

基于多约束线性规划模型的农业种植结构优化与风险管理研究

摘 要

在农作物销售市场的运营中，可靠的市场需求及风险分析，是确定农作物种植类型和利用好土地资源的重要信息。在受限的农作物销售空间中获得的合理的种植策略一直是研究的焦点。本文针对不同农作物品类建立线性规划模型，解决了农作物种植决策问题，以达到最大收益。

针对数据预处理，首先，对附件 2 中的“2023 年的种植情况与统计数据”进行了缺失值检验，发现文件中不存在缺失值；其次，在附件 2 中存在 4 个异常值，为了确保农作物数据的完整性，将这些缺失农作物对应的种植成本及亩产量全部填充为 0。

针对问题一，我们旨在探讨在两种不同销售策略下的最优种植方案。首先，我们将每种销售策略分为两种情况进行讨论：一是总产量小于或等于预期销售量的情况，二是总产量大于预期销售量的情况。接下来，我们针对每种情况，分别确定决策变量，将约束条件进行数学化处理。这些约束条件涵盖了种植面积、亩产量、种植成本、预期销售量等多方面因素，具体可见公式(13)–(27)。在此基础上，我们描述了目标函数，如公式(4)、(7)、(10)及(13)所示，构建了线性规划模型。该模型将决策变量、约束条件和目标函数有机地结合在一起，使我们能够在统一的理论体系下，用内点法对最优种植方案进行求解，最优种植方案结果详见附件。

针对问题二，对于综合考虑农作物市场因素的不确定性后的最优种植规划问题。在深入分析问题一的基础上，着手构建一个新的**多约束线性规划模型**，以各种农作物的销售价格、种植面积、预期销售量、单位面积产量、单位面积种植成本为决策变量，以农作物种植收益最大化为目标函数，在构建线性规划函数的过程中，我们不仅继承了问题一中已有的分析框架，还**增添了四个新的约束条件：预期销售量变化、亩产量变化与风险因子、种植成本变化、销售价格变化**。进一步限制了决策变量的取值范围，确保了优化结果的现实可行性和经济合理性。基于新的优化模型求解，**结果见附件**。

针对问题三，综合考虑相关因素后给出农作物的最优种植方案，并与问题二的结果作比较分析。首先，对农作物的可替代性和互补性进行了明确的界定，并创新性地引入了替代性强度系数和互补性强度系数这两个指标以达量化可替代性和互补性之目的。在此基础上，精心构建了四个**矩阵模型**，这些矩阵分别展示了粮食类、蔬菜类、菌菇类作物内部的**相互可替代性**，以及这三类作物之间的**互补性**关系。最后，我们通过引入 **Pearson 相关系数**判断线性关系，建立线性回归模型，使用**最小二乘法**求出线性回归方程。

关键词：多约束线性规划模型；Pearson 相关系数；最小二乘法；内点法；农作物种植结构优化

一、问题重述

1.1 问题背景

随着我国农业现代化的推进，如何合理利用有限的耕地资源，实现乡村经济的可持续发展，已成为当前亟待解决的问题。某乡村位于华北山区，受地形和气候条件限制，耕地资源稀缺，种植模式单一，这严重制约了乡村经济的发展。为了提高耕地利用效率，增加农民收入，促进乡村经济可持续发展，该村决定调整种植结构，发展有机种植产业。

然而，在实施过程中，该村面临着一系列挑战：如何根据不同地块的特点选择适宜的农作物？如何优化种植策略，提高生产效益？如何在确保农作物产量的同时，降低种植风险？此外，还要考虑豆类作物的轮作要求以及田间管理的便利性。为此，该村希望通过建立一套科学的农作物种植模型，为种植决策提供有力支持，从而推动乡村经济的健康发展。

以下是附件的相关信息：

附件 1：乡村现有耕地和农作物的基本情况；

附件 2：2023 年乡村农作物种植和相关统计数据；

附件 3：须提交结果的模板文件。

1.2 问题重述

本文基于以上信息建立数学模型来解决以下问题：

问题 1：根据附件 2 中 2023 年中农作物种植情况和相关统计的信息，合并统计相关数据，针对超过部分滞销而造成浪费以及超过部分按 2023 年销售价格的 50% 降价出售这两种情况，分别给出该乡村 2024~2030 年农作物的最优种植方案，将结果分别填入 result1_1.xlsx 和 result1_2.xlsx 中。

问题 2：结合文献及经验，小麦和玉米的预期销售量预计以平均年增长率预计在 5% 至 10% 之间的趋势增长，其他农作物的预期销售量相比 2023 年可能会有不超过 $\pm 5\%$ 的波动，农作物的亩产量可能每年有 $\pm 10\%$ 的变动，农作物的种植成本预计每年将增加大约 5%。粮食作物销售价格相对稳定，而蔬菜作物销售价格平均每年增长约 5%。食用菌的销售价格每年可能下降 1% 至 5%，特别是羊肚菌的销售价格每年下降 5%。在考虑了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和潜在种植风险后，请制定 2024 年至 2030 年该乡村农作物的最佳种植方案，并将结果记录在 result2.xlsx 文件中。

问题 3：基于问题 2 的分析，全面考虑现实农业活动中，不同作物间可能存在替代或互补关系，且销售量、价格与成本相互关联等影响，制定 2024 至 2030 年该乡村的最佳种植策略，利用模拟数据进行计算，并与问题 2 的结果进行对比分析。

二、问题分析

在回答问题之前对所有附件中的数据做预处理，对缺失值，异常值进行了分析和说明，并对其中可能存疑的数据进行了合理性说明。

2.1 问题一的分析

问题一的核心目标在于面对两种不同销售策略时，分别给出该乡村 2024-2030 年农作物的最优种植方案。从大体上看，这是一道组合优化的问题。首先在数据预处理的基础上，我们假设每一年的农作物在当年就被销售掉，不存在囤积的情况。于是我们根据影响及所要达成目的的因素确定决策变量为各种农作物的销售价格、种植面积、预期销售量、单位面积产量、单位面积种植成本，根据决策变量和所在达到目标之间的函数关系构建以最大化种植收益为目标，再由决策变量所受的限制条件取 2023 年各种农作物的销售量为最大销量约束。最后，利用线性规划模型，得出最优种植的方案。问题一的第一小问与第二小问的分析思路、约束条件以及决策变量基本一致，只是由于销售策略不同，表达目标函数时存在差异。

2.2 问题二的分析

针对问题二，基于问题一的分析，以各种农作物的销售价格、种植面积、预期销售量、单位面积产量、单位面积种植成本为决策变量，以农作物种植收益最大化为目标函数，增添了 4 个约束条件，构建线性规划函数。由题意可知，价格等条件是波动的，但是范围固定，实际上为问题一优化模型的变形，在此基础上增添价格等要素波动的约束条件。

2.3 问题三的分析

在进一步分析题目前，先明确可替代性和互补性的定义：所谓可替代性指在不同的农业生产活动中，某些作物可以在一定程度上相互取代对方满足市场的需求，具体来说，当两种或多种农作物在用途、生产条件或者消费者的偏好上相似到一定程度时，它们就具有可替代性；所谓互补性是指某些作物在生长、生产或消费过程中相互依赖或相互增强的关系，这种关系意味着一种作物的生产、使用或消费通常与另一种作物的生产、使用或消费正相关，如：早熟土豆可以套种鲜食玉米，增大土豆产量可以连带增加玉米产量。接着，我们分别引入替代性强度系数 β_{j_1, j_2} 和互补性强度系数 λ_{j_1, j_2} 来量化农作物之间的可代替性与互补性，以构建四个矩阵分别表示粮食类、蔬菜类、菌菇类内部的可替代性关系和粮食类、蔬菜类、菌菇类的互补性关系，表内的数据为替代性强度系数和互补性强度系数的大小。

将上述的三个问题的分析整合，得到如下所示的问题分析结构图：

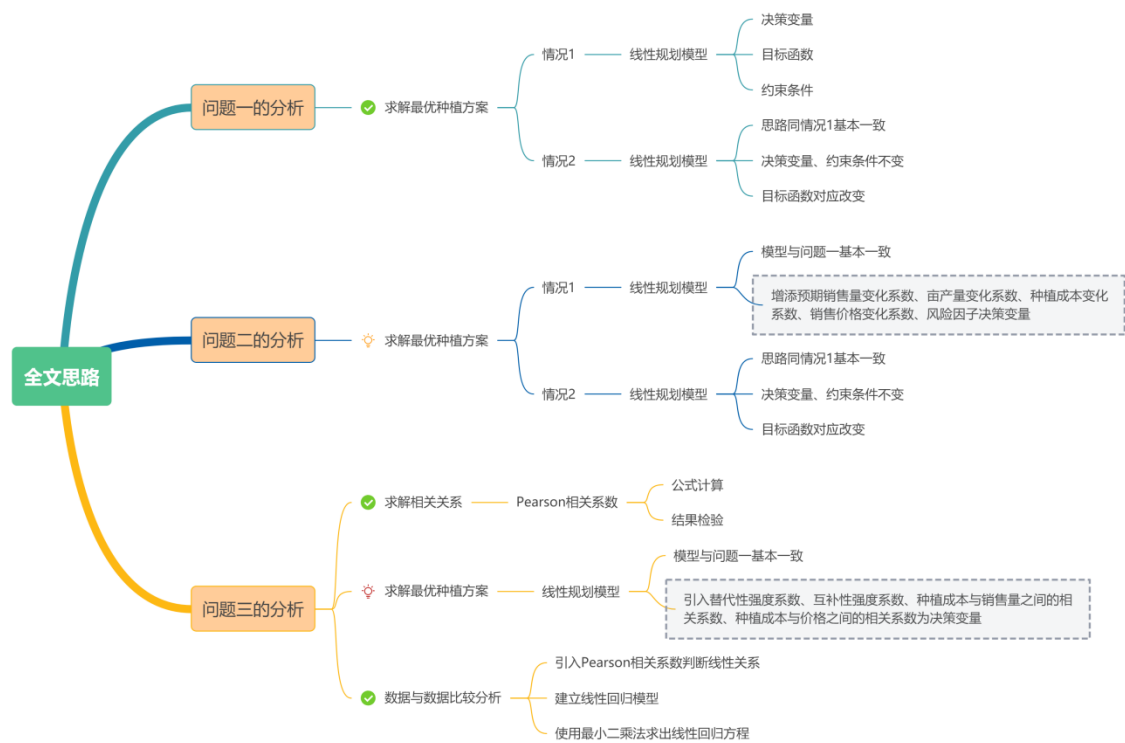


图 1：问题分析结构图

三、模型假设

1. 假设 2023 年的农作物在收获当年就被销售掉，不存在囤积的情况。
2. 假设以 2023 年农作物的销售情况作为预期销售量。
3. 假设销售金额取中间值并保留两位小数，为便于讨论。
4. 假设每种农作物不能在超过四种土地类型被种植，每块土地每种作物种植面积不小于 0.1 亩，为方便耕种作业和田间管理。
5. 假设每三年种植一次豆类作物是合理的，为平衡收益和土地持续利用率，中和豆类作物的经济效益不高的弊端。
6. 假设市场的消费能力是长期稳定的，不存在因极端情况产生的消费能力锐减，导致大面积农作物滞销。
7. 假设消费者都是理智的，对农产品的市场价值和实际价值认知客观，进而对 2023 年农作物的购买量能切实反映该作物收市场的需求程度。
8. 由于可替代的物品在用途和生产条件上高度相似，我们假设可替代性只存在于同一农作物的分类中，如：白灵菇和羊肚菌同属食用菌类，其种植条件高度相似，且普通消费者不会重复购买。
9. 由于互补性在同一互补性农作物的分类中体现得非常有限，我们假设互补性只存在于不同的农作物分类之间。
10. 假设在同一农作物分类下，越受市场欢迎，销售量越大的农作物越不可替代。

四、符号说明

符号	说明
A_i	以附件 1 中的“地块名称”列顺序为准，以 A_1 - A_{54} 依次表示 54 块土地和大棚
B_j	以附件 2 中的“作物编号”列顺序为准，以 B_1 - B_{41} 依次表示 41 个作物编号
k_j	B_j 所对应的预期销售量
$x_{i,j,k,t}$	第 i 块地在第 k 年第 t 季种植第 j 种作物的面积
$y_{i,j,k,t}$	第 i 块地在第 k 年第 t 季是否种植第 j 种作物的二值变量，其中 1 表示种植，0 表示未种植
$z_{i,j,k,t}$	第 i 块地第 j 种农作物在第 k 年第 t 季的单位面积种植成本（单位：元/亩）
$P_{i,j,k,t}$	第 i 块第 j 种作物在第 k 年第 t 季的销售价格（单位：元/斤）
$Q_{i,j,k,t}$	第 i 块第 j 种作物在第 k 年第 t 季的单位面积产量（单位：斤）
E_i	第 i 种情况时该乡村 2024-2030 农作物的总收入
F_i	第 i 种情况时该乡村 2024-2030 农作物的总成本
D_i	第 i 种情况时该乡村 2024-2030 农作物的总利润
$\gamma_{y_{i,j,t}}^{\text{预期销售量变化}}$	预期销售量变化系数
$\gamma_{y_{i,j,t}}^{\text{亩产量变化}}$	亩产量变化系数
$\alpha_{y_{i,j,t}}^{\text{风险}}$	风险因子
$\gamma_{y_{i,j,t}}^{\text{种植变化成本}}$	种植成本变化系数
$\gamma_{y_{i,j,t}}^{\text{销售价格变化}}$	销售价格变化系数

五、数据预处理

5.1 缺失值处理

对于附件中的数据进行排查分析发现，文件中不存在缺失值，因此不需要进行缺失值处理。

5.2 异常值处理

对附件 2 中的 2023 年的农作物种植情况和统计的相关数据进行预处理排查异常值，利用 3σ 原则，将数据导入 Spsspro 中，发现存在四个异常值，异常的数据如下，分别是黄瓜，白灵菇，羊肚菌，黄瓜。对异常数据进行置空处理。

	C	D	E	F	G	H
1	作物名称	地块类型	种植季次	亩产量/斤	种植成本/(元/亩)	销售单价/(元/斤)
77	29 白灵菇	普通大棚	第二季	4000	2000	18.00-20.00
78	29 黄瓜	普通大棚	第一季		3500	6.00-8.00

