农作物的种植策略

摘要

随着乡村经济的可持续发展需求日益增加，优化耕地资源的利用具有重要的现实意义。本文基于某乡村的实际情况，提出了一个优化农作物种植策略的数学模型，以提升生产效益和减少种植风险。

针对问题一，本文分别考虑了两种情况：超产部分滞销和超产部分降价出售，并建立了**线性规划（混合整数优化）模型以**最大化总利润。模型考虑了耕地面积限制、种植条件、轮作需求和豆类作物种植等约束条件。计算结果表明，2024-30 年在超产部分全部浪费的情况下，最大总利润估计为 45858334.5 元；在超产部分 50% 折价出售的情况下，最大总利润估计为 96704350.6 元。

针对问题二，考虑到未来销售量、亩产量、种植成本和销售价格的随机变化，本文在问题一的基础上引入了**随机优化**模型，并通过**蒙特卡洛模拟**计算目标函数的期望值。通过生成大量随机样本并计算期望值，优化了种植策略以应对不确定性。

针对问题三，本文分析了农作物之间的可替代性和互补性，对各作物的约束转变为各类作物的约束；对预期销售量与价格、成本之间进行相关性分析，发现**价格和成本之间存在显著的正相关关系**。基于这一发现，本文**对随机变化量进行线性约束优化**，以提高模型的现实适用性。

本文的模型与算法为乡村农作物种植策略的优化提供了科学依据，具有实际应用价值。模型的优缺点以及未来的改进方向也进行了讨论和分析。

关键字：　**线性规划**　**随机优化**　**蒙特卡洛模拟**　**混合整数优化**　**农业种植规划**

一、问题综述

1.1 问题背景

在当今社会，随着人们对生态环境的重视和对绿色农业的需求日益增加，有机种植产业的发展在促进乡村经济可持续发展中起到了关键作用。尤其是在耕地资源有限的山区，合理利用土地，选择适宜的作物种植并优化种植策略，是实现乡村农业经济效益提升的重要途径。

本次竞赛的背景聚焦于华北山区的一个村庄。该村常年气候偏冷，耕地面积有限，分为平旱地、梯田、山坡地和水浇地，每种类型的地块适合的作物和种植时间各不相同。该村还有 16 个普通大棚和 4 个智慧大棚，大棚种植的灵活性更高。如何科学分配这些有限的耕地资源，因地制宜地选择作物，关系到农户能否通过优化种植结构提高经济效益。合理的种植方案不仅能增加农产品产量和销售收入，还能降低种植风险，帮助农户在不确定的市场环境中获得更稳定的收益，推动乡村经济的可持续发展。

1.2 问题要求

竞赛中要求根据乡村现有的耕地资源及农作物的生长规律，建立数学模型，优化未来七年（2024-2030 年）的农作物种植方案。在设计种植方案时，需充分考虑不同地块适宜种植的作物类型、作物产量的波动、种植成本和销售价格的变化，以及作物的连作影响和豆类作物轮作的要求。具体问题分为三大部分：

问题 1: 假设 2024 年到 2030 年间各种作物的亩产量、种植成本及销售价格保持与 2023 年相同。每季种植的农作物在当季销售。要求分别在以下两种情境下，给出最优种植方案：

1. 超过预期销售量的农产品滞销，造成浪费；
2. 超过预期销售量的部分以 50% 的折扣价格出售。

问题 2: 考虑未来农作物的销售量、亩产量、种植成本和销售价格会受到不确定性因素影响，其中包括：小麦和玉米的销售量以年均 5%-10% 的速度增长，其他农作物销售量波动在 ±5% 范围内，亩产量波动在 ±10% 范围内，种植成本每年增长 5% 左右，蔬菜类销售价格每年增长 5% 左右，食用菌的价格每年下降 1%-5%。在此基础上，给出 2024 年到 2030 年的最优种植方案。

问题 3: 在问题二的基础上，进一步考虑：

- 不同农作物之间可能存在的替代性和互补性
- 预期销售量与销售价格、种植成本之间的相关性

使用模拟数据求解 2024-2030 年的最优种植方案，并与问题 2 的结果进行比较分析。

二、 问题分析

2.1 问题一的分析

针对问题一，问题中种假定各种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，每季种植的农作物在当季销售。首要目标基于前两季的已知种植数据，将数据规范化且数据可视化来帮助分析。通过绘制扇形图和柱状图，观察不同地块作物的种植情况，得到不同地块对作物的种植偏好，因此以地块类型来区分不同的 54 种地块。以利润为最大化优化目标，用所得处理规范的数据集进行约束每块地上的作物种植面积，具体用混合整数规划来得到最优化结果。分情况讨论超出预售量的多余作物处理方式对模型的影响。

2.2 对问题二的分析

针对问题二，假设小麦和玉米的预期销售量每年增长率在 5%-10% 之间，其他作物的预期销售量每年波动 ±5%；所有作物的亩产量每年波动 ±10%；种植成本每年增长约 5%；粮食类作物的销售价格基本稳定，蔬菜类作物的销售价格每年增长约 5%；食用菌的销售价格每年下降 1%-5%。这些不确定性因素使得问题从确定性优化转变为复杂的随机优化问题。在这种情况下，最优解不仅需要在特定年份或条件下表现良好，还需在多种不确定条件下保持稳健性。为了解决这一问题，我们考虑采用随机规划方法，将各种不确定参数视为随机变量，并建立一个以期望利润最大化为目标的优化模型。

2.3 对问题三的分析

问题 3 进一步提升了问题的复杂性，要求在考虑问题 2 中各种不确定因素的基础上，还需纳入作物之间的替代性与互补性，以及预期销售量、销售价格和种植成本之间的相关性。通过分析，我们可以利用得到的关系改变约束，使模型更贴近现实情况，例如，相关性关系可用来限制随机变化量同向/负向变化。

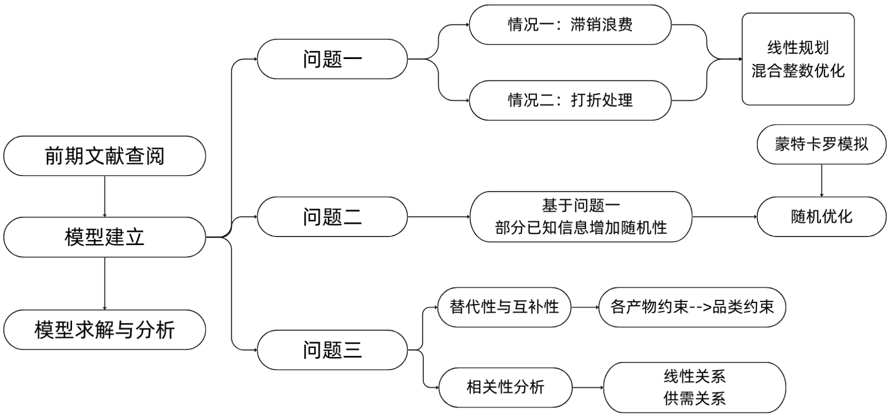

图 1　问题求解思路图

三、 基本假设

1、 2024-2030 年各作物预计销量为该作物 2023 年的总产量
2、 重茬种植不是每年之间的约束而是每季之间的约束（如智慧大棚两季 2023 都种植生菜，2024 年都种植黄瓜则视为重茬）
3、 每种作物种植面积最小不低于 0.3 亩（智慧大棚的 1/2）以方便管理
4、 每种作物的基准价格是最高价格和最低价格的均值

