16:返回结果

输出最优种植策略及其经济效益

利用蒙特卡洛模拟求解，具体的种植策略见附件，求得情况 1 和情况 2 的收益分别为收益 1 和收益 2，列出年份 2024 至 2030 年的具体收益情况如下表：

年份	2024	2025	2026	2027	2028	2029	2030
收益 1	5873613	5955049	5958657	5954838	5973633	6041416	5904912
收益 2	5918823	5893306	5853169	5946593	5981521	5937903	5910530

两种情况的 7 年总收益分别是 41662119.8 元与 41441844 元。经过对 23 年中心数据的统计，得到当年的收益为 5926348 元。对比规划的种植方案和 2023 年实际方案的收益，发现规划的年收益和 23 年实际收益相差较小，这进一步验证了种植策略的现实意义

5.2.3.结果合理性分析

为检验结果的准确性，本文通过检验正态分布检验，如下图。得到通过该模型在两种情况下得到的 p-value 随着经济效益整体整体上都呈现正态分布，但是情况 2 中的 p-value 为 0.82808 远大于情况 1 中的 0.35315，可见情况 2 更加接受正态分布的假设。

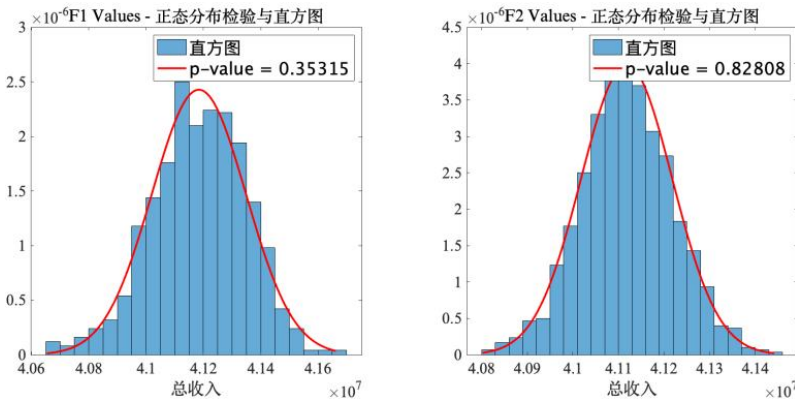


图 7 正态分布检验图

为进一步探究两种情况的优劣，本文再次对 1000 次蒙特卡洛模拟的经济效益进行可视化，如图所示：

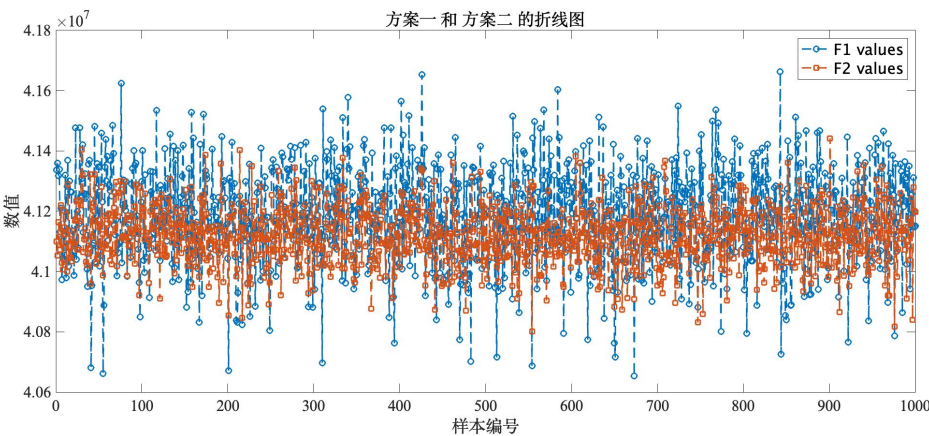


图 8 蒙特卡洛模拟可视化图

显然相对于情况 1，情况 2 的经济收益波动性较弱，有较好的稳健型，所以从稳定收益的角度来看应该选择情况 2 的过量半价处理。但是在于部分条件下，选择情况 1 的过量作废，可以提升经济效益的边际，尝试获得更高的收益。

5.3 敏感性分析

在蒙特卡罗模拟中，以经济效益为特征建立标准化灵敏度分析模型。该模型在鲁棒性评估中效果优越。标准灵敏度 S_i 为：

$$S_i = \frac{\Delta f_i / f_0}{\Delta x_i / x_0} \quad (5.31)$$

其中， Δf_i ， Δx_i 分别为模拟的输出变化和输入变化， f_0 和 x_0 分别为原始输出和输入。标准灵敏度 S_i 越接近 0 则模型的鲁棒性越好。

对销售单价进行 5% 的扰动后，进行蒙特卡罗模拟，并计算特征的平均鲁棒性。

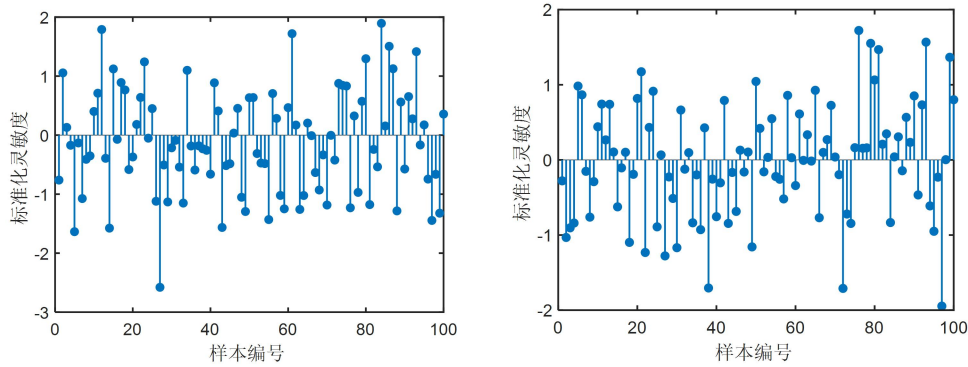


图 9 鲁棒性分析图

如左图为情景 1 的标准化灵敏度，右图为情景 2 的。观察图像可知，两种情景的标准化灵敏度均在 0 值上下，情景 1 的平均标准灵敏度为 0.1713，情景 2 的为 -0.154，两情景的标准灵敏度均较小，模型都具有良好的鲁棒性。但是情景 2 的种植策略比情景 1 的更稳定。

六、问题二的模型建立与求解

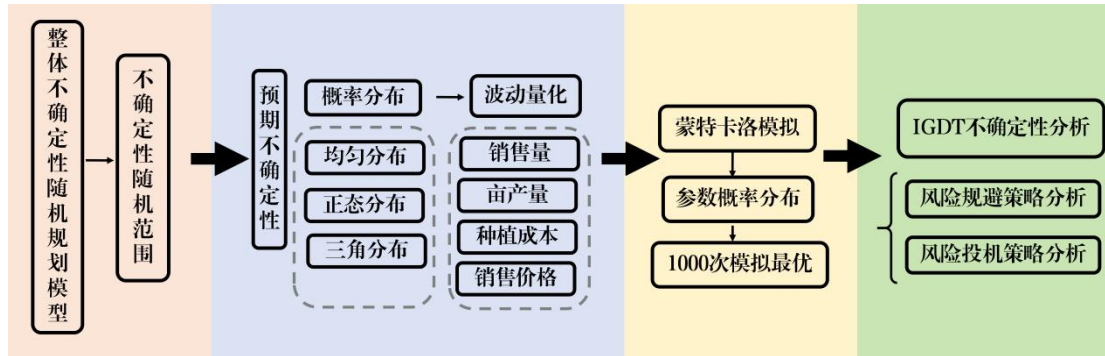


图 10 问题二流程图

6.1 整体不确定性随机规划模型建立

根据题意，综合考虑各农作物的预期销售量、亩产量、种植成本和销售价格的随机波动性以及潜在风险，可以得到最后的目标函数如下：

$$\max \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t - g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \right) \right)$$

$$s.t. \begin{cases} S_{ijk}^{t+1} = L_s \cdot S_{ijk}^t \\ P_{ijk}^{t+1} = L_p \cdot P_{ijk}^t \\ D_{ijk}^{t+1} = L_d \cdot D_{ijk}^t \\ C_{ijk}^{t+1} = L_c \cdot C_{ijk}^t \\ \text{公式(5.5)} \sim \text{公式(5.25)} \end{cases} \quad (6.1)$$

其中, S_{ijk}^t 为第 t 年, 第 j 种作物在第 k 季度种植第 j 种作物的销售价格; P_{ijk}^t 为第 t 年, 第 i 个地块在第 k 季度种植第 j 种作物的亩产量; D_{ijk}^{2023} 为第 j 种作物的市场预期销售量; C_{ijk}^t 为第 t 年, 第 i 个地块在第 k 季度种植第 j 种作物的种植成本; L_s, L_p, L_d, L_c 分别为与销售价格、亩产量、预期销售量以及种植成本的不确定性随机范围。

6.2 基于 IGDT 分析的多阶段随机规划模型

6.2.1 考虑预期不确定性的多阶段随机规划模型

预期不确定性是指具有波动特征可以大致量化范围的性质^[3], 而本题的预期不确定性因素主要为预期销售量、亩产量、种植成本以及销售价格的不确定性。

1. 随机变量的概率分布函数

跟据题目, 本文首先必须探究各个预期不确定性因素的随机分布函数, 以之量化各因素的取值范围与概率, 如下为 3 个常见的概率分布函数的相关信息及适用条件:

均匀分布: 本文数据主要为连续型数据, 故而此处均匀分布仅讨论连续型均匀分布。该分布主要适用于**固定均匀区间** $[a, b]$ 中的数据。其概率密度函数和累计分布函数如下:

概率密度函数 $f(x)$:

$$f(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & x \notin [a, b] \end{cases} \quad (6.2)$$

累计分布函数 $F(x)$:

$$F(x) = \begin{cases} 0 & x \in (-\infty, a) \\ \frac{x-a}{b-a} & x \in [a, b] \\ 1 & x \in (b, +\infty) \end{cases} \quad (6.3)$$

正态分布: 最常见的连续型概率分布之一, 也叫高斯分布。其有**明确的平均值与标准差**, 且波动呈现关于均值的**对称性**。其概率密度函数和累计分布函数如下:

概率密度函数 $f(x)$:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (6.4)$$

其中， μ 为正态分布的均值； σ 为正态分布的标准差；

累计分布函数 $F(x)$:

$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x - \mu}{\sigma \sqrt{2}} \right) \right] \quad (6.5)$$

其中， $\operatorname{erf}(z)$ 为误差函数，记为：

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \quad (6.6)$$

三角分布:一种连续概率分布、常用于近似未知分布，一般由三个参数构成，即最小值 a ，最大值 b ，以及最可能值 c 。其概率密度函数和累计分布函数如下：

概率密度函数 $f(x)$:

$$f(x) = \begin{cases} 0 & x \in (-\infty, a) \\ \frac{2(x-a)}{(b-a)(c-a)} & x \in [a, c) \\ \frac{2(b-x)}{(b-a)(b-c)} & x \in [c, b] \\ 0 & x \in (b, +\infty) \end{cases} \quad (6.7)$$

累计分布函数 $F(x)$:

$$F(x) = \begin{cases} 0 & x \in (-\infty, a) \\ \frac{(x-a)^2}{(b-a)(c-a)} & x \in [a, c) \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & x \in [c, b] \\ 1 & x \in (b, +\infty) \end{cases} \quad (6.8)$$

2.预期不确定性波动范围量化

接着对具体的不定性因素进行量化范围，具体如下：

● 预期销售量的不确定性

由题意得，小麦和玉米未来的平均年增长率介于 5%~10%，记为 $\alpha_{jd}(j=6,7)$ ，假定服从均匀分布；其余农作物销售量每年有 $\pm 5\%$ 的变化，记为 $\alpha_{jd}(j \neq 6,7)$ ，因其呈现对称性，且有明确的平均值与标准差，故服从正态分布。因此，可得预期销售量的不确定性波动范围 L_d 为：

$$L_d = \left\{ L_{d,T} \left| \begin{array}{ll} L_{d,T} \sim \bigcup (\inf \alpha_{jd}, \sup \alpha_{jd}) & j = 6, 7 \\ L_{d,T} \sim N \left(\mu, \sigma_d^2 \right), \text{其中 } \sigma_d = \frac{\alpha_{jd} \times \mu}{a} & j \neq 6, 7 \end{array} \right. \right\} \quad (6.9)$$

其中， $\inf \alpha_{jd}$ 和 $\sup \alpha_{jd}$ 分别代表 α_{jd} 的上下确界； a 为正态分布 95% 置信区间的分界点； μ 为销售量平均值。

● 亩产量的不确定性

已知农作物亩产量受气候等因素影响，每年有 $\pm 10\%$ 的变化，记为 α_{jp} ，因其波动呈现对称性，且有明确的平均值与标准差。因此， α_{jp} 服从正态分布，可得亩产量的不确定性波动范围 L_p 为：

$$L_p = \left\{ L_{p,T} \left| L_{p,T} \sim N\left(\mu, \sigma_p^2\right), \text{其中} \sigma_p = \frac{\alpha_{jp} \times \mu}{a}, \forall j \right. \right\} \quad (6.10)$$

其中， a 为正态分布 95%置信区间的分界点， μ 为亩产量平均值。

● 种植成本的不确定性

受市场条件影响，农作物的种植成本平均每年增长率为 5% 左右，记为 α_{jc} 。假设价格增长最可能接近 α_{jc} ，上下限设为 $0.8\alpha_{jc}$ 和 $1.2\alpha_{jc}$ 。因此，其随机变量服从三角分布，反映了偏向 α_{jc} 的特征。故而，可得种植成本的不确定性波动范围 L_c 为：

$$L_c = \left\{ L_{c,T} \left| L_{c,T} \sim \text{angular}\left(0.8\alpha_{jc}, 1.2\alpha_{jc}, \alpha_{jc}\right), \forall j \right. \right\} \quad (6.11)$$

● 销售价格的不确定性

各不同类型农作物销售价格差异很大，为便于计算统一记为 α_{js} 。粮食类作物销售价格基本稳定，服从正态分布，($j=1, \dots, 16$)；蔬菜类平均每年增长 5% ($j=17, \dots, 37$)，随机变量服从三角分布；食用菌销售价格每年下降 1% 到 5% ($j=38, 39, 40$)，随机变量服从均匀分布，其中羊肚菌销售价格下降幅度固定为 5% ($j=41$)，随机变量服从固定比例衰退模型。故而，可得销售价格的不确定性波动范围 L_s 为：

$$L_s = \left\{ L_{s,T} \left| \begin{array}{ll} L_{s,T} \sim N\left(\mu, \sigma_s^2\right), \text{其中} \sigma_s = \frac{\alpha_{js} \times \mu}{a} & j \in \{1, \dots, 16\} \\ L_{s,T} \sim \text{Triangular}\left(0.8\alpha_{js}, 1.2\alpha_{js}, \alpha_{js}\right) & j \in \{17, \dots, 37\} \\ L_{s,T} \sim \bigcup\left(\inf \alpha_{js}, \sup \alpha_{js}\right) & j \in \{38, 39, 40\} \\ L_{s,T} = \left(1 - \alpha_{js}\right)^t L_{s0} & j = 41 \end{array} \right. \right\} \quad (6.12)$$

其中， L_d, L_p, L_c, L_s 分别为预期销售量、亩产量、种植成本以及销售价格的不确定性波动范围； $L_{d,T}, L_{p,T}, L_{c,T}, L_{s,T}$ 分别为预期销售量、亩产量、种植成本以及销售价格的实际值。

综上，可得各不确定性因素的随机变量服从的分布类型，具体如下表所示：

表 7 随机变量分布表

作物类型		预期销售量	亩产量	种植成本	销售价格
粮食	小麦和玉米	均匀分布	正态分布	三角分布	正态分布
	其余粮食	正态分布			三角分布
蔬菜					固定比例衰退
食用菌	羊肚菌				均匀分布
	其余食用菌				

6.2.2 IGDT 不确定性分析

信息差决策理论 (IGDT) 是一种用于应对不确定性决策的方法，适合在面对深度不确定性时帮助决策者做出稳健和机会性导向的决策^[4]。已知种植过程中

存在潜在风险，为应对该未知因素的风险，分别通过风险规避策略和风险投机策略对模型的稳健性和机会性分析，以提升对于危机的应对能力。

1.风险规避策略分析

风险规避模型，即鲁棒性优化模型，考虑各因素不确定性都处于最坏的情况下，分析能够保持最低收益目标的最大综合不确定范围。

$$\alpha(X) = \max\{u \mid P(f(X, \alpha) \geq r) \geq 1 - \kappa\} \quad (6.13)$$

其中， u 是不确定性水平，表示不确定性波动范围； X 为对应的种植策略； $P(f(X, \alpha) \geq R)$ 是决策 X 在不确定性 α 下满足标准收益 r 的概率； κ 是容忍的风险阈值，表示允许的失败概率。

2.风险投机策略分析

风险投机模型，即机会性优化模型，考虑各因素不确定性都处于最好的情况下，分析可以获得的最高收益时的最大综合不确定范围。

$$\beta(X) = \min\{u \mid P(f(X, \alpha) \geq R) \geq \rho\} \quad (6.14)$$

其中， u 是不确定性水平，表示不确定性波动范围； X 为对应的种植策略； $P(f(X, \alpha) \geq R)$ 是决策 X 在不确定性 α 下满足较高收益 r 的概率； ρ 是机会性阈值，表示实现高收益的概率。

6.3 模型求解

由于预期销售量、亩产量、种植成本和销售单价随前一年的数据滚动变化。6.2.1 中根据随机变量滚动变化的情况，分别分析它们的概率分布函数，并量化预期不确定性波动范围。以预期销售量为例，下图为不同作物在 23 年至 24 年的数值呈复杂的动态变化，如下图：

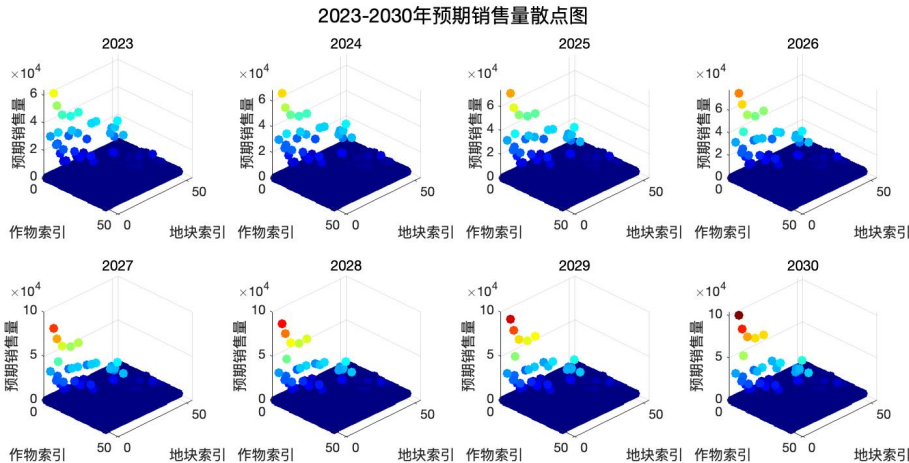


图 11 预期销售量散点图

其中，小麦和玉米的预期销量变化明显，逐步增大。其他作物的预期销售量在上下波动变化。

根据不同随机变量的波动范围在对应概率分布函数上取 100 个初始值，组合得到多种初始情景。用滚动迭代后面年份的情景组合。用蒙特卡罗模拟种植方案，在多个情景计算目标函数，并最终得到适应多种情景的最优种植方案。具体方案见附件，并得到 2024~2030 年七年的总收益为 40904868.8 元，具体如下：

年份	2024	2025	2026	2027	2028	2029	2030
收益	5823151	5791443	5860212	5861223	5858059	5841374	5869407