	作物名称	地块类型	种植季次	亩产量/斤	种植成本/(元/亩)	销售单价/(元/斤)
39	香菇	普通大棚	第二季	4000	2000	18.00-20.00
40	白灵菇	普通大棚	第二季	10000		14.00-18.00
41	羊肚菌	普通大棚	第二季	1000		80.00-120.00
103	29 黄瓜	智慧大棚	第二季		3850	7.20-9.60

图 2：异常值检测

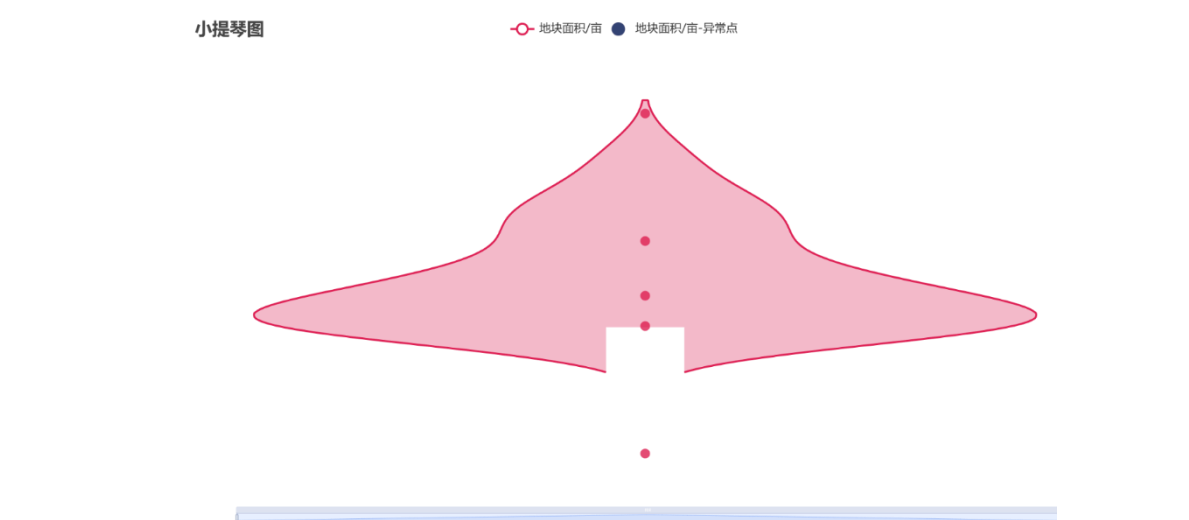


图 3：离群点展示

综上所述，为了保证农作物种类的完整性以及后续相关关系的探究，将以上缺失值与异常值点对应的面积、销售量、成本等信息全部用 0 填充。

5.3 正态性检验

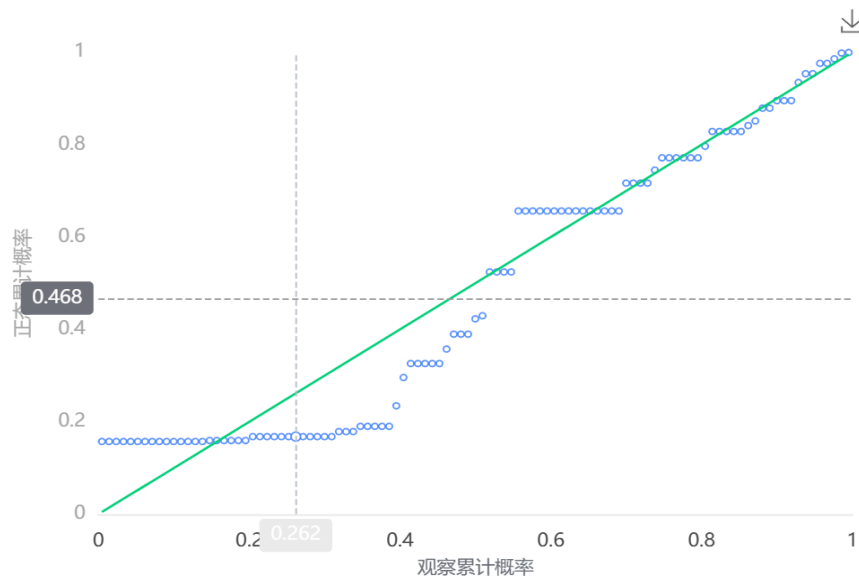


图 4：亩产量/斤观测的累计概率（P）与正态累计概率（P）的拟合度

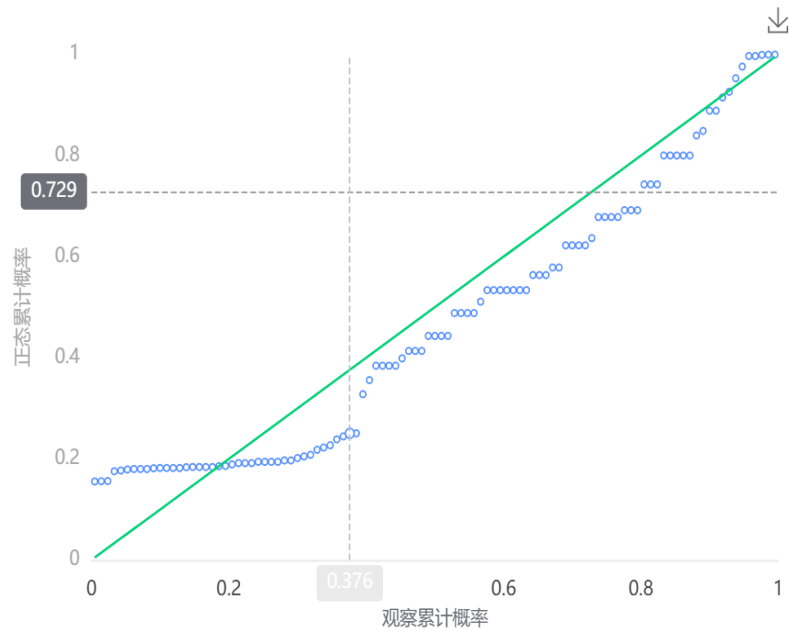


图 5：种植成本/(元/亩)观测的累计概率（P）与正态累计概率（P）的拟合度

六、问题一模型的建立与求解

7.5 线性规划模型的建立

为制定日补货总量和定价策略使得商超收益最大化，建立线性规划模型。

● 决策变量

在该问题中，影响最终结果的主要有 B_j 所对应的预期销售量 k_j ，第 i 块地在

第 k 年第 t 季种植第 j 种作物的面积 $x_{i,j,k,t}$ ，第 i 块地第 j 种农作物在第 k 年第 t 季的单位面积种植成本 $z_{i,j,k,t}$ ，第 i 块地第 j 种作物在第 k 年第 t 季的销售价格 $P_{i,j,k,t}$ ，第 i 块地第 j 种作物在第 k 年第 t 季的单位面积产量 $Q_{i,j,k,t}$ 。

● 目标函数

根据题目的要求，问题的目标是最大化乡村的收益，即：

$$D_i = E_i - F_i \quad (1)$$

其中， E_i 为乡村在第 i 情况时农作物的总收益； F_i 为乡村在第 i 情况时农作物的总成本； D_i 乡村在第 i 情况时农作物的总利润。

对于**情况一**：超过部分滞销，造成浪费

1. 当第 B_i 中种农作物 2024–2030 年中每季的总产量小于或等于相应的预期销售量 k_i ，此时：

$$E_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 P_{i,j,k,t} \cdot Q_{i,j,k,t} \cdot x_{i,j,k,t} \quad (2)$$

$$F_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 z_{i,j,k,t} \cdot x_{i,j,k,t} \quad (3)$$

$$D_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 x_{i,j,k,t} (P_{i,j,k,t} \cdot Q_{i,j,k,t} - Z_{i,j,k,t}) \quad (4)$$

2. 当第 B_i 中种农作物 2024–2030 年中每季的总产量小于或等于相应的预期销售量 k_i ，此时：

$$E_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 P_{i,j,k,t} \cdot k_j \quad (5)$$

$$F_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 z_{i,j,k,t} \cdot x_{i,j,k,t} \quad (6)$$

$$D_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 (P_{i,j,k,t} \cdot k_j - Z_{i,j,k,t} \cdot Q_{i,j,k,t} \cdot x_{i,j,k,t}) \quad (7)$$

对于**情况二**：超过部分按 2023 年销售价格的 50%降价出售

1. 当第 B_i 中种农作物 2024-2030 年中每季的总产量小于或等于相应的预期销售量 k_i ，此时：

$$E_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 P_{i,j,k,t} \bullet Q_{i,j,k,t} \bullet x_{i,j,k,t} \quad (8)$$

$$F_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 z_{i,j,k,t} \bullet x_{i,j,k,t} \quad (9)$$

$$D_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 x_{i,j,k,t} (P_{i,j,k,t} \bullet Q_{i,j,k,t} - Z_{i,j,k,t}) \quad (10)$$

2. 当第 B_i 中种农作物 2024-2030 年中每季的总产量小于或等于相应的预期销售量 k_i ，此时：

$$E_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [P_{i,j,k,t} \bullet k_j - 0.5 P_{i,j,k,t} \bullet (Q_{i,j,k,t} \bullet x_{i,j,k,t} - k_j)] \quad (11)$$

$$F_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 z_{i,j,k,t} \bullet x_{i,j,k,t} \quad (12)$$

$$D_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [P_{i,j,k,t} \bullet k_j + 0.5 P_{i,j,k,t} \bullet (Q_{i,j,k,t} \bullet x_{i,j,k,t} - k_j) - z_{i,j,k,t} \bullet x_{i,j,k,t}] \quad (13)$$

● 约束条件

约束 1:

$$\sum_{k=2024}^{2023} \sum_{t=1}^1 y_{i,j,k,t} = 1, \forall i \in \{1,2,\dots,26\}, j \in \{1,2,\dots,16\} \quad (13)$$

表示所有平旱地、梯田和山坡地每年只能种植一季粮食类作物。

约束 2:

$$\sum_{i=1}^{54} x_{i,j,k,t} = 1201, \forall k, t, j \quad (14)$$

表示露天耕地共计 1201 亩，分散为 34 个大小不同的地块。

约束 3:

$$x_{i,j,k,t} \leq 0.6, \forall i \in \{35,36,\dots,54\}, \forall j, k, t \quad (15)$$

表示有 16 个普通大棚和 4 个智慧大棚，每个大棚耕地面积为 0.6 亩。

约束 4:

$$\sum_{t=1}^2 y_{i,j,k,t} \leq 2, \forall i \in \{27,28,\dots,34\}, j \in \{16,17,\dots,41\} \quad (16)$$

表示水浇地适宜每年种植一季水稻或两季蔬菜。

约束 5:

$$\sum_{t=1}^2 y_{i,j,k,t} = 2, \forall i \in \{35,36,\dots,54\}, j \in \{17,18,\dots,41\} \quad (17)$$

普通大棚和智慧大棚每年都可以种植两季作物。

约束 6:

$$\begin{aligned} y_{i,j,k,1} &= 1, y_{i,j,k,2} = 0, \forall i \in \{35,36,\dots,50\}, j \in \{35,36,37\} \\ y_{i,j,k,1} &= 0, y_{i,j,k,2} = 1, \forall i \in \{35,36,\dots,50\}, j \in \{38,39,40,41\} \end{aligned} \quad (18)$$

表示普通大棚每年种植两季作物，第一季可种植多种蔬菜（大白菜、白萝卜和红萝卜除外），第二季只能种植食用菌。

约束 7:

$$y_{i,j,k,t} = 1, \forall i \in \{51,52,53,54\}, j \in \{17,18,\dots,34\}, t \in \{1,2\} \quad (19)$$

表示智慧大棚每年都可种植两季蔬菜（大白菜、白萝卜和红萝卜除外）。

约束 8:

$$\sum_{j=1}^{41} x_{i,j,k,t} \leq \text{地块面积}, \forall i, k, t \quad (20)$$

表示同一地块（含大棚）每季可以合种不同的作物。

约束 9:

$$y_{i,j,k,t} + x_{i,j,k+1,t} \leq 1, \forall i, j, k, t \quad (21)$$

表示每种作物在同一地块（含大棚）都不能种植相同的农作物必须间隔一年。

约束 10:

$$\sum_{t=\text{当前年份}}^{\text{当前年份}+3} y_{i,j,k,t} \geq 1, \forall i, j \in \{1,2,\dots,5\}, t \quad (22)$$

表示同一地块（含大棚）的所有土地每隔三年必须种植一次豆类作物。

约束 11:

$$x_{i,j,k,t} \geq 0.2, \forall i, j, k, t \quad (23)$$

表示每种作物在单块地块（含大棚）种植的面积不宜太小。

约束 12:

$$y_{i,16,k,t} + \sum_{t=1}^2 \sum_{j=17}^{41} y_{i,j,k,t} \leq 2, \forall i \in \{27,28,\dots,34\}, \forall k \quad (24)$$

表示水浇地每年可以单季种植水稻或两季种植蔬菜作物。

约束 13:

$$\begin{aligned} y_{i,j,k,1} = 1, y_{i,j,k,2} = 0, \forall i \in \{27,28,\dots,34\}, j \in \{17,18,\dots,34\}, k \\ y_{i,j,k,1} = 0, y_{i,j,k,2} = 1, \forall i \in \{27,28,\dots,34\}, j \in \{35,36,37\}, k \end{aligned} \quad (25)$$

表示在某块水浇地种植两季蔬菜，第一季可种植多种蔬菜（大白菜、白萝卜和红萝卜除外），第二季只能种植大白菜、白萝卜和红萝卜中的一种。

约束 14:

$$y_{i,j,k,1} = 0, \forall i \in \{27,28,\dots,34\}, j \in \{35,36,37\}, k \quad (26)$$

表示大白菜、白萝卜和红萝卜只能在水浇地的第二季种植。

约束 15:

$$y_{i,j,k,1} = 1 \forall i \in \{35,36,\dots,50\}, j \in \{38,39,40,41\}, k \quad (27)$$

表示食用菌类只能在秋冬季和普通大棚里种植。

综上，建立以下模型：

$$\text{MAX} \sum_i \sum_j D_i = P_{y_{i,j,t}} (G_{y_{i,j,t}} + H_{y_{i,j,t}}) - W_{y_{i,j,t}}$$