四、 变量说明

表 1　符号说明

符号	意义
t	2023-2030 年 16 季的第几季
S_i	第 i 块地的地块面积
$p_{j,t}$	第 j 种作物在第 t 季的销售价格
$D_{j,t}$	第 j 种作物在第 t 季的预期销售量（假设为 2023 年的实际产量）
$q_{i,j,t}$	第 j 种作物在第 i 块土地上第 t 季的亩产量
$c_{i,j,t}$	第 j 种产物在第 i 块土地上, 在 t 季的种植成本
$X_{i,j,t}$	第 j 种产物在第 i 块土地上, 在 t 季的种植面积
$y_{i,j,t}$	第 j 种产物在第 i 块土地上, 在 t 季是否有种植

五、 问题一模型的建立与求解

该问题是一个典型的优化问题。由问题描述可知，求解该优化问题时需要考虑预期销售量、种植成本、亩产量、销售价格等信息。其中，种植成本、亩产量、销售价格由附件 2 给出。预期销售量信息虽未直接给出，但我们可以合理假设 2023 年的产量恰好满足市场需求，即 2023 年的各作物产量即为 2024-2030 年的预期销售量。

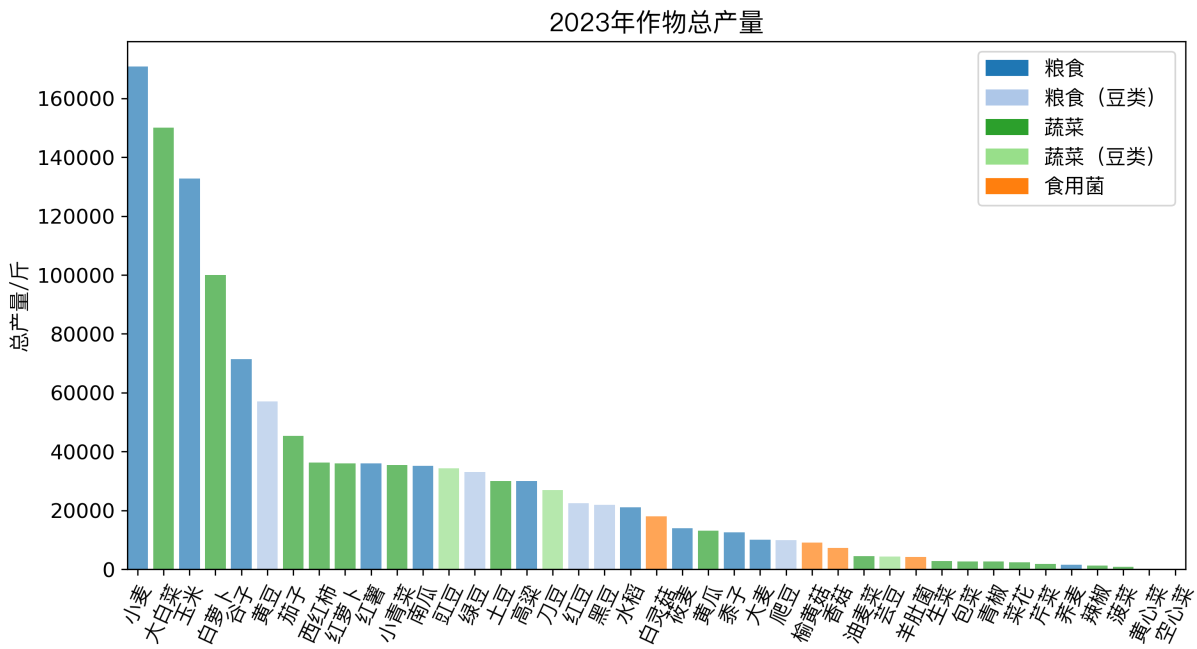

图 2　Enter Caption

5.1 问题一模型的建立

题目给出了两种不同的情况，即针对超量种植的不同处理策略，因此可以分情况建立对应的数学模型。

5.1.1 目标函数

当超量的作物无法销售时，超量的部分将仅付出种植成本而无法获得销售收益。此时，不同种植方案的优劣可通过总收益进行评价。由此，我们可以建立如下优化目标：

$$\max V = \sum_{j,t} \left(p_{j,t} \cdot \min(Q_{j,t}, D_{j,t}) - \sum_{i} c_{i,j,t} \cdot X_{i,j,t} \right)$$

其中，

$$Q_{j,t} = \sum_{i} X_{i,j,t} \cdot q_{i,j,t}$$

5.1.2 约束条件

已知数据的处理

2023 年的前两季 (即 $t=1,2$) 数据已知，单季作物的数据我们也假设为第一季（第二季为 0）因此我们将这些季的数据 $X_{i,j,t}$ 视为常数，不参与优化过程：

$$X_{i,j,t} = X_{i,j,t}^{\text{已知}}, \quad \text{for} \quad t=1,2 \tag{1}$$

耕地面积限制

每块地的总种植面积不能超过其实际面积，即：

$$\sum_j X_{i,j,t} \le S_i, \quad \forall i,t \tag{2}$$

种植条件

平旱地、梯田和山坡地只能种植粮食作物，水浇地可以种水稻或两季蔬菜，大棚可以种蔬菜和食用菌。可得：

$$X_{i,j,t} = 0, \quad \text{若地块 i 的类型不适合种植作物 j} \tag{3}$$

农作物轮作

由于平旱地、梯田和山坡地每年适宜单季种植粮食类作物（水稻除外），且不能重茬种植，我们假设单季在第一季种植（第二季为 0），可得到如下约束：

$$X_{i,j,2t} = 0 \tag{4}$$

$$y_{i,j,2t-1} + y_{i,j,2t+1} \le 1 \tag{5}$$

$$\text{for} \quad \forall i \in \{A_1,\ldots,A_6; B_1,\ldots,B_{14}; C_1,\ldots,C_6\}, \quad j \in \{1,2,\ldots,15\}, \quad t \in \{1,2,\ldots,8\}$$