由于本题为一个不确定性的随机规划模型，将年收益绘制随时间变化的曲线图，并以 95%的置信区间描述多次蒙特卡罗模拟取值的波动区间得到图像如下：

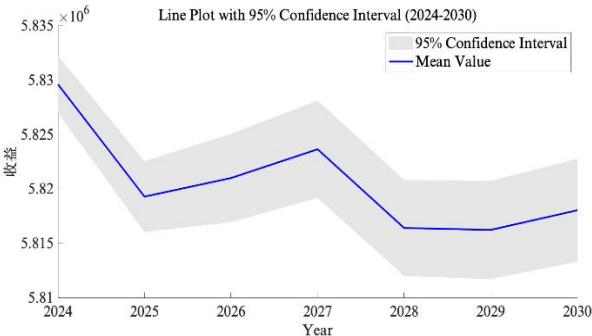


图 12 波动区间图

根据图像，整体的年收益大致随时间呈现下降趋势，且时间越晚，数据的不确定性和波动性呈增大趋势。这是由于销售量，亩产量，销售价格等在当前年份随前一年份呈滚动变化，这导致了时间靠后的的随机变量滚动迭代次数过多，因此增加了不确定性和波动性。

6.4 模型检验

相较于第一问，本文增加了多种参数的不确定性，在随机参数中动态求解最优策略，因此本题的策略相较于第一问，有更好的稳健性和适应性，但收益大小关系不确定。

本题的策略收益为 54466827.24，相较于第一问的收益 59755608.28 结果较小。为分析稳健性，在蒙特卡罗分析中，以经济效益为特征，建立标准化灵敏度分析模型，解得每个样本的标准化灵敏度如下图：

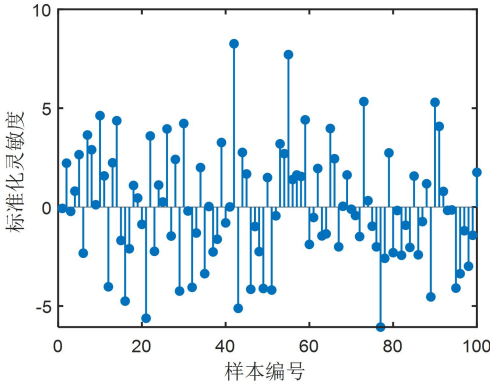


图 13 问题二鲁棒性图

观察图像可知，灵敏度在 0 值附件分布，说明该策略具有较好的鲁棒性。对标准化灵敏度求均值，为 0.03169，相较于问题一的平均标准化灵敏性更靠近 0，均有更高的鲁棒性。符合模型的稳健性更高，效益不定的先决判断，模型合理。

七、问题三的模型建立与求解

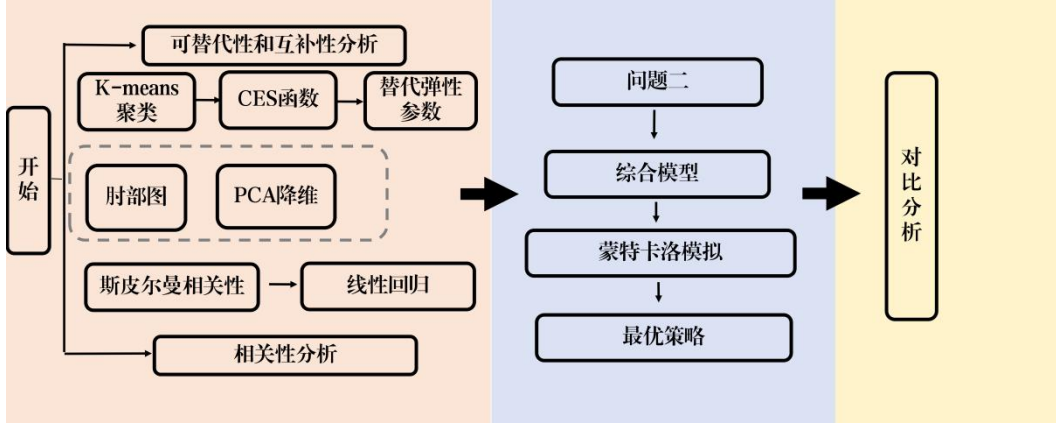


图 14 问题三流程图

7.1 可替代性和互补性分析

结合 4.3，以地块类型为依据，将作物分为旱地粮食，非季节性蔬菜，季节性蔬菜，食用菌和水生粮食五类。观察数据，显然可得季节性蔬菜，食用菌和水田粮食的种类较少，研究价值较弱，为简化计算，不讨论其可替代性和互补性。

7.1.1 模型的建立

1. K-means 聚类

K-means 聚类是将数据集划分为 k 个到簇心距离最小的簇的聚类方法^[5]。过程如下：

对于每个样本 i ，计算其到每个簇中心的欧几里德距离。将样本分配给距离最近的簇中心 j 。对于每个簇 C_j ，计算簇内所有样本的平均值，并将其作为新的簇中心。更新簇中心的公式为：

$$\mu_i = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \quad (7.1)$$

其中， $|C_j|$ 表示簇 C_j 中样本的数量， x_i 为簇 C_j 中的样本。

接着不断迭代并检查模型的收敛性，若收敛效果好，则停止迭代更新簇中心，得到最终的聚类结果。每次迭代中，K-means 的目标是最小化簇内平方和 J ，目标函数公式为：

$$J = \sum_{j=1}^k \sum_{e_i \in C_j} \|e_i - \mu_j\|^2 \quad (7.2)$$

其中， C_j 为第 j 个簇， μ_j 为第 j 个簇的质心， e_i 是簇 C_j 中的数据点。

2. 肘部图法

肘部图法是在 K-means 聚类的基础上引入不同聚类数量 k 值，绘制对应最小簇内平方和 J 与聚类数量 k 的肘部图像，其拐点对应的聚类数量 k 即为最佳聚类。二阶差分在曲线中表现为变化的加速度，本题选用二阶差分法寻找拐点，具体公式如下：

$$\begin{aligned} \Delta x_i &= x_{i+1} - x_i \\ \Delta(\Delta x) &= \Delta x_{i+1} - \Delta x_i \end{aligned} \quad (7.3)$$

其中 Δx_i 为一阶差分， $\Delta(\Delta x_i)$ 为二阶差分。

计算各聚类数量 k 处的二阶差分，最大二阶差分处的 k 值即为最佳聚类数量。

3. 成分分析

主要成分分析(PCA)可以将高维的数据通过线性变化映射到低维空间，并且最大可能的保留原始数据的信息和特征。本题选用 PCA 将特征降为二维，并绘制聚类散点图。

首先对于给定的 $a \times b$ 维特征数据集 X_p ，求每个特征的均值 μ ，将每个样本的特征减去对应均值，获得去中心化特征数据集 X_d ：

$$X_d = X_p - \mu \quad (7.4)$$

接着将去中心化数据集 X_d 转换成协方差矩阵，公式如下：

$$X_l = \frac{1}{n-1} X_d^T X_d \quad (7.5)$$

对协方差矩阵计算特征值和特征向量，取最大的 i 个特征值对应的特征向量组成矩阵 W ，则降维后的数据集 X_b 为：

$$X_b = X_p W \quad (7.6)$$

4. 恒定替代弹性函数

恒定替代弹性函数(CES)是一种效用函数，可以灵活分析多种要素间的替代性和互补性^[6]。

本题将同一聚类中的作物进行 CES 函数分析，分析不同作物之间的替代性和互补性。其一般函数形式如下：

$$Y = A[\alpha X_1^\delta + (1-\alpha)X_2^\delta]^{\frac{1}{\delta}} \quad (7.7)$$

其中， Y 是产出， A 是效用规模， X_1 和 X_2 是投入要素， α 是要素的相对重要性， δ 是替代弹性参数。 δ 的范围为0到1，越接近0则两要素互补性越强，越靠近1则两要素替代性越强。

7.1.2 模型的求解

因为经济效益对于种植策略的选择至关重要，本题的特征值除了亩产量 p ，销售单价 s 和销售成本 c 外，还增加单位效益作为评价指标。单位效益 o 为：

$$o = sp - c \quad (7.8)$$

种植策略的规划目的是求最大收益，因此本题规定替代性和互补性的应对策略为：替代性高的作物组合中选择单位效益高的作物，互补性高的作物组合连续种植。

以旱地粮食为例，旱地粮食可以在山坡地平旱地和梯田上种植，分别取这三种地块类型对应的亩产量 p ，销售单价 s 、销售成本 c 作为特征。

1.K-means 聚类

先用肘部图法确定最佳聚类，根据二阶差分结果，如下列左图所示，为两类。接着以之为聚类数，用 K-means 聚类后，用成分分析将特征值映射到二维，并绘制散点图，如下列右图。最终聚类的结果见附件。

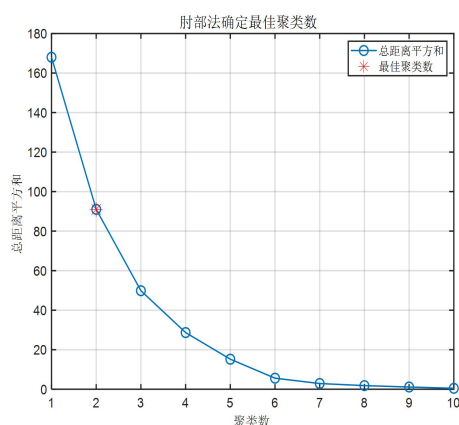


图 15 肘部图

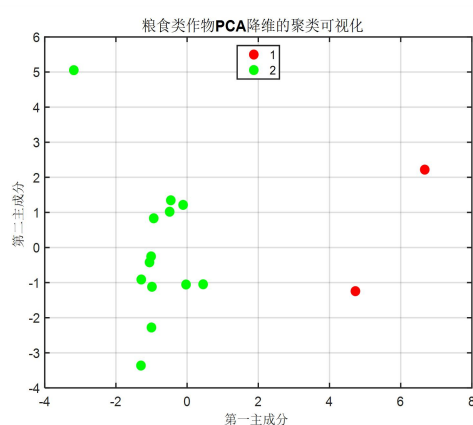


图 16 PCA 聚类可视化

2. 恒定替代弹性函数

对聚类 1 和聚类 2 中的作物分别用恒定替代弹性函数计算替代弹性参数 δ 。直接用特征平均值分析弹性参数 δ 结果集中在 0.5 左右，效果较差。进一步将特征用 PCA 映射到 2 维，即 P_{c1} , P_{c2} 。则公式中的值为：

$$X_1 = P_{c1}, X_2 = p_{c1}, Y = \frac{P_{c1} + P_{c2} + p_{c1} + p_{c1}}{4} \quad (7.9)$$

则(7.9)可进一步推进为：

$$\frac{P_{c1} + P_{c2} + p_{c1} + p_{c1}}{4} = A[\alpha P_{c1}^\delta + (1-\alpha)p_{c1}^\delta]^{\frac{1}{\delta}} \quad (7.10)$$

由于 A 表示生产规模，对作物的替代性和互补性没有影响，不妨设 A 为 1。则求解替代弹性参数 δ 可进一步转化成求单目标优化模型中获得。单目标优化模型为：

$$\begin{aligned} \min & \frac{P_{c1} + P_{c2} + p_{c1} + p_{c1}}{4} - [\alpha P_{c1}^\delta + (1-\alpha)p_{c1}^\delta]^{\frac{1}{\delta}} \\ \text{s.t.} & \begin{cases} 0 \leq \alpha \leq 1 \\ 1 \leq \delta \leq 1 \end{cases} \end{aligned} \quad (7.11)$$

设定初始 α 和 δ 均为 0.5，选用 SciPy 库中的 minimize 函数来寻找最优解，结果弹性参数 δ 依然集中在 0.5 左右，较为集中。为避免局部最优解，对初始 α 和 δ 做随机搜索，求解得出的 α 和 δ 做均值处理，并动态调整随机搜索，直到求得 α 和 δ 变化程度小于 0.01。

在聚类 1 中选取 δ 最小的三个组合，和 δ 最大的三个组合，加上聚类 2 的组合情况，数据如下图：

表 8 聚类 2 互补性替代性判断表

作物组合	α	δ	单位收益 1	单位收益 2
(1, 3)	0.680905835	0.044598754	550	2500
(3, 8)	0.236372782	0.016617381	2500	1920
(3, 11)	0.286374935	0.025807516	2500	2800
(8, 10)	0.115927835	0.983941095	1920	2890
(9, 15)	0.434063165	0.983644252	2900	1150
(10, 11)	0.66580826	0.980346102	2890	2800

(12, 13)	0.32076872	0.488080467	2550	3250
----------	------------	-------------	------	------

(12,13)为聚类 2 的组合, δ 接近 0.5, 不呈现明显替代性和互补性。在聚类 1 取的组合中, 作物组合(1,3)、(3,8)、(3,11)呈现强互补性, 在决策中应当连续种植。作物组合(8,10)、(9,15)、(10,11)呈现强替代性, 在决策中遇到这种作物组合选择时, 应当分别选择单位收益更高的 10、9、10。

在旱地粮食中, 编号 1 至 5 为豆类, 观察呈现互补性的三组作物, 发现均存在豆类, 这符合题中所给条件, 含有豆类根茎的土壤利于其他作物生长, 验证了模型的合理性和准确性。

对于非季节性蔬菜, 可以种植在水浇地第一季, 普通大棚第一季, 智慧大棚第一季和智慧大棚第二季, 分别取这四种情况对应的亩产量 p , 销售单价 s 、销售成本 c 作为特征。采用与旱地粮食相同的处理求解过程, 结果最佳聚类数为 2, 获得不同组合作物的 δ 值、 α 值和对应组合单位收益 (见附件)。选取部分数据展示, 聚类 2 中选取 δ 最小的三个组合, 和 δ 最大的三个组合, 加上聚类 1 的组合情况, 数据如下图:

表 9 聚类 1 互补性替代性判断表

作物组合	α	δ	单位收益 1	单位收益 2
(17, 19)	0.390867038	0.061430336	22210	15190
(18, 24)	0.516606339	0.073245268	11800	10365
(18, 34)	0.520298016	0.04918819	11800	19585
(17, 26)	0.643326302	0.986189147	22210	20790
(18, 21)	0.750216712	0.997286213	11800	12190
(22, 24)	0.913862391	0.996120434	36440	10365
(29, 32)	0.918532149	0.839859411	83862.5	30500

(29,32)为聚类 1 的组合, δ 为 0.839, 呈现明显替代性, 在决策中遇到这种作物组合选择时, 应当分别选择单位收益更高的 29 作物。在聚类 2 取的组合中, 作物组合(17,19)、(18,24)、(18,34)呈现强互补性, 在决策中应当连续种植。作物组合(17,26)、(18,21)、(22,24)呈现强替代性, 在决策中遇到这种作物组合选择时, 应当分别选择单位收益更高的 17、21、22。

在非季节性蔬菜中, 编号 17 至 19 为豆类, 观察呈现互补性的三组作物, 发现也均存在豆类, 这符合含有豆类根茎的土壤利于其他作物生长情况, 验证了模型的准确性和合理性。

通过对旱地粮食和非季节性蔬菜的分析, 互补性的探究体现了豆类作物的重要性, 为简化计算, 将决策中连续种植强互补性作物组合进一步抽象成增加种植豆类作物的次数。即将豆类作物轮作限制修改为至少每隔一年种植一次豆类:

$$\begin{cases} \sum_{t=2024}^{t+1} \sum_{j \in \text{豆类}} y_{ji}^t \geq 1, \forall i \in \text{单季度种植地块}, \forall t = 2024, \dots, 2028 \\ \sum_{t=2024}^{t+1} \sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^t \geq 1, \forall i \in \text{双季度种植地块}, \forall t = 2024, \dots, 2028 \end{cases} \quad (7.12)$$

7.2 相关性分析

7.2.1 斯皮尔曼相关性分析

斯皮尔曼相关系数: 又称秩相关系数, 秩次指该样本在全部样本从小到大排序中所处的位置^[7]。当变量不服从正态分布, 或者是分类、等级变量时, 可以采

用 斯皮尔曼相关分析，其公式为：

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (7.13)$$

其中， n 是样本的数量, d 代表数据之间的等级差。

斯皮尔曼相关分析通过等级排序的方式将数值转化为了等级排序，进而应用在非正态性的数据中进行相关性检验。

为探究预期销售量、销售价格和种植成本之间的相关性，本文以附件 2 的 107 个对应数据为基础，通过 spearman 相关性系数进行分析。由于这里的销售单价为不确定性参数，本文提取随机范围的中间值，最小值，最大值以及区间大小作为代替。以此做出相关性热力图并进行 P 值检验，具体结果如图，(其中 * 表示 <0.05 ，** 表示 <0.01 ，*** 表示 <0.005 ，**** 表示 <0.001)：

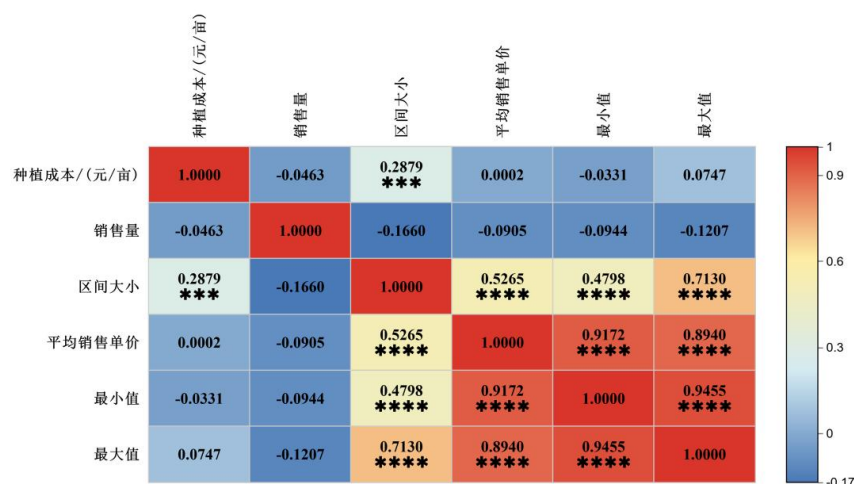


图 17 斯皮尔曼相关性热力图

1.销售量与单价、种植成本间的相关性分析

观察图像，首先观察具有较高显著性关系的区间大小与种植成本间的关系，显然，发现两者之间有 0.2879 的正相关性，可以认定种植成本的提升与单价波动范围为扩大有一定的相互促进作用。而其余情况下都呈现负相关性，除却区间大小与销售量间有 -0.1660 的负相关性，其余都只有 -0.12~-0.0002 的相关性。

2.销售量与单价、种植成本间关系函数的构建

由上一部分的分析可以得到作为销售单价代替因素之一的区间大小与种植成本和销售量都有较好的线性相关性。为了尽可能准确的探究两者之间的关系，先做出两种情况下的散点图发现做图中，显然两图中序号 11、38、40、41 的点都明显偏移其他数据点，可以判断其为异常点，本文决定剔除这些点，调整后的数据更加均匀，有助于获得更准确的关系。