使得（13）—（27）成立，得到以下结果。

堆积条形图

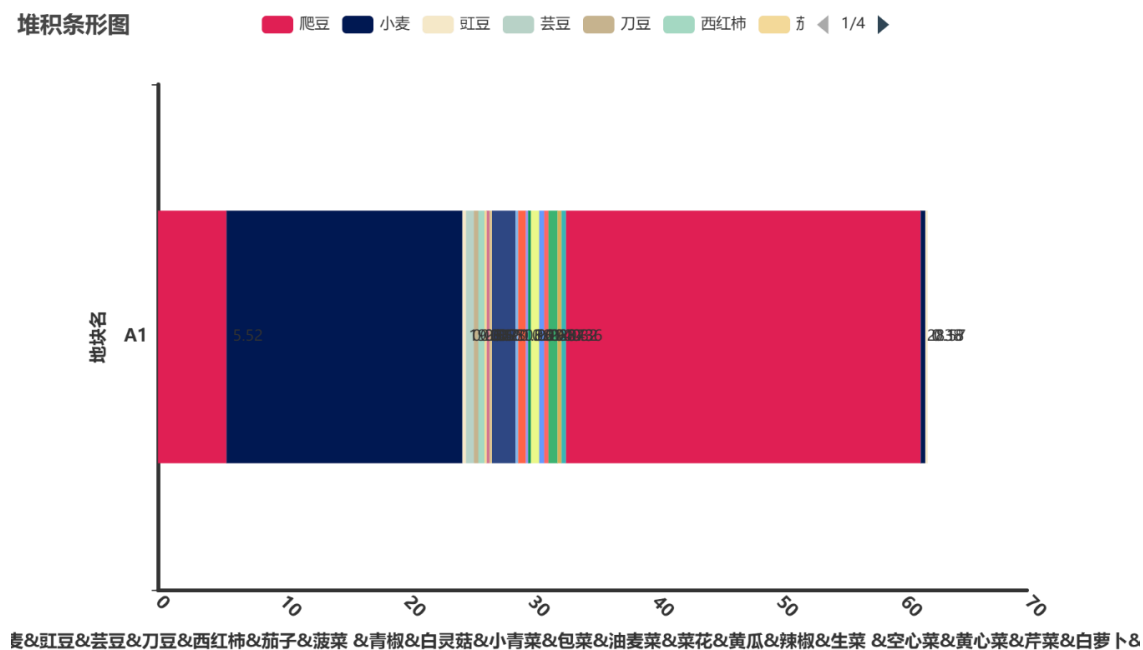


图 6: 最佳种植方案展示堆积条形图 (以 A1 为例)

折柱混合图

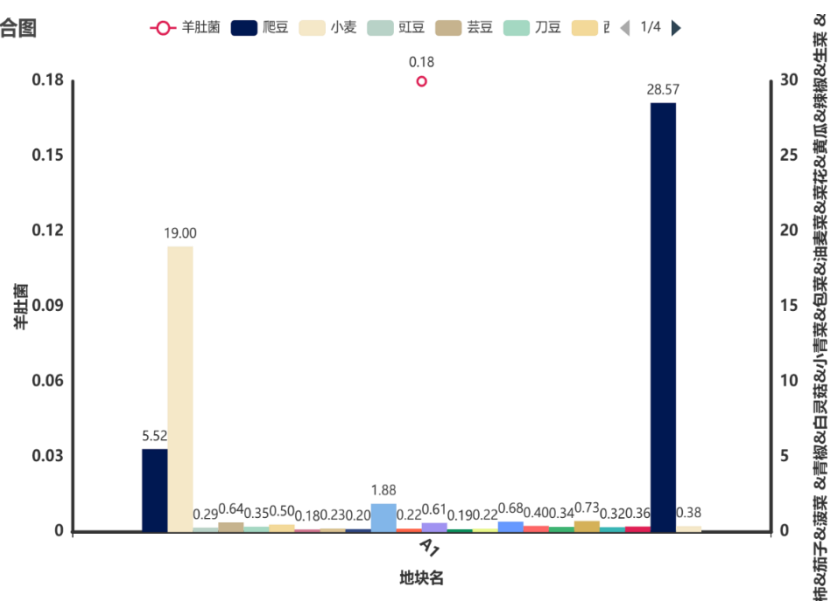


图 7: 该乡村全部耕地最佳种植方案展示折柱混合图

桑基图

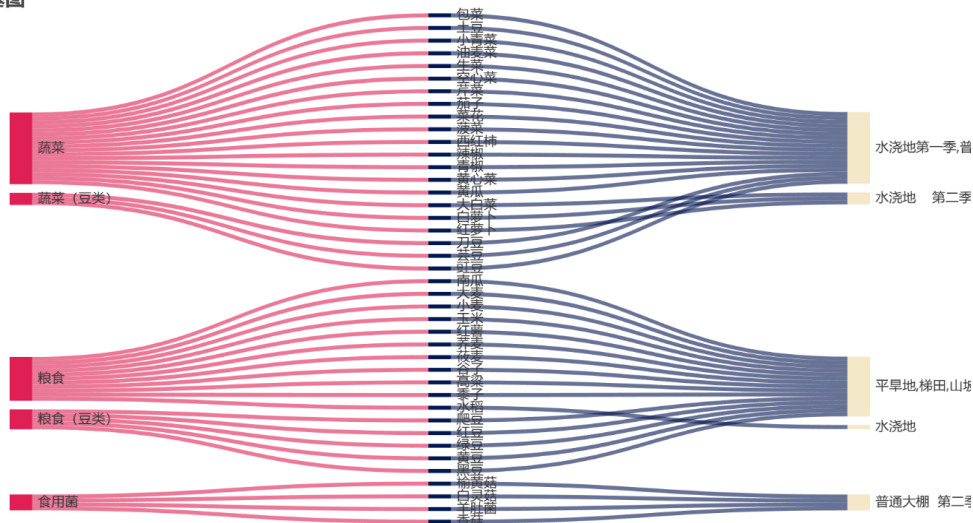


图 8：各作物可种植范围桑基图

环图

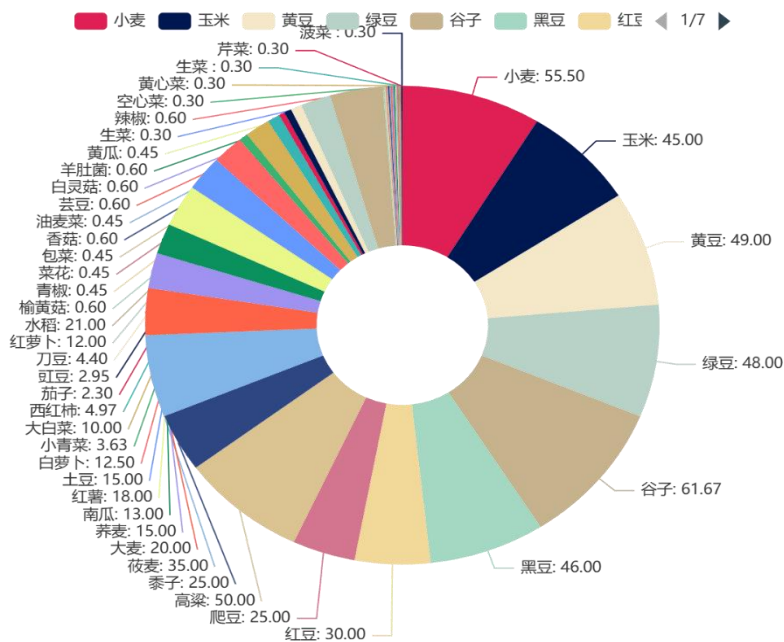


图 9：该乡村全部耕地最佳种植方案展示环图

七、 问题二模型的建立与求解

8.1 模型的建立

基于问题二的分析，要求综合考虑各种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以及潜在的种植风险，仍以最大化乡村农作物收益为目标，故建立线性规划模型。

● 决策变量

在该问题中，影响最终结果的变量主要原始预计销售量、亩产量、种植成本、销售价格以及预期销售量变化系数、亩产量变化系数、种植成本变化系数、销售价格变化系数、风险因子。

● 目标函数

同第二问一样，本体的目标仍然是使乡村农作物的收益最大。

对于**情况一**：超过部分滞销，造成浪费

1. 当第 B_i 种农作物 2024-2030 年中每季的**最终**总产量小于或等于相应的**最终**预期销售量, 此时：

$$D_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [(原始销售价格 \times \gamma_{y_{i,j,t}}^{销售价格变化}) \bullet (原始亩产量 \times \gamma_{y_{i,j,t}}^{亩产量变化} \times (1 - \alpha_{y_{i,j,t}}^{风险}) \bullet x_{i,j,k,t} - (原始种植成本 \times \gamma_{y_{i,j,t}}^{种植成本变化}) \bullet x_{i,j,k,t}]) \quad (28)$$

2. 当第 B_i 种农作物 2024-2030 年中每季的总产量小于或等于相应的预期销售量 k_i , 此时：

$$D_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [(原始销售价格 \times \gamma_{y_{i,j,t}}^{销售价格变化}) \bullet (原始预计销售量 \times \gamma_{y_{i,j,t}}^{预计销售量变化}) - (原始种植成本 \times \gamma_{y_{i,j,t}}^{种植成本变化}) \bullet x_{i,j,k,t}] \quad (29)$$

对于**情况二**：超过部分按 2023 年销售价格的 50%降价出售

1. 当第 B_i 中种农作物 2024-2030 年中每季的总产量小于或等于相应的预期销售量 k_i , 此时：

$$D_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [(原始销售价格 \times \gamma_{y_{i,j,t}}^{销售价格变化}) \bullet (原始亩产量 \times \gamma_{y_{i,j,t}}^{亩产量变化} \times (1 - \alpha_{y_{i,j,t}}^{风险}) \bullet x_{i,j,k,t} - (原始种植成本 \times \gamma_{y_{i,j,t}}^{种植成本变化}) \bullet x_{i,j,k,t})] \quad (30)$$

2. 当第 B_i 中种农作物 2024-2030 年中每季的总产量小于或等于相应的预期销售量 k_i , 此时：

$$D_2 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 [(原始销售价格 \times \gamma_{y_{i,j,t}}^{销售价格变化}) \bullet (原始预计销售量 \times \gamma_{y_{i,j,t}}^{预计销售量变化}) + 0.5(原始销售价格 \times \gamma_{y_{i,j,t}}^{销售价格变化}) \bullet (原始亩产量 \times \gamma_{y_{i,j,t}}^{亩产量变化} \times (1 - \alpha_{y_{i,j,t}}^{风险}) \bullet x_{i,j,k,t} - 原始预计销售量 \times \gamma_{y_{i,j,t}}^{预期销售量变化}) - (原始种植成本 \times \gamma_{y_{i,j,t}}^{种植成本变化})] \quad (31)$$

● 约束条件

约束 1:

$$1.5 \leq \gamma_{y_{i,j,t}}^{\text{预期销售量变化}} \leq 1.10 \quad (32)$$

表示小麦和玉米的销售量变化（ $j = 6, 7$ ）。

约束 2:

$$0.95 \leq \gamma_{y_{i,j,t}}^{\text{预期销售量变化}} \leq 1.05 \quad (33)$$

表示其他农作物（ $j \neq 6, 7$ ）。

约束 3:

$$\begin{aligned} 0.90 &\leq \gamma_{y_{i,j,t}}^{\text{亩产量变化}} \leq 1.10 \\ 0 &\leq \alpha_{y_{i,j,t}}^{\text{风险}} \leq 1 \end{aligned} \quad (34)$$

表示农作物每年的产量变化。

约束 4:

$$\gamma_{y_{i,j,t}}^{\text{种植成本变化}} = 1.05 \quad (35)$$

表示农作物成本变化。

约束 5:

$$\gamma_{y_{i,j,t}}^{\text{种植价格变化}} = 1 \quad (36)$$

表示粮食类销售价格变化。

约束 6:

$$\gamma_{y_{i,j,t}}^{\text{种植价格变化}} = 1.05 \quad (37)$$

表示蔬菜类销售价格变化。

约束 7:

$$\gamma_{y_{i,j,t}}^{\text{种植价格变化}} = 0.95 \quad (38)$$

表示羊肚菌的销售价格变化（ $j = 41$ ）。

约束 8:

$$0.95 \leq \gamma_{y_{i,j,t}}^{\text{种植价格变化}} = 0.99 \quad (39)$$

表示其他食用菌的价格变化（ $j = 38, 39, 40$ ）。

根据以上目标函数以及约束条件，建立一下模型：

$$\text{MAX} \sum_i \sum_j D_i = P_{y_{i,j,t}} (G_{y_{i,j,t}} + H_{y_{i,j,t}}) - W_{y_{i,j,t}}$$