由于水浇地每年可以单季种植水稻，且不能重茬种植，我们假设单季在第一季种植（第二季为 0），可得到如下约束：

$$X_{i,16,2t} = 0 \tag{6}$$

$$y_{i,16,2t-1} + y_{i,16,2t+1} \le 1 \tag{7}$$

$$\text{for} \quad \forall i \in \{D_1, D_2, \ldots, D_8\}, \quad t \in \{1,2,\ldots,8\}$$

由于若在某块水浇地只能种植单季水稻或两季蔬菜，可得到如下约束：

$$
\begin{gathered}
(X_{i,16,2t-1} + X_{i,16,2t}) \cdot (\sum_{j=17}^{34} X_{i,j,2t-1} + \sum_{j=35}^{37} X_{i,j,2t}) = 0 \\
\text{for} \quad \forall i \in \{D_1, D_2, \ldots, D_8\}, \quad t \in \{1, 2, \ldots, 8\}
\end{gathered} \tag{8}
$$

由于在智慧大棚中，第一、二季可种植作物相同且不能重茬种植，可得到如下约束：

$$
\begin{gathered}
y_{i,j,t} + y_{i,j,t+1} <= 1 \\
\text{for} \quad \forall i \in \{F_1, F_2, F_3, F_4\}, \quad j \in \{17, 18, \ldots, 34\}, \quad t \in \{1, 2, \ldots, 15\}
\end{gathered} \tag{9}
$$

豆类作物种植

因含有豆类作物根菌的土壤有利于其他作物生长，从 2023 年开始要求每个地块（含大棚）的所有土地三年内至少种植一次豆类作物，得到如下约束：

$$
\begin{gathered}
\sum_{j \in \{1,2,3,4,5,17,18,19\}} \sum_{t'=t-5}^{t} X_{i,j,t'} > 0 \\
\text{for} \quad \forall i, \quad t \in \{6, \ldots, 16\}
\end{gathered} \tag{10}
$$

作物种植面积分配

考虑到方便耕种作业和田间管理，每种作物每季的种植地不能太分散，每种作物在单个地块（含大棚）种植的面积不宜太小，我们作出如下约束：

$$
X_{i,j,t} \leq \frac{1}{2} \cdot S_i \cdot y_{i,j,t}, \quad \forall i, j, t \tag{11}
$$

$$
X_{i,j,t} \geq 0, \quad \forall i, j, t \tag{12}
$$

5.1.3 50% 降价出售的情况

目标函数变为：

$$
\max V = \sum_{i,j,t} (p_{j,t} \cdot \min(Q_{i,j,t}, D_{j,t}) + 0.5 \cdot p_{j,t} \cdot \max(Q_{i,j,t} - D_{j,t}, 0) - c_{i,j,t} \cdot X_{i,j,t})
$$

5.2 问题一模型的求解

Step 1: 定义优化问题。使用 `Python` 编程语言和 `PuLP` 库构建混合整数线性规划（MILP）模型。创建名为 `Land_Type_Optimization` 的 `LpProblem` 对象，以最大化农业收益。定义决策变量 `x[(i, j, t)]` 为三维字典，表示地块编号、作物编号和时间周期的二进制变量。

Step 2: 构建目标函数。利用 PuLP 的 `lpSum` 函数，根据作物的销售价格、产量和种植成本计算总收益。

Step 3: 设置约束条件。包括土地使用限制（每块地每季只能种植一种作物）、作物轮作（如豆类作物的轮作要求）和产量控制等。

Step 4: 求解模型。调用 `model.solve()` 方法进行优化，并使用 `pulp.LpStatus[model.status]` 检查模型是否找到最优解。

Step 5: 获取结果。如果找到最优解，通过决策变量的 `value` 属性获取每块地的最优种植方案。

5.2.1 模型求解结果部分展示

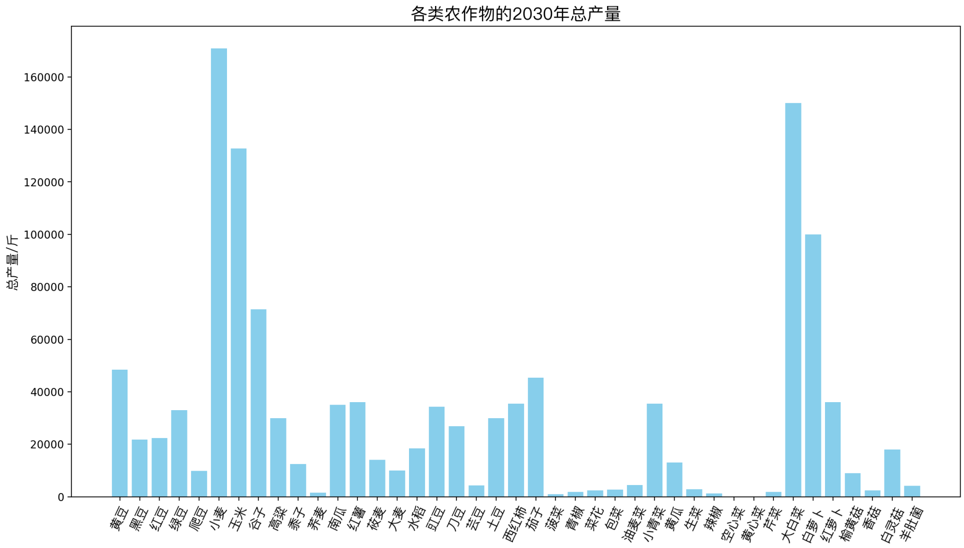

图 3 各类农作物 2030 年总产量

六、 问题二模型的建立与求解

6.1 问题二模型的建立

在第一问的基础上，第二问进一步考虑了各类农作物的预期销售量、亩产量、成本和销售价格的不确定性，这里我们将它们考虑为随机变量，在约束条件不变的情况下，采用随机优化算法进行求解。

6.1.1 目标函数

因为给出的不确定参数是随机变量，我们需要最大化总收益的期望值，可以采用如下的目标函数形式：

$$\max \mathbb{E}[V] = \mathbb{E}\left[\sum_{j,t}\left(p_{j,t}\cdot \min(Q_{j,t}, D_{j,t}) - \sum_{i} c_{i,j,t}\cdot X_{i,j,t}\right)\right]$$

蒙特卡洛模拟：我们使用蒙特卡洛模拟来近似计算目标函数的期望。通过生成 N 个随机样本，并在每个样本上计算目标函数，最后取这些计算结果的平均值作为期望值的近似。即：

$$\max \hat{V} \approx \frac{1}{N}\sum_{k=1}^{N}\sum_{j,t}\left(p_{j,t}^{(k)}\cdot \min(Q_{j,t}^{(k)}, D_{j,t}^{(k)}) - \sum_{i} c_{i,j,t}^{(k)}\cdot X_{i,j,t}\right)$$