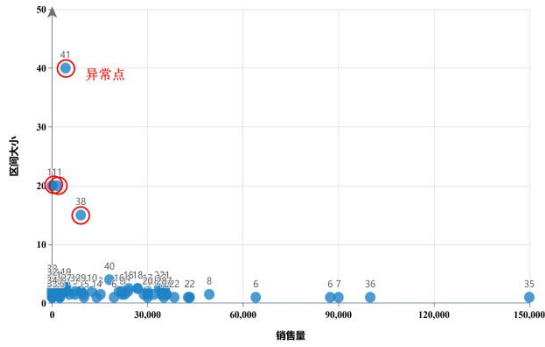


图 18 区间关于销售量的散点图

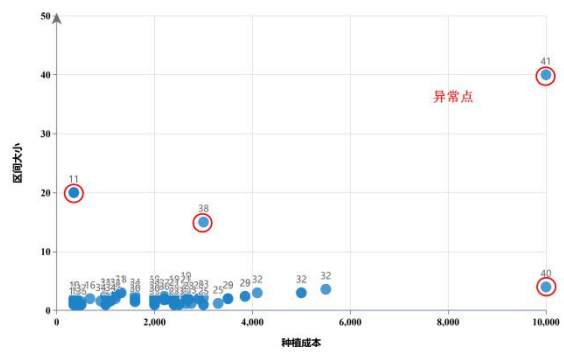
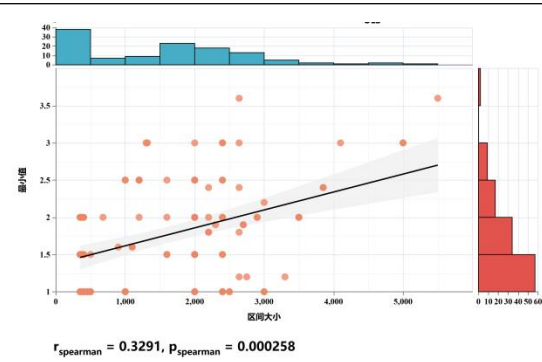
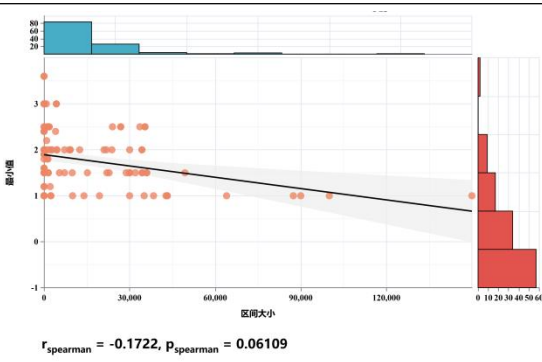


图 19 区间关于种植成本的散点图

故而，可以将处理后的数据重新通过线性回归，最后构建出两种情况下区间大小分别与种植成本、销售量间的关系式：

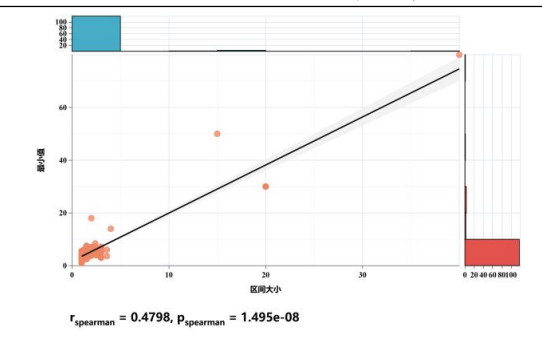
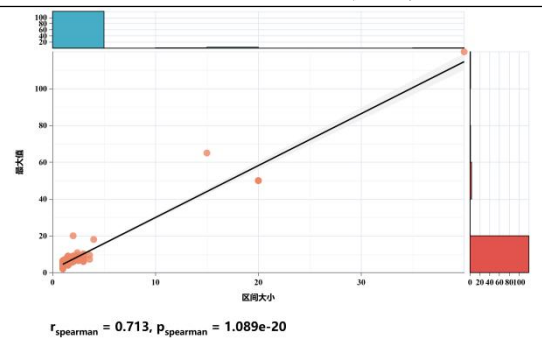
表 10 关系函数表

区间大小与种植成本	区间大小与销售量
 <p>$r_{\text{spearman}} = 0.3291, p_{\text{spearman}} = 0.000258$</p>	 <p>$r_{\text{spearman}} = -0.1722, p_{\text{spearman}} = 0.06109$</p>
$L = f_1(C) = 2.409 \times 10^{-4}C + 1.377$	$L = f_1(D) = -8.196 \times 10^{-6}D + 1.894$
L 为区间大小，C 为种植成本大小	L 为区间大小，D 为种植成本大小

3.销售单价范围转换

由图 12 可以看出，对于代替销售单价特征的各因素间都有显著性的强正相关性，大部分都超过 0.7。为了将数据转化为问题二模型中可用的数据，可以据此构建出区间大小与销售单价最值间的关系：

表 11 销售单价转换表

区间大小与区间最小值	区间大小与区间最大值
 <p>$r_{\text{spearman}} = 0.4798, p_{\text{spearman}} = 1.495e-08$</p>	 <p>$r_{\text{spearman}} = 0.713, p_{\text{spearman}} = 1.089e-20$</p>
$m = f_3(L) = 1.824L + 1.692$	$M = f_4(L) = 2.824L + 1.692$
L 为区间大小，m 为区间最小值	L 为区间大小，M 为区间最小值

此时，销售单价的区间为：

$$[m, M] = [f_3(a_1 f_1(C) + b_1 f_2(C)), f_4(a_1 f_1(C) + b_1 f_2(C))] \quad (7.14)$$

其中, a, b 分别为区间大小 L 受种植成本 C 和销售量 D 的影响权重。

7.3 综合策略模型

7.3.1 综合策略模型的建立

在 6.1 的基础上, 综合考虑各种农作物之间的可替代性与互补性, 以及预期销售量与销售价格、种植成本间的相关性, 最后得到如下模型:

$$\begin{aligned} \max \quad & \sum_{t=2024}^{2030} \left(\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^2 \left(S_{ijk}^t \cdot P_{ijk}^t \cdot D_{ijk}^{2023} - C_{ijk}^t \cdot x_{ijk}^t - g(P_{ijk}^t \cdot x_{ijk}^t - D_{ijk}^{2023}) \right) \right) \\ s.t. \quad & \begin{cases} S_{ijk}^{t+1} = L_s \cdot S_{ijk}^t \\ P_{ijk}^{t+1} = L_p \cdot P_{ijk}^t \\ D_{ijk}^{t+1} = L_d \cdot D_{ijk}^t \\ C_{ijk}^{t+1} = L_c \cdot C_{ijk}^t \\ S_{ijk}^t \in [f_3(a_1 f_1(C) + b_1 f_2(C)), f_4(a_1 f_1(C) + b_1 f_2(C))] \\ \sum_t \sum_{j \in \text{豆类}} y_{ij}^t \geq 1, \forall i \in \text{单季度种植地块}, \forall t = 2024, \dots, 2028 \\ \sum_t \sum_{j \in \text{豆类}} \sum_{k=1}^2 y_{ijk}^t \geq 1, \forall i \in \text{双季度种植地块}, \forall t = 2024, \dots, 2028 \\ \text{公式(5.5)} \sim \text{公式(5.25)} \end{cases} \end{aligned} \quad (7.15)$$

7.3.2 综合策略模型的求解

同样通过蒙特卡罗模拟可以得到 2024 部分种植策略如下(完整结果见附件):

表 12 2024 年部分种植策略

	地块名	黄豆	黑豆	红豆	绿豆	爬豆	小麦	...	榆黄菇	香菇	白灵菇	羊肚菌
第一季	A1	0	80	0	0	0	0	...	0	0	0	0
	A2	0	55	0	0	0	0	...	0	0	0	0
	A3	35	0	0	0	0	0	...	0	0	0	0
	A4	0	36	0	0	0	0	...	0	0	0	0
	A5	0	0	0	0	34	0	...	0	0	0	0
	A6	0	55	0	0	0	0	...	0	0	0	0
...
第二季	0	0	0	0	0	0	0	...	0	0.6	0	0.6
	0	0	0	0	0	0	0	...	0	0.6	0	0
	0	0	0	0	0	0	0	...	0	0	0	0
	0	0	0	0	0	0	0	...	0	0	0	0
	0	0	0	0	0	0	0	...	0	0	0	0
	0	0	0	0	0	0	0	...	0	0	0	0

7.4 对比分析

7.4.1 最优策略收益对比

本文根据问题二，问题三的最优策略，得到了 2024~2030 年内各年的对应的最大收益如下表所示：

表 13 各年最大收益表

年份	2024	2025	2026	2027	2028	2029	2030	2024
问题二	5823151	5791443	5860212	5861223	5858059	5841374	5869407	40904869
问题三	5838207	5849695	5816897	5834828	5855246	5799346	5792304	40786522

显然可以发现，相较于问题二，问题三的收益普遍低于问题二，且两者之间的收益差距随着时间的推移也在不断增大。

7.4.2 波动性对比

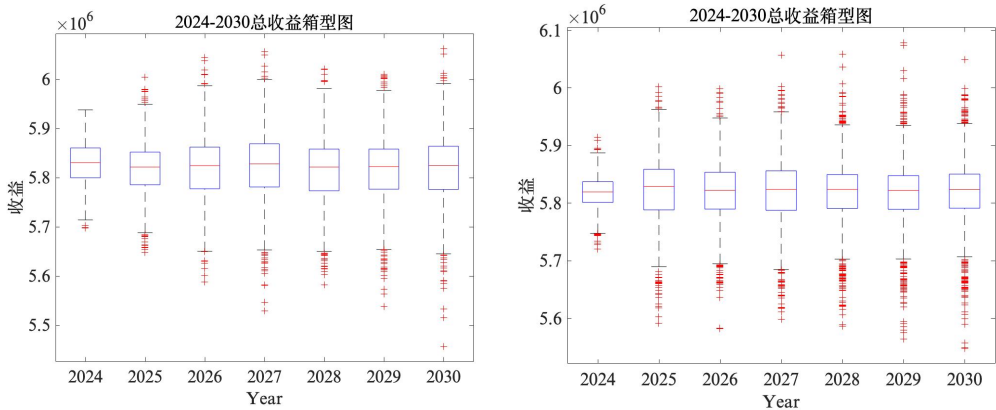


图 20 收益箱型图

为了探究问题二和问题三收益函数的稳定性，本文通过箱型图来比较波动性大小。整体而言，两者的收益分布都较为稳定且总趋势稳定。但是明显在问题三的图像中在各个年份的离群点都偏多，波动性更强，收益分布相对较为离散，而问题二的收益较为稳定。

7.5 模型检验

本题在第二问的基础上，增加了作物之间的可替代性和相关性抉择，并且量化了销量，销售成本和种植成本之间的关系，这些更深精度的定义，减少了变量的维度并增加了关系，使模型更容易适应外界因素的变化，具有更强的鲁棒性。在蒙特卡罗分析中，以经济效益为特征，建立标准化灵敏度分析模型，定量研究每个样本的标准化灵敏度，灵敏度分析模型的公式为：

$$S_i = \frac{\Delta f_i / f_0}{\Delta x_i / x_0} \quad (7.16)$$

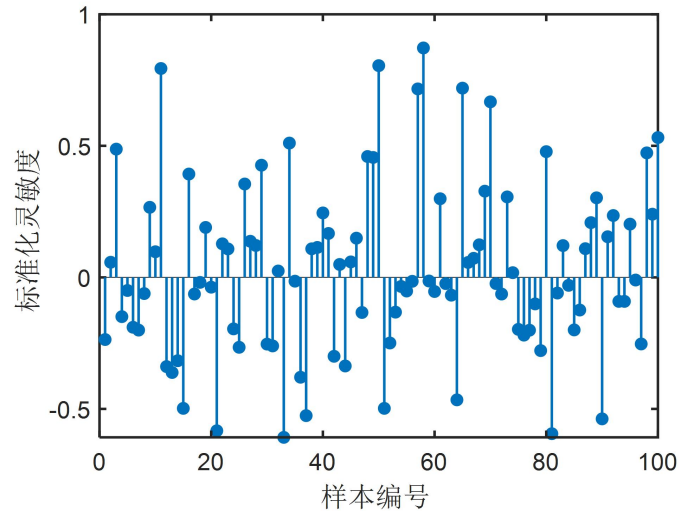


图 21 问题三鲁棒性图

观察图像可知，该模型的标准化灵敏度在 0 值上下大致呈均匀分布，且均与 0 值之间的误差在 1 以内，相较于前两问的模型，灵敏度分布更加集中，鲁棒性更强。平均标准化灵敏度为 0.028902，大于前两问模型的平均标准化灵敏度，稳健性最高，符合问题模型设想，模型合理。

八、模型的评价

8.1 模型的优势

- (1) 使用滚动优化模型，不断修正以趋于最优策略，高效准确。
- (2) 使用多种检验方式，使得模型更加准确可靠，更有说服力。
- (3) 使用蒙特卡罗模拟求解不确定性随机变量问题，极大地节省了算力与时间。

8.2 模型的劣势

- (1) 对单一问题适配性高，对于其余的问题普遍性相对较弱。

参考文献

- [1] 李化.蒙特卡洛模拟法在计算风场超越概率发电量中的应用[J/OL].南方能源设,1-9[2024-09-08].
- [2] 雷杨,赵纪峰,丁石川,等.考虑双重不确定性的区域综合能源系统多阶段滚动随机规划[J].电力系统自动化,2023,47(20):53-63.
- [3] 黄瑛杰.考虑预期不确定和非预期不确定的输配网多阶段随机规划[J].青海电力,2024,43(S1):15-21.DOI:10.15919/j.cnki.qhep.2024.S1.003.
- [4] Yongxiu He, Yuan Lyu, Yiran Che,Operational optimization of combined cooling, heat and power system based on information gap decision theory method considering probability distribution,Sustainable Energy Technologies and Assessments,Volume 51,2022,101977,ISSN 2213-1388,
- [5] 张林灵,郑焜.基于 PCA 算法和 K-means 聚类算法的医用耗材库存分类管理研究[J].中国医疗设备,2022,37(01):5-8+19.
- [6] 刘克寅,宣勇,池仁勇.企业内外部 R&D 战略的互补性与替代性研究——基于中国大中型工业企业的行业数据分析[J].研究与发展管理,2015,27(06):1-9.
- [7] Yatmar H, Ramli M I, Pasra M, et al. The Relationship Between Socio-Demographic Environment and Trip Preferences Towards Relaxation Facilities in Urban Area (Case Study: Modern Café in Makassar)[C]//IOP Conference Series: Earth and Environmental Science. IOP Publishing, 2022, 1000(1): 012016.

附录

附录 A 支撑材料文件列表

(1) 数据集


-  亩产量-1季度
-  亩产量-2季度
-  相关性分析数据
-  销售价格-1季度
-  销售价格-2季度
-  销售量-1季度
-  销售量-2季度
-  种植成本-1季度
-  种植成本-2季度
-  种植面积-1季度
-  种植面积-2季度


(2) 图


- | | |
|--|---|
|  表10 右图 |  图 11 预期销售量散点图 |
|  表10 左图 |  图 12 波动区间图 |
|  表11 右图 |  图 13 问题二鲁棒性图 |
|  表11左图 |  图 14 问题三流程图 |
|  图 1 整体流程图 |  图 15 肘部图 |
|  图 2 箱线图(1) |  图 16 PCA聚类可视化 |
|  图 2 箱线图(2) |  图 17 斯皮尔曼相关性热力图 |
|  图 2 箱线图(3) |  图 18 区间关于销售量的散点图 |
|  图 3 问题一流程图 |  图 19 区间关于种植成本的散点图 |
|  图 4 第一季各作物排列图 |  图 20 收益箱型图 |
|  图 5 第二季各作物排列图 |  图 21 问题三鲁棒性图 |
|  图 6 滚动优化原理图 |  问题1流程图原文件 |
|  图 7 正态分布检验图 |  问题2流程图原文件 |
|  图 8 蒙特卡洛模拟可视化图 |  问题3流程图原文件 |
|  图 9 鲁棒性分析图 |  整体流程图 |
|  图 10 问题二流程图 | |


(3) 表


 表 1 符号列表


 表 2 细化分类表


 表 3 作物细化分类表


 表 4 特殊作物列表


 表 5 异常值列表


 表 6 蒙特卡洛算法伪代码


 表 7 随机变量分布表


 表 8 聚类2互补性替代性判断表

 表 9 聚类1互补性替代性判断表


 表 10 关系函数表


 表 11 销售单价转换表


 表 12 2024年部分种植策略

 表 13 各年最大收益表

(4) 完整代码路径

 问题一

 问题二

 问题三

 问题四

附录 B 核心程序列表

附录 1		运行程序: MATLAB	
程序编号	1	说明	问题一(主程序)
<pre> 1. % 保障不同电脑结果可以复现 2. rng(2) 3. clc 4. clear 5. for j=1:1:1000 6. %% 参数初始化 7. N = 54; % 地块或大棚总数 8. M = 41; % 作物总数 9. T = 8; % 时间范围 (2024-2030 年, 共 7 年) 10. % 地块和作物信息 (2023 年数据) 11. A = xlsread('A.xlsx', 'E2:B55'); % 每个地块的总面积 (N x 1) 12. % P2023 >> 2023 年亩产量 (N x M x 2) 13. P2023(:, :, 1) = xlsread('P2023-1.xlsx', 'B2:BC42'); % P2023 的 (N*M, 1) 14. P2023(:, :, 2) = xlsread('P2023-2.xlsx', 'B2:BC42'); % P2023 的 (N*M, 2) 15. % C2023 = 2023 年种植成本 (N x M x 2) 16. C2023(:, :, 1) = xlsread('C2023-1.xlsx', 'B2:BC42'); 17. C2023(:, :, 2) = xlsread('C2023-2.xlsx', 'B2:BC42'); 18. % S2023 = 2023 年销售价格 (N x M x 2) 19. S2023(:, :, 1) = xlsread('S2023-1.xlsx', 'B2:BC42'); 20. S2023(:, :, 2) = xlsread('S2023-2.xlsx', 'B2:BC42'); 21. % D2023 = 2023 年市场预期销售量 (N x M x 2) 22. D2023(:, :, 1) = xlsread('D2023-1.xlsx', 'B2:BC42'); 23. D2023(:, :, 2) = xlsread('D2023-2.xlsx', 'B2:BC42'); 24. % W2023 >> 2023 年亩数 (N x M x 2) 25. W2023(:, :, 1) = xlsread('W2023-1.xlsx', 'B2:BC42'); 26. W2023(:, :, 2) = xlsread('W2023-2.xlsx', 'B2:BC42'); 27. % 初始条件 (2023 年数据用于每年的决策) 28. P = repmat(P2023, 1, 1, 1, T-1); % 每一年的亩产量与 2023 年相同 29. C = repmat(C2023, 1, 1, 1, T-1); % 每一年的种植成本与 2023 年相同 30. S = repmat(S2023, 1, 1, 1, T-1); % 每一年的销售价格与 2023 年相同 </pre>			

```

31. D = repmat(D2023, 1, 1, 1, T-1); % 每一年的市场需求量与 2023 年相同
32. W = repmat(W2023, 1, 1, 1, T-1); % 每一年的市场需求量与 2023 年相同

33. % 地块类型信息
34. single_crop_land = [1:26]; % 单季度种植地块索引（平旱地、梯田、山坡地）
35. water_land = [27:34]; % 水浇地索引（既可以种一季水稻，也可以种两季蔬菜）
36. double_crop_land = [35:54]; % 双季度种植地块索引（普通大棚和智慧大棚）

37. % 豆类作物索引
38. legume_indices_grains = [1:5]; % 粮食豆类作物索引
39. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引
40. legume_indices_all = [legume_indices_grains,
    legume_indices_vegetables]; % 所有豆类作物索引

41. x_solution = zeros(N, M, 2, T);
42. y_solution = zeros(N, M, 2, T);

43. x_solution(:, :, :, 1) = W2023; % 带入 2023 年数据进行生成

44. %% 粮食豆类作物索引
45. planting_amount = zeros(N, M); % 初始化种植量矩阵

46. single_crop_land = [1:26]; % 单季度种植地块索引（平旱地、梯田、山坡地）
47. vegetable_indices = [1:15]; % 单季度地块可以种植的粮食作物索引

48. for t = 2:T
49. planting_amount = zeros(N, M); % 初始化种植量矩阵
50. for land_idx = single_crop_land
51. total_area = A(land_idx); % 当前地块面积
52. remaining_area = total_area; % 初始化剩余面积
53. % 动态重茬限制，确保第 t 年和第 t-1 年作物不同
54. previous_year_crop = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1 年
    种植的作物索引
55. available_crops = setdiff(vegetable_indices, previous_year_crop); % 去除
    上一年种植的作物
56. expected_sales1 = sum(W2023(single_crop_land, vegetable_indices, 1), 1); %
    获取第一季度预期销售量
57. remaining_sales = expected_sales1; % 初始化剩余的预期销售量

