基于多目标优化下的农作物种植规划

摘要

在农业经济日益发展的背景下，本文主要针对华北山区某乡村面临的**耕地资源有限、农作物生长限制**等挑战，分析并提出了科学的农作物种植策略规划方法。研究的主要目标是探讨在不同条件背景限制下，实现 2024 至 2030 年间的农作物**总收益最大化和种植风险最小化**。

问题一：在各农作物未来的预期销售量、种植成本、亩产量和销售价格保持稳定的情况下，从**单目标优化模型**的角度切入，考虑如何分配可实现七年的总利润最大化，也就是问题的目标函数。在对原始表格信息进行**预处理**和**可视化分析**，了解不同农作物的特性以及它们与地块之间的种植联系后，根据题中信息依次判断出相对于农作物和不同地块的**约束条件**。再在题目的两种不能正常销售的情况，设置关于超出部分售价的比例系数 k（0 和 0.5），推出不同的**总利润目标函数方程**，找到在约束条件下的利润净额最大的种植规划方案。

问题二：为了考虑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，我们利用**正态分布**模拟了四要素在题中所给范围内的动态波动，对于最大化总利润的目标函数进行重新设定。再利用**多目标优化模型**把潜在的种植风险量化，具体反映在计算不同方案下最终总利润额的标准差和方差。最后利用**线性加权法**将多个目标函数组合成一个单一的标量目标函数来简化问题，得出最优种植方案。

问题三：在探究农作物替代性和互补性上，先对所有农作物进行分类，再对大类植物通过构建**相关系数矩阵**来判断其相互之间的相关性，借助正相关和负相关的程度来代表替代性和互补性；在探究预期销售量、销售价格、种植成本之间相关性上，利用**多元回归模型**，分析不同在作物中这三种的相关程度。综合多种性质，在前文的基础中再次对作物种植进行规划，根据总利润和稳定性两大目标因素与问题二对比分析。

通过综合考虑农作物之间的相互作用和市场因素的动态变化，可以显著提高农作物的总利润，并增强年度生产的稳定性。这一发现表明，科学化的种植规划对于提升农业生产的经济效益和可持续性具有重要作用。本研究不仅为华北山区的乡村提供了一套切实可行的种植策略，也为其他地区的农作物种植规划提供了理论支持和实践指导。

关键词：农作物种植规划；目标优化模型；相关性分析；多目标优化

基于多目标优化下的农作物种植规划

一、问题重述

1.1 问题背景

近年来，在助力乡村经济的背景下，农作物种植的可持续发展备受关注[1]。有机种植通过避免使用化学合成的农药和化肥，通过轮作、间作、绿肥种植等措施增加土壤有机质含量，改善土壤结构，提高土壤肥力，从而实现了土地的长期可持续利用[2]。但因耕地资源有限、地块类型多样、市场需求多变及种植成本与销售价格波动等多重挑战[3]，有机种植产业的发展任重道远。为了充分利用现有资源，提高农业生产效益，并降低种植风险，华北山区的一个乡村决定通过科学的种植策略规划，为 2024 至 2030 年当地农作物种植制定详细方案。这一规划不仅需要考虑到不同地块的种植适宜性，还需根据作物生长规律，合理安排轮作制度，确保土壤健康与作物产量。同时，面对市场需求的各种风险与未知变动，乡村必须灵活调整种植结构，从而最大化经济收益。此外，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、种植成本之间的相关性，也是制定种植策略时不可忽视的重要因素。通过进行数学建模，乡村将能够利用历史数据进行未来预测，深入分析各种农作物种植方案的经济效益与风险，从而制定出既符合市场需求又能够保障农民收益的最优种植策略。这一过程不仅有助于推动乡村经济的可持续发展，还将为政策制定者、投资者和公众提供有价值的参考，共同助力乡村振兴与农业现代化进程[4]。

1.2 问题提出

在解决“农作物的种植策略”这一数模赛题时，我们需要构建综合的数学模型来优化农作物的种植方案，设定最大化总收益为目标函数，考虑作物销售收入与种植成本之差。

问题一：基于现有数据，在多个约束条件的前提下，构建一个数学模型，从而优化 2024 年至 2030 年的农作物种植方案，以实现总收益最大化。

问题二：考虑多因素的不确定性，设计一个数学模型来评估这些不确定性对农作物种植策略的影响，量化每种方案的风险与收益，从而得出该村在 2024 至 2030 年间农作物的最优种植方案。

问题二：建立一个数学模型，综合考虑农作物之间可能存在的各种经济相关性，制定出既能最大化总收益又能降低种植风险的农作物种植策略。

二、问题分析

2.1 问题一的分析

本题设定农作物未来年销售、成本、产量、价格均稳定，事实上是需要构建静态环境下的利润最大化模型。我们借鉴了前人如冯利平小麦模拟模型[5]、郝小宇双层规划优化种植结构[6]、陈兆波多目标优化模型[7]的经验，利用多源数据预处理与可视化，深入探究农作物特性与地块关联。问题中蕴含的多重约束条件直接影响了种植策略的设计，我们精准分析约束条件，确保模型反映实际限制。考虑到题目中提出的两种特殊情况——超出预期销售量的农作物将面临滞销风险或需以半价销售，我们灵活地在模型中引入了相应的约束条件调整：对于前者，我们确保 2024 年至 2030 年间各农作物的总种植量维持在与 2023 年相同的水平，以规避不必要的库存积压；对于后者，我们通过构建新的总利润目标函数方程，来捕捉半价销售策略对最终利润净额的影响。综合上述分析与优化技术，我们制定出在限定条件下实现利润最大化的种植规划方案。