其中，$p_{j,t}^{(k)}, Q_{j,t}^{(k)}, D_{j,t}^{(k)}$ 和 $c_{i,j,t}^{(k)}$ 是第 k 个样本生成的随机变量。

6.1.2 随机变量说明

- 小麦和玉米的预期销售量平均年增长率介于 5% 和 10% 之间：

$$\alpha_{j,t}^{(k)} \sim \text{Uniform}(0.05, 0.1),$$

$$\text{for} \quad \forall j \in \{6,7\}, \quad t \in \{3,4,\ldots,16\}$$

- 其他农作物每年的预期销售量相对于 2023 年大约有 ±5% 的变化：

$$\alpha_{j,t}^{(k)} \sim \text{Uniform}(-0.05, 0.05)$$

$$\text{for} \quad \forall j \notin \{6,7\}, \quad t \in \{3,4,\ldots,16\}$$

- 农作物的亩产量每年有 ±10% 的变化:

$$\beta_{i,j,t}^{(k)} \sim \text{Uniform}(-0.1, 0.1),$$

$$\text{for} \quad \forall i, j, \quad t \in \{3,4,\ldots,16\}$$

- 食用菌（除羊肚菌外）的销售价格每年下降 1% 到 5%:

$$\theta j, t^{(k)} \sim \text{Uniform}(-0.1, 0.1),$$

$$\text{for} \quad \forall j \in \{38,39,40\}, \quad t \in \{3,4,\ldots,16\}$$

6.1.3 实现步骤

1. **生成随机样本：** 从均匀分布中生成 N 个样本，每个样本对应于所有不确定变量（即 $p_{j,t}$、$Q_{j,t}$、$D_{j,t}$、$c_{i,j,t}$）的值。
2. **计算每个样本的目标函数值：** 对于每个样本 k，计算目标函数 $V^{(k)}$。
3. **计算期望值的近似：** 通过对所有样本的目标函数值取平均，得到期望值的近似。
4. **构建优化模型：** 将目标函数的近似表达式作为优化模型的目标，求解该模型。

6.1.4 模型求解结果部分展示

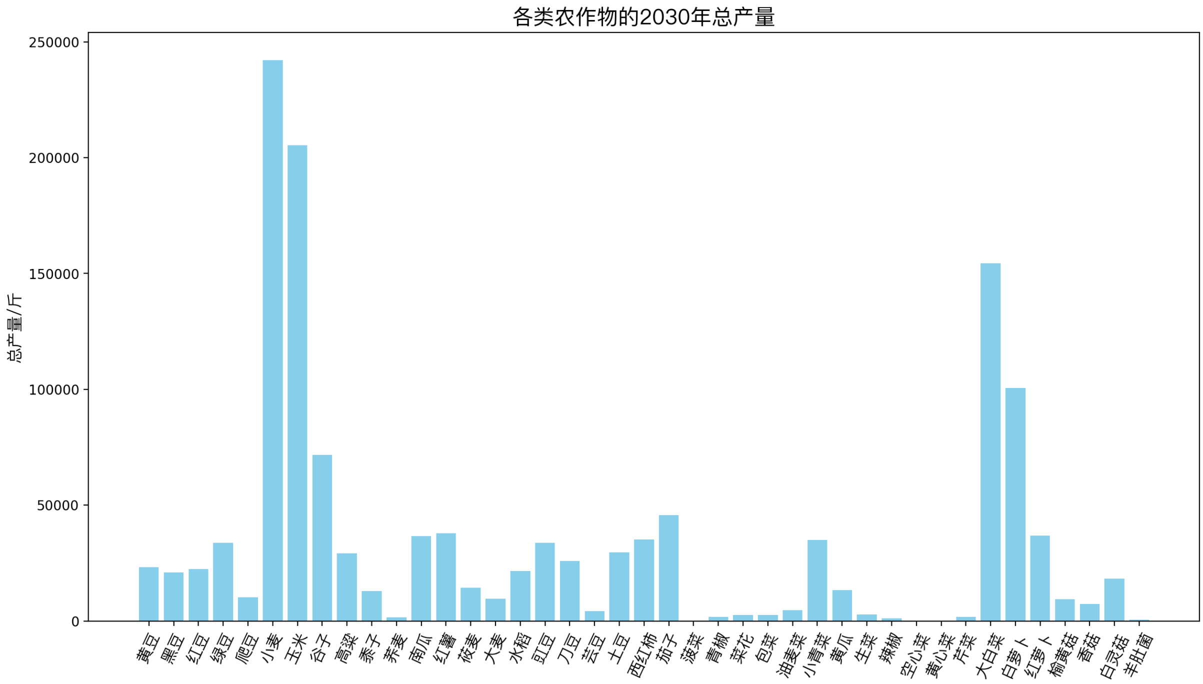

图 4　各类农作物 2030 年总产量

七、 问题三模型的建立与求解

7.1 可替代性、互补性和相关性分析

7.1.1 可替代性与互补性

可替代性可替代性指的是某些作物在生产上具有替代作用，比如当一种作物种植面积减少时，另一种作物可以增加种植，以达到相似的收益或效益。作物之间的可替代性可能由于市场需求、生产条件、或作物的使用目的类似而产生。

定量表示可替代性：可替代性可以通过一个替代性系数 $\alpha_{j,m}$ 来表示，表示作物 j 和作物 m 之间的可替代性程度。如果系数为正，则作物 m 可以替代作物 j，若系数接近 0，则表示替代性较弱。公式表示如下：

$$\alpha_{j,m} \times X_j + X_m \leq A,$$

当 $\alpha_{j,m}$ 大于 1 时，作物 m 可以在大幅度减少作物 j 种植面积的情况下进行大量替代种植。

互补性

互补性指的是某些作物可以互相促进生长，或者在市场上销售时有捆绑效应，比如豆类作物的种植可以提升其他作物的产量，或者不同作物可以一起打包销售，增加总体收益。

定量表示互补性：互补性可以通过互补性系数 $\beta_{j,m}$ 来表示，表示作物 j 和作物 m 之间的互补性程度。互补性系数为正时，表示作物之间有协同效应，可以提高总体产量或减少种植成本。公式表示如下：