```

```

58. % 在选择作物之前，检查该物种在上一年是否种植超过预期销售量
59. overplanting_last_year = zeros(1, M); % 记录上一年种植是否超过预期销售量
60. for crop_idx = vegetable_indices
61. planted_last_year = x_solution(land_idx, crop_idx, 1, t-1); % 上一年种植
    量
62. if planted_last_year >= expected_sales1(crop_idx)
63. overplanting_last_year(crop_idx) = 1; % 标记超过的作物
64. end
65. end
66. % 豆类轮作：只要过去两年中有一年种植过豆类，就不强制豆类种植
67. legume_rotated = false;
68. if t > 2 % 检查过去两年是否有豆类作物
69. % 遍历过去两年
70. for year_check = t-1:-1:max(1, t-2)
71. if any(x_solution(land_idx, legume_indices_grains, 1, year_check))
72. legume_rotated = true; % 过去两年中有一年种植豆类，设置标志
73. break; % 一旦找到一年种植了豆类，跳出循环
74. end
75. end
76. end
77. % 如果过去两年没有种植豆类，则强制种植豆类并占用全部面积
78. if t > 2 && ~legume_rotated
79. % 确保选择一个豆类作物并将所有面积用于种植
80. current_year_legume_choices = setdiff(legume_indices_grains,
    find(x_solution(land_idx, legume_indices_grains, 1, t)));
81. if ~isempty(current_year_legume_choices)
82. legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
83. else
84. legume_crop_idx =
    legume_indices_grains(randi(length(legume_indices_grains)));
85. end

86. % **防止豆类分配超过剩余面积**
87. planting_amount(land_idx, legume_crop_idx) = min(remaining_area,
    total_area);
88. remaining_area = 0; % 该地块的所有面积都用于豆类作物

89. else
90. % 如果豆类已经轮作或豆类不需要种植，按照原有规则进行作物选择
91. % 动态选择作物种植，随机选择，但要避免超过预期销售量的作物
92. while remaining_area > 0 && ~isempty(available_crops)
93. % 排除超过预期销售量的作物

```



```

94. available_crops = setdiff(available_crops, find(overplanting_last_year));

95. % 如果没有可用作物了, 跳出循环
96. if isempty(available_crops)
97. break;
98. end

99. % 随机选择一个可用作物
100. random_idx = randi(length(available_crops));
101. crop_idx = available_crops(random_idx);

102. expected_sales = remaining_sales(crop_idx); % 获取该作物的剩余预期销售量

103. % **根据剩余面积进行种植, 确保不会超过地块总面积**
104. if expected_sales >= 0.5 * total_area
105. planting_area = min(remaining_area, total_area); % 确保种植面积不会超过剩
    余面积
106. planting_amount(land_idx, crop_idx) = planting_area;
107. remaining_area = remaining_area - planting_area;
108. elseif expected_sales > 0 && expected_sales < 0.5 * total_area
109. planting_area = min(remaining_area, 0.5 * total_area); % 确保种植面积不会
    超过剩余面积
110. planting_amount(land_idx, crop_idx) = planting_area;
111. remaining_area = remaining_area - planting_area;
112. end

113. % 更新剩余预期销售量
114. remaining_sales(crop_idx) = remaining_sales(crop_idx) -
    planting_amount(land_idx, crop_idx);

115. % 如果当前作物的种植量超过或达到了预期销售量, 移除该作物
116. if remaining_sales(crop_idx) <= 0
117. available_crops = setdiff(available_crops, crop_idx); % 将超出预期销售量的
    作物移除
118. end

119. % 如果剩余面积已分配完, 跳出循环
120. if remaining_area <= 0
121. break;
122. end
123. end
124. end

125. % **随机分配剩余面积给其他非豆类作物, 只要前一年没有种过**
126. if remaining_area > 0

```

```

127. remaining_available_crops = setdiff(vegetable_indices,
    previous_year_crop); % 确保该作物前一年没有种植过
128. if ~isempty(remaining_available_crops)
129. random_crop_idx =
    remaining_available_crops(randi(length(remaining_available_crops))); %
    随机选择没有种植过的作物
130. else
131. random_crop_idx = vegetable_indices(randi(length(vegetable_indices))); %
    如果所有作物都种植过，随机选择一个作物
132. end
133. % **确保不会超过剩余面积**
134. planting_area = min(remaining_area, total_area);
135. planting_amount(land_idx, random_crop_idx) = planting_area; % 将剩余面积
    分配给这个作物
136. % 更新剩余面积
137. remaining_area = remaining_area - planting_amount(land_idx,
    random_crop_idx);
138. end
139. clear available_crops current_year_legume_choices previous_year_crop;
140. end
141. % 更新 x_solution 和 y_solution
142. x_solution(single_crop_land, :, 1, t) =
    planting_amount(single_crop_land, :);
143. y_solution(single_crop_land, :, 1, t) = x_solution(single_crop_land, :, 1,
    t) > 0;
144. % 第二季度不种植作物
145. x_solution(single_crop_land, :, 2, t) = 0;

146. clear legume_rotated remaining_area;
147. end

148. %
149. %% 水浇地种植方案规划
150. water_land = [27:34]; % 水浇地索引
151. vegetable_indices1 = [17:34]; % 第一季度可以种植的蔬菜索引
152. rice_index = 16; % 水稻的索引
153. vegetable_indices2 = [35, 36, 37]; % 第二季度可以种植的蔬菜索引
154. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

155. for t = 2:T
156. for land_idx = water_land
157. if x_solution(land_idx, rice_index, 1, t-1) > 0
158. % 如果前一年种植了水稻，则当前年必须种植蔬菜
159. choice = 2;

```

```

160. else
161. % 随机选择种植水稻（1）或两季蔬菜（2）
162. choice = randi([1, 2]);
163. end

164. if choice == 1
165. % 选择种植水稻
166. total_area = A(land_idx); % 获取当前地块面积
167. x_solution(land_idx, rice_index, 1, t) = total_area;
168. y_solution(land_idx, rice_index, 1, t) = 1; % 标记为种植水稻
169. % 第二季度不能种植任何作物
170. x_solution(land_idx, :, 2, t) = 0;
171. else
172. % 选择种植两季蔬菜
173. total_area = A(land_idx); % 获取当前地块面积

174. %% 第一季度种植规划
175. remaining_area = total_area; % 初始化剩余面积
176. previous_year_crop_1 = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1
    年第一季度种植的作物索引
177. previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1)); % 获取第 t-1
    年第二季度种植的作物索引
178. available_veg1 = setdiff(vegetable_indices1, [previous_year_crop_2]); %
    去除上一年第二季度种植的作物

179. % 动态重茬限制，首先确保土地有剩余面积
180. for veg_idx = available_veg1
181. if remaining_area <= 0
182. break; % 如果没有剩余面积，停止分配作物
183. end

184. % 获取到目前为止的总种植量（在前面所有土地上）
185. total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 1,
    t));

186. % 获取第一季度预期销售量
187. expected_sales_veg1 = W2023(land_idx, veg_idx, 1);

188. % 物种选择约束：检查前面土地上累计种植量是否超过预期销售量
189. if total_planted_so_far >= expected_sales_veg1
190. continue; % 跳过该作物，不进行当前年的种植
191. end

```

```

192.% 确保种植面积不超过剩余面积
193.if expected_sales_veg1 >= 0.5 * total_area
194.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
195.x_solution(land_idx, veg_idx, 1, t) = planting_area;
196.remaining_area = remaining_area - planting_area; % 更新剩余面积
197.elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area
198.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
199.x_solution(land_idx, veg_idx, 1, t) = planting_area;
200.remaining_area = remaining_area - planting_area; % 更新剩余面积
201.end
202.end

203.%% 第二季度种植规划
204.remaining_area = total_area; % 重新初始化剩余面积
205.available_veg2 = setdiff(vegetable_indices2, find(x_solution(land_idx, :,
    1, t))); % 去除本年第一季度种植的作物

206.for veg_idx = available_veg2
207.if remaining_area <= 0
208.break; % 如果没有剩余面积，停止分配作物
209.end

210.% 获取到目前为止的总种植量（在前面所有土地上）
211.total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 2,
    t));

212.% 获取第二季度预期销售量
213.expected_sales_veg2 = W2023(land_idx, veg_idx, 2);

214.% % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
215.% if total_planted_so_far >= expected_sales_veg2
216.% continue; % 跳过该作物，不进行当前年的种植
217.% end

218.% **确保种植面积不超过剩余面积**
219.if expected_sales_veg2 >= 0.5 * total_area
220.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
221.x_solution(land_idx, veg_idx, 2, t) = planting_area;
222.remaining_area = remaining_area - planting_area; % 更新剩余面积
223.elseif expected_sales_veg2 > 0 && expected_sales_veg2 < 0.5 * total_area
224.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积

```

```

225. x_solution(land_idx, veg_idx, 2, t) = planting_area;
226. remaining_area = remaining_area - planting_area; % 更新剩余面积
227. end
228. end

229. %% 检查过去两年是否种植豆类作物
230. legume_rotated = false; % 用来判断是否种植了豆类作物
231. for year_check = t-1:max(1, t-2)
232. if any(x_solution(land_idx, legume_indices_vegetables, :, year_check))
233. legume_rotated = true;
234. break;
235. end
236. end

237. if ~legume_rotated
238. % 如果过去两年内没有种植过豆类作物，优先种植豆类作物
239. % 检查当前季度是否已经种植了豆类作物
240. current_year_legume_choices = setdiff(legume_indices_vegetables,
    find(x_solution(land_idx, legume_indices_vegetables, :, t)));

241. if ~isempty(current_year_legume_choices)
242. % 如果有未种植的豆类作物，选择其中一个
243. legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
244. else
245. % 如果所有豆类作物都已经种植过，则随机选择一个
246. legume_crop_idx =
    legume_indices_vegetables(randi(length(legume_indices_vegetables)));
247. end

248. % **确保豆类的种植面积不超过总面积**
249. planting_area = min(remaining_area, total_area);
250. x_solution(land_idx, legume_crop_idx, 1, t) = planting_area; % 第一季度种
    植豆类作物
251. remaining_area = remaining_area - planting_area; % 更新剩余面积

252. % 确保第二季度清除豆类种植（如有）
253. x_solution(land_idx, legume_indices_vegetables, 2, t) = 0;
254. end
255. end

256. % 更新第一季度和第二季度的种植状态
257. y_solution(land_idx, vegetable_indices1, 1, t) = x_solution(land_idx,

```

```

    vegetable_indices1, 1, t) > 0;
258. y_solution(land_idx, vegetable_indices2, 2, t) = x_solution(land_idx,
    vegetable_indices2, 2, t) > 0;
259. end
260. end

261. %% 普通大棚的种植选择限制
262. common_greenhouse_indices = 35:50; % 普通大棚的索引
263. mushroom_indices = 38:41; % 食用菌的索引
264. vegetable_indices = 17:34; % 蔬菜的索引
265. excluded_veg_indices = 35:37; % 大白菜、白萝卜和红萝卜的索引
266. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

267. for t = 2:T
268. for land_idx = common_greenhouse_indices
269. total_area = A(land_idx); % 获取当前地块面积
270. remaining_area = total_area; % 初始化剩余面积
271. %% 第一季度只能种植蔬菜（大白菜、白萝卜和红萝卜除外）
272. % 获取前一年第二季度种植的作物索引，确保跨年度不重茬
273. previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1));
274. available_veg1 = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_crop_2]);

275. % 动态重茬处理
276. for veg_idx = available_veg1
277. % 获取到目前为止的总种植量（在前面所有土地上）
278. total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), veg_idx, 1, t));

279. expected_sales_veg1 = sum(W2023(land_idx, veg_idx, 1), 1); % 获取第一季度
    预期销售量

280. % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
281. if total_planted_so_far >= expected_sales_veg1
282. continue; % 如果超过了预期销售量，跳过该作物
283. end

284. if expected_sales_veg1 >= 0.5 * total_area
285. % 如果预期销售量大于 50%的地块面积，则种植 100%
286. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
287. x_solution(land_idx, veg_idx, 1, t) = planting_area;
288. remaining_area = remaining_area - planting_area; % 更新剩余面积
289. elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area

```

```

290.% 如果预期销售量小于 50%，则种植 50%
291.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
292.x_solution(land_idx, veg_idx, 1, t) = planting_area;
293.remaining_area = remaining_area - planting_area; % 更新剩余面积
294.end
295.if remaining_area <= 0
296.break; % 如果剩余面积不足，停止分配
297.end
298.end
299.%% 如果第一季度没有种满，继续随机选择一种蔬菜补足剩余面积
300.if remaining_area > 0
301.available_vegetables = setdiff(vegetable_indices, excluded_veg_indices);
302.while remaining_area > 0 && ~isempty(available_vegetables)
303.veg_choice =
    available_vegetables(randi(length(available_vegetables))); % 随机选择一种蔬菜
304.expected_sales_veg1 = W2023(land_idx, veg_choice, 1); % 获取蔬菜的预期销售量
305.% **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
306.total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), veg_choice, 1, t));
307.if total_planted_so_far >= expected_sales_veg1
308.available_vegetables = setdiff(available_vegetables, veg_choice); % 如果超过了预期销售量，跳过该作物
309.continue;
310.end
311.% 检查是否可以继续种植该蔬菜，且不超过预期销售量
312.if x_solution(land_idx, veg_choice, 1, t) + remaining_area <=
    expected_sales_veg1
313.x_solution(land_idx, veg_choice, 1, t) = x_solution(land_idx, veg_choice,
    1, t) + remaining_area;
314.remaining_area = 0; % 地块已满
315.else
316.additional_area = expected_sales_veg1 - x_solution(land_idx, veg_choice,
    1, t);
317.x_solution(land_idx, veg_choice, 1, t) = expected_sales_veg1;
318.remaining_area = remaining_area - additional_area;
319.available_vegetables = setdiff(available_vegetables, veg_choice);
320.end
321.end
322.end
323.% 第一季度不允许种植食用菌

```

```

324. x_solution(land_idx, mushroom_indices, 1, t) = 0;
325. %% 第二季度只能种植食用菌，动态重茬处理
326. remaining_area = total_area;
327. available_mushrooms = mushroom_indices;

328. for mush_idx = available_mushrooms
329. % 获取到目前为止的总种植量（在前面所有土地上）
330. total_planted_so_far =
        sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), mush_idx, 2, t));

331. expected_sales_mush = W2023(land_idx, mush_idx, 2); % 获取第二季度预期销售量
332. % 物种选择约束：检查前面土地上累计种植量是否超过预期销售量
333. if total_planted_so_far >= expected_sales_mush
334. continue; % 如果超过了预期销售量，跳过该作物
335. end

336. if expected_sales_mush >= 0.5 * total_area
337. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
338. x_solution(land_idx, mush_idx, 2, t) = planting_area;
339. remaining_area = remaining_area - planting_area; % 更新剩余面积
340. elseif expected_sales_mush > 0 && expected_sales_mush < 0.5 * total_area
341. planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
342. x_solution(land_idx, mush_idx, 2, t) = planting_area;
343. remaining_area = remaining_area - planting_area; % 更新剩余面积
344. end

345. if remaining_area <= 0
346. break;
347. end
348. end

349. %% 如果第二季度没有种满，随机选择一种食用菌进行补足
350. if remaining_area > 0
351. mush_choice = mushroom_indices(randi(length(mushroom_indices)));
352. x_solution(land_idx, mush_choice, 2, t) = remaining_area;
353. y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
354. end

355. % 第二季度其他作物的种植面积必须为 0
356. x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

```



```

357.%% 如果第二季度没有种满，随机选择一种食用菌进行补足
358.if remaining_area > 0
359.mush_choice = mushroom_indices(randi(length(mushroom_indices)));
360.x_solution(land_idx, mush_choice, 2, t) = remaining_area;
361.y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
362.end

363.% 第二季度其他作物的种植面积必须为0
364.x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

365.%% 检查过去两年是否种植豆类作物，优先种植豆类作物
366.legume_rotated = false;
367.if t >= 3 % 从第三年开始检查过去两年
368.for year_check = t-1:-1:max(1, t-2)
369.if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
    || ...
370.any(x_solution(land_idx, legume_indices_vegetables, 2, year_check))
371.legume_rotated = true; % 如果过去两年中的任意季度种植了豆类，设置标志
372.break; % 一旦找到豆类种植，停止检查
373.end
374.end
375.end

376.% 如果过去两年中没有种植过豆类，则优先种植豆类
377.if t >= 3 && ~legume_rotated
378.% 如果过去两年没有任何季度种植豆类作物，选择豆类作物
379.current_year_legume_choices = setdiff(legume_indices_vegetables,
    find(x_solution(land_idx, legume_indices_vegetables, 1, t)));

380.if ~isempty(current_year_legume_choices)
381.% 随机选择一个没有种植的豆类作物
382.legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
383.else
384.% 如果所有豆类作物都已经种植过，随机选择一个豆类作物
385.legume_crop_idx =
    legume_indices_vegetables(randi(length(legume_indices_vegetables)));
386.end

387.% 替换为豆类作物，清除当前第一季度的其他作物
388.x_solution(land_idx, :, 1, t) = 0; % 清除当前第一季度的作物
389.x_solution(land_idx, legume_crop_idx, 1, t) = total_area; % 使用整个地块
    种植豆类作物

```

```

390. end

391. %% 清除循环变量，防止影响下一次循环
392. clear legume_rotated、current_year_legume_choices、legume_crop_idx、
    year_check ;
393. end
394. end

395. %% 智慧大棚的种植选择限制
396. smart_greenhouse_indices = 51:54; % 智慧大棚索引
397. vegetable_indices = 17:34; % 蔬菜索引
398. excluded_veg_indices = 35:37; % 排除的大白菜、白萝卜和红萝卜
399. legume_indices_vegetables = [17, 18, 19]; % 豆类蔬菜作物索引
400. for t = 2:T
401. for land_idx = smart_greenhouse_indices
402. total_area = A(land_idx); % 获取当前地块的总面积
403. half_area = total_area / 2; % 每个蔬菜占用一半面积
404. % 获取前一年第二季度的作物，确保跨年度不重茬
405. previous_year_second_season_crop = find(x_solution(land_idx, :, 2, t-1));
406. available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_second_season_crop]);
407. %% 第一季度种植
408. if length(available_vegetables) >= 2
409. % 随机选择两个不同的蔬菜
410. veg_choices =
        available_vegetables(randperm(length(available_vegetables), 2));
411. % 分别种植两种蔬菜，各占 50%的面积
412. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
413. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
414. y_solution(land_idx, veg_choices(1), 1, t) = 1;
415. y_solution(land_idx, veg_choices(2), 1, t) = 1;
416. end

417. % 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
418. x_solution(land_idx, excluded_veg_indices, 1, t) = 0;

419. %% 第二季度种植
420. available_vegetables = setdiff(vegetable_indices,
    excluded_veg_indices); % 重新获取可种植的蔬菜
421. if length(available_vegetables) >= 2
422. % 随机选择两个不同的蔬菜
423. veg_choices =
        available_vegetables(randperm(length(available_vegetables), 2));
424. % 分别种植两种蔬菜，各占 50%的面积

```

```

425. x_solution(land_idx, veg_choices(1), 2, t) = half_area;
426. x_solution(land_idx, veg_choices(2), 2, t) = half_area;
427. y_solution(land_idx, veg_choices(1), 2, t) = 1;
428. y_solution(land_idx, veg_choices(2), 2, t) = 1;
429. end

430. % 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
431. x_solution(land_idx, excluded_veg_indices, 2, t) = 0;

432. %% 动态重茬与豆类轮作规则
433. legume_rotated = false;
434. % 从第三年开始检查过去两年是否种植过豆类
435. if t >= 3
436. for year_check = t-1:t-2
437. % 检查过去两年中的**任意一个季度**是否种植过豆类
438. if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
    || ... % 检查第一季度
439. any(x_solution(land_idx, legume_indices_vegetables, 2, year_check)) % 检
    查第二季度
440. legume_rotated = true;
441. break;
442. end
443. end
444. end

445. if t >= 3
446. if ~legume_rotated
447. % 如果过去两年内没有种植过豆类作物，优先种植豆类作物
448. current_year_legume_choices = setdiff(legume_indices_vegetables, ...
449. find(x_solution(land_idx, legume_indices_vegetables, :, t)));
450. if ~isempty(current_year_legume_choices)
451. % 随机选择一个没有种植的豆类作物
452. legume_crop_idx =
        current_year_legume_choices(randi(length(current_year_legume_choices)))
        ;
453. % 第一季度和第二季度种植豆类作物，各占 50%的面积
454. x_solution(land_idx, legume_crop_idx, 1, t) = half_area;
455. x_solution(land_idx, legume_crop_idx, 2, t) = half_area;
456. y_solution(land_idx, legume_crop_idx, 1, t) = 1;
457. y_solution(land_idx, legume_crop_idx, 2, t) = 1;
458. end
459. else

460. % 如果过去两年种植过豆类作物，则种植其他蔬菜作物