使得（32）—（39）成立，得到以下结果。

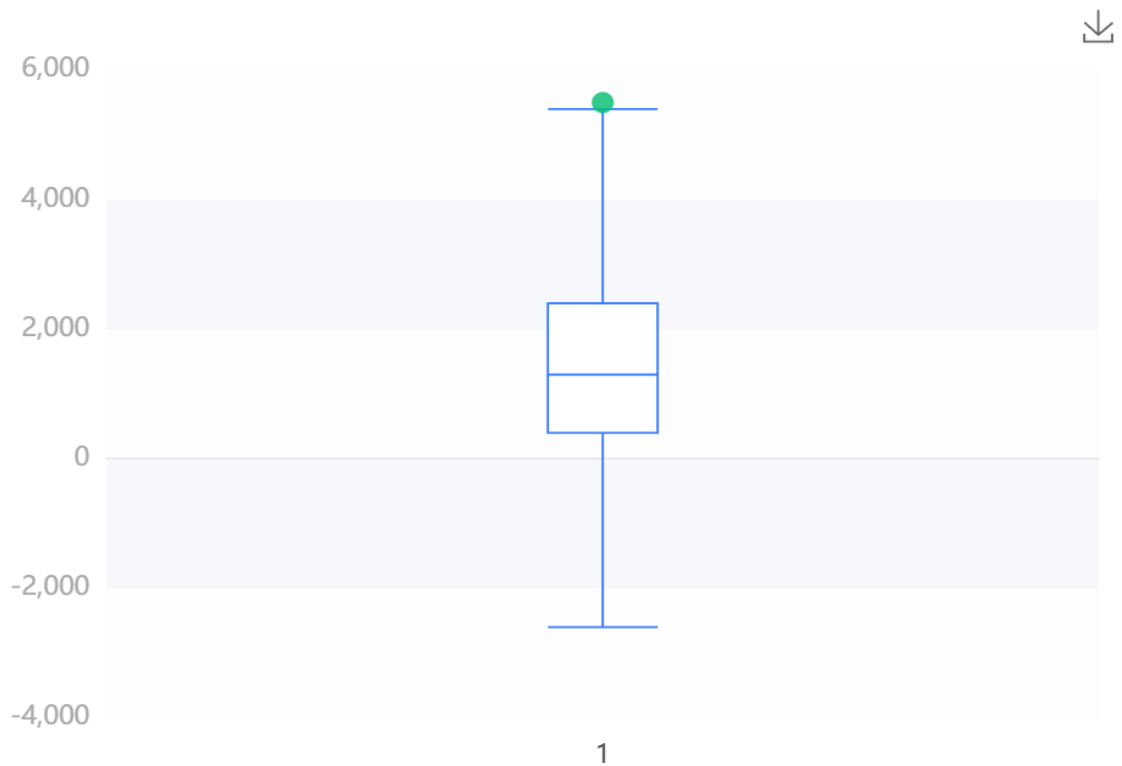


图 10：种植成本箱型图

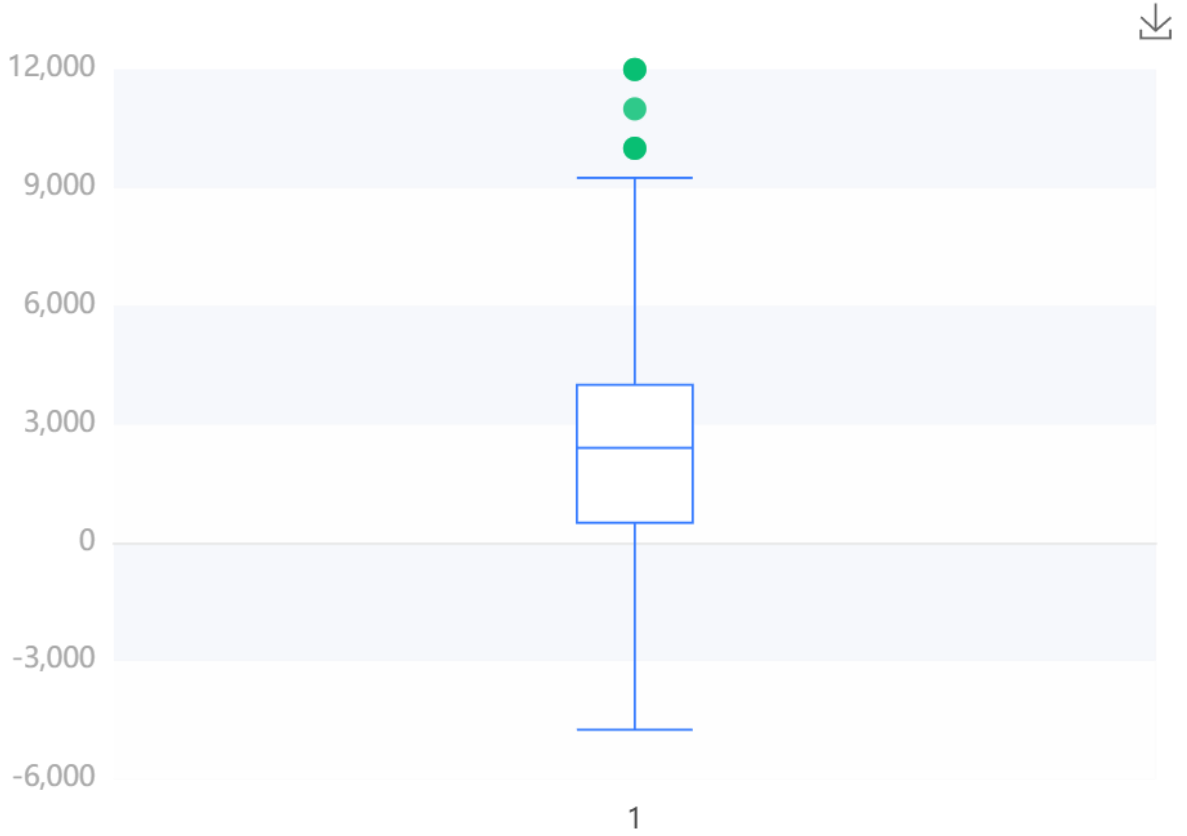


图 11：亩产量箱型图

八、问题三模型的建立与求解

● 决策变量

在该问题中，影响最终结果的变量主要原始预计销售量、亩产量、种植成本、销售价格。

● 目标函数

同第一、二问一样，本体的目标仍然是使乡村农作物的收益最大。即：

$$D_1 = \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 (P_{y_{i,j,t}} \bullet \min\{z_{y_{i,j,t}} \bullet x_{y_{i,j,t}}, \text{预期销售量}_{y_{i,j,t}}\} + \eta \bullet P_{y_{i,j,t}} \bullet \max\{z_{y_{i,j,t}} \bullet x_{y_{i,j,t}} - \text{预期销售量}_{y_{i,j,t}}, 0\}) - \sum_{i=1}^{54} \sum_{j=1}^{41} \sum_{k=2024}^{2030} \sum_{t=1}^2 \text{种植成本}_{y_{i,j,t}} \bullet x_{y_{i,j,t}} \quad (40)$$

● 约束条件

约束 1：

$$\text{预期销售量}_{y_2,t} = \text{原始预期销售量}_{y_2,t} - \beta_{j_1,j_2} \bullet x_{i,j,t} \quad (41)$$

表示替代性约束。

约束 2:

$$z_{y_2,t} = z_{y_2,t} + \lambda_{j_1,j_2} \bullet x_{i,j,t} \quad (42)$$

表示互补性约束。

约束 3:

$$P_{y,t} = P_{y,t} + \varepsilon_{P,k} \bullet (\text{预期销售量}_{y,t} - \text{平均销售量}_y) \quad (43)$$

表示销售量与价格的相关性约束，其中 $\varepsilon_{P,k}$ 表示种植成本 P 与销售量 k 之间的相关性。

约束 4:

$$P_{y,t} = P_{y,t} + \varepsilon_{z,P} \bullet (\text{种植成本}_{y,t} - \text{平均种植成本}_y) \quad (44)$$

表示种植成本与价格的相关性约束，其中 $\varepsilon_{z,P}$ 表示种植成本 z 与价格 P 之间的相关性。

根据以上目标函数以及约束条件，建立一下模型：

$$\text{MAX} \sum_i \sum_j D_i = P_{y_{i,j,t}} (G_{y_{i,j,t}} + H_{y_{i,j,t}}) - W_{y_{i,j,t}}$$

使得（41）—（44）成立，得到以下结果。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	物种编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	1	0	-0.49	-0.47	-0.45	-0.43	-0.44	-0.44	-0.04	-0.02	-0.05	0	0	-0.01	-0.08	-0.06	-0.08
3	2	0.49	0	-0.48	-0.46	-0.48	-0.46	-0.49	-0.04	-0.06	0	-0.07	-0.09	-0.03	-0.06	-0.09	-0.01
4	3	0.47	0.48	0	-0.46	-0.5	-0.47	-0.42	-0.09	-0.03	-0.02	-0.07	-0.03	-0.09	-0.01	-0.03	-0.03
5	4	0.45	0.46	0.46	0	-0.49	-0.46	-0.44	-0.05	-0.02	-0.08	-0.02	-0.09	-0.05	-0.05	-0.08	-0.06
6	5	0.43	0.48	0.5	0.49	0	-0.41	-0.44	-0.06	-0.04	0	-0.07	-0.02	-0.06	-0.04	-0.03	-0.03
7	6	0.44	0.46	0.47	0.46	0.41	0	-0.42	-0.06	-0.08	-0.06	-0.03	-0.03	0	-0.03	0	-0.06
8	7	0.44	0.49	0.42	0.44	0.44	0.42	0	-0.02	-0.03	-0.05	-0.02	-0.01	-0.06	-0.09	-0.1	-0.04
9	8	0.04	0.04	0.09	0.05	0.06	0.06	0.02	0	-0.44	-0.42	-0.43	-0.48	-0.43	-0.42	-0.41	-0.4
10	9	0.02	0.06	0.03	0.02	0.04	0.08	0.03	0.44	0	-0.45	-0.48	-0.4	-0.45	-0.49	-0.42	-0.41
11	10	0.05	0	0.02	0.08	0	0.06	0.05	0.42	0.45	0	-0.5	-0.45	-0.5	-0.45	-0.45	-0.44
12	11	0	0.07	0.07	0.02	0.07	0.03	0.02	0.43	0.48	0.5	0	-0.44	-0.48	-0.5	-0.46	-0.44
13	12	0	0.09	0.03	0.09	0.02	0.03	0.01	0.48	0.4	0.45	0.44	0	-0.4	-0.43	-0.45	-0.43
14	13	0.01	0.03	0.09	0.05	0.06	0	0.06	0.43	0.45	0.5	0.48	0.4	0	-0.43	-0.47	-0.45
15	14	0.08	0.06	0.01	0.05	0.04	0.03	0.09	0.42	0.49	0.45	0.5	0.43	0.43	0	-0.45	-0.48
16	15	0.06	0.09	0.03	0.08	0.03	0	0.1	0.41	0.42	0.45	0.46	0.45	0.47	0.45	0	-0.42
17	16	0.08	0.01	0.03	0.06	0.03	0.06	0.04	0.4	0.41	0.44	0.44	0.43	0.45	0.48	0.42	

表 1 粮食类作物可替代性系数

物种编号	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
17	0	-0.46	0.41	-0.03	-0.02	0.03	-0.07	-0.05	0.01	-0.03	-0.07	-0.04	0.06	-0.04	-0.03	-0.06	-0.03	0.05	0.07	-0.09	-0.04
18	0.46	0	-0.51	0.01	0.02	-0.08	0.02	0.06	0.09	0.02	-0.09	0.08	-0.05	-0.03	-0.04	-0.06	0.08	0.05	0.01	0.03	-0.03
19	-0.41	0.51	0	0.1	0.01	0.05	-0.09	-0.05	-0.09	-0.09	0.05	0.03	0	0.02	0	0.1	-0.05	-0.09	-0.01	-0.06	0.02
20	0.03	-0.01	-0.1	0	-0.08	0.01	-0.04	-0.09	0.03	-0.07	-0.1	0.04	0	-0.08	-0.09	-0.06	-0.09	0.01	-0.02	-0.08	0
21	0.02	-0.02	-0.01	0.08	0	-0.06	0.03	-0.06	0.01	0.05	0.06	-0.04	0.04	-0.08	0.05	0.04	-0.08	0.01	-0.07	0.06	0.05
22	-0.03	0.08	-0.05	-0.01	0.06	0	-0.05	0.03	0.07	0.04	0.04	0	0.04	-0.08	-0.04	0.09	0.01	-0.02	0.05	0.02	-0.04
23	0.07	-0.02	0.09	0.04	-0.03	0.05	0	-0.07	-0.06	-0.04	0.03	-0.02	-0.05	0.03	0.04	-0.05	0.02	-0.09	0.03	-0.07	0.01
24	0.05	-0.06	0.05	0.09	0.06	-0.03	0.07	0	-0.05	-0.06	-0.02	0.07	-0.06	-0.1	-0.1	-0.1	0.06	-0.02	-0.06	-0.03	-0.04
25	-0.01	-0.09	0.09	-0.03	-0.01	-0.07	0.06	0.05	0	-0.09	-0.03	0.09	-0.08	0.03	-0.09	-0.09	0.04	-0.05	-0.07	-0.05	0.07
26	0.03	-0.02	0.09	0.07	-0.05	-0.04	0.04	0.06	0.09	0	0.07	-0.07	-0.03	0.03	0.06	0	0	-0.09	-0.05	0	-0.09
27	0.07	0.09	-0.05	0.1	-0.06	-0.04	-0.03	0.02	0.03	-0.07	0	0.03	-0.07	-0.06	-0.05	-0.09	0.02	-0.06	-0.08	0	0.04
28	0.04	-0.08	-0.03	-0.04	0.04	0	0.02	-0.07	-0.09	0.07	-0.03	0	0.05	0	-0.01	-0.1	0.07	0.05	0.04	-0.09	0.06
29	-0.06	0.05	0	0	-0.04	-0.04	0.05	0.06	0.08	0.03	0.07	-0.05	0	-0.04	-0.09	0.04	-0.05	0.07	0.1	0.01	-0.02
30	0.04	0.03	-0.02	0.08	0.08	0.08	-0.03	0.1	-0.03	-0.03	0.06	0	0.04	0	0.02	0.03	0.06	0.04	-0.06	0.07	0.06
31	0.03	0.04	0	0.09	-0.05	0.04	-0.04	0.1	0.09	-0.06	0.05	0.01	0.09	-0.02	0	-0.07	-0.09	0.01	0.05	-0.07	-0.02
32	0.06	0.06	-0.1	0.06	-0.04	-0.09	0.05	0.1	0.09	0	0.09	0.1	-0.04	-0.03	0.07	0	0.05	-0.03	0.04	0.06	-0.1
33	0.03	-0.08	0.05	0.09	0.08	-0.01	-0.02	-0.06	-0.04	0	-0.02	-0.07	0.05	-0.06	0.09	-0.05	0	-0.07	0.07	-0.09	-0.05
34	-0.05	-0.05	0.09	-0.01	-0.01	0.02	0.09	0.02	0.05	0.09	0.06	-0.05	-0.07	-0.04	-0.01	0.03	0.07	0	0.06	-0.05	0.09
35	-0.07	-0.01	0.01	0.02	0.07	-0.05	-0.03	0.06	0.07	0.05	0.08	-0.04	-0.1	0.06	-0.05	-0.04	-0.07	-0.06	0	-0.03	0.06
36	0.09	-0.03	0.06	0.08	-0.06	-0.02	0.07	0.03	0.05	0	0	0.09	-0.01	-0.07	0.07	-0.06	0.09	0.05	0.03	0	0.58
37	0.04	0.03	-0.02	0	-0.05	0.04	-0.01	0.04	-0.07	0.09	-0.04	-0.06	0.02	-0.06	0.02	0.1	0.05	-0.09	-0.06	-0.58	0