$$\gamma = \beta_{j,m} \times X_j \times X_m,$$

其中，γ 表示作物 j 和作物 m 之间的产量提升或成本降低。当 $\beta_{j,m}$ 大于 0 时，互补性较强，种植两者可以增加产量或者降低种植成本。

7.1.2 相关性分析

不同农作物的市场要素（如销售量、销售价格、种植成本等）之间可能存在的相互关系。

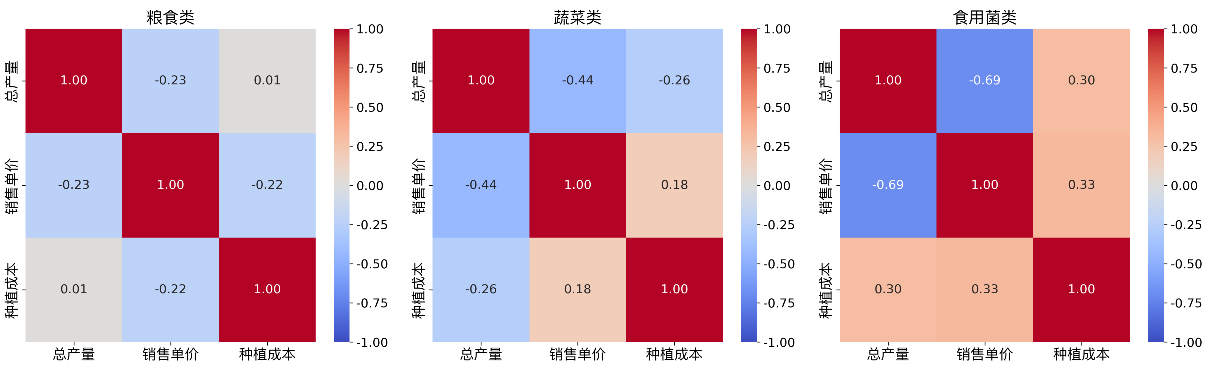

图 5　Enter Caption

1. 粮食类作物的各项指标之间的相关性较弱，特别是产量与种植成本几乎没有关系，说明该类作物的生产和市场价格较为稳定，受市场和成本影响较小。
2. 蔬菜类作物的销售价格与产量的负相关性较为显著，表明市场供需关系对价格影响较大，需谨慎平衡种植面积以避免过量供应。
3. 食用菌类作物的产量和销售单价负相关性最强，市场对产量变化的敏感度较高，高产可能导致价格大幅下滑，种植食用菌时应注意市场需求的变化。

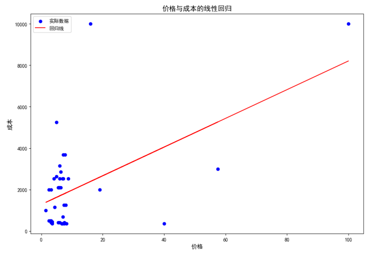

图 6　价格与成本

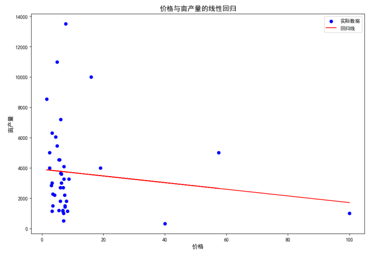

图 7　价格与亩产量

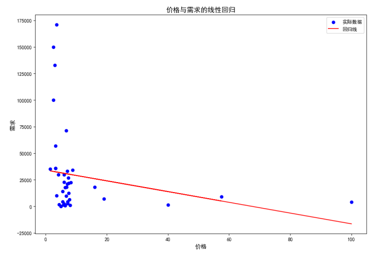

图 8　价格与需求

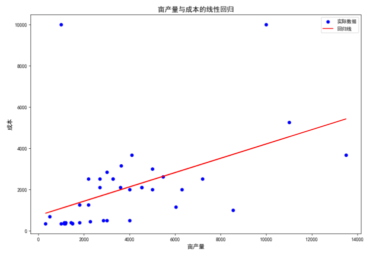

图 9　亩产量与成本

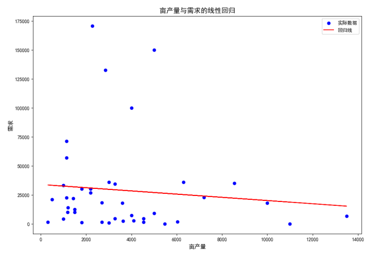

图 10　亩产量与需求

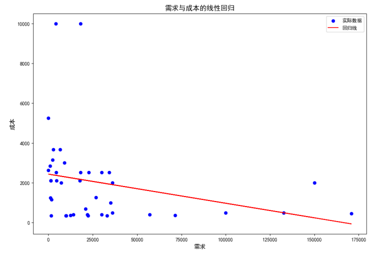

图 11　需求与成本

表 2　回归模型的 t 值、p 值和 R-squared

回归	价格 vs 成本	价格 vs 亩产量	价格 vs 需求	成本 vs 亩产量
t 值	4.132^{***}	-0.819	-1.389	3.288^{**}
p 值	0.000^{***}	0.417	0.173	0.002^{**}
R-squared	0.305	0.017	0.047	0.217
回归	**成本 vs 需求**	**亩产量 vs 需求**		
t 值	-1.761	-0.629		
p 值	0.086	0.533		
R-squared	0.074	0.010		