```

```

461. available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_second_season_crop]); % 排除已经种植过的作物
462. if length(available_vegetables) >= 2
463. % 随机选择两个满足重茬限制的不同蔬菜
464. veg_choices =
    available_vegetables(randperm(length(available_vegetables), 2));
465. % 分别种植两种蔬菜, 各占 50%的面积
466. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
467. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
468. y_solution(land_idx, veg_choices(1), 1, t) = 1;
469. y_solution(land_idx, veg_choices(2), 1, t) = 1;
470. end
471. end
472. end

473. %% 清除循环变量, 防止影响下一次循环
474. clear veg_choices available_vegetables current_year_legume_choices
    legume_crop_idx;
475. end
476. end

477. % F11 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4);
478. % F22 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P .*
    x_solution - D , 1), 2), 3),4);

479. x_solution1=x_solution(:, :, :, 2:8);
480. F11 = sum(sum(sum(sum(S .* P .* W , 1), 2), 3),4) - sum(sum(sum(sum(C .*
    x_solution1, 1), 2), 3),4);
481. F22 = sum(sum(sum(sum(S .* P .* W , 1), 2), 3),4) - sum(sum(sum(sum(C .*
    x_solution1, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P .* x_solution1 - D , 1),
    2), 3),4);

482. % 保存 F1 和 F2 的值
483. F1_values(j) = F11(:, :, 1);
484. F2_values(j) = F22(:, :, 1)
485. x(:, :, :, :, j) = x_solution;

486. % 找到 F1 和 F2 最大值及其对应的 x_solution
487. [~, max_F1_index] = max(F1_values);
488. [~, max_F2_index] = max(F2_values);

489. % 输出 F1 最大和 F2 最大时的种植方案

```

```

490.best_x_solution_F1 = x(:, :, :, :, max_F1_index);
491.best_x_solution_F2 = x(:, :, :, :, max_F2_index);

492.end

493.%% 保存结果到 Excel
494.filename = '情况 1_OptimizationResultsF1.xlsx'; % 定义输出 Excel 文件名
495.for t = 1:T
496.sheetName = ['Year ' num2str(t)]; % 定义工作表名
497.% 获取 N x M x 1 和 N x M x 2 的部分
498.x_solutionPart1 = best_x_solution_F1(:, :, 1, t); % N x M x 1
499.x_solutionPart2 = best_x_solution_F1(27:end, :, 2, t); % N x M x 2
500.% 拼接上下两部分
501.x_solutionCombined1 = [x_solutionPart1; x_solutionPart2]; % 先是
    xOptPart1, 在下面加上 xOptPart2
502.% 输出到 Excel 文件中
503.writematrix(x_solutionCombined1, filename, 'Sheet', sheetName);
504.end
505.disp('Final x solution has been saved to Excel file1.');
```



```

506.filename = '情况 1_OptimizationResultsF2.xlsx'; % 定义输出 Excel 文件名
507.for t = 1:T
508.sheetName = ['Year ' num2str(t)]; % 定义工作表名
509.% 获取 N x M x 1 和 N x M x 2 的部分
510.x_solutionPart11 = best_x_solution_F2(:, :, 1, t); % N x M x 1
511.x_solutionPart22 = best_x_solution_F2(27:end, :, 2, t); % N x M x 2
512.% 拼接上下两部分
513.x_solutionCombined2 = [x_solutionPart11; x_solutionPart22]; % 先是
    xOptPart1, 在下面加上 xOptPart2
514.% 输出到 Excel 文件中
515.writematrix(x_solutionCombined2, filename, 'Sheet', sheetName);
516.end
517.disp('Final x solution has been saved to Excel file.');
```



```

518.save x x
519.save best_x_solution_F1 best_x_solution_F1
520.save best_x_solution_F2 best_x_solution_F2
521.save F1_values F1_values
522.save F2_values F2_values
```

附录 2		运行程序：MATLAB	
程序编号	2	说明	问题二(主程序)
<pre> 1. % 保障不同电脑结果可以复现 2. rng(1) 3. clc 4. clear 5. for j=1:1:1000 6. %% 参数初始化 7. N = 54; % 地块或大棚总数 8. M = 41; % 作物总数 9. T = 8; % 时间范围（2024-2030 年，共 7 年） 10. % 地块和作物信息（2023 年数据） 11. A = xlsread('A.xlsx', 'E2:B55'); % 每个地块的总面积（N x 1） 12. % P2023 >> 2023 年亩产量（N x M x 2） 13. P2023(:, :, 1) = xlsread('P2023-1.xlsx', 'B2:BC42'); % P2023 的（N*M, 1） 14. P2023(:, :, 2) = xlsread('P2023-2.xlsx', 'B2:BC42'); % P2023 的（N*M, 2） 15. % C2023 = 2023 年种植成本（N x M x 2） 16. C2023(:, :, 1) = xlsread('C2023-1.xlsx', 'B2:BC42'); 17. C2023(:, :, 2) = xlsread('C2023-2.xlsx', 'B2:BC42'); 18. % S2023 = 2023 年销售价格（N x M x 2） 19. S2023(:, :, 1) = xlsread('S2023-1.xlsx', 'B2:BC42'); 20. S2023(:, :, 2) = xlsread('S2023-2.xlsx', 'B2:BC42'); 21. % D2023 = 2023 年市场预期销售量（N x M x 2） 22. D2023(:, :, 1) = xlsread('D2023-1.xlsx', 'B2:BC42'); 23. D2023(:, :, 2) = xlsread('D2023-2.xlsx', 'B2:BC42'); 24. % W2023 >> 2023 年亩数（N x M x 2） 25. W2023(:, :, 1) = xlsread('W2023-1.xlsx', 'B2:BC42'); 26. W2023(:, :, 2) = xlsread('W2023-2.xlsx', 'B2:BC42'); 27. % 初始条件（2023 年数据用于每年的决策） 28. P = repmat(P2023, 1, 1, 1, T); % 每一年的亩产量与 2023 年相同 29. C = repmat(C2023, 1, 1, 1, T); % 每一年的种植成本与 2023 年相同 30. S = repmat(S2023, 1, 1, 1, T); % 每一年的销售价格与 2023 年相同 31. D = repmat(D2023, 1, 1, 1, T); % 每一年的市场需求量与 2023 年相同 32. W = repmat(W2023, 1, 1, 1, T); % 每一年的市场需求量与 2023 年相同 </pre>			

```

33. %% 销售价格修改
34. % 设置三角分布的参数，均值为 5%，上下限为 0.8*5%和 1.2*5%
35. lower_bound = 0.8 * 0.05; % 下限 (0.8 * 5%)
36. upper_bound = 1.2 * 0.05; % 上限 (1.2 * 5%)
37. mode_value = 0.05; % 模式值 (5%)

38. % 设置正态分布的参数，均值为 3%，标准差为 1%（即下降 1%到 5%的范围）
39. mean_decline = 0.03; % 均值 3% 下降
40. std_dev = 0.01; % 标准差 1%

41. % 修改第 17 到第 41 个农作物的销售价格
42. for t = 2:T
43. % 针对第 41 个作物的固定 5%下降
44. S(:, 41, 1, t) = S(:, 41, 1, t-1) * 0.95; % 第一季度
45. S(:, 41, 2, t) = S(:, 41, 2, t-1) * 0.95; % 第二季度
46. % 针对第 30 到第 38 个作物的随机下降
47. for crop_idx = 30:38
48. % 随机生成一个下降比例，确保下降比例在 1%到 5%之间
49. decline_rate = 0.01 + (0.05 - 0.01) * rand; % 生成 1%到 5%之间的均匀分布随机数
50. % 更新第 crop_idx 个农作物的销售价格
51. S(:, crop_idx, 1, t) = S(:, crop_idx, 1, t-1) * (1 - decline_rate); % 第一季度
52. S(:, crop_idx, 2, t) = S(:, crop_idx, 2, t-1) * (1 - decline_rate); % 第二季度
53. end
54. % 针对第 17 到第 37 个蔬菜类作物的三角分布增长
55. for crop_idx = 17:37
56. % 生成三角分布的增长比例
57. r = rand; % 随机数 [0, 1]
58. if r < (mode_value - lower_bound) / (upper_bound - lower_bound)
59. growth_rate = lower_bound + sqrt(r * (upper_bound - lower_bound) * (mode_value - lower_bound));
60. else
61. growth_rate = upper_bound - sqrt((1 - r) * (upper_bound - lower_bound) * (upper_bound - mode_value));
62. end
63. % 更新第 crop_idx 个蔬菜类作物的销售价格
64. S(:, crop_idx, 1, t) = S(:, crop_idx, 1, t-1) * (1 + growth_rate); % 第一季度
65. S(:, crop_idx, 2, t) = S(:, crop_idx, 2, t-1) * (1 + growth_rate); % 第二季度
66. end
67. end

```

```

68. %% 销售量修改
69. % 设置小麦和玉米的增长率范围
70. lower_bound_growth = 0.05; % 5%
71. upper_bound_growth = 0.10; % 10%

72. % 设置其余农作物的±5%的变化范围
73. lower_bound_sales = 0.95; % -5% 下降
74. upper_bound_sales = 1.05; % +5% 增长

75. % 针对所有农作物预期销售量的修改
76. for t = 2:T
77. for crop_idx = 1:41
78. if crop_idx == 6 || crop_idx == 7 % 小麦(6)和玉米(7)
79. % 生成 5%到 10%之间的均匀分布增长率
80. growth_rate = lower_bound_growth + (upper_bound_growth -
    lower_bound_growth) * rand;
81. % 更新小麦和玉米的预期销售量
82. D(:, crop_idx, 1, t) = D(:, crop_idx, 1, t-1) * (1 + growth_rate); % 第一
    季度
83. D(:, crop_idx, 2, t) = D(:, crop_idx, 2, t-1) * (1 + growth_rate); % 第二
    季度
84. else % 其余农作物
85. % 生成±5%之间的均匀分布变化
86. sales_change_rate = lower_bound_sales + (upper_bound_sales -
    lower_bound_sales) * rand;
87. % 更新其他农作物的预期销售量
88. D(:, crop_idx, 1, t) = D(:, crop_idx, 1, t-1) * sales_change_rate; % 第一
    季度
89. D(:, crop_idx, 2, t) = D(:, crop_idx, 2, t-1) * sales_change_rate; % 第二
    季度
90. end
91. end
92. end
93. %% 亩产量修改

94. % 设置增长和下降的不同区间
95. growth_lower_bound = 1.00; % 增长下限 +5%
96. growth_upper_bound = 1.10; % 增长上限 +10%
97. decline_lower_bound = 0.90; % 下降下限 -10%
98. decline_upper_bound = 1.00; % 下降上限 -5%

```



```

99. % 针对所有农作物（1 到 41）的亩产量修改
100. for t = 2:T
101. % 随机决定该年是增长还是下降
102. is_growth = rand > 0.5; % 如果随机数大于 0.5，则为增长，否则为下降
103. for crop_idx = 1:41
104. if is_growth
105. % 生成增长比例，均匀分布在 0%到 10%之间
106. yield_change_rate = growth_lower_bound + (growth_upper_bound -
        growth_lower_bound) * rand;
107. else
108. % 生成下降比例，均匀分布在 0%到-10%之间
109. yield_change_rate = decline_lower_bound + (decline_upper_bound -
        decline_lower_bound) * rand;
110. end
111. % 更新第 crop_idx 个作物的亩产量
112. P(:, crop_idx, 1, t)= P(:, crop_idx,1, t-1) * yield_change_rate;
113. P(:, crop_idx, 2, t)= P(:, crop_idx,2, t-1) * yield_change_rate;
114. end
115. end

116. %% 种植成本
117. % 设置三角分布的参数
118. lower_bound = 0.8 * 0.05; % 下限 (0.8 * 5%)
119. upper_bound = 1.2 * 0.05; % 上限 (1.2 * 5%)
120. mode_value = 0.05; % 模式值 (5%)

121. % 针对所有农作物（1 到 41）的种植成本修改
122. for t = 2:T
123. for crop_idx = 1:41
124. % 生成三角分布的增长比例
125. r = rand; % 随机数 [0, 1]
126. if r < (mode_value - lower_bound) / (upper_bound - lower_bound)
127. cost_growth_rate = lower_bound + sqrt(r * (upper_bound - lower_bound) *
        (mode_value - lower_bound));
128. else
129. cost_growth_rate = upper_bound - sqrt((1 - r) * (upper_bound - lower_bound)
        * (upper_bound - mode_value));
130. end
131. % 更新第 crop_idx 个作物的种植成本
132. C(:, crop_idx, 1, t) = C(:, crop_idx, 1, t) * (1 + cost_growth_rate);
133. C(:, crop_idx, 2, t) = C(:, crop_idx, 2, t) * (1 + cost_growth_rate);
134. end
135. end

```

```

136.% 地块类型信息
137.single_crop_land = [1:26]; % 单季度种植地块索引（平旱地、梯田、山坡地）
138.water_land = [27:34]; % 水浇地索引（既可以种一季水稻，也可以种两季蔬菜）
139.double_crop_land = [35:54]; % 双季度种植地块索引（普通大棚和智慧大棚）

140.% 豆类作物索引
141.legume_indices_grains = [1:5]; % 粮食豆类作物索引
142.legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引
143.legume_indices_all = [legume_indices_grains,
    legume_indices_vegetables]; % 所有豆类作物索引

144.x_solution = zeros(N, M, 2, T);
145.y_solution = zeros(N, M, 2, T);

146.x_solution(:,:,1) = W2023; % 带入 2023 年数据进行生成

147.%% 粮食豆类作物索引
148.planting_amount = zeros(N, M); % 初始化种植量矩阵

149.single_crop_land = [1:26]; % 单季度种植地块索引（平旱地、梯田、山坡地）
150.vegetable_indices = [1:15]; % 单季度地块可以种植的粮食作物索引

151.for t = 2:T
152.W2023 = x_solution(:,:,t-1);
153.planting_amount = zeros(N, M); % 初始化种植量矩阵
154.for land_idx = single_crop_land
155.total_area = A(land_idx); % 当前地块面积
156.remaining_area = total_area; % 初始化剩余面积
157.% 动态重茬限制，确保第 t 年和第 t-1 年作物不同
158.previous_year_crop = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1
    年种植的作物索引
159.available_crops = setdiff(vegetable_indices, previous_year_crop); % 去除
    上一年种植的作物
160.expected_sales1 = sum(W2023(single_crop_land, vegetable_indices, 1),
    1); % 获取第一季度预期销售量
161.remaining_sales = expected_sales1; % 初始化剩余的预期销售量

162.% 在选择作物之前，检查该物种在上一年是否种植超过预期销售量
163.overplanting_last_year = zeros(1, M); % 记录上一年种植是否超过预期销售量
164.for crop_idx = vegetable_indices

```

```

165.planted_last_year = x_solution(land_idx, crop_idx, 1, t-1); % 上一年种植
    量
166.if planted_last_year >= expected_sales1(crop_idx)
167.overplanting_last_year(crop_idx) = 1; % 标记超过的作物
168.end
169.end
170.% **豆类轮作检查逻辑**：只要过去两年中有一年种植过豆类，就不强制豆类种植
171.legume_rotated = false;
172.if t > 2 % 检查过去两年是否有豆类作物
173.% 遍历过去两年
174.for year_check = t-1:-1:max(1, t-2)
175.if any(x_solution(land_idx, legume_indices_grains, 1, year_check))
176.legume_rotated = true; % 过去两年中有一年种植豆类，设置标志
177.break; % 一旦找到一年种植了豆类，跳出循环
178.end
179.end
180.end
181.% 如果过去两年没有种植豆类，则强制种植豆类并占用全部面积
182.if t > 2 && ~legume_rotated
183.% 确保选择一个豆类作物并将所有面积用于种植
184.current_year_legume_choices = setdiff(legume_indices_grains,
    find(x_solution(land_idx, legume_indices_grains, 1, t)));
185.if ~isempty(current_year_legume_choices)
186.legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
187.else
188.legume_crop_idx =
    legume_indices_grains(randi(length(legume_indices_grains)));
189.end

190.% **防止豆类分配超过剩余面积**
191.planting_amount(land_idx, legume_crop_idx) = min(remaining_area,
    total_area);
192.remaining_area = 0; % 该地块的所有面积都用于豆类作物

193.else
194.% 如果豆类已经轮作或豆类不需要种植，按照原有规则进行作物选择
195.% 动态选择作物种植，随机选择，但要避免超过预期销售量的作物
196.while remaining_area > 0 && ~isempty(available_crops)
197.% 排除超过预期销售量的作物
198.available_crops = setdiff(available_crops,
    find(overplanting_last_year));

```

```

199.% 如果没有可用作物了, 跳出循环
200.if isempty(available_crops)
201.break;
202.end

203.% 随机选择一个可用作物
204.random_idx = randi(length(available_crops));
205.crop_idx = available_crops(random_idx);

206.expected_sales = remaining_sales(crop_idx); % 获取该作物的剩余预期销售量

207.% **根据剩余面积进行种植, 确保不会超过地块总面积**
208.if expected_sales >= 0.5 * total_area
209.planting_area = min(remaining_area, total_area); % 确保种植面积不会超过剩
    余面积
210.planting_amount(land_idx, crop_idx) = planting_area;
211.remaining_area = remaining_area - planting_area;
212.elseif expected_sales > 0 && expected_sales < 0.5 * total_area
213.planting_area = min(remaining_area, 0.5 * total_area); % 确保种植面积不会
    超过剩余面积
214.planting_amount(land_idx, crop_idx) = planting_area;
215.remaining_area = remaining_area - planting_area;
216.end

217.% 更新剩余预期销售量
218.remaining_sales(crop_idx) = remaining_sales(crop_idx) -
    planting_amount(land_idx, crop_idx);

219.% 如果当前作物的种植量超过或达到了预期销售量, 移除该作物
220.if remaining_sales(crop_idx) <= 0
221.available_crops = setdiff(available_crops, crop_idx); % 将超出预期销售量
    的作物移除
222.end

223.% 如果剩余面积已分配完, 跳出循环
224.if remaining_area <= 0
225.break;
226.end
227.end
228.end

229.% **随机分配剩余面积给其他非豆类作物, 只要前一年没有种过**
230.if remaining_area > 0
231.remaining_available_crops = setdiff(vegetable_indices,
    previous_year_crop); % 确保该作物前一年没有种植过

```

```

232. if ~isempty(remaining_available_crops)
233. random_crop_idx =
        remaining_available_crops(randi(length(remaining_available_crops))); %
        随机选择没有种植过的作物
234. else
235. random_crop_idx = vegetable_indices(randi(length(vegetable_indices))); %
        如果所有作物都种植过，随机选择一个作物
236. end
237. % **确保不会超过剩余面积**
238. planting_area = min(remaining_area, total_area);
239. planting_amount(land_idx, random_crop_idx) = planting_area; % 将剩余面积
        分配给这个作物
240. % 更新剩余面积
241. remaining_area = remaining_area - planting_amount(land_idx,
        random_crop_idx);
242. end
243. clear available_crops current_year_legume_choices previous_year_crop;
244. end
245. % 更新 x_solution 和 y_solution
246. x_solution(single_crop_land, :, 1, t) =
        planting_amount(single_crop_land, :);
247. y_solution(single_crop_land, :, 1, t) = x_solution(single_crop_land, :, 1,
        t) > 0;
248. % 第二季度不种植作物
249. x_solution(single_crop_land, :, 2, t) = 0;

250. clear legume_rotated remaining_area;
251. end

252. %
253. %% 水浇地种植方案规划
254. water_land = [27:34]; % 水浇地索引
255. vegetable_indices1 = [17:34]; % 第一季度可以种植的蔬菜索引
256. rice_index = 16; % 水稻的索引
257. vegetable_indices2 = [35, 36, 37]; % 第二季度可以种植的蔬菜索引
258. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

259. for t = 2:T
260. for land_idx = water_land
261. if x_solution(land_idx, rice_index, 1, t-1) > 0
262. % 如果前一年种植了水稻，则当前年必须种植蔬菜
263. choice = 2;
264. else
265. % 随机选择种植水稻（1）或两季蔬菜（2）