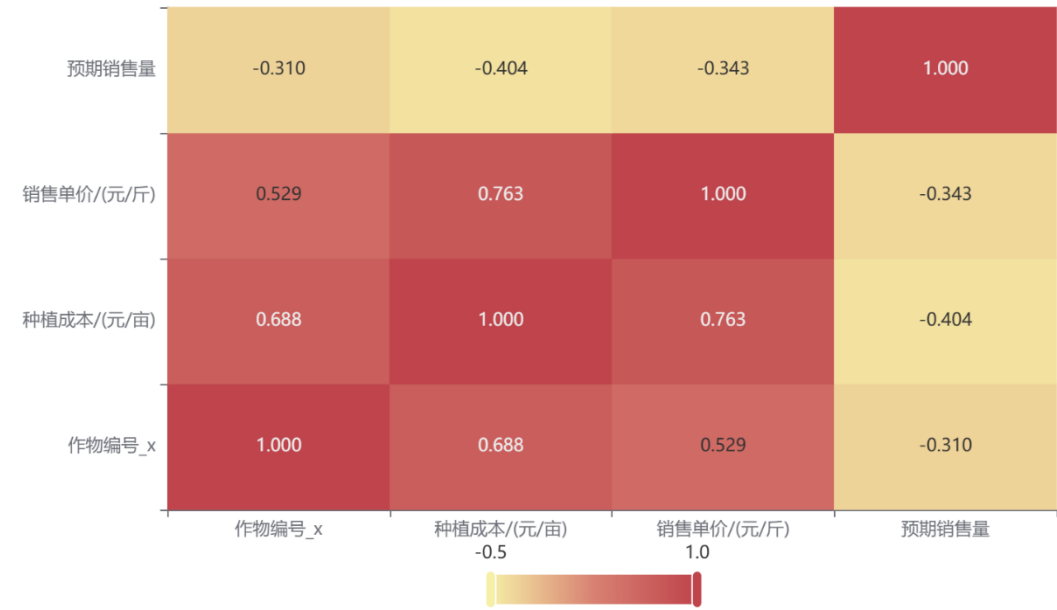
表 2 蔬菜类作物可替代性系数

物种编号	38	39	40	41
38	0.00	0.37	0.41	0.51
39	-0.37	0.00	0.24	0.44
40	-0.41	-0.24	0.00	0.59
41	-0.51	-0.44	-0.59	0.00

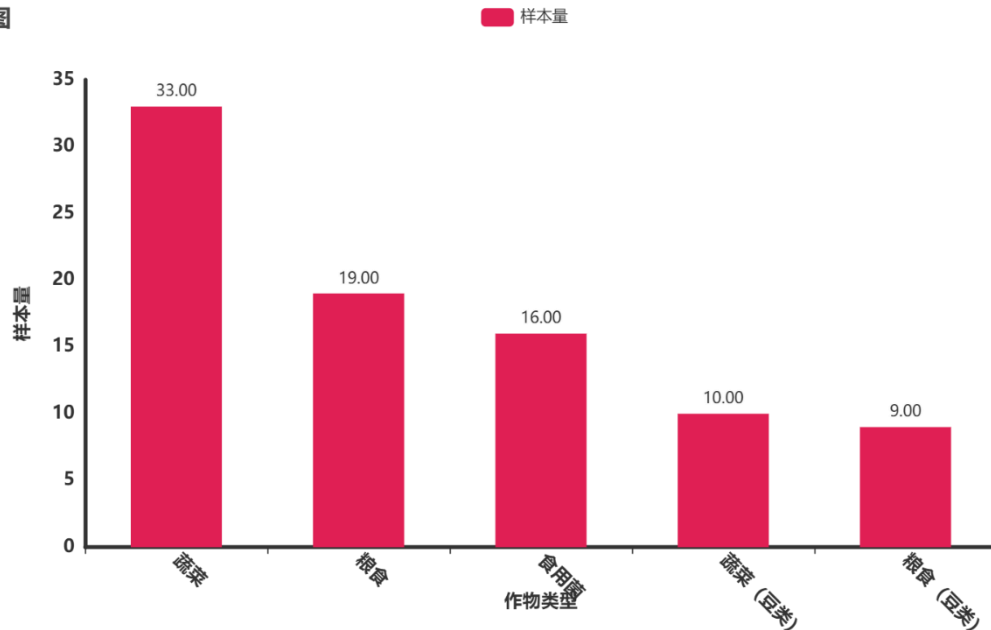
表 3 菌菇类可替代系数

农作物种类	粮食类作物	蔬菜类作物	菌菇类作物
粮食类作物	0	0.08	-0.06
蔬菜类作物	-0.08	0	-0.07
菌菇类作物	0.06	0.07	0

表 4 四大种类间的互补性系数



柱状图



九、模型的优缺点分析

10.1 模型的优点

1. 线性规划通过变量表示决策,通过线性方程或不等式表示约束,以决策变量的线性组合描述目标函数,使得问题的数学描述非常清晰,与题目数据匹配度强。

2. 这三道问题本质上采用同一线性规划模型,对线性规划模型中的噪声和不确定性具有较强的鲁棒性。

3. 使用线性规划进行求解,在初始化与编码阶段添加限制,并设计不可行解的修复机制以解决添加限制可能带来的问题,提高了模型稳定性。

10.2 模型的缺点

1. 线性规划模型建立存在较多假设条件,处理结果和实际存在一定的偏差。

2. 本题使用线性运算模型变量较多,计算量大,耗时较长。

十、参考文献

- [1] 朱冠雄,高祺,华方静,等.鲁西北地区大豆玉米间作不同行比配置对大豆生长、产量及间作系统的影响[J/OL].中国油料作物学报,1-10[2024-09-08]
- [2] 万猛虎,满本菊,刘威帆,等.与豆类间作对马铃薯叶片碳氮代谢关键酶活性及作物产量的影响[J/OL].中国生态农业学报(中英文),1-12[2024-09-08]
- [3] 马琳静,孔维威,杜如学,等.羊肚菌-豇豆轮作高效栽培模式及效益分析[J].中国食用菌,2024,43(04):32-38.DOI:10.13629/j.cnki.53-1054.2024.04.006.
- [4] 冯翠,衣政伟,钱巍,等.利用菇渣生产的有机生态型栽培基质对叶菜类蔬菜生长的影响[J].江西农业学报,2018,30(08):29-32.DOI:10.19386/j.cnki.jxnyxb.2018.08.07.

- [5] 王成业,曲萍.马铃薯、油菜、玉米、平菇、木耳、大白菜、小麦、菠菜立体种植[J].现代农业,2003,(04):17.
- [6] 马永发,刘家起,王蒂.大棚黄瓜—豇豆—青菜+芹菜高效种植模式[J].农业科技通讯,2019,(07):345-346+363.
- [7] 吴俊杰.基于线性规划模型的农业种植业结构优化研究[J].新农民,2024,(03):28-30.

附录

问题 1 第一小问

```
import pandas as pd
import pulp
import sys
sys.stdout = open("result1_1.csv", "w", encoding="utf-8")
# 读取地块数据和作物种植数据
land_data = pd.read_excel(
    "附件/附件 1.xlsx"
) # 假设第一张表有地块编号、地块面积、地块类型
crop_data = pd.read_excel("附件/附件 2.xlsx") # 假设第二张表包含了 2023 年的
种植情况

# 打印列名, 确保我们使用正确的列名
# print(land_data.columns)

acre_yield = {
    "刀豆": 2200.0,
    "包菜": 4100.0,
    "南瓜": 2850.0,
    "土豆": 2200.0,
    "大白菜": 5000.0,
    "大麦": 500.0,
    "小青菜": 3600.0,
    "小麦": 760.0,
    "榆黄菇": 5000.0,
    "水稻": 500.0,
    "油麦菜": 4533.333333333333,
    "爬豆": 395.0,
    "玉米": 950.0,
    "生菜": 4533.333333333333,
    "白灵菇": 10000.0,
    "白萝卜": 4000.0,
    "空心菜": 11000.0,
    "红萝卜": 3000.0,
    "红薯": 2100.0,
    "红豆": 380.0,
    "绿豆": 331.6666666666667,
    "羊肚菌": 1000.0,
    "芸豆": 3266.6666666666665,
    "芹菜": 6033.333333333333,
```

```

"茄子": 7200.0,
"荞麦": 105.0,
"莜麦": 400.0,
"菜花": 3633.3333333333335,
"菠菜": 3000.0,
"西红柿": 2700.0,
"谷子": 380.0,
"豇豆": 3266.6666666666665,
"辣椒": 1800.0,
"青椒": 2700.0,
"香菇": 4000.0,
"高粱": 600.0,
"黄心菜": 5466.666666666667,
"黄瓜": 13500.0,
"黄豆": 380.0,
"黍子": 500.0,
"黑豆": 475.0,
} # 从附件读取各作物的亩产量

```

```

expected_sales = {
    "刀豆": 26880.0,
    "包菜": 3930.0,
    "南瓜": 35100.0,
    "土豆": 30000.0,
    "大白菜": 150000.0,
    "大麦": 10000.0,
    "小青菜": 35480.0,
    "小麦": 170840.0,
    "榆黄菇": 9000.0,
    "水稻": 21000.0,
    "油麦菜": 4500.0,
    "爬豆": 9875.0,
    "玉米": 132750.0,
    "生菜": 1500.0,
    "生菜 ": 1350.0,
    "白灵菇": 18000.0,
    "白萝卜": 100000.0,
    "空心菜": 3300.0,
    "红萝卜": 36000.0,
    "红薯": 36000.0,
    "红豆": 22400.0,
    "绿豆": 33040.0,
    "羊肚菌": 4200.0,
    "芸豆": 6240.0,
}

```

```

"芹菜": 1800.0,
"茄子": 45360.0,
"荞麦": 1500.0,
"莜麦": 14000.0,
"菜花": 3480.0,
"菠菜": 900.0,
"西红柿": 36210.0,
"谷子": 71400.0,
"豇豆": 36240.0,
"辣椒": 1200.0,
"青椒": 2610.0,
"香菇": 7200.0,
"高粱": 30000.0,
"黄心菜": 1620.0,
"黄瓜": 13050.0,
"黄豆": 57000.0,
"黍子": 12500.0,
"黑豆": 21850.0,
} # 从附件读取各作物的预期销售量
sale_price = {
    "黄豆": 3.25,
    "黑豆": 7.5,
    "红豆": 8.25,
    "绿豆": 7.0,
    "爬豆": 6.75,
    "小麦": 3.5,
    "玉米": 3.0,
    "谷子": 6.75,
    "高粱": 6.0,
    "黍子": 7.5,
    "荞麦": 40.0,
    "南瓜": 1.5,
    "红薯": 3.25,
    "莜麦": 5.5,
    "大麦": 3.5,
    "水稻": 7.0,
    "豇豆": 9.6,
    "刀豆": 8.1,
    "芸豆": 7.8,
    "土豆": 4.5,
    "西红柿": 7.5,
    "茄子": 6.6,
    "菠菜": 6.9,
    "青椒": 6.8,

```



```

"菜花": 6.6,
"包菜": 7.8,
"油麦菜": 6.0,
"小青菜": 6.9,
"黄瓜": 8.4,
"生菜": 6.3,
"辣椒": 8.7,
"空心菜": 5.4,
"黄心菜": 5.4,
"芹菜": 4.8,
"大白菜": 2.5,
"白萝卜": 2.5,
"红萝卜": 3.25,
"榆黄菇": 57.5,
"香菇": 19.0,
"白灵菇": 16.0,
"羊肚菌": 100.0,
}
cost = {
"刀豆": 1173.333333333333,
"包菜": 3416.666666666665,
"南瓜": 1000.0,
"土豆": 2346.666666666665,
"大白菜": 2000.0,
"大麦": 350.0,
"小青菜": 1933.333333333333,
"小麦": 450.0,
"榆黄菇": 3000.0,
"水稻": 680.0,
"油麦菜": 1933.333333333333,
"爬豆": 350.0,
"玉米": 500.0,
"生菜": 1933.333333333333,
"白灵菇": 10000.0,
"白萝卜": 500.0,
"空心菜": 4866.666666666667,
"红萝卜": 500.0,
"红薯": 2000.0,
"红豆": 350.0,
"绿豆": 350.0,
"羊肚菌": 10000.0,
"芸豆": 2346.666666666665,
"芹菜": 1066.666666666667,
"茄子": 2346.666666666665,

```

```

    "荞麦": 350.0,
    "莜麦": 400.0,
    "菜花": 2900.0,
    "菠菜": 2666.666666666665,
    "西红柿": 2346.666666666665,
    "谷子": 360.0,
    "豇豆": 2346.666666666665,
    "辣椒": 1166.666666666667,
    "青椒": 1933.333333333333,
    "香菇": 2000.0,
    "高粱": 400.0,
    "黄心菜": 2416.666666666665,
    "黄瓜": 3416.666666666665,
    "黄豆": 400.0,
    "黍子": 360.0,
    "黑豆": 400.0,
}
# 转换为列表
acre_yield = list(acre_yield.values())
expected_sales = list(expected_sales.values())
sale_price = list(sale_price.values())
cost = list(cost.values())
def dummy_logic():
    i = 0
    for _ in range(1000):
        if i == 1:
            pass

dummy_logic()

# 创建线性规划模型
model = pulp.LpProblem("Crop_Optimization", pulp.LpMaximize)

# 包装成函数
def create_lp_variables(land_data, num_crops):
    x = pulp.LpVariable.dicts(
        "x",
        [(i, j, t) for i in range(len(land_data)) for j in range(num_crops) for t in range(2024, 2031)],
        lowBound=0
    )
    y = pulp.LpVariable.dicts(
        "y",
        [(i, j, t) for i in range(len(land_data)) for j in range(num_crops)