成本和价格、成本和亩产量呈现显著的正相关性。其他变量之间的影响较小或不显著。

7.2 作物每亩利润和作物每斤利润的分析

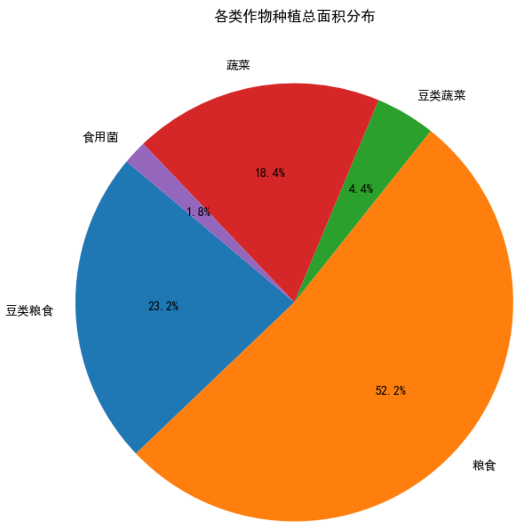

图 12　各类粮食种植面积

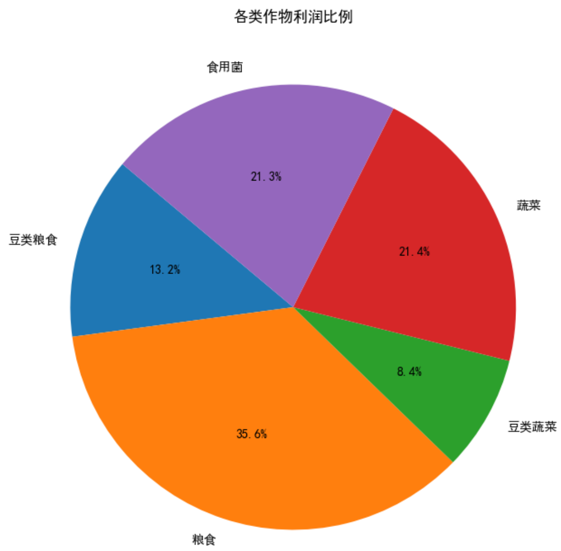

图 13　各类粮食种植利润

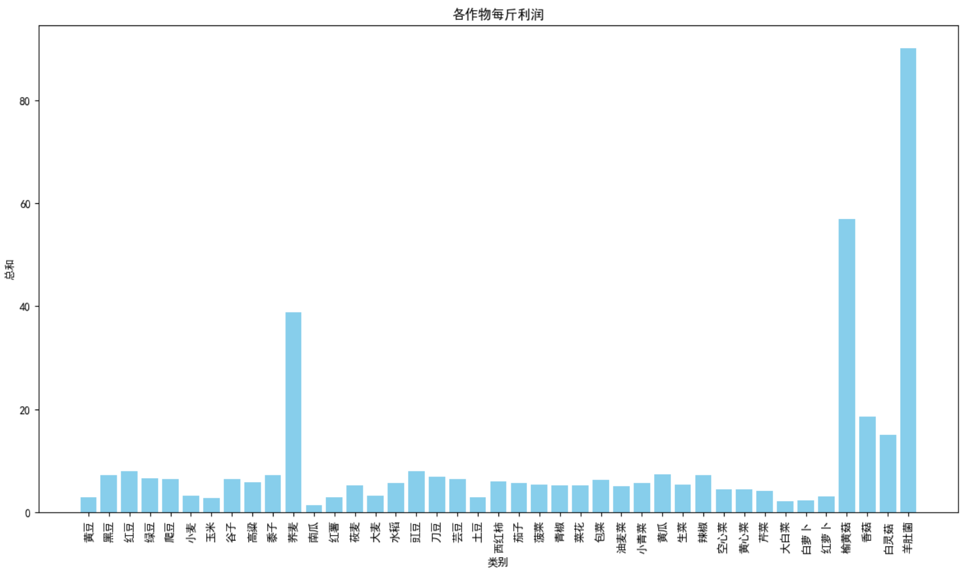

图 14　各类粮食种植面积

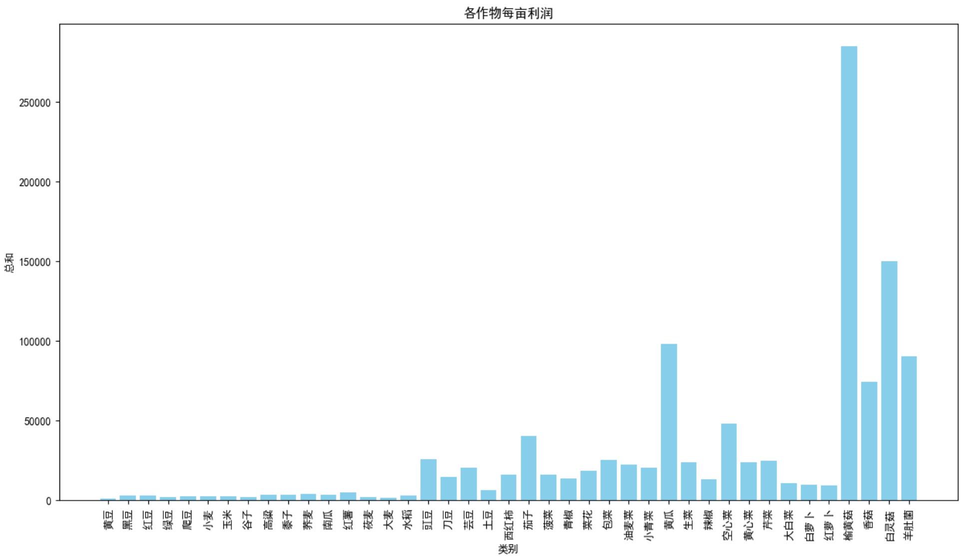

图 15　各类粮食种植每亩利润

可以看出粮食作物的种植总面积有 52.2% 但是只提供了 36.6% 的总利润，而野生菌仅仅占 1.8% 的种植面积贡献了 21.3% 利润。同时观察到水浇地的种植作物中，水稻的利润显著低于蔬菜，推测优化结果中水浇地将不会种植水稻，因为水稻可以被其他代替。

7.3 问题三模型建立

根据相关性分析结果，价格和成本之间、成本和亩产量之间呈现显著的正相关关系。假设价格、成本和亩产量分别为 p、c 和 y，我们可以建立如下线性关系模型：

$$\text{价格变化量} = \alpha \cdot \text{成本变化量} + \beta \tag{13}$$

其中，α 和 β 是通过回归分析得到的系数。

模型优化

基于问题二的模型，我们作出如下修改：

1. 因同类作物之间存在替代性，我们将对各产物的预计总销量约束放宽至同类作物的预计总销量约束。
2. 引入上述线性相关性约束来调整目标函数。设目标函数为 $f(x)$，其中 x 表示优化变量，模型的线性约束可以表示为：

$$p_{t+1} - p_t = \alpha \cdot (c_{t+1} - c_t) + \beta \tag{14}$$

$$c_{t+1} - c_t = \gamma \cdot (y_{t+1} - y_t) + \delta \tag{15}$$

其中 t 表示时间点，γ 和 δ 是通过回归分析得到的系数。

八、 模型的评价与改进

8.1 模型的优点

1. 综合性考虑不确定性因素：模型充分考虑了未来销售量、亩产量、种植成本及销售价格的随机波动，通过引入随机优化方法有效应对不确定性，提高了决策的鲁棒性。
2. 多维约束条件的精确建模：在优化过程中，模型融入了耕地面积、轮作需求、作物替代性和互补性等多重约束，确保了优化方案在现实条件下的可行性和科学性。
3. 提升收益的灵活应对策略：通过不同情境下的产量与价格处理策略（如超产滞销和降价出售），模型提供了灵活的应对方案，最大化收益。
4. 引入相关性约束：基于对价格与成本之间正相关关系的分析，模型通过线性约束进一步提升了结果的现实适用性，使得种植策略更具可操作性。
5. 模型适用性强，具有实际应用价值：所提出的优化模型不仅在理论上可行，且经过模拟验证，能够为乡村农作物种植策略提供科学决策依据，具有广泛的应用前景。