```

```

266.choice = randi([1, 2]);
267.end

268.if choice == 1
269.% 选择种植水稻
270.total_area = A(land_idx); % 获取当前地块面积
271.x_solution(land_idx, rice_index, 1, t) = total_area;
272.y_solution(land_idx, rice_index, 1, t) = 1; % 标记为种植水稻
273.% 第二季度不能种植任何作物
274.x_solution(land_idx, :, 2, t) = 0;
275.else
276.% 选择种植两季蔬菜
277.total_area = A(land_idx); % 获取当前地块面积

278.%% 第一季度种植规划
279.remaining_area = total_area; % 初始化剩余面积
280.previous_year_crop_1 = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1
    年第一季度种植的作物索引
281.previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1)); % 获取第 t-1
    年第二季度种植的作物索引
282.available_veg1 = setdiff(vegetable_indices1, [previous_year_crop_2]); %
    去除上一年第二季度种植的作物

283.% 动态重茬限制，首先确保土地有剩余面积
284.for veg_idx = available_veg1
285.if remaining_area <= 0
286.break; % 如果没有剩余面积，停止分配作物
287.end

288.% 获取到目前为止的总种植量（在前面所有土地上）
289.total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 1,
    t));

290.% 获取第一季度预期销售量
291.expected_sales_veg1 = W2023(land_idx, veg_idx, 1);

292.% **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
293.if total_planted_so_far >= expected_sales_veg1
294.continue; % 跳过该作物，不进行当前年的种植
295.end

296.% **确保种植面积不超过剩余面积**
297.if expected_sales_veg1 >= 0.5 * total_area

```

```

298.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
299.x_solution(land_idx, veg_idx, 1, t) = planting_area;
300.remaining_area = remaining_area - planting_area; % 更新剩余面积
301.elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area
302.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
303.x_solution(land_idx, veg_idx, 1, t) = planting_area;
304.remaining_area = remaining_area - planting_area; % 更新剩余面积
305.end
306.end

307.%% 第二季度种植规划
308.remaining_area = total_area; % 重新初始化剩余面积
309.available_veg2 = setdiff(vegetable_indices2, find(x_solution(land_idx, :,
    1, t))); % 去除本年第一季度种植的作物

310.for veg_idx = available_veg2
311.if remaining_area <= 0
312.break; % 如果没有剩余面积, 停止分配作物
313.end

314.% 获取到目前为止的总种植量 (在前面所有土地上)
315.total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 2,
    t));

316.% 获取第二季度预期销售量
317.expected_sales_veg2 = W2023(land_idx, veg_idx, 2);

318.%% **物种选择约束**: 检查前面土地上累计种植量是否超过预期销售量
319.% if total_planted_so_far >= expected_sales_veg2
320.% continue; % 跳过该作物, 不进行当前年的种植
321.% end

322.%% **确保种植面积不超过剩余面积**
323.if expected_sales_veg2 >= 0.5 * total_area
324.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
325.x_solution(land_idx, veg_idx, 2, t) = planting_area;
326.remaining_area = remaining_area - planting_area; % 更新剩余面积
327.elseif expected_sales_veg2 > 0 && expected_sales_veg2 < 0.5 * total_area
328.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
329.x_solution(land_idx, veg_idx, 2, t) = planting_area;
330.remaining_area = remaining_area - planting_area; % 更新剩余面积

```

```

331. end
332. end

333. %% 检查过去两年是否种植豆类作物
334. legume_rotated = false; % 用来判断是否种植了豆类作物
335. for year_check = t-1:max(1, t-2)
336. if any(x_solution(land_idx, legume_indices_vegetables, :, year_check))
337. legume_rotated = true;
338. break;
339. end
340. end

341. if ~legume_rotated
342. % 如果过去两年内没有种植过豆类作物，优先种植豆类作物
343. % 检查当前季度是否已经种植了豆类作物
344. current_year_legume_choices = setdiff(legume_indices_vegetables,
    find(x_solution(land_idx, legume_indices_vegetables, :, t)));

345. if ~isempty(current_year_legume_choices)
346. % 如果有未种植的豆类作物，选择其中一个
347. legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
348. else
349. % 如果所有豆类作物都已经种植过，则随机选择一个
350. legume_crop_idx =
    legume_indices_vegetables(randi(length(legume_indices_vegetables)));
351. end

352. % **确保豆类的种植面积不超过总面积**
353. planting_area = min(remaining_area, total_area);
354. x_solution(land_idx, legume_crop_idx, 1, t) = planting_area; % 第一季度种
    植豆类作物
355. remaining_area = remaining_area - planting_area; % 更新剩余面积

356. % 确保第二季度清除豆类种植（如有）
357. x_solution(land_idx, legume_indices_vegetables, 2, t) = 0;
358. end
359. end

360. % 更新第一季度和第二季度的种植状态
361. y_solution(land_idx, vegetable_indices1, 1, t) = x_solution(land_idx,
    vegetable_indices1, 1, t) > 0;
362. y_solution(land_idx, vegetable_indices2, 2, t) = x_solution(land_idx,

```



```

        vegetable_indices2, 2, t) > 0;
363. end
364. end

365. %% 普通大棚的种植选择限制
366. common_greenhouse_indices = 35:50; % 普通大棚的索引
367. mushroom_indices = 38:41; % 食用菌的索引
368. vegetable_indices = 17:34; % 蔬菜的索引
369. excluded_veg_indices = 35:37; % 大白菜、白萝卜和红萝卜的索引
370. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

371. for t = 2:T
372. for land_idx = common_greenhouse_indices
373. total_area = A(land_idx); % 获取当前地块面积
374. remaining_area = total_area; % 初始化剩余面积
375. %% 第一季度只能种植蔬菜（大白菜、白萝卜和红萝卜除外）
376. % 获取前一年第二季度种植的作物索引，确保跨年度不重茬
377. previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1));
378. available_veg1 = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_crop_2]);

379. % 动态重茬处理
380. for veg_idx = available_veg1
381. % 获取到目前为止的总种植量（在前面所有土地上）
382. total_planted_so_far =
        sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), veg_idx, 1, t));

383. expected_sales_veg1 = sum(W2023(land_idx, veg_idx, 1), 1); % 获取第一季度
    预期销售量

384. % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
385. if total_planted_so_far >= expected_sales_veg1
386. continue; % 如果超过了预期销售量，跳过该作物
387. end

388. if expected_sales_veg1 >= 0.5 * total_area
389. % 如果预期销售量大于 50% 的地块面积，则种植 100%
390. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
391. x_solution(land_idx, veg_idx, 1, t) = planting_area;
392. remaining_area = remaining_area - planting_area; % 更新剩余面积
393. elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area
394. % 如果预期销售量小于 50%，则种植 50%
395. planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面

```

```

    积
396.x_solution(land_idx, veg_idx, 1, t) = planting_area;
397.remaining_area = remaining_area - planting_area; % 更新剩余面积
398.end
399.if remaining_area <= 0
400.break; % 如果剩余面积不足，停止分配
401.end
402.end
403.%% 如果第一季度没有种满，继续随机选择一种蔬菜补足剩余面积
404.if remaining_area > 0
405.available_vegetables = setdiff(vegetable_indices, excluded_veg_indices);
406.while remaining_area > 0 && ~isempty(available_vegetables)
407.veg_choice =
    available_vegetables(randi(length(available_vegetables))); % 随机选择一种
    蔬菜
408.expected_sales_veg1 = W2023(land_idx, veg_choice, 1); % 获取蔬菜的预期销
    售量
409.% **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
410.total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indic
    es==land_idx)-1), veg_choice, 1, t));
411.if total_planted_so_far >= expected_sales_veg1
412.available_vegetables = setdiff(available_vegetables, veg_choice); % 如果
    超过了预期销售量，跳过该作物
413.continue;
414.end
415.% 检查是否可以继续种植该蔬菜，且不超过预期销售量
416.if x_solution(land_idx, veg_choice, 1, t) + remaining_area <=
    expected_sales_veg1
417.x_solution(land_idx, veg_choice, 1, t) = x_solution(land_idx, veg_choice,
    1, t) + remaining_area;
418.remaining_area = 0; % 地块已满
419.else
420.additional_area = expected_sales_veg1 - x_solution(land_idx, veg_choice,
    1, t);
421.x_solution(land_idx, veg_choice, 1, t) = expected_sales_veg1;
422.remaining_area = remaining_area - additional_area;
423.available_vegetables = setdiff(available_vegetables, veg_choice);
424.end
425.end
426.end
427.% 第一季度不允许种植食用菌
428.x_solution(land_idx, mushroom_indices, 1, t) = 0;
429.%% 第二季度只能种植食用菌，动态重茬处理

```

```

430.remaining_area = total_area;
431.available_mushrooms = mushroom_indices;

432.for mush_idx = available_mushrooms
433.% 获取到目前为止的总种植量（在前面所有土地上）
434.total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), mush_idx, 2, t));

435.expected_sales_mush = W2023(land_idx, mush_idx, 2); % 获取第二季度预期销售量
436.% **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
437.if total_planted_so_far >= expected_sales_mush
438.continue; % 如果超过了预期销售量，跳过该作物
439.end

440.if expected_sales_mush >= 0.5 * total_area
441.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
442.x_solution(land_idx, mush_idx, 2, t) = planting_area;
443.remaining_area = remaining_area - planting_area; % 更新剩余面积
444.elseif expected_sales_mush > 0 && expected_sales_mush < 0.5 * total_area
445.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
446.x_solution(land_idx, mush_idx, 2, t) = planting_area;
447.remaining_area = remaining_area - planting_area; % 更新剩余面积
448.end

449.if remaining_area <= 0
450.break;
451.end
452.end

453.%% 如果第二季度没有种满，随机选择一种食用菌进行补足
454.if remaining_area > 0
455.mush_choice = mushroom_indices(randi(length(mushroom_indices)));
456.x_solution(land_idx, mush_choice, 2, t) = remaining_area;
457.y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
458.end

459.% 第二季度其他作物的种植面积必须为 0
460.x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

461.%% 如果第二季度没有种满，随机选择一种食用菌进行补足
462.if remaining_area > 0

```

```

463.mush_choice = mushroom_indices(randi(length(mushroom_indices)));
464.x_solution(land_idx, mush_choice, 2, t) = remaining_area;
465.y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
466.end

467.% 第二季度其他作物的种植面积必须为 0
468.x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

469.%% 检查过去两年是否种植豆类作物，优先种植豆类作物
470.legume_rotated = false;
471.if t >= 3 % 从第三年开始检查过去两年
472.for year_check = t-1:-1:max(1, t-2)
473.if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
    || ...
474.any(x_solution(land_idx, legume_indices_vegetables, 2, year_check))
475.legume_rotated = true; % 如果过去两年中的任意季度种植了豆类，设置标志
476.break; % 一旦找到豆类种植，停止检查
477.end
478.end
479.end

480.% 如果过去两年中没有种植过豆类，则优先种植豆类
481.if t >= 3 && ~legume_rotated
482.% 如果过去两年没有任何季度种植豆类作物，选择豆类作物
483.current_year_legume_choices = setdiff(legume_indices_vegetables,
    find(x_solution(land_idx, legume_indices_vegetables, 1, t)));

484.if ~isempty(current_year_legume_choices)
485.% 随机选择一个没有种植的豆类作物
486.legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
487.else
488.% 如果所有豆类作物都已经种植过，随机选择一个豆类作物
489.legume_crop_idx =
    legume_indices_vegetables(randi(length(legume_indices_vegetables)));
490.end

491.% 替换为豆类作物，清除当前第一季度的其他作物
492.x_solution(land_idx, :, 1, t) = 0; % 清除当前第一季度的作物
493.x_solution(land_idx, legume_crop_idx, 1, t) = total_area; % 使用整个地块
    种植豆类作物
494.end

```

```

495.%% 清除循环变量，防止影响下一次循环
496. clear legume_rotated、current_year_legume_choices、legume_crop_idx、
    year_check ;
497. end
498. end

499.%% 智慧大棚的种植选择限制
500. smart_greenhouse_indices = 51:54; % 智慧大棚索引
501. vegetable_indices = 17:34; % 蔬菜索引
502. excluded_veg_indices = 35:37; % 排除的大白菜、白萝卜和红萝卜
503. legume_indices_vegetables = [17, 18, 19]; % 豆类蔬菜作物索引
504. for t = 2:T
505. for land_idx = smart_greenhouse_indices
506. total_area = A(land_idx); % 获取当前地块的总面积
507. half_area = total_area / 2; % 每个蔬菜占用一半面积
508. % 获取前一年第二季度的作物，确保跨年度不重茬
509. previous_year_second_season_crop = find(x_solution(land_idx, :, 2, t-1));
510. available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_second_season_crop]);
511. %% 第一季度种植
512. if length(available_vegetables) >= 2
513. % 随机选择两个不同的蔬菜
514. veg_choices =
        available_vegetables(randperm(length(available_vegetables), 2));
515. % 分别种植两种蔬菜，各占 50%的面积
516. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
517. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
518. y_solution(land_idx, veg_choices(1), 1, t) = 1;
519. y_solution(land_idx, veg_choices(2), 1, t) = 1;
520. end

521. % 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
522. x_solution(land_idx, excluded_veg_indices, 1, t) = 0;

523. %% 第二季度种植
524. available_vegetables = setdiff(vegetable_indices,
    excluded_veg_indices); % 重新获取可种植的蔬菜
525. if length(available_vegetables) >= 2
526. % 随机选择两个不同的蔬菜
527. veg_choices =
        available_vegetables(randperm(length(available_vegetables), 2));
528. % 分别种植两种蔬菜，各占 50%的面积
529. x_solution(land_idx, veg_choices(1), 2, t) = half_area;
530. x_solution(land_idx, veg_choices(2), 2, t) = half_area;

```

```

531.y_solution(land_idx, veg_choices(1), 2, t) = 1;
532.y_solution(land_idx, veg_choices(2), 2, t) = 1;
533.end

534.% 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
535.x_solution(land_idx, excluded_veg_indices, 2, t) = 0;

536.%% 动态重茬与豆类轮作规则
537.legume_rotated = false;
538.% 从第三年开始检查过去两年是否种植过豆类
539.if t >= 3
540.for year_check = t-1:t-2
541.% 检查过去两年中的**任意一个季度**是否种植过豆类
542.if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
    || ... % 检查第一季度
543.any(x_solution(land_idx, legume_indices_vegetables, 2, year_check)) % 检
    查第二季度
544.legume_rotated = true;
545.break;
546.end
547.end
548.end

549.if t >= 3
550.if ~legume_rotated
551.% 如果过去两年内没有种植过豆类作物，优先种植豆类作物
552.current_year_legume_choices = setdiff(legume_indices_vegetables, ...
553.find(x_solution(land_idx, legume_indices_vegetables, :, t)));
554.if ~isempty(current_year_legume_choices)
555.% 随机选择一个没有种植的豆类作物
556.legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
557.% 第一季度和第二季度种植豆类作物，各占 50%的面积
558.x_solution(land_idx, legume_crop_idx, 1, t) = half_area;
559.x_solution(land_idx, legume_crop_idx, 2, t) = half_area;
560.y_solution(land_idx, legume_crop_idx, 1, t) = 1;
561.y_solution(land_idx, legume_crop_idx, 2, t) = 1;
562.end
563.else

564.% 如果过去两年种植过豆类作物，则种植其他蔬菜作物
565.available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_second_season_crop]); % 排除已经种植过的作物

```

```

566. if length(available_vegetables) >= 2
567. % 随机选择两个满足重茬限制的不同蔬菜
568. veg_choices =
    available_vegetables(randperm(length(available_vegetables), 2));
569. % 分别种植两种蔬菜，各占 50%的面积
570. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
571. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
572. y_solution(land_idx, veg_choices(1), 1, t) = 1;
573. y_solution(land_idx, veg_choices(2), 1, t) = 1;
574. end
575. end
576. end

577. %% 清除循环变量，防止影响下一次循环
578. clear veg_choices available_vegetables current_year_legume_choices
    legume_crop_idx;
579. end
580. end

581. % F11 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4);
582. % F22 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P .*
    x_solution - D , 1), 2), 3),4);
583. %
584. x_solution1=x_solution(:, :, :, 2:8);
585. D1=D(:, :, :, 2:8);
586. S1=S(:, :, :, 2:8);
587. P1=P(:, :, :, 2:8);
588. W1=W(:, :, :, 2:8);
589. C1=C(:, :, :, 2:8);
590. F11 = sum(sum(sum(sum(S1 .* P1 .* W1 , 1), 2), 3),4) - sum(sum(sum(sum(C1 .*
    x_solution1, 1), 2), 3),4);
591. F22 = sum(sum(sum(sum(S1 .* P1 .* W1 , 1), 2), 3),4) - sum(sum(sum(sum(C1 .*
    x_solution1, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P1 .* x_solution1 - D1 ,
    1), 2), 3),4);

592. % 保存 F1 和 F2 的值
593. F1_values(j) = F11(:, :, 1)
594. F2_values(j) = F22(:, :, 1);
595. x(:, :, :, j) = x_solution;

```

```

596.% 找到 F1 和 F2 最大值及其对应的 x_solution
597.[~, max_F1_index] = max(F1_values);
598.[~, max_F2_index] = max(F2_values);

599.% 输出 F1 最大和 F2 最大时的种植方案
600.best_x_solution_F1 = x(:, :, :, :, max_F1_index);
601.best_x_solution_F2 = x(:, :, :, :, max_F2_index);

602.end

603.%% 保存结果到 Excel
604.filename = '情况 1_OptimizationResultsF1.xlsx'; % 定义输出 Excel 文件名
605.for t = 1:T
606.sheetName = ['Year ' num2str(t)]; % 定义工作表名
607.% 获取 N x M x 1 和 N x M x 2 的部分
608.x_solutionPart1 = best_x_solution_F1(:, :, 1, t); % N x M x 1
609.x_solutionPart2 = best_x_solution_F1(27:end, :, 2, t); % N x M x 2
610.% 拼接上下两部分
611.x_solutionCombined1 = [x_solutionPart1; x_solutionPart2]; % 先是
    xOptPart1, 在下面加上 xOptPart2
612.% 输出到 Excel 文件中
613.writematrix(x_solutionCombined1, filename, 'Sheet', sheetName);
614.end
615.disp('Final x solution has been saved to Excel file1.');
```



```

616.filename = '情况 1_OptimizationResultsF2.xlsx'; % 定义输出 Excel 文件名
617.for t = 1:T
618.sheetName = ['Year ' num2str(t)]; % 定义工作表名
619.% 获取 N x M x 1 和 N x M x 2 的部分
620.x_solutionPart11 = best_x_solution_F2(:, :, 1, t); % N x M x 1
621.x_solutionPart22 = best_x_solution_F2(27:end, :, 2, t); % N x M x 2
622.% 拼接上下两部分
623.x_solutionCombined2 = [x_solutionPart11; x_solutionPart22]; % 先是
    xOptPart1, 在下面加上 xOptPart2
624.% 输出到 Excel 文件中
625.writematrix(x_solutionCombined2, filename, 'Sheet', sheetName);
626.end
627.disp('Final x solution has been saved to Excel file.');
```



```

628.save x x
629.save best_x_solution_F1 best_x_solution_F1
630.save best_x_solution_F2 best_x_solution_F2
```



```
631. save F1_values F1_values
632. save F2_values F2_values
```