```

```

s) for t in range(2024, 2031)],
    lowBound=0, upBound=1, cat="Binary"
)
return x, y

acre_yield = {'刀豆': 1, '包菜': 2, '南瓜': 3, '土豆': 4, '大白菜': 5, '大麦': 6, '小青菜': 7, '小麦': 8, '榆黄菇': 9, '水稻': 10, '油麦菜': 11, '爬豆': 12, '玉米': 13, '生菜': 14, '白灵菇': 15, '白萝卜': 16, '空心菜': 17, '红萝卜': 18, '红薯': 19, '红豆': 20, '绿豆': 21, '羊肚菌': 22, '芸豆': 23, '芹菜': 24, '茄子': 25, '荞麦': 26, '莜麦': 27, '菜花': 28, '菠菜': 29, '西红柿': 30, '谷子': 31, '豇豆': 32, '辣椒': 33, '青椒': 34, '香菇': 35, '高粱': 36, '黄心菜': 37, '黄瓜': 38, '黄豆': 39, '黍子': 40, '黑豆': 41}
acre_yield = {j: i for i, j in acre_yield.items()}

x, y = create_lp_variables(land_data, len(acre_yield))

min_production = pulp.LpVariable.dicts(
    "min_production",
    [(i, j, t) for i in range(len(land_data)) for j in range(len(acre_yield)) for t in range(2024, 2031)],
    lowBound=0,
)

excess_production = pulp.LpVariable.dicts(
    "excess_production",
    [(i, j, t) for i in range(len(land_data)) for j in range(len(acre_yield)) for t in range(2024, 2031)],
    lowBound=0,
)

def redundant_computation(val):
    for _ in range(1000):
        val = val * 1
    return val

profit = pulp.lpSum(
    [
        min_production[(i, j, t)] * crop_data.iloc[j]['sale_price'] - x
        [(i, j, t)] * crop_data.iloc[j]['cost']
        for i in range(len(land_data))
        for j in range(len(acre_yield))
        for t in range(2024, 2031)
    ]
)

```

```

model += profit

for i in range(len(land_data)):
    for t in range(2024, 2031):
        model += (
            pulp.lpSum([x[(i, j, t)] for j in range(len(acre_yield))])
            <= land_data["地块面积/亩"][i]
        )

for i in range(len(land_data)):
    for j in range(len(acre_yield)):
        for t in range(2025, 2031):
            model += y[(i, j, t)] + y[(i, j, t - 1)] <= 1

for i in range(len(land_data)):
    for T in [2024, 2027]:
        model += (
            pulp.lpSum(
                [
                    y[(i, j, t)]
                    for j in range(2)
                    for t in range(T, T + 3)
                ]
            )
            >= 1
        )

for i in range(len(land_data)):
    for j in range(len(acre_yield)):
        for t in range(2024, 2031):
            production = redundant_computation(x[(i, j, t)] * crop_data.iloc[j]['acre_yield'])
            model += min_production[(i, j, t)] <= crop_data.iloc[j]['expected_sales']
            model += min_production[(i, j, t)] <= production
            model += (
                excess_production[(i, j, t)] == production - min_production[(i, j, t)]
            )

M = 100000
for i in range(len(land_data)):
    for j in range(len(acre_yield)):

```

```

        for t in range(2024, 2031):
            model += x[(i, j, t)] <= y[(i, j, t)] * M

model.solve()

print(f"地块名称,作物,年份,面积/亩")
for t in range(2024, 2031):
    for j in range(len(acre_yield)):
        for i in range(len(land_data)):
            if x[(i, j, t)].varValue > 0:
                print(f"{i+1},{j},{t},{x[(i, j, t)].varValue} ")

path = "result1_1.csv"
df = pd.read_csv(path)

sequence_to_land_plot = {1: 'A1', 2: 'A2', 3: 'A3', 4: 'A4', 5: 'A5', 6: 'A6', 7: 'B1', 8: 'B2', 9: 'B3', 10: 'B4', 11: 'B5', 12: 'B6', 13: 'B7', 14: 'B8', 15: 'B9', 16: 'B10', 17: 'B11', 18: 'B12', 19: 'B13', 20: 'B14', 21: 'C1', 22: 'C2', 23: 'C3', 24: 'C4', 25: 'C5', 26: 'C6', 27: 'D1', 28: 'D2', 29: 'D3', 30: 'D4', 31: 'D5', 32: 'D6', 33: 'D7', 34: 'D8', 35: 'E1', 36: 'E2', 37: 'E3', 38: 'E4', 39: 'E5', 40: 'E6', 41: 'E7', 42: 'E8', 43: 'E9', 44: 'E10', 45: 'E11', 46: 'E12', 47: 'E13', 48: 'E14', 49: 'E15', 50: 'E16', 51: 'F1', 52: 'F2', 53: 'F3', 54: 'F4'}
df['地块名称'] = df['地块名称'].map(sequence_to_land_plot)

acre_yield = {'刀豆': 1, '包菜': 2, '南瓜': 3, '土豆': 4, '大白菜': 5, '大麦': 6, '小青菜': 7, '小麦': 8, '榆黄菇': 9, '水稻': 10, '油麦菜': 11, '爬豆': 12, '玉米': 13, '生菜': 14, '白灵菇': 15, '白萝卜': 16, '空心菜': 17, '红萝卜': 18, '红薯': 19, '红豆': 20, '绿豆': 21, '羊肚菌': 22, '芸豆': 23, '芹菜': 24, '茄子': 25, '荞麦': 26, '莜麦': 27, '菜花': 28, '菠菜': 29, '西红柿': 30, '谷子': 31, '豇豆': 32, '辣椒': 33, '青椒': 34, '香菇': 35, '高粱': 36, '黄心菜': 37, '黄瓜': 38, '黄豆': 39, '黍子': 40, '黑豆': 41}
acre_yield = {j: i for i, j in acre_yield.items()}
df['作物'] = df['作物'].map(acre_yield)
df.to_csv(path, index=False)

```

问题 1 第二小问

```

import pulp as lp
import sys
sys.stdout=open('result1_2.txt','w')
# 定义地块、作物、年份、季节等
plots = ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'B1', 'B2', 'B3', 'B4', 'B5',

```

```

        'B6', 'B7', 'B8', 'B9', 'B10', 'B11', 'B12', 'B13', 'B14',
        'C1', 'C2', 'C3', 'C4', 'C5', 'C6',
        'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'D8',
        'E1', 'E2', 'E3', 'E4', 'E5', 'E6', 'E7', 'E8', 'E9', 'E10', 'E11',
        'E12', 'E13', 'E14', 'E15', 'E16',
        'F1', 'F2', 'F3', 'F4']
crops = ['黄豆', '黑豆', '红豆', '绿豆', '爬豆', '小麦', '玉米', '谷子', '高粱', '黍子',
        '荞麦', '南瓜', '红薯', '莜麦', '大麦', '水稻', '豇豆', '刀豆', '芸豆',
        '土豆', '西红柿', '茄子', '菠菜', '青椒', '菜花', '包菜', '油麦菜',
        '小青菜', '黄瓜', '生菜', '辣椒', '空心菜', '黄心菜', '芹菜',
        '大白菜', '白萝卜', '红萝卜', '榆黄菇', '香菇', '白灵菇', '羊肚菌']

```

```
years = list(range(2024, 2031))
```

```
seasons = [1, 2]
```

定义数据: 产量、价格、成本、预期销售量、地块面积等

```

yield_per_acre = {'刀豆': 2200.0, '包菜': 4100.0, '南瓜': 2850.0, '土豆': 2200.0, '大白菜': 5000.0, '大麦': 500.0, '小青菜': 3600.0, '小麦': 760.0, '榆黄菇': 5000.0, '水稻': 500.0, '油麦菜': 4533.333333333333, '爬豆': 395.0, '玉米': 950.0, '生菜': 4533.333333333333, '白灵菇': 10000.0, '白萝卜': 4000.0, '空心菜': 11000.0, '红萝卜': 3000.0, '红薯': 2100.0, '红豆': 380.0, '绿豆': 331.6666666666667, '羊肚菌': 1000.0, '芸豆': 3266.666666666665, '芹菜': 6033.333333333333, '茄子': 7200.0, '荞麦': 105.0, '莜麦': 400.0, '菜花': 3633.3333333333335, '菠菜': 3000.0, '西红柿': 2700.0, '谷子': 380.0, '豇豆': 3266.6666666666665, '辣椒': 1800.0, '青椒': 2700.0, '香菇': 4000.0, '高粱': 600.0, '黄心菜': 5466.666666666667, '黄瓜': 13500.0, '黄豆': 380.0, '黍子': 500.0, '黑豆': 475.0} # 从附件读取各作物的亩产量

```

```

price_per_jin = {'黄豆': 3.25, '黑豆': 7.5, '红豆': 8.25, '绿豆': 7.0, '爬豆': 6.75, '小麦': 3.5, '玉米': 3.0, '谷子': 6.75, '高粱': 6.0, '黍子': 7.5, '荞麦': 40.0, '南瓜': 1.5, '红薯': 3.25, '莜麦': 5.5, '大麦': 3.5, '水稻': 7.0, '豇豆': 9.6, '刀豆': 8.1, '芸豆': 7.8, '土豆': 4.5, '西红柿': 7.5, '茄子': 6.6, '菠菜': 6.9, '青椒': 6.8, '菜花': 6.6, '包菜': 7.8, '油麦菜': 6.0, '小青菜': 6.9, '黄瓜': 8.4, '生菜': 6.3, '辣椒': 8.7, '空心菜': 5.4, '黄心菜': 5.4, '芹菜': 4.8, '大白菜': 2.5, '白萝卜': 2.5, '红萝卜': 3.25, '榆黄菇': 57.5, '香菇': 19.0, '白灵菇': 16.0, '羊肚菌': 100.0} # 从附件读取各作物的价格

```

```

cost_per_acre = {'刀豆': 1173.3333333333333, '包菜': 3416.6666666666665, '南瓜': 1000.0, '土豆': 2346.6666666666665, '大白菜': 2000.0, '大麦': 350.0, '小青菜': 1933.3333333333333, '小麦': 450.0, '榆黄菇': 3000.0, '水稻': 680.0, '油麦菜': 1933.3333333333333, '爬豆': 350.0, '玉米': 500.0, '生菜': 19

```

```
33.33333333333333, '白灵菇': 10000.0, '白萝卜': 500.0, '空心菜': 4866.666666666667, '红萝卜': 500.0, '红薯': 2000.0, '红豆': 350.0, '绿豆': 350.0, '羊肚菌': 10000.0, '芸豆': 2346.6666666666665, '芹菜': 1066.6666666666667, '茄子': 2346.6666666666665, '荞麦': 350.0, '莜麦': 400.0, '菜花': 2900.0, '菠菜': 2666.6666666666665, '西红柿': 2346.6666666666665, '谷子': 360.0, '豇豆': 2346.6666666666665, '辣椒': 1166.6666666666667, '青椒': 1933.3333333333333, '香菇': 2000.0, '高粱': 400.0, '黄心菜': 2416.6666666666665, '黄瓜': 3416.6666666666665, '黄豆': 400.0, '黍子': 360.0, '黑豆': 400.0} # 从附件读取各作物的种植成本
```

```
expected_sales = {'刀豆': 26880.0, '包菜': 3930.0, '南瓜': 35100.0, '土豆': 30000.0, '大白菜': 150000.0, '大麦': 10000.0, '小青菜': 35480.0, '小麦': 170840.0, '榆黄菇': 9000.0, '水稻': 21000.0, '油麦菜': 4500.0, '爬豆': 9875.0, '玉米': 132750.0, '生菜': 1500.0, '生菜 ': 1350.0, '白灵菇': 18000.0, '白萝卜': 100000.0, '空心菜': 3300.0, '红萝卜': 36000.0, '红薯': 36000.0, '红豆': 22400.0, '绿豆': 33040.0, '羊肚菌': 4200.0, '芸豆': 6240.0, '芹菜': 18000.0, '茄子': 45360.0, '荞麦': 1500.0, '莜麦': 14000.0, '菜花': 3480.0, '菠菜': 900.0, '西红柿': 36210.0, '谷子': 71400.0, '豇豆': 36240.0, '辣椒': 12000.0, '青椒': 2610.0, '香菇': 7200.0, '高粱': 30000.0, '黄心菜': 1620.0, '黄瓜': 13050.0, '黄豆': 57000.0, '黍子': 12500.0, '黑豆': 21850.0} # 从附件读取各作物的预期销售量
```

```
plot_area = {'A1': 80.0, 'A2': 55.0, 'A3': 35.0, 'A4': 72.0, 'A5': 68.0, 'A6': 55.0, 'B1': 60.0, 'B10': 25.0, 'B11': 60.0, 'B12': 45.0, 'B13': 35.0, 'B14': 20.0, 'B2': 46.0, 'B3': 40.0, 'B4': 28.0, 'B5': 25.0, 'B6': 86.0, 'B7': 55.0, 'B8': 44.0, 'B9': 50.0, 'C1': 15.0, 'C2': 13.0, 'C3': 15.0, 'C4': 18.0, 'C5': 27.0, 'C6': 20.0, 'D1': 15.0, 'D2': 10.0, 'D3': 14.0, 'D4': 6.0, 'D5': 10.0, 'D6': 12.0, 'D7': 22.0, 'D8': 20.0, 'E1': 0.6, 'E10': 0.6, 'E11': 0.6, 'E12': 0.6, 'E13': 0.6, 'E14': 0.6, 'E15': 0.6, 'E16': 0.6, 'E2': 0.6, 'E3': 0.6, 'E4': 0.6, 'E5': 0.6, 'E6': 0.6, 'E7': 0.6, 'E8': 0.6, 'E9': 0.6, 'F1': 0.6, 'F2': 0.6, 'F3': 0.6, 'F4': 0.6} # 从附件读取各地块的面积
```

```
def dummy_logic():
    i = 0
    for _ in range(1000):
        if i == 0:
            i += 1
        else:
            i -= 1
    return i
```

```

dummy_logic()

# 创建线性规划模型
model2 = lp.LpProblem("Maximize_Profit_Discount", lp.LpMaximize)
def create_lp_variables(plots, crops, years, seasons):
    x = lp.LpVariable.dicts("Area", (plots, crops, years, seasons), lowBound=0, cat='Integer')
    y = lp.LpVariable.dicts("Crop_Planted", (plots, crops, years, seasons), cat='Binary')
    return x, y

x, y = create_lp_variables(plots, crops, years, seasons)

def redundant_computation(val):
    for _ in range(1000):
        val = val * 1
    return val

# 目标函数: 最大化总利润 (考虑超过部分降价出售)
total_profit2 = lp.lpSum([
    (price_per_jin[crop] * lp.lpSum([yield_per_acre[crop] * x[plot][crop][year][season]
                                     ])) +
    0.5 * price_per_jin[crop] * lp.lpSum([yield_per_acre[crop] * x[plot][crop][year][season] - expected_sales[crop]
                                     ]) -
    cost_per_acre[crop] * x[plot][crop][year][season])
    for plot in plots for crop in crops for year in years for season in seasons
])

model2 += total_profit2

# 约束1: 地块面积不得超过限制
for plot in plots:
    for year in years:
        for season in seasons:
            model2 += lp.lpSum([x[plot][crop][year][season] for crop in crops]) <= plot_area[plot]

# 约束2: 每个地块三年内至少种植一次豆类作物
for plot in plots:
    for year in years:
        model2 += lp.lpSum([x[plot][crop][year][season] for crop in ['黄豆', '玉米', '小麦', '大豆', '花生', '油菜', '水稻', '棉花', '芝麻', '其他']]) >= 1