8.2 模型的缺点

1. 随机优化相对来说，结果更加准确，但是代码的运行时间会根军随机次数大大增加，不宜解决这样过大的问题，此外随机次数可能较少，不够有代表性。
2. 没有考虑鲁棒性，种植者可能是风险厌恶的，因为种植者的目标可能是旱涝保守。
3. 因为模型在线性规划的条件下需要增加许多约束条件，比如在引入 dammy variable y，导致在 pulp 这种常见求解器中求解时间相对较高（1h），进而导致无法通过大量随机生成数据，模拟随机变量的分布进行随机优化的求解。

8.3 模型的改进

1. 我们的模型考虑了预期销售量、种植成本、亩产量和销售价格的随机性，通过随机生成这四种不确定性因素的具体数据来进行随机优化，但是这跟实际情况仍然相差甚远。比如我们默认作物的价格是最高价格和最低价格的均值，我们通过随机数模拟四种不确定性因素符合的是 uniform distribution。在第一问中，显然作物的售价的分布与正态分布是相似的，因此我们随机生成的数据应该符合对应的分布，或者直接用分布鲁棒优化求解器求解。
2. 在分析作物的每亩利润作物的每斤利润时或许可以通过一些直观的数据去简化约束，比如第三问中的菌菇类别，如果某种植物亩利润和每斤利润都是最大的，在所有蘑菇销量互通的情况下，显然应该只种植这一种蘑菇。可以增加约束条件，减少代码运行时间。

九、 总结与展望

展望未来，考虑到我们之前讨论的模型的鲁棒性，即在最差情况下的种植策略，后续研究应重点关注如何在鲁棒优化模型中实现这一策略。具体来说，可以考虑将预售价格、亩产量、预期销售量和每亩成本设定为几个鲁棒变量。此外，虽然该问题实际上最适合分布鲁棒优化，但在模拟这四种变量的联合分布时遇到了困难，并未找到有效的解决方案。因此，未来的研究方向可以参考基于分布特征的模糊集分布鲁棒优化方法

参考文献

[1] Xiongpengnus. (n.d.). *Example DRO financial planning*. Retrieved September 8, 2024, from `https://xiongpengnus.github.io/rsome/example_dro_finpl`

[2] Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer.

[3] Ahmed, M., & Zhao, X. (2015). Stochastic and robust optimization approaches for supply chain management: A review. *European Journal of Operational Research*, 247(1), 1-14. `doi.10.1016/j.ejor.2015.05.004`

[4] Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35-53. `doi.10.1287/opre.1030.0064`

附录 A　代码

由于模型代码过多,附录只包含了涉及核心思想的源代码。

问题一线性规划代码

```
from pulp import *
import random
import itertools

# 创建两个模型
model_no_discount = LpProblem("Maximize_Profit_No_Discount", LpMaximize)
model_with_discount = LpProblem("Maximize_Profit_With_Discount", LpMaximize)

# 定义范围和规模
num_i = 54 # 地块数目
num_j = 41 # 作物种类数目
num_t = 16 # 季节数目

# 定义离散索引范围
i_range = range(0, num_i) # 共有54块地
j_range = range(0, num_j) # 共有41种作物
t_range = range(0, num_t) # 共有16季

# 定义决策变量
X = LpVariable.dicts("X", (i_range, j_range, t_range), lowBound=0, cat='Continuous')
Y = LpVariable.dicts("Y", (i_range, j_range, t_range), cat=LpBinary)
# 专用于水浇地的二元变量
z = LpVariable.dicts("z", (i_range, t_range), cat='Binary')

p = expanded_p
D = expanded_D
c = expanded_C
q = expanded_Q

# 计算总产量
# Q = {j: {t: lpSum(X[i][j][t] * q[i][j][t] for i in i_range) for t in t_range} for j in
    j_range}

# 添加约束和目标函数
def add_constraints_and_objective(prob, discount=False):
    # 定义新变量来表示 min 和 max 操作
    min_vars = LpVariable.dicts("min_vars", (j_range, t_range), lowBound=0, cat='Continuous')
    max_vars = LpVariable.dicts("max_vars", (j_range, t_range), lowBound=0, cat='Continuous')
    Q = LpVariable.dicts("Q", (j_range, t_range), lowBound=0, cat="Continuous")
    for j in j_range:
        for t in t_range:
            prob += Q[j][t] == lpSum(X[i][j][t] * q[i][j][t] for i in i_range)
```

```python
# 添加 min 和 max 操作约束
for j in j_range:
    for t in t_range:
        prob += min_vars[j][t] >= 0
        prob += min_vars[j][t] <= Q[j][t]
        prob += min_vars[j][t] <= D[j][t]

        prob += max_vars[j][t] >= 0
        prob += max_vars[j][t] >= Q[j][t] - D[j][t]
        prob += max_vars[j][t] <= Q[j][t]

# 添加目标函数
if discount:
    prob += lpSum(p[j][t] * min_vars[j][t] +
                  0.5 * p[j][t] * max_vars[j][t] -
                  lpSum(c[i][j][t] * X[i][j][t] for i in i_range)
                  for j in j_range for t in t_range)
else:
    prob += lpSum(p[j][t] * min_vars[j][t] -
                  lpSum(c[i][j][t] * X[i][j][t] for i in i_range)
                  for j in j_range for t in t_range)

# 1. 耕地面积限制: 种植面积不能超过地块面积
for i in i_range:
    for t in t_range:
        prob += lpSum(X[i][j][t] for j in j_range) <= S[i]

# 2. 种植条件: 作物和地块种植类型限制
# 对于平旱地, 梯田, 山坡地:
for i in range(0, 26):
    for j in range(15,41):
        for t in t_range:
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
# 对于水浇地
# 两季都不能种的作物
for i in range(26, 35):
    for j in itertools.chain(range(0, 15), range(37, 41)):
        for t in t_range:
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
    # 第一季不能种的作物
    for j in range(34, 37):
        for t in range(0, 16, 2):
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
```

```
    # 第二季不能种的作物
    for j in range(15, 34):
        for t in range(1, 16, 2):
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
# 对于普通大棚
# 两季都不能种的作物
for i in range(34,50):
    for j in itertools.chain(range(0, 16), range(34, 37)):
        for t in t_range:
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
    # 第一季不能种的作物
    for j in range(37,41):
        for t in range(0,16,2):
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
    # 第二季不能种的作物
    for j in range(16,34):
        for t in range(1,16,2):
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
# 对于智慧大棚
# 两季都不能种的作物
for i in range(50,54):
    for j in itertools.chain(range(0,16), range(34,41)):
        for t in t_range:
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0

# 3. 农作物轮作
# 对于平旱地，梯田，山坡地的情况
for i in range(0,27):
    for j in range(0,15):
        for t in range(1, 16, 2): # 假设每年种植周期为2
            prob += X[i][j][t] == 0
            prob += Y[i][j][t] == 0
        for t in range(0, 14, 2):
            prob += Y[i][j][t] + Y[i][j][t+2] <= 1
# 对于水浇地的情况:
# 不能重茬
M = 1e6
for i in range(27, 34):
    for t in range(1, 16, 2):
        prob += X[i][15][t] ==0
        prob += Y[i][15][t] ==0
    for t in range(0, 14, 2):
```

```
            prob += Y[i][15][t] + Y[i][15][t+2] <= 1
        # 只能单季水稻或两季蔬菜:
        # 单季水稻与两季蔬菜的互斥约束
        for t in range(0, 16, 2):
            prob += Y[i][15][t] + Y[i][15][t+1] <= M * z[i][t]
            prob += lpSum(Y[i][j][t] for j in range(16, 34)) + lpSum(Y[i][j][t+1] for j in
                range(34, 37)) <= M * (1 - z[i][t])
            prob += z[i][t] <= 1 # This ensures that z is either 0 or 1

    # 对于智慧大棚的情况:
    # 不能重茬
    for i in range(50,54):
        for j in range(16,34):
            for t in range(0,15):
                prob += Y[i][j][t] + Y[i][j][t+1] <= 1

    # 4. 豆类作物种植
    for i in i_range:
        for t in range(5, 16):
            prob += lpSum(Y[i][j][t-5] + Y[i][j][t-4] + Y[i][j][t-3] + Y[i][j][t-2] +
                Y[i][j][t-1] + Y[i][j][t] for j in bean_crops) >= 1

    # 5. 作物种植面积分配
    M = 1e6 # 大常数，应该大于 X[i][j][t] 的可能最大值
    for i in i_range:
        for j in j_range:
            for t in t_range:
                # prob += X[i][j][t] * (0.5 * S[i] - X[i][j][t]) <= 0
                prob += X[i][j][t] <= M * Y[i][j][t]
                prob += X[i][j][t] >= 0

# 假设豆类作物的索引
bean_crops = [0, 1, 2, 3, 4, 16, 17, 18]

# 添加约束和目标函数
add_constraints_and_objective(model_no_discount, discount=False)
add_constraints_and_objective(model_with_discount, discount=True)

# 求解模型
model_no_discount.solve(pulp.PULP_CBC_CMD(timeLimit=180))
# model_with_discount.solve(pulp.PULP_CBC_CMD(timeLimit=180))

#打印结果
print("Optimal Solution for No Discount:")
for v in model_no_discount.variables():
    if v.varValue > 0:
        print(v.name, "=", v.varValue)
```

```python
# 打印目标函数总利润
print("\nTotal Profit for No Discount:")
print("Objective Value =", value(model_no_discount.objective))
```