附录 3		运行程序：MATLAB	
程序编号	3	说明	问题三(主程序)
<pre> 1. % 保障不同电脑结果可以复现 2. rng(1) 3. clc 4. clear 5. for j=1:1:1000 6. %% 参数初始化 7. N = 54; % 地块或大棚总数 8. M = 41; % 作物总数 9. T = 8; % 时间范围（2024-2030 年，共 7 年） 10. % 地块和作物信息（2023 年数据） 11. A = xlsread('A.xlsx', 'E2:B55'); % 每个地块的总面积（N x 1） 12. % P2023 >> 2023 年亩产量（N x M x 2） 13. P2023(:, :, 1) = xlsread('P2023-1.xlsx', 'B2:BC42'); % P2023 的（N*M, 1） 14. P2023(:, :, 2) = xlsread('P2023-2.xlsx', 'B2:BC42'); % P2023 的（N*M, 2） 15. % C2023 = 2023 年种植成本（N x M x 2） 16. C2023(:, :, 1) = xlsread('C2023-1.xlsx', 'B2:BC42'); 17. C2023(:, :, 2) = xlsread('C2023-2.xlsx', 'B2:BC42'); 18. % S2023 = 2023 年销售价格（N x M x 2） 19. S2023(:, :, 1) = xlsread('S2023-1.xlsx', 'B2:BC42'); 20. S2023(:, :, 2) = xlsread('S2023-2.xlsx', 'B2:BC42'); 21. % Smax2023 = 2023 年最大销售价格（N x M x 2） 22. Smax2023(:, :, 1) = xlsread('Smax2023-1.xlsx', 'B2:BC42'); 23. Smax2023(:, :, 2) = xlsread('Smax2023-2.xlsx', 'B2:BC42'); 24. % Smin2023 = 2023 年最小销售价格（N x M x 2） 25. Smin2023(:, :, 1) = xlsread('Smin2023-1.xlsx', 'B2:BC42');</pre>			

```

26. Smin2023(:, :, 2) = xlsread('Smin2023-2.xlsx', 'B2:BC42');

27. % D2023 = 2023 年市场预期销售量 (N x M x 2)
28. D2023(:, :, 1) = xlsread('D2023-1.xlsx', 'B2:BC42');
29. D2023(:, :, 2) = xlsread('D2023-2.xlsx', 'B2:BC42');

30. % W2023 >> 2023 年亩数 (N x M x 2)
31. W2023(:, :, 1) = xlsread('W2023-1.xlsx', 'B2:BC42');
32. W2023(:, :, 2) = xlsread('W2023-2.xlsx', 'B2:BC42');

33. % 初始条件 (2023 年数据用于每年的决策)
34. P = repmat(P2023, 1, 1, 1, T); % 每一年的亩产量与 2023 年相同
35. C = repmat(C2023, 1, 1, 1, T); % 每一年的种植成本与 2023 年相同
36. S = repmat(S2023, 1, 1, 1, T); % 每一年的销售价格与 2023 年相同
37. Smax = repmat(Smax2023, 1, 1, 1, T); % 每一年的销售价格与 2023 年相同
38. Smin = repmat(Smin2023, 1, 1, 1, T); % 每一年的销售价格与 2023 年相同
39. D = repmat(D2023, 1, 1, 1, T); % 每一年的市场需求量与 2023 年相同
40. W = repmat(W2023, 1, 1, 1, T); % 每一年的市场需求量与 2023 年相同

41. %% 销售量修改
42. % 设置小麦和玉米的增长率范围
43. lower_bound_growth = 0.05; % 5%
44. upper_bound_growth = 0.10; % 10%

45. % 设置其余农作物的±5%的变化范围
46. lower_bound_sales = 0.95; % -5% 下降
47. upper_bound_sales = 1.05; % +5% 增长

48. % 针对所有农作物预期销售量的修改
49. for t = 2:T
50. for crop_idx = 1:41
51. if crop_idx == 6 || crop_idx == 7 % 小麦(6)和玉米(7)
52. % 生成 5%到 10%之间的均匀分布增长率
53. growth_rate = lower_bound_growth + (upper_bound_growth -
        lower_bound_growth) * rand;
54. % 更新小麦和玉米的预期销售量
55. D(:, crop_idx, 1, t) = D(:, crop_idx, 1, t-1) * (1 + growth_rate); % 第一
        季度
56. D(:, crop_idx, 2, t) = D(:, crop_idx, 2, t-1) * (1 + growth_rate); % 第二

```

```

    季度
57. else % 其余农作物
58. % 生成±5%之间的均匀分布变化
59. sales_change_rate = lower_bound_sales + (upper_bound_sales -
    lower_bound_sales) * rand;
60. % 更新其他农作物的预期销售量
61. D(:, crop_idx, 1, t) = D(:, crop_idx, 1, t-1) * sales_change_rate; % 第一
    季度
62. D(:, crop_idx, 2, t) = D(:, crop_idx, 2, t-1) * sales_change_rate; % 第二
    季度
63. end

64. %% 根据更新后的种植面积 D 计算销售区间 L
65. for n = 1:N
66. D_current = D(n, crop_idx, 1, t); % 获取当前种植面积
67. L_D = -8.196 * 10^(-6) * D_current + 1.894; % 根据 D 计算 L

68. % 计算销售价格区间最小值 m 和最大值 M
69. m_D = 1.824 * L_D + 1.692; % 根据 L_D 计算区间最小值
70. M_D = 2.824 * L_D + 1.692; % 根据 L_D 计算区间最大值

71. % 更新 Smin 和 Smax
72. Smin(n, crop_idx, 1, t) = m_D;
73. Smax(n, crop_idx, 1, t) = M_D;
74. Smin(n, crop_idx, 2, t) = m_D;
75. Smax(n, crop_idx, 2, t) = M_D;
76. end
77. end
78. end

79. %% 亩产量修改

80. P = repmat(P2023, 1, 1, 1, T).*1.15; % 由于豆类种植间隔变小，所以产量增加
81. % 设置增长和下降的不同区间
82. growth_lower_bound = 1.00; % 增长下限 +5%
83. growth_upper_bound = 1.10; % 增长上限 +10%
84. decline_lower_bound = 0.90; % 下降下限 -10%
85. decline_upper_bound = 1.00; % 下降上限 -5%

86. % 针对所有农作物（1 到 41）的亩产量修改
87. for t = 2:T
88. % 随机决定该年是增长还是下降
89. is_growth = rand > 0.5; % 如果随机数大于 0.5，则为增长，否则为下降

```

```

90. for crop_idx = 1:41
91. if is_growth
92. % 生成增长比例，均匀分布在 0%到 10%之间
93. yield_change_rate = growth_lower_bound + (growth_upper_bound -
    growth_lower_bound) * rand;
94. else
95. % 生成下降比例，均匀分布在 0%到-10%之间
96. yield_change_rate = decline_lower_bound + (decline_upper_bound -
    decline_lower_bound) * rand;
97. end
98. % 更新第 crop_idx 个作物的亩产量
99. P(:, crop_idx, 1, t)= P(:, crop_idx,1, t-1) * yield_change_rate;
100. P(:, crop_idx, 2, t)= P(:, crop_idx,2, t-1) * yield_change_rate;
101. end
102. end

103. %% 种植成本 C 与销售区间 L 修改
104. % 设置三角分布的参数
105. lower_bound = 0.8 * 0.05; % 下限 (0.8 * 5%)
106. upper_bound = 1.2 * 0.05; % 上限 (1.2 * 5%)
107. mode_value = 0.05; % 模式值 (5%)

108. for t = 2:T
109. for crop_idx = 1:41
110. for n = 1:N
111. % 生成三角分布的增长比例 (种植成本 C)
112. r = rand; % 随机数 [0, 1]
113. if r < (mode_value - lower_bound) / (upper_bound - lower_bound)
114. cost_growth_rate = lower_bound + sqrt(r * (upper_bound - lower_bound) *
    (mode_value - lower_bound));
115. else
116. cost_growth_rate = upper_bound - sqrt((1 - r) * (upper_bound - lower_bound)
    * (upper_bound - mode_value));
117. end

118. % 更新种植成本
119. C(n, crop_idx, 1, t) = C(n, crop_idx, 1, t-1) * (1 + cost_growth_rate);
120. C(n, crop_idx, 2, t) = C(n, crop_idx, 2, t-1) * (1 + cost_growth_rate);

121. %% 第一步：根据种植成本 C 计算销售区间 L
122. L_C = 2.409 * 10^(-4) * C(n, crop_idx, 1, t) + 1.377; % 根据 C 计算 L
123. % 计算初次的销售价格区间最小值 m 和最大值 M
124. m_C = 1.824 * L_C + 1.692; % 根据 L_C 计算区间最小值
125. M_C = 2.824 * L_C + 1.692; % 根据 L_C 计算区间最大值

```

```

126.% 更新 Smin 和 Smax (根据 C 计算)
127.Smin(n, crop_idx, 1, t) = m_C;
128.Smax(n, crop_idx, 1, t) = M_C;
129.Smin(n, crop_idx, 2, t) = m_C;
130.Smax(n, crop_idx, 2, t) = M_C;

131.end
132.end
133.end

134.%% 销售价格修改

135.% 设置三角分布的参数, 均值为 5%, 上下限为 0.8*5%和 1.2*5%
136.lower_bound = 0.8 * 0.05; % 下限 (0.8 * 5%)
137.upper_bound = 1.2 * 0.05; % 上限 (1.2 * 5%)
138.mode_value = 0.05; % 模式值 (5%)

139.% 设置正态分布的参数, 均值为 3%, 标准差为 1% (即下降 1%到 5%的范围)
140.mean_decline = 0.03; % 均值 3% 下降
141.std_dev = 0.01; % 标准差 1%

142.S = (Smin+Smax)/2;

143.% 修改第 17 到第 41 个农作物的销售价格
144.for t = 2:T

145.% 针对第 41 个作物的固定 5%下降
146.S(:, 41, 1, t) = S(:, 41, 1, t-1) * 0.95; % 第一季度
147.S(:, 41, 2, t) = S(:, 41, 2, t-1) * 0.95; % 第二季度
148.% 针对第 30 到第 38 个作物的随机下降
149.for crop_idx = 30:38
150.% 随机生成一个下降比例, 确保下降比例在 1%到 5%之间
151.decline_rate = 0.01 + (0.05 - 0.01) * rand; % 生成 1%到 5%之间的均匀分布随机数
152.% 更新第 crop_idx 个农作物的销售价格
153.S(:, crop_idx, 1, t) = S(:, crop_idx, 1, t-1) * (1 - decline_rate); % 第一季度
154.S(:, crop_idx, 2, t) = S(:, crop_idx, 2, t-1) * (1 - decline_rate); % 第

```

```

    二季度
155. end
156. % 针对第 17 到第 37 个蔬菜类作物的三角分布增长
157. for crop_idx = 17:37
158. % 生成三角分布的增长比例
159. r = rand; % 随机数 [0, 1]
160. if r < (mode_value - lower_bound) / (upper_bound - lower_bound)
161. growth_rate = lower_bound + sqrt(r * (upper_bound - lower_bound) *
    (mode_value - lower_bound));
162. else
163. growth_rate = upper_bound - sqrt((1 - r) * (upper_bound - lower_bound) *
    (upper_bound - mode_value));
164. end
165. % 更新第 crop_idx 个蔬菜类作物的销售价格
166. S(:, crop_idx, 1, t) = S(:, crop_idx, 1, t-1) * (1 + growth_rate); % 第
    一季度
167. S(:, crop_idx, 2, t) = S(:, crop_idx, 2, t-1) * (1 + growth_rate); % 第
    二季度
168. end

169. end

170. %% 地块类型信息
171. single_crop_land = [1:26]; % 单季度种植地块索引 (平旱地、梯田、山坡地)
172. water_land = [27:34]; % 水浇地索引 (既可以种一季水稻, 也可以种两季蔬菜)
173. double_crop_land = [35:54]; % 双季度种植地块索引 (普通大棚和智慧大棚)

174. % 豆类作物索引
175. legume_indices_grains = [1:5]; % 粮食豆类作物索引
176. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引
177. legume_indices_all = [legume_indices_grains,
    legume_indices_vegetables]; % 所有豆类作物索引

178. x_solution = zeros(N, M, 2, T);
179. y_solution = zeros(N, M, 2, T);

180. x_solution(:, :, :, 1) = W2023; % 带入 2023 年数据进行生成

```

```

181. %% 粮食豆类作物索引
182. planting_amount = zeros(N, M); % 初始化种植量矩阵

183. single_crop_land = [1:26]; % 单季度种植地块索引（平旱地、梯田、山坡地）
184. vegetable_indices = [1:15]; % 单季度地块可以种植的粮食作物索引

185. for t = 2:T
186. W2023 = x_solution(:, :, :, t-1);

187. planting_amount = zeros(N, M); % 初始化种植量矩阵
188. for land_idx = single_crop_land
189. total_area = A(land_idx); % 当前地块面积
190. remaining_area = total_area; % 初始化剩余面积
191. % 动态重茬限制，确保第 t 年和第 t-1 年作物不同
192. previous_year_crop = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1
    年种植的作物索引
193. available_crops = setdiff(vegetable_indices, previous_year_crop); % 去除
    上一年种植的作物
194. expected_sales1 = sum(W2023(single_crop_land, vegetable_indices, 1),
    1); % 获取第一季度预期销售量
195. remaining_sales = expected_sales1; % 初始化剩余的预期销售量

196. % 在选择作物之前，检查该物种在上一年是否种植超过预期销售量
197. overplanting_last_year = zeros(1, M); % 记录上一年种植是否超过预期销售量
198. for crop_idx = vegetable_indices
199. planted_last_year = x_solution(land_idx, crop_idx, 1, t-1); % 上一年种植
    量
200. if planted_last_year >= expected_sales1(crop_idx)
201. overplanting_last_year(crop_idx) = 1; % 标记超过的作物
202. end
203. end
204. % **豆类轮作检查逻辑**：只要过去两年中有一年种植过豆类，就不强制豆类种植
205. legume_rotated = false;
206. if t > 1 % 检查过去两年是否有豆类作物
207. % 遍历过去两年
208. for year_check = t-1
209. if any(x_solution(land_idx, legume_indices_grains, 1, year_check))
210. legume_rotated = true; % 过去两年中有一年种植豆类，设置标志
211. break; % 一旦找到一年种植了豆类，跳出循环
212. end
213. end

```

```

214. end
215. % 如果过去两年没有种植豆类，则强制种植豆类并占用全部面积
216. if t > 1 && ~legume_rotated
217. % 确保选择一个豆类作物并将所有面积用于种植
218. current_year_legume_choices = setdiff(legume_indices_grains,
    find(x_solution(land_idx, legume_indices_grains, 1, t)));
219. if ~isempty(current_year_legume_choices)
220. legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
221. else
222. legume_crop_idx =
    legume_indices_grains(randi(length(legume_indices_grains)));
223. end

224. % **防止豆类分配超过剩余面积**
225. planting_amount(land_idx, legume_crop_idx) = min(remaining_area,
    total_area);
226. remaining_area = 0; % 该地块的所有面积都用于豆类作物

227. else
228. % 如果豆类已经轮作或豆类不需要种植，按照原有规则进行作物选择
229. % 动态选择作物种植，随机选择，但要避免超过预期销售量的作物
230. while remaining_area > 0 && ~isempty(available_crops)
231. % 排除超过预期销售量的作物
232. available_crops = setdiff(available_crops,
    find(overplanting_last_year));

233. % 如果没有可用作物了，跳出循环
234. if isempty(available_crops)
235. break;
236. end

237. % 随机选择一个可用作物
238. random_idx = randi(length(available_crops));
239. crop_idx = available_crops(random_idx);

240. expected_sales = remaining_sales(crop_idx); % 获取该作物的剩余预期销售量

241. % **根据剩余面积进行种植，确保不会超过地块总面积**
242. if expected_sales >= 0.5 * total_area
243. planting_area = min(remaining_area, total_area); % 确保种植面积不会超过剩
    余面积
244. planting_amount(land_idx, crop_idx) = planting_area;

```



```

245.remaining_area = remaining_area - planting_area;
246.elseif expected_sales > 0 && expected_sales < 0.5 * total_area
247.    planting_area = min(remaining_area, 0.5 * total_area); % 确保种植面积不会
    超过剩余面积
248.    planting_amount(land_idx, crop_idx) = planting_area;
249.    remaining_area = remaining_area - planting_area;
250.end

251.% 更新剩余预期销售量
252.remaining_sales(crop_idx) = remaining_sales(crop_idx) -
    planting_amount(land_idx, crop_idx);

253.% 如果当前作物的种植量超过或达到了预期销售量，移除该作物
254.if remaining_sales(crop_idx) <= 0
255.    available_crops = setdiff(available_crops, crop_idx); % 将超出预期销售量的
    作物移除
256.end

257.% 如果剩余面积已分配完，跳出循环
258.if remaining_area <= 0
259.    break;
260.end
261.end
262.end

263.% **随机分配剩余面积给其他非豆类作物，只要前一年没有种过**
264.if remaining_area > 0
265.    remaining_available_crops = setdiff(vegetable_indices,
        previous_year_crop); % 确保该作物前一年没有种植过
266.    if ~isempty(remaining_available_crops)
267.        random_crop_idx =
            remaining_available_crops(randi(length(remaining_available_crops))); %
            随机选择没有种植过的作物
268.    else
269.        random_crop_idx = vegetable_indices(randi(length(vegetable_indices))); %
            如果所有作物都种植过，随机选择一个作物
270.    end
271.% **确保不会超过剩余面积**
272.    planting_area = min(remaining_area, total_area);
273.    planting_amount(land_idx, random_crop_idx) = planting_area; % 将剩余面积
        分配给这个作物
274.% 更新剩余面积
275.    remaining_area = remaining_area - planting_amount(land_idx,
        random_crop_idx);
276.end

```

```

277. clear available_crops current_year_legume_choices previous_year_crop;
278. end
279. % 更新 x_solution 和 y_solution
280. x_solution(single_crop_land, :, 1, t) =
    planting_amount(single_crop_land, :);
281. y_solution(single_crop_land, :, 1, t) = x_solution(single_crop_land, :, 1,
    t) > 0;
282. % 第二季度不种植作物
283. x_solution(single_crop_land, :, 2, t) = 0;

284. clear legume_rotated remaining_area;
285. end

286. %
287. %% 水浇地种植方案规划
288. water_land = [27:34]; % 水浇地索引
289. vegetable_indices1 = [17:34]; % 第一季度可以种植的蔬菜索引
290. rice_index = 16; % 水稻的索引
291. vegetable_indices2 = [35, 36, 37]; % 第二季度可以种植的蔬菜索引
292. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

293. for t = 2:T
294. for land_idx = water_land
295. if x_solution(land_idx, rice_index, 1, t-1) > 0
296. % 如果前一年种植了水稻，则当前年必须种植蔬菜
297. choice = 2;
298. else
299. % 随机选择种植水稻（1）或两季蔬菜（2）
300. choice = randi([1, 2]);
301. end

302. if choice == 1
303. % 选择种植水稻
304. total_area = A(land_idx); % 获取当前地块面积
305. x_solution(land_idx, rice_index, 1, t) = total_area;
306. y_solution(land_idx, rice_index, 1, t) = 1; % 标记为种植水稻
307. % 第二季度不能种植任何作物
308. x_solution(land_idx, :, 2, t) = 0;
309. else
310. % 选择种植两季蔬菜
311. total_area = A(land_idx); % 获取当前地块面积

312. %% 第一季度种植规划
313. remaining_area = total_area; % 初始化剩余面积