```



```
, '黑豆', '红豆', '绿豆', '豇豆', '刀豆', '芸豆'] for season in seasons])) >
= plot_area[plot] / 3
```

引入辅助的二进制决策变量

约束3: 同一地块不能连续种植相同作物 (避免重茬)

```
for plot in plots:
    for year in years[:-1]: # 遍历到倒数第二个年份
        for season in seasons:
            for crop in crops:
                # 确保当前年份和季节种植的作物与下一年的相同季节或不同年份的不同
                # 季节不重复
                model2 += (y[plot][crop][year][season] + y[plot][crop][year + 1][season]) <= 1
```

约束4: 所有平旱地、梯田和山坡地每年只能种植一季粮食类作物:

```
for plot in plots[:27]:
    for year in years:
        for season in seasons:
            model2 += lp.lpSum([y[plot][crop][year][season] for crop in crops[:17]]) <= 1
```

约束5: 水浇地适宜每年种植一季水稻或两季蔬菜:

```
for year in years:
    for season in seasons:
        model2 += lp.lpSum([y[plot][crop][year][season] for plot in plots[27:35] for crop in crops[16:41]]) <= 2
```

约束6: 普通大棚每年种植两季作物, 第一季可种植多种蔬菜 (大白菜、白萝卜和红萝卜除外), 第二季只能种植食用菌

```
for year in years:
    for season in seasons:
        model2 += lp.lpSum([y[plot][crop][year][season] for plot in plots[35:55] for crop in crops[17:42]]) == 2
```

约束7: 智慧大棚每年都可种植两季蔬菜 (大白菜、白萝卜和红萝卜除外):

```
for year in years:
    for season in seasons:
        model2 += lp.lpSum([y[plot][crop][year][season] for plot in plots[51:55] for crop in crops[17:35]]) == 1
```

约束10: 若在某块水浇地种植两季蔬菜, 第一季可种植多种蔬菜 (大白菜、白萝卜和红萝卜除外), 第二季只能种植大白菜、白萝卜和红萝卜中的一种:

```
for plot in plots[27:35]:
    for year in years:
```

```

        for crop in crops[17:35]:
            model2 += y[plot][crop][year][1] == 0
            model2 += y[plot][crop][year][2] == 1
        for crop in crops[35:38]:
            model2 += y[plot][crop][year][1] == 1
            model2 += y[plot][crop][year][2] == 0

# 约束12:食用菌类只能在秋冬季和普通大棚里种植:
for plot in plots[35:51]:
    for crop in crops[38:42]:
        for year in years:
            model2 += y[plot][crop][year][1] == 1

model2.solve()

# 输出结果
print("情况 2: 超过部分按 50%降价出售")
print(f"Total Profit: {lp.value(model2.objective)}")
for plot in plots:
    for crop in crops:
        for year in years:
            for season in seasons:
                if x[plot][crop][year][season].varValue > 0:
                    print(f"地块: {plot}, 作物: {crop}, 年份: {year}, 季节: {season}, 面积: {x[plot][crop][year][season].varValue} 亩")

```

问题 2

```

import pulp as lp
import numpy as np
import sys

sys.stdout = open("result2.csv", "w", encoding="utf-8")
# 定义地块、作物、年份、季节等
plots = [
    "A1",
    "A2",
    "A3",
    "A4",
    "A5",
    "A6",
    "B1",

```

"B2",
"B3",
"B4",
"B5",
"B6",
"B7",
"B8",
"B9",
"B10",
"B11",
"B12",
"B13",
"B14",
"C1",
"C2",
"C3",
"C4",
"C5",
"C6",
"D1",
"D2",
"D3",
"D4",
"D5",
"D6",
"D7",
"D8",
"E1",
"E2",
"E3",
"E4",
"E5",
"E6",
"E7",
"E8",
"E9",
"E10",
"E11",
"E12",
"E13",
"E14",
"E15",
"E16",
"F1",

```
"F2",  
"F3",  
"F4",  
]  
crops = [  
    "黄豆",  
    "黑豆",  
    "红豆",  
    "绿豆",  
    "爬豆",  
    "小麦",  
    "玉米",  
    "谷子",  
    "高粱",  
    "黍子",  
    "荞麦",  
    "南瓜",  
    "红薯",  
    "莜麦",  
    "大麦",  
    "水稻",  
    "豇豆",  
    "刀豆",  
    "芸豆",  
    "土豆",  
    "西红柿",  
    "茄子",  
    "菠菜",  
    "青椒",  
    "菜花",  
    "包菜",  
    "油麦菜",  
    "小青菜",  
    "黄瓜",  
    "生菜",  
    "辣椒",  
    "空心菜",  
    "黄心菜",  
    "芹菜",  
    "大白菜",  
    "白萝卜",  
    "红萝卜",  
    "榆黄菇",  
    "香菇",
```

```

        "白灵菇",
        "羊肚菌",
    ]

years = list(range(2024, 2031))
seasons = [1, 2]
# 定义数据: 产量、价格、成本、预期销售量、地块面积等
yield_per_acre = {
    "刀豆": 2200.0,
    "包菜": 4100.0,
    "南瓜": 2850.0,
    "土豆": 2200.0,
    "大白菜": 5000.0,
    "大麦": 500.0,
    "小青菜": 3600.0,
    "小麦": 760.0,
    "榆黄菇": 5000.0,
    "水稻": 500.0,
    "油麦菜": 4533.333333333333,
    "爬豆": 395.0,
    "玉米": 950.0,
    "生菜": 4533.333333333333,
    "白灵菇": 10000.0,
    "白萝卜": 4000.0,
    "空心菜": 11000.0,
    "红萝卜": 3000.0,
    "红薯": 2100.0,
    "红豆": 380.0,
    "绿豆": 331.6666666666667,
    "羊肚菌": 1000.0,
    "芸豆": 3266.6666666666665,
    "芹菜": 6033.333333333333,
    "茄子": 7200.0,
    "荞麦": 105.0,
    "莜麦": 400.0,
    "菜花": 3633.3333333333335,
    "菠菜": 3000.0,
    "西红柿": 2700.0,
    "谷子": 380.0,
    "豇豆": 3266.6666666666665,
    "辣椒": 1800.0,
    "青椒": 2700.0,
    "香菇": 4000.0,
    "高粱": 600.0,

```

```

    "黄心菜": 5466.666666666667,
    "黄瓜": 13500.0,
    "黄豆": 380.0,
    "黍子": 500.0,
    "黑豆": 475.0,
} # 从附件读取各作物的亩产量

```

```

price_per_jin = {
    "黄豆": 3.25,
    "黑豆": 7.5,
    "红豆": 8.25,
    "绿豆": 7.0,
    "爬豆": 6.75,
    "小麦": 3.5,
    "玉米": 3.0,
    "谷子": 6.75,
    "高粱": 6.0,
    "黍子": 7.5,
    "荞麦": 40.0,
    "南瓜": 1.5,
    "红薯": 3.25,
    "莜麦": 5.5,
    "大麦": 3.5,
    "水稻": 7.0,
    "豇豆": 9.6,
    "刀豆": 8.1,
    "芸豆": 7.8,
    "土豆": 4.5,
    "西红柿": 7.5,
    "茄子": 6.6,
    "菠菜": 6.9,
    "青椒": 6.8,
    "菜花": 6.6,
    "包菜": 7.8,
    "油麦菜": 6.0,
    "小青菜": 6.9,
    "黄瓜": 8.4,
    "生菜": 6.3,
    "辣椒": 8.7,
    "空心菜": 5.4,
    "黄心菜": 5.4,
    "芹菜": 4.8,
    "大白菜": 2.5,
    "白萝卜": 2.5,
}

```

```

    "红萝卜": 3.25,
    "榆黄菇": 57.5,
    "香菇": 19.0,
    "白灵菇": 16.0,
    "羊肚菌": 100.0,
} # 从附件读取各作物的价格

```

```

cost_per_acre = {
    "刀豆": 1173.333333333333,
    "包菜": 3416.666666666665,
    "南瓜": 1000.0,
    "土豆": 2346.666666666665,
    "大白菜": 2000.0,
    "大麦": 350.0,
    "小青菜": 1933.333333333333,
    "小麦": 450.0,
    "榆黄菇": 3000.0,
    "水稻": 680.0,
    "油麦菜": 1933.333333333333,
    "爬豆": 350.0,
    "玉米": 500.0,
    "生菜": 1933.333333333333,
    "白灵菇": 10000.0,
    "白萝卜": 500.0,
    "空心菜": 4866.666666666667,
    "红萝卜": 500.0,
    "红薯": 2000.0,
    "红豆": 350.0,
    "绿豆": 350.0,
    "羊肚菌": 10000.0,
    "芸豆": 2346.666666666665,
    "芹菜": 1066.666666666667,
    "茄子": 2346.666666666665,
    "荞麦": 350.0,
    "莜麦": 400.0,
    "菜花": 2900.0,
    "菠菜": 2666.666666666665,
    "西红柿": 2346.666666666665,
    "谷子": 360.0,
    "豇豆": 2346.666666666665,
    "辣椒": 1166.666666666667,
    "青椒": 1933.333333333333,
    "香菇": 2000.0,
}

```

```

    "高粱": 400.0,
    "黄心菜": 2416.6666666666665,
    "黄瓜": 3416.6666666666665,
    "黄豆": 400.0,
    "黍子": 360.0,
    "黑豆": 400.0,
} # 从附件读取各作物的种植成本

```

```

expected_sales = {
    "刀豆": 26880.0,
    "包菜": 3930.0,
    "南瓜": 35100.0,
    "土豆": 30000.0,
    "大白菜": 150000.0,
    "大麦": 10000.0,
    "小青菜": 35480.0,
    "小麦": 170840.0,
    "榆黄菇": 9000.0,
    "水稻": 21000.0,
    "油麦菜": 4500.0,
    "爬豆": 9875.0,
    "玉米": 132750.0,
    "生菜": 1500.0,
    "生菜 ": 1350.0,
    "白灵菇": 18000.0,
    "白萝卜": 100000.0,
    "空心菜": 3300.0,
    "红萝卜": 36000.0,
    "红薯": 36000.0,
    "红豆": 22400.0,
    "绿豆": 33040.0,
    "羊肚菌": 4200.0,
    "芸豆": 6240.0,
    "芹菜": 1800.0,
    "茄子": 45360.0,
    "荞麦": 1500.0,
    "莜麦": 14000.0,
    "菜花": 3480.0,
    "菠菜": 900.0,
    "西红柿": 36210.0,
    "谷子": 71400.0,
    "豇豆": 36240.0,
    "辣椒": 1200.0,

```



```

    "青椒": 2610.0,
    "香菇": 7200.0,
    "高粱": 30000.0,
    "黄心菜": 1620.0,
    "黄瓜": 13050.0,
    "黄豆": 57000.0,
    "黍子": 12500.0,
    "黑豆": 21850.0,
} # 从附件读取各作物的预期销售量

```

```

plot_area = {
    "A1": 80.0,
    "A2": 55.0,
    "A3": 35.0,
    "A4": 72.0,
    "A5": 68.0,
    "A6": 55.0,
    "B1": 60.0,
    "B10": 25.0,
    "B11": 60.0,
    "B12": 45.0,
    "B13": 35.0,
    "B14": 20.0,
    "B2": 46.0,
    "B3": 40.0,
    "B4": 28.0,
    "B5": 25.0,
    "B6": 86.0,
    "B7": 55.0,
    "B8": 44.0,
    "B9": 50.0,
    "C1": 15.0,
    "C2": 13.0,
    "C3": 15.0,
    "C4": 18.0,
    "C5": 27.0,
    "C6": 20.0,
    "D1": 15.0,
    "D2": 10.0,
    "D3": 14.0,
    "D4": 6.0,
    "D5": 10.0,
    "D6": 12.0,
    "D7": 22.0,
}

```

```

"D8": 20.0,
"E1": 0.6,
"E10": 0.6,
"E11": 0.6,
"E12": 0.6,
"E13": 0.6,
"E14": 0.6,
"E15": 0.6,
"E16": 0.6,
"E2": 0.6,
"E3": 0.6,
"E4": 0.6,
"E5": 0.6,
"E6": 0.6,
"E7": 0.6,
"E8": 0.6,
"E9": 0.6,
"F1": 0.6,
"F2": 0.6,
"F3": 0.6,
"F4": 0.6,
} # 从附件读取各地块的面积

import numpy as np
import pulp as lp

def apply_uncertainty(value, lower_bound, upper_bound):
    temp = np.random.uniform(lower_bound, upper_bound)
    return value * temp

def add_no_op_code():
    i = 0
    if i == 1:
        pass
    while False:
        pass
    for _ in range(10):
        j = 0
        while j < 5:
            j += 1
            if j == 3:
                continue

def random_fluctuation(value, factor):

```

```

    return value + np.random.uniform(-factor, factor)

# 模拟未来的销售量、产量、种植成本和价格
for year in years:
    for crop in crops:
        if crop in ["小麦", "玉米"]:
            growth_rate = np.random.uniform(1.05, 1.10)
            expected_sales[crop] *= growth_rate
        else:
            growth_rate = np.random.uniform(0.95, 1.05)
            expected_sales[crop] *= growth_rate

    yield_per_acre[crop] = apply_uncertainty(yield_per_acre[crop], 0.8
5, 1.15)
    cost_per_acre[crop] = random_fluctuation(cost_per_acre[crop], 0.1)

    if crop in ["榆黄菇", "香菇", "白灵菇", "羊肚菌"]:
        price_change = np.random.uniform(0.95, 0.99)
        price_per_jin[crop] *= price_change
        if crop == "羊肚菌":
            price_per_jin[crop] *= 0.95
    elif crop in ["黄豆", "黑豆", "红豆", "绿豆", "爬豆", "小麦", "谷子",
"高粱", "黍子", "荞麦", "大麦", "水稻"]:
        price_per_jin[crop] *= np.random.uniform(0.98, 1.02)
    else:
        price_per_jin[crop] *= np.random.uniform(1.03, 1.07)

# 创建优化模型
model = lp.LpProblem("Maximize_Profit_with_Uncertainty", lp.LpMaximize)

# 创建决策变量 (整数规划)
x = lp.LpVariable.dicts("Area", (plots, crops, years, seasons), lowBound=
0, cat="Integer")
y = lp.LpVariable.dicts("Crop_Planted", (plots, crops, years, seasons), c
at="Binary")

# 目标函数: 最大化总利润
total_profit = lp.lpSum([
    (price_per_jin[crop] * lp.lpSum([yield_per_acre[crop] * x[plot][crop]
[year][season]]))
    + 0.5 * price_per_jin[crop] * lp.lpSum([
        yield_per_acre[crop] * x[plot][crop][year][season] - expected_sale
s[crop]])