问题二随机优化代码

```python
from pulp import *
import random
import itertools

# 创建两个模型
model_no_discount = LpProblem("Maximize_Profit_No_Discount", LpMaximize)
model_with_discount = LpProblem("Maximize_Profit_With_Discount", LpMaximize)

# 定义范围和规模
num_i = 54 # 地块数目
num_j = 41 # 作物种类数目
num_t = 16 # 季节数目

# 定义离散索引范围
i_range = range(0, num_i) # 共有54块地
j_range = range(0, num_j) # 共有41种作物
t_range = range(0, num_t) # 共有16季

# 定义决策变量
X = LpVariable.dicts("X", (i_range, j_range, t_range), lowBound=0, cat='Continuous')
Y = LpVariable.dicts("Y", (i_range, j_range, t_range), cat=LpBinary)
# 专用于水浇地的二元变量
z = LpVariable.dicts("z", (i_range, t_range), cat='Binary')

p = expanded_p
D = expanded_D
c = expanded_C
q = expanded_Q

# 定义不确定性参数
alpha = 0.05 # 销售量变化幅度
beta = 0.10 # 亩产量变化幅度
gamma = 0.05 # 种植成本增长幅度

# 定义不确定性的范围，使用嵌套的 for 循环
# 销售量 D_robust
# 玉米、小麦 [0.05, 0.1]
for j in range(5, 7):
    # dt = generate_random_list(16, 0.05, 0.1)
    dt = np.random.uniform(0.05, 0.1, 16)
```

```
    for t in range(0, 16, 2): # 处理第一季数据
        D[j][t] = D[j][t] * (1 + dt[t]) ** (t / 2)
    for t in range(1, 16, 2): # 处理第二季数据
        D[j][t] = D[j][t] * (1 + dt[t]) ** ((t - 1) / 2)
# 其他农作物 [-0.05, 0.05]
for j in itertools.chain(range(0,5), range(7,41)):
    # dt = generate_random_list(16, 0.05, 0.1)
    dt = np.random.uniform(-0.05, 0.05, 16)
    for t in range(0,16): # 处理季数据 (第一季)
        D[j][t] = D[j][t] * (1 - dt[t])


# 亩产量 q_robust [-0.1, 0.1]
for i in i_range:
    # qt = generate_random_list(16, -0.1, 0.1)
    qt = np.random.uniform(-0.1, 0.1, 16)
    for j in j_range:
        for t in range(0, 16, 2): # 第一季
            q[i][j][t] = q[i][j][t] * (1 - qt[t]) ** (t / 2)
        for t in range(1, 16, 2): # 第二季
            q[i][j][t] = q[i][j][t] * (1 - qt[t]) ** ((t - 1) / 2)

# 种植成本 c_robust 0.05
for i in i_range:
    for j in j_range:
        for t in range(0,16,2):
            c[i][j][t] = c[i][j][t] * (1 + gamma) ** (t/2)
        for t in range(1,16,2):
            c[i][j][t] = c[i][j][t] * (1 + gamma) ** ((t-1)/2)

# 蔬菜类作物 每年增长5%
for j in range(16, 37):
    for t in range(0,16,2):
        p[j][t] = p[j][t] * (1 + 0.05) ** (t/2)
    for t in range(1,16,2):
        p[j][t] = p[j][t] * (1 + 0.05) ** ((t-1)/2)
# 食用菌每年下降1%~5%(除羊肚菌外)
for j in range(37, 40):
    # pt = generate_random_list(16, -0.05, -0.01)
    pt = np.random.uniform(-0.05, -0.01, 16)
    for t in range(0,16,2):
        p[j][t] = p[j][t] * (1 - pt[t]) ** (t/2)
    for t in range(1,16,2):
        p[j][t] = p[j][t] * (1 - pt[t]) ** ((t-1)/2)
# 羊肚菌 -0.05
for t in range(0,16,2):
    p[40][t] = p[j][t] * (1 -0.05) ** (t/2)
```

```python
for t in range(1,16,2):
    p[40][t] = p[j][t] * (1 -0.05) ** ((t-1)/2)
```

1. 传统机器学习

2.LSTM