```

```

314. previous_year_crop_1 = find(x_solution(land_idx, :, 1, t-1)); % 获取第 t-1
    年第一季度种植的作物索引
315. previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1)); % 获取第 t-1
    年第二季度种植的作物索引
316. available_veg1 = setdiff(vegetable_indices1, [previous_year_crop_2]); %
    去除上一年第二季度种植的作物

317. % 动态重茬限制，首先确保土地有剩余面积
318. for veg_idx = available_veg1
319. if remaining_area <= 0
320. break; % 如果没有剩余面积，停止分配作物
321. end

322. % 获取到目前为止的总种植量（在前面所有土地上）
323. total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 1,
    t));

324. % 获取第一季度预期销售量
325. expected_sales_veg1 = W2023(land_idx, veg_idx, 1);

326. % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
327. if total_planted_so_far >= expected_sales_veg1
328. continue; % 跳过该作物，不进行当前年的种植
329. end

330. % **确保种植面积不超过剩余面积**
331. if expected_sales_veg1 >= 0.5 * total_area
332. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
333. x_solution(land_idx, veg_idx, 1, t) = planting_area;
334. remaining_area = remaining_area - planting_area; % 更新剩余面积
335. elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area
336. planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面
    积
337. x_solution(land_idx, veg_idx, 1, t) = planting_area;
338. remaining_area = remaining_area - planting_area; % 更新剩余面积
339. end
340. end

341. %% 第二季度种植规划
342. remaining_area = total_area; % 重新初始化剩余面积
343. available_veg2 = setdiff(vegetable_indices2, find(x_solution(land_idx, :,
    1, t))); % 去除本年第一季度种植的作物

```

```

344. for veg_idx = available_veg2
345. if remaining_area <= 0
346. break; % 如果没有剩余面积，停止分配作物
347. end

348. % 获取到目前为止的总种植量（在前面所有土地上）
349. total_planted_so_far =
    sum(x_solution(water_land(1:find(water_land==land_idx)-1), veg_idx, 2,
    t));

350. % 获取第二季度预期销售量
351. expected_sales_veg2 = W2023(land_idx, veg_idx, 2);

352. % % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
353. % if total_planted_so_far >= expected_sales_veg2
354. % continue; % 跳过该作物，不进行当前年的种植
355. % end

356. % **确保种植面积不超过剩余面积**
357. if expected_sales_veg2 >= 0.5 * total_area
358. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
359. x_solution(land_idx, veg_idx, 2, t) = planting_area;
360. remaining_area = remaining_area - planting_area; % 更新剩余面积
361. elseif expected_sales_veg2 > 0 && expected_sales_veg2 < 0.5 * total_area
362. planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面
    积
363. x_solution(land_idx, veg_idx, 2, t) = planting_area;
364. remaining_area = remaining_area - planting_area; % 更新剩余面积
365. end
366. end

367. %% 检查过去两年是否种植豆类作物
368. legume_rotated = false; % 用来判断是否种植了豆类作物
369. for year_check = t-1
370. if any(x_solution(land_idx, legume_indices_vegetables, :, year_check))
371. legume_rotated = true;
372. break;
373. end
374. end

375. if ~legume_rotated
376. % 如果过去两年内没有种植过豆类作物，优先种植豆类作物
377. % 检查当前季度是否已经种植了豆类作物
378. current_year_legume_choices = setdiff(legume_indices_vegetables,

```

```

        find(x_solution(land_idx, legume_indices_vegetables, :, t)));

379. if ~isempty(current_year_legume_choices)
380. % 如果有未种植的豆类作物，选择其中一个
381. legume_crop_idx =
        current_year_legume_choices(randi(length(current_year_legume_choices)))
        ;
382. else
383. % 如果所有豆类作物都已经种植过，则随机选择一个
384. legume_crop_idx =
        legume_indices_vegetables(randi(length(legume_indices_vegetables)));
385. end

386. % **确保豆类的种植面积不超过总面积**
387. planting_area = min(remaining_area, total_area);
388. x_solution(land_idx, legume_crop_idx, 1, t) = planting_area; % 第一季度种
        植豆类作物
389. remaining_area = remaining_area - planting_area; % 更新剩余面积

390. % 确保第二季度清除豆类种植（如有）
391. x_solution(land_idx, legume_indices_vegetables, 2, t) = 0;
392. end
393. end

394. % 更新第一季度和第二季度的种植状态
395. y_solution(land_idx, vegetable_indices1, 1, t) = x_solution(land_idx,
        vegetable_indices1, 1, t) > 0;
396. y_solution(land_idx, vegetable_indices2, 2, t) = x_solution(land_idx,
        vegetable_indices2, 2, t) > 0;
397. end
398. end

399. %% 普通大棚的种植选择限制
400. common_greenhouse_indices = 35:50; % 普通大棚的索引
401. mushroom_indices = 38:41; % 食用菌的索引
402. vegetable_indices = 17:34; % 蔬菜的索引
403. excluded_veg_indices = 35:37; % 大白菜、白萝卜和红萝卜的索引
404. legume_indices_vegetables = [17, 18, 19]; % 蔬菜豆类作物索引

405. for t = 2:T
406. for land_idx = common_greenhouse_indices
407. total_area = A(land_idx); % 获取当前地块面积
408. remaining_area = total_area; % 初始化剩余面积
409. %% 第一季度只能种植蔬菜（大白菜、白萝卜和红萝卜除外）

```

```

410.% 获取前一年第二季度种植的作物索引，确保跨年度不重茬
411.previous_year_crop_2 = find(x_solution(land_idx, :, 2, t-1));
412.available_veg1 = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_crop_2]);

413.% 动态重茬处理
414.for veg_idx = available_veg1
415.% 获取到目前为止的总种植量（在前面所有土地上）
416.total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), veg_idx, 1, t));

417.expected_sales_veg1 = sum(W2023(land_idx, veg_idx, 1), 1); % 获取第一季度
    预期销售量

418.% **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
419.if total_planted_so_far >= expected_sales_veg1
420.continue; % 如果超过了预期销售量，跳过该作物
421.end

422.if expected_sales_veg1 >= 0.5 * total_area
423.% 如果预期销售量大于 50%的地块面积，则种植 100%
424.planting_area = min(remaining_area, total_area); % 确保不会超出总面积
425.x_solution(land_idx, veg_idx, 1, t) = planting_area;
426.remaining_area = remaining_area - planting_area; % 更新剩余面积
427.elseif expected_sales_veg1 > 0 && expected_sales_veg1 < 0.5 * total_area
428.% 如果预期销售量小于 50%，则种植 50%
429.planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面积
430.x_solution(land_idx, veg_idx, 1, t) = planting_area;
431.remaining_area = remaining_area - planting_area; % 更新剩余面积
432.end
433.if remaining_area <= 0
434.break; % 如果剩余面积不足，停止分配
435.end
436.end

437.%% 如果第一季度没有种满，继续随机选择一种蔬菜补足剩余面积
438.if remaining_area > 0
439.available_vegetables = setdiff(vegetable_indices, excluded_veg_indices);
440.while remaining_area > 0 && ~isempty(available_vegetables)
441.veg_choice =
    available_vegetables(randi(length(available_vegetables))); % 随机选择一种
    蔬菜
442.expected_sales_veg1 = W2023(land_idx, veg_choice, 1); % 获取蔬菜的预期销

```

```

    销售量
443. % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
444. total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), veg_choice, 1, t));
445. if total_planted_so_far >= expected_sales_veg1
446. available_vegetables = setdiff(available_vegetables, veg_choice); % 如果
    超过了预期销售量，跳过该作物
447. continue;
448. end
449. % 检查是否可以继续种植该蔬菜，且不超过预期销售量
450. if x_solution(land_idx, veg_choice, 1, t) + remaining_area <=
    expected_sales_veg1
451. x_solution(land_idx, veg_choice, 1, t) = x_solution(land_idx, veg_choice,
    1, t) + remaining_area;
452. remaining_area = 0; % 地块已满
453. else
454. additional_area = expected_sales_veg1 - x_solution(land_idx, veg_choice,
    1, t);
455. x_solution(land_idx, veg_choice, 1, t) = expected_sales_veg1;
456. remaining_area = remaining_area - additional_area;
457. available_vegetables = setdiff(available_vegetables, veg_choice);
458. end
459. end
460. end
461. % 第一季度不允许种植食用菌
462. x_solution(land_idx, mushroom_indices, 1, t) = 0;
463. %% 第二季度只能种植食用菌，动态重茬处理
464. remaining_area = total_area;
465. available_mushrooms = mushroom_indices;

466. for mush_idx = available_mushrooms
467. % 获取到目前为止的总种植量（在前面所有土地上）
468. total_planted_so_far =
    sum(x_solution(common_greenhouse_indices(1:find(common_greenhouse_indices==land_idx)-1), mush_idx, 2, t));

469. expected_sales_mush = W2023(land_idx, mush_idx, 2); % 获取第二季度预期销
    销售量
470. % **物种选择约束**：检查前面土地上累计种植量是否超过预期销售量
471. if total_planted_so_far >= expected_sales_mush
472. continue; % 如果超过了预期销售量，跳过该作物
473. end

```

```

474. if expected_sales_mush >= 0.5 * total_area
475. planting_area = min(remaining_area, total_area); % 确保不会超出总面积
476. x_solution(land_idx, mush_idx, 2, t) = planting_area;
477. remaining_area = remaining_area - planting_area; % 更新剩余面积
478. elseif expected_sales_mush > 0 && expected_sales_mush < 0.5 * total_area
479. planting_area = min(remaining_area, 0.5 * total_area); % 确保不会超出总面
    积
480. x_solution(land_idx, mush_idx, 2, t) = planting_area;
481. remaining_area = remaining_area - planting_area; % 更新剩余面积
482. end

483. if remaining_area <= 0
484. break;
485. end
486. end

487. %% 如果第二季度没有种满，随机选择一种食用菌进行补足
488. if remaining_area > 0
489. mush_choice = mushroom_indices(randi(length(mushroom_indices)));
490. x_solution(land_idx, mush_choice, 2, t) = remaining_area;
491. y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
492. end

493. % 第二季度其他作物的种植面积必须为 0
494. x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

495. %% 如果第二季度没有种满，随机选择一种食用菌进行补足
496. if remaining_area > 0
497. mush_choice = mushroom_indices(randi(length(mushroom_indices)));
498. x_solution(land_idx, mush_choice, 2, t) = remaining_area;
499. y_solution(land_idx, mush_choice, 2, t) = 1; % 更新种植状态
500. end

501. % 第二季度其他作物的种植面积必须为 0
502. x_solution(land_idx, setdiff(1:M, mushroom_indices), 2, t) = 0;

503. %% 检查过去两年是否种植豆类作物，优先种植豆类作物
504. legume_rotated = false;
505. if t >= 2 % 从第三年开始检查过去两年
506. for year_check = t-1
507. if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
        || ...
508. any(x_solution(land_idx, legume_indices_vegetables, 2, year_check))
509. legume_rotated = true; % 如果过去两年中的任意季度种植了豆类，设置标志

```



```

510. break; % 一旦找到豆类种植，停止检查
511. end
512. end
513. end

514. % 如果过去两年中没有种植过豆类，则优先种植豆类
515. if t >= 2 && ~legume_rotated
516. % 如果过去两年没有任何季度种植豆类作物，选择豆类作物
517. current_year_legume_choices = setdiff(legume_indices_vegetables,
    find(x_solution(land_idx, legume_indices_vegetables, 1, t)));

518. if ~isempty(current_year_legume_choices)
519. % 随机选择一个没有种植的豆类作物
520. legume_crop_idx =
    current_year_legume_choices(randi(length(current_year_legume_choices)))
    ;
521. else
522. % 如果所有豆类作物都已经种植过，随机选择一个豆类作物
523. legume_crop_idx =
    legume_indices_vegetables(randi(length(legume_indices_vegetables)));
524. end

525. % 替换为豆类作物，清除当前第一季度的其他作物
526. x_solution(land_idx, :, 1, t) = 0; % 清除当前第一季度的作物
527. x_solution(land_idx, legume_crop_idx, 1, t) = total_area; % 使用整个地块
    种植豆类作物
528. end

529. %% 清除循环变量，防止影响下一次循环
530. clear legume_rotated、current_year_legume_choices、legume_crop_idx、
    year_check ;
531. end
532. end

533. %% 智慧大棚的种植选择限制
534. smart_greenhouse_indices = 51:54; % 智慧大棚索引
535. vegetable_indices = 17:34; % 蔬菜索引
536. excluded_veg_indices = 35:37; % 排除的大白菜、白萝卜和红萝卜
537. legume_indices_vegetables = [17, 18, 19]; % 豆类蔬菜作物索引
538. for t = 2:T
539. for land_idx = smart_greenhouse_indices
540. total_area = A(land_idx); % 获取当前地块的总面积
541. half_area = total_area / 2; % 每个蔬菜占用一半面积
542. % 获取前一年第二季度的作物，确保跨年度不重茬

```

```

543. previous_year_second_season_crop = find(x_solution(land_idx, :, 2, t-1));
544. available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
    previous_year_second_season_crop]);
545. %% 第一季度种植
546. if length(available_vegetables) >= 2
547. % 随机选择两个不同的蔬菜
548. veg_choices =
    available_vegetables(randperm(length(available_vegetables), 2));
549. % 分别种植两种蔬菜, 各占 50%的面积
550. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
551. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
552. y_solution(land_idx, veg_choices(1), 1, t) = 1;
553. y_solution(land_idx, veg_choices(2), 1, t) = 1;
554. end

555. % 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
556. x_solution(land_idx, excluded_veg_indices, 1, t) = 0;

557. %% 第二季度种植
558. available_vegetables = setdiff(vegetable_indices,
    excluded_veg_indices); % 重新获取可种植的蔬菜
559. if length(available_vegetables) >= 2
560. % 随机选择两个不同的蔬菜
561. veg_choices =
    available_vegetables(randperm(length(available_vegetables), 2));
562. % 分别种植两种蔬菜, 各占 50%的面积
563. x_solution(land_idx, veg_choices(1), 2, t) = half_area;
564. x_solution(land_idx, veg_choices(2), 2, t) = half_area;
565. y_solution(land_idx, veg_choices(1), 2, t) = 1;
566. y_solution(land_idx, veg_choices(2), 2, t) = 1;
567. end

568. % 确保大白菜、白萝卜和红萝卜不在智慧大棚的两季中种植
569. x_solution(land_idx, excluded_veg_indices, 2, t) = 0;

570. %% 动态重茬与豆类轮作规则
571. legume_rotated = false;
572. % 从第三年开始检查过去两年是否种植过豆类
573. if t >= 2
574. for year_check = t-1
575. % 检查过去两年中的**任意一个季度**是否种植过豆类
576. if any(x_solution(land_idx, legume_indices_vegetables, 1, year_check))
    || ... % 检查第一季度
577. any(x_solution(land_idx, legume_indices_vegetables, 2, year_check)) % 检

```

```

    查第二季度
578. legume_rotated = true;
579. break;
580. end
581. end
582. end

583. if t >= 2
584. if ~legume_rotated
585. % 如果过去两年内没有种植过豆类作物，优先种植豆类作物
586. current_year_legume_choices = setdiff(legume_indices_vegetables, ...
587. find(x_solution(land_idx, legume_indices_vegetables, :, t)));
588. if ~isempty(current_year_legume_choices)
589. % 随机选择一个没有种植的豆类作物
590. legume_crop_idx =
        current_year_legume_choices(randi(length(current_year_legume_choices)))
        ;
591. % 第一季度和第二季度种植豆类作物，各占 50%的面积
592. x_solution(land_idx, legume_crop_idx, 1, t) = half_area;
593. x_solution(land_idx, legume_crop_idx, 2, t) = half_area;
594. y_solution(land_idx, legume_crop_idx, 1, t) = 1;
595. y_solution(land_idx, legume_crop_idx, 2, t) = 1;
596. end
597. else

598. % 如果过去两年种植过豆类作物，则种植其他蔬菜作物
599. available_vegetables = setdiff(vegetable_indices, [excluded_veg_indices,
        previous_year_second_season_crop]); % 排除已经种植过的作物
600. if length(available_vegetables) >= 2
601. % 随机选择两个满足重茬限制的不同蔬菜
602. veg_choices =
        available_vegetables(randperm(length(available_vegetables), 2));
603. % 分别种植两种蔬菜，各占 50%的面积
604. x_solution(land_idx, veg_choices(1), 1, t) = half_area;
605. x_solution(land_idx, veg_choices(2), 1, t) = half_area;
606. y_solution(land_idx, veg_choices(1), 1, t) = 1;
607. y_solution(land_idx, veg_choices(2), 1, t) = 1;
608. end
609. end
610. end

611. %% 清除循环变量，防止影响下一次循环
612. clear veg_choices available_vegetables current_year_legume_choices
        legume_crop_idx;

```

```

613. end
614. end

615. % F11 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4);
616. % F22 = sum(sum(sum(sum(S .* P .* x_solution , 1), 2), 3),4) -
    sum(sum(sum(sum(C .* x_solution, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P .*
    x_solution - D , 1), 2), 3),4);
617. %
618. x_solution1=x_solution(:, :, :, 2:8);

619. D1=D(:, :, :, 2:8);
620. S1=S(:, :, :, 2:8);
621. P1=P(:, :, :, 2:8);
622. W1=W(:, :, :, 2:8);
623. C1=C(:, :, :, 2:8);

624. F11 = sum(sum(sum(sum(S1 .* P1 .* W1 , 1), 2), 3),4) - sum(sum(sum(sum(C1 .*
    x_solution1, 1), 2), 3),4);
625. F22 = sum(sum(sum(sum(S1 .* P1 .* W1 , 1), 2), 3),4) - sum(sum(sum(sum(C1 .*
    x_solution1, 1), 2), 3),4) + 0.5*sum(sum(sum(sum(P1 .* x_solution1 - D1 ,
    1), 2), 3),4);

626. % 保存 F1 和 F2 的值
627. F1_values(j) = F11(:, :, 1);
628. F2_values(j) = F22(:, :, 1);
629. x(:, :, :, j) = x_solution;

630. % 找到 F1 和 F2 最大值及其对应的 x_solution
631. [~, max_F1_index] = max(F1_values);
632. [~, max_F2_index] = max(F2_values);

633. % 输出 F1 最大和 F2 最大时的种植方案
634. best_x_solution_F1 = x(:, :, :, max_F1_index);
635. best_x_solution_F2 = x(:, :, :, max_F2_index);

636. end

637. %% 保存结果到 Excel

```

```

638.filename = '情况 1_OptimizationResultsF1.xlsx'; % 定义输出 Excel 文件名
639.for t = 1:T
640.sheetName = ['Year ' num2str(t)]; % 定义工作表名
641.% 获取 N x M x 1 和 N x M x 2 的部分
642.x_solutionPart1 = best_x_solution_F1(:, :, 1, t); % N x M x 1
643.x_solutionPart2 = best_x_solution_F1(27:end, :, 2, t); % N x M x 2
644.% 拼接上下两部分
645.x_solutionCombined1 = [x_solutionPart1; x_solutionPart2]; % 先是
    xOptPart1, 在下面加上 xOptPart2
646.% 输出到 Excel 文件中
647.writematrix(x_solutionCombined1, filename, 'Sheet', sheetName);
648.end
649.disp('Final x solution has been saved to Excel file1.');
```



```

650.filename = '情况 1_OptimizationResultsF2.xlsx'; % 定义输出 Excel 文件名
651.for t = 1:T
652.sheetName = ['Year ' num2str(t)]; % 定义工作表名
653.% 获取 N x M x 1 和 N x M x 2 的部分
654.x_solutionPart11 = best_x_solution_F2(:, :, 1, t); % N x M x 1
655.x_solutionPart22 = best_x_solution_F2(27:end, :, 2, t); % N x M x 2
656.% 拼接上下两部分
657.x_solutionCombined2 = [x_solutionPart11; x_solutionPart22]; % 先是
    xOptPart1, 在下面加上 xOptPart2
658.% 输出到 Excel 文件中
659.writematrix(x_solutionCombined2, filename, 'Sheet', sheetName);
660.end
661.disp('Final x solution has been saved to Excel file.');
```



```

662.save x x
663.save best_x_solution_F1 best_x_solution_F1
664.save best_x_solution_F2 best_x_solution_F2
665.save F1_values F1_values
666.save F2_values F2_values
```