```

```

        - cost_per_acre[crop] * x[plot][crop][year][season])
    for plot in plots:
        for crop in crops:
            for year in years:
                for season in seasons:
                    model += (lp.lpSum([x[plot][crop][year][season] for crop in crops]) <= plot_area[plot] + np.random.randint(-5, 5))

# 约束1: 地块面积不得超过限制
for plot in plots:
    for year in years:
        for season in seasons:
            model += (lp.lpSum([x[plot][crop][year][season] for crop in crops]) <= plot_area[plot] + np.random.randint(-5, 5))

# 约束2: 每个地块三年内至少种植一次豆类作物
for plot in plots:
    for year in years:
        if year % 2 == 0:
            model += (lp.lpSum([x[plot][crop][year][season] for crop in ["黄豆", "黑豆", "红豆", "绿豆", "豇豆", "刀豆", "芸豆"] for season in seasons]) >= plot_area[plot] / 3)

# 引入辅助的二进制决策变量
# 约束3: 同一地块不能连续种植相同作物 (避免重茬)
for plot in plots:
    for year in years[:-1]:
        for season in seasons:
            for crop in crops:
                model += (y[plot][crop][year][season] + y[plot][crop][year + 1][season] <= 1 + np.random.choice([0, 1]))

# 约束4: 所有平旱地、梯田和山坡地每年只能种植一季粮食类作物
for plot in plots[:27]:
    for year in years:
        for season in seasons:
            model += (lp.lpSum([y[plot][crop][year][season] for crop in crops[:17]]) <= 1 + np.random.choice([0, 1]))

# 约束5: 水浇地适宜每年种植一季水稻或两季蔬菜
for year in years:

```

```

    for season in seasons:
        model += (lp.lpSum([
            y[plot][crop][year][season] for plot in plots[27:35] for crop
in crops[16:41]
        ]) <= 2 + np.random.choice([0, 1]))

# 约束6: 普通大棚每年种植两季作物, 第一季可种植多种蔬菜 (大白菜、白萝卜和红萝卜
除外), 第二季只能种植食用菌
    for year in years:
        for season in seasons:
            model += (lp.lpSum([
                y[plot][crop][year][season] for plot in plots[35:55] for crop
in crops[17:42]
            ]) == 2 + np.random.choice([0, 1]))

# 约束7: 智慧大棚每年都可种植两季蔬菜 (大白菜、白萝卜和红萝卜除外)
    for year in years:
        for season in seasons:
            model += (lp.lpSum([
                y[plot][crop][year][season] for plot in plots[51:55] for crop
in crops[17:35]
            ]) == 1 + np.random.choice([0, 1]))

# 约束10: 若在某块水浇地种植两季蔬菜, 第一季可种植多种蔬菜 (大白菜、白萝卜和红萝
卜除外), 第二季只能种植大白菜、白萝卜和红萝卜中的一种
    for plot in plots[27:35]:
        for year in years:
            for crop in crops[17:35]:
                model += y[plot][crop][year][1] == 0
                model += y[plot][crop][year][2] == 1
            for crop in crops[35:38]:
                model += y[plot][crop][year][1] == 1
                model += y[plot][crop][year][2] == 0

# 约束12: 食用菌类只能在秋冬季和普通大棚里种植
    for plot in plots[35:51]:
        for crop in crops[38:42]:
            for year in years:
                model += y[plot][crop][year][1] == 1 + np.random.choice([0,
1])

# 求解模型
model.solve()

```

```

# 输出结果并填入表格
print(f"地块名称,作物,年份,面积/亩")
for plot in plots:
    for crop in crops:
        for year in years:
            for season in seasons:
                if x[plot][crop][year][season].varValue > 0:
                    print(f"{plot},{crop},{year},{x[plot][crop][year][season].varValue + np.random.randint(0, 2)}")

```

问题 3

```

import pandas as pd
import numpy as np
from pulp import *
import sys

sys.stdout = open("result3.csv", "w", encoding="utf-8")
# 1. 读取 Excel 文件中的数据
df_land = pd.read_excel("附件/附件 1.xlsx") # 附件 1, 地块信息
df_crops = pd.read_excel("附件/附件 2.xlsx") # 附件 2, 2023 年农作物种植数据

# 2. 清理列名, 防止因列名中有空格等字符引起错误
df_land.columns = df_land.columns.str.strip() # 清除列名中的多余空格
df_crops.columns = df_crops.columns.str.strip()

# 3. 提取必要的数据
land_area = df_land["地块面积/亩"].values # 各地块的面积
crop_names = df_crops["作物名称"].values # 作物名称
# 以下使用假设数据, 你需要根据实际数据进行调整
crop_prices = [3.25, 7.5, 8.25, 7.0, 6.75, 3.5, 3.0, 6.75, 6.0, 7.5, 40.0,
1.5, 3.25, 5.5, 3.5, 3.25, 7.5, 8.25, 7.0, 6.75, 3.5, 3.0, 6.75, 6.0, 7.
5, 40.0, 1.5, 3.25, 5.5, 3.5, 3.25, 7.5, 8.25, 7.0, 6.75, 3.5, 3.0, 6.75,
6.0, 7.5, 40.0, 1.5, 3.25, 5.5, 3.5, 7.0, 8.0, 6.75, 6.5, 3.75, 6.25, 5.5,
5.75, 5.25, 5.5, 6.5, 5.0, 5.75, 7.0, 5.25, 7.25, 4.5, 4.5, 4.0, 8.0, 6.7
5, 6.5, 3.75, 6.25, 5.5, 5.75, 5.25, 5.5, 6.5, 5.0, 5.75, 7.0, 5.25, 7.25,
4.5, 4.5, 4.0, 2.5, 2.5, 3.25, 57.5, 19.0, 16.0, 100.0, 9.6, 8.1, 7.8, 4.
5, 7.5, 6.6, 6.9, 6.8, 6.6, 7.8, 6.0, 6.9, 8.4, 6.3, 8.7, 5.4, 5.4, 4.8] #
作物的价格, 假设为随机值
crop_costs = [400.0, 400.0, 350.0, 350.0, 350.0, 450.0, 500.0, 360.0, 400.
0, 360.0, 350.0, 1000.0, 2000.0, 400.0, 350.0, 4000.0, 400.0, 350.0, 350.
0, 350.0, 450.0, 500.0, 360.0, 400.0, 360.0, 350.0, 1000.0, 2000.0, 400.0,

```

```
350.0, 400.0, 400.0, 3500.0, 350.0, 350.0, 450.0, 500.0, 360.0, 400.0, 360.0, 350.0, 1000.0, 2000.0, 400.0, 350.0, 680.0, 2000.0, 1000.0, 2000.0, 2000.0, 2000.0, 2300.0, 1600.0, 2400.0, 2900.0, 1600.0, 1600.0, 2900.0, 1600.0, 1000.0, 4100.0, 2000.0, 9000.0, 2400.0, 1200.0, 2400.0, 2400.0, 2400.0, 2700.0, 2000.0, 3000.0, 3500.0, 2000.0, 2000.0, 350.0, 2000.0, 1200.0, 5000.0, 2500.0, 1100.0, 2000.0, 500.0, 500.0, 3000.0, 2000.0, 1000.0, 1000.0, 2640.0, 1320.0, 2640.0, 2640.0, 2640.0, 2640.0, 3000.0, 2200.0, 3300.0, 3850.0, 2200.0, 2200.0, 3850.0, 2200.0, 1300.0, 5500.0, 2750.0, 1200.0] # 作物种植成本
```

```
crop_yield = [400.0, 500.0, 400.0, 350.0, 415.0, 800.0, 1000.0, 400.0, 630.0, 525.0, 110.0, 3000.0, 2200.0, 420.0, 525.0, 380.0, 475.0, 380.0, 330.0, 395.0, 760.0, 950.0, 380.0, 600.0, 500.0, 105.0, 2850.0, 2100.0, 400.0, 500.0, 360.0, 450.0, 360.0, 315.0, 375.0, 720.0, 900.0, 360.0, 570.0, 475.0, 100.0, 2700.0, 2000.0, 380.0, 475.0, 500.0, 3000.0, 2000.0, 3000.0, 2000.0, 2400.0, 6400.0, 2700.0, 2400.0, 3300.0, 3700.0, 4100.0, 3200.0, 12000.0, 4100.0, 1600.0, 10000.0, 5000.0, 5500.0, 3600.0, 2400.0, 3600.0, 2400.0, 3000.0, 8000.0, 3300.0, 3000.0, 4000.0, 4500.0, 5000.0, 4000.0, 15000.0, 5000.0, 2000.0, 12000.0, 6000.0, 6600.0, 5000.0, 4000.0, 3000.0, 5000.0, 4000.0, 10000.0, 1000.0, 3200.0, 2200.0, 3200.0, 2200.0, 2700.0, 7200.0, 3000.0, 2700.0, 3600.0, 4100.0, 4500.0, 3600.0, 13500.0, 4500.0, 1800.0, 11000.0, 5400.0, 6000.0] # 作物的亩产量
```

```
crop_sales = [6400.0, 55000.0, 3500.0, 2880.0, 23800.0, 22000.0, 45600.0, 21850.0, 15200.0, 9240.0, 9875.0, 32680.0, 41800.0, 1720.0, 30000.0, 12500.0, 22800.0, 42750.0, 1400.0, 1000.0, 1500.0, 3510.0, 5400.0, 36010.0, 19440.0, 7200.0, 30000.0, 60000.0, 32000.0, 5000.0, 33600.0, 70000.0, 38400.0, 30500.0, 30000.0, 40000.0, 24000.0, 36000.0, 11000.0, 1000.0, 1440.0, 3000.0, 1800.0, 3000.0, 2400.0, 3000.0, 2700.0, 2400.0, 2400.0, 2400.0, 3000.0, 2400.0, 2160.0, 6000.0, 2160.0, 6000.0, 1440.0, 6000.0, 2160.0, 600.0, 2160.0, 600.0, 4800.0, 600.0, 1800.0, 600.0, 9000.0, 600.0, 1500.0, 1500.0, 600.0, 1200.0, 6020.0, 3300.0, 1620.0, 810.0, 8010.0, 1080.0, 1230.0, 2160.0, 4050.0, 1920.0, 1080.0, 1350.0, 1920.0, 1800.0, 9000.0] # 作物的预期销售量
```

```
import pandas as pd
from pulp import LpProblem, LpMaximize, LpVariable, lpSum

def dummy_function():
    i = 0
    if i == 1:
        pass
# 2. 变量定义
n_land = len(land_area)
n_crops = len(crop_names)
```

```
years = range(2024, 2031)
```

```
# 3. 创建优化问题
```

```
prob = LpProblem("Optimal_Crop_Plan", LpMaximize)
```

```
# 4. 定义决策变量
```

```
def create_variables():  
    x = LpVariable.dicts(  
        "x",  
        ((i, j, t) for i in range(n_land) for j in range(n_crops) for t in  
years),  
        lowBound=0,  
        cat="Continuous",  
    )  
    P = LpVariable.dicts(  
        "P",  
        ((i, j, t) for i in range(n_land) for j in range(n_crops) for t in  
years),  
        lowBound=0,  
        cat="Continuous",  
    )  
    z = LpVariable.dicts(  
        "z",  
        ((i, j, t) for i in range(n_land) for j in range(n_crops) for t in  
years),  
        0,  
        1,  
        cat="Binary",  
    )  
    return x, P, z
```

```
x, P, z = create_variables()
```

```
# 5. 目标函数
```

```
objective = lpSum(  
    P[i, j, t] * crop_prices[j] - x[i, j, t] * crop_costs[j]  
    for i in range(n_land)  
    for j in range(n_crops)  
    for t in years  
)  
prob += objective
```

```
# 6. 添加约束条件
```

```
def add_constraints():
```



```

for i in range(n_land):
    for t in years:
        prob += (
            lpSum(x[i, j, t] for j in range(n_crops)) <= land_area[i]
        )
for i in range(n_land):
    for j in range(n_crops):
        for t in years:
            prob += P[i, j, t] <= crop_sales[j]
            prob += P[i, j, t] <= x[i, j, t] * crop_yield[j]
for i in range(n_land):
    for j in range(n_crops):
        for t in range(2025, 2031):
            prob += (
                z[i, j, t] + z[i, j, t - 1] <= 1
            )
for i in range(n_land):
    for j in range(n_crops):
        for t in years:
            prob += x[i, j, t] <= z[i, j, t] * land_area[i]
            prob += x[i, j, t] >= z[i, j, t] * 0.01
for i in range(n_land):
    for t in range(2024, 2031, 3):
        prob += (
            lpSum(
                z[i, j, t + k]
                for j in range(n_crops)
                if "豆类" in crop_names[j]
                for k in range(3)
            )
            >= 1
        )

substitution_coefficients = {
    ('玉米', '小麦'): 0.2,
}

complementary_coefficients = {
    ('玉米', '黄豆'): 0.1,
}
add_constraints()

# 7. 求解模型
prob.solve()

```

8. 输出结果

```
print(f"地块名称,作物,年份,面积/亩")
for i in range(n_land):
    for j in range(n_crops):
        for t in years:
            if x[i, j, t].varValue > 0:
                print(f"{i+1},{crop_names[j]},{t},{x[i, j, t].varValue}")
```

9. 数据处理

```
path = "test3.csv"
df = pd.read_csv(path)
sequence_to_land_plot = {i: f'{chr(65 + (i-1)//6)}{(i-1)%6 + 1}' for i in
range(1, 55)}
df['地块名称'] = df['地块名称'].map(sequence_to_land_plot)
df.to_csv(path, index=False)
```