2.2 问题二的分析

在探讨问题二时，需全面分析农作物种植中的多重不确定性，如销量、产量、成本及价格波动，并考虑潜在风险。此分析扩展了问题一的逻辑框架，强调多目标优化，即在最大化经济收益的同时最小化种植风险。我们采用优化规划模型，构建目标函数与约束条件，针对销售量、产量等不确定性，结合华北山区实际，通过概率分布模型量化这些不确定因素。为了使风险最小化，我们计算不同种植方案的总利润标准差与方差。经过参考化学[8]、工程[9]、经济[10]等领域的多目标优化经验，我们决定利用线性加权法将多个目标函数组合成一个单一的标量目标函数来简化问题，分析各方案在不确定环境下的稳健性，力求选出高收益低风险的种植策略，以应对复杂多变的外部环境。

2.3 问题三的分析

第三问在前两问基础上，进一步探究农作物间可替代性与互补性，以及销售量、价格与成本间的相关性对种植规划的影响。我们通过相关系数量化作物间的关联性，相关系数接近 1 或-1 的值表明强互补或替代关系。回归分析常被用来探究经济领域的财政安排[11]。我们利用回归分析揭示销售量、价格与成本间的复杂数学联系，如线性、多项式或指数关系。最后使用优化模型帮助我们在考虑农作物之间的可替代性和互补性，销售量、销售价格和种植成本之间的相关性的情况下，找到最佳的农作物种植组合和种植面积。

本文的研究路径如图 1 所示：

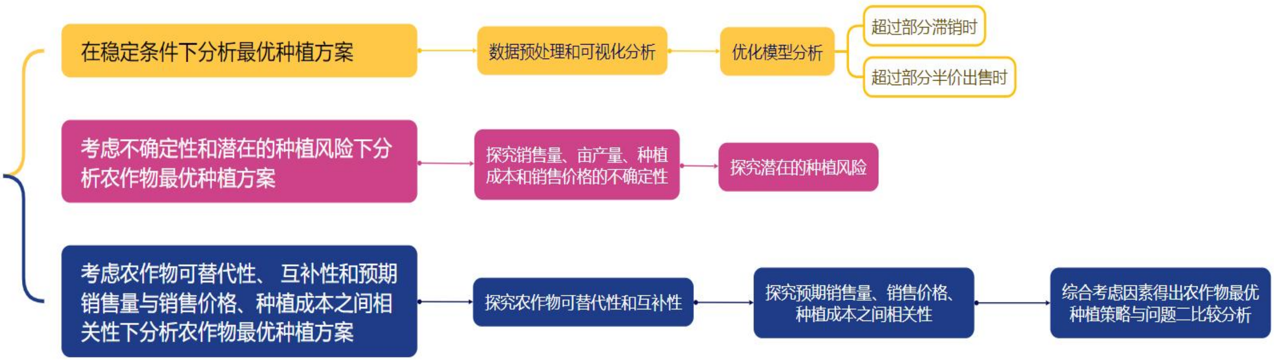


图 1 研究路径

三、模型假设

1. 假设 2023 年农民总产量与当年市场需求大致相同。
2. 假设一种农作物种植在同一种类的地块中（除大棚）等于或多于 4 个地块区域定义为种植分散，且大棚作物不考虑分散性。
3. 假设农作物销售单价为其浮动范围的平均值。
4. 假设单个地块存在一部分种植某特定作物时，默认整个地块收到该作物种植的影响，如第二年该地块整个部分都不能再种该作物。
5. 假设农作物预计销售额、成本、亩产量等增长率在区间范围内符合正态分布。

四、符号说明

符号	含义
x_{ijt}	决策变量
D_i	地块类型
$p^{y}_{i,j,t}$	第 y 年里第 i 块地块上第 t 季农作物 j 的售价
$v^{y}_{i,j,t}$	农作物 j 的亩产量
$c^{y}_{i,j,t}$	在地块 i 上种植农作物 j 的每亩成本
k	超出预期销售量部分农作物的售价占正常售价的比例
μ	各农作物各因素的增长率均值

五、模型构建与求解

5.1 问题 1——在农作物未来销售预期、种植成本、亩产量和销售价格稳定的情况下分析最优种植方案

5.1.1 数据预处理和可视化分析

我们首先对题目所提供的数据表格进行了详尽的预处理与分析工作。这一过程旨在确保数据的准确性和一致性，为后续分析奠定坚实基础。随后，基于预处理后得到的精炼数据表，我们进一步实施了可视化分析策略，以期更直观地揭示数据背后的规律与趋势。

由统计软件绘制出农作物总产量柱形图，我们可以直观地看出作物总产量的分配，如下图所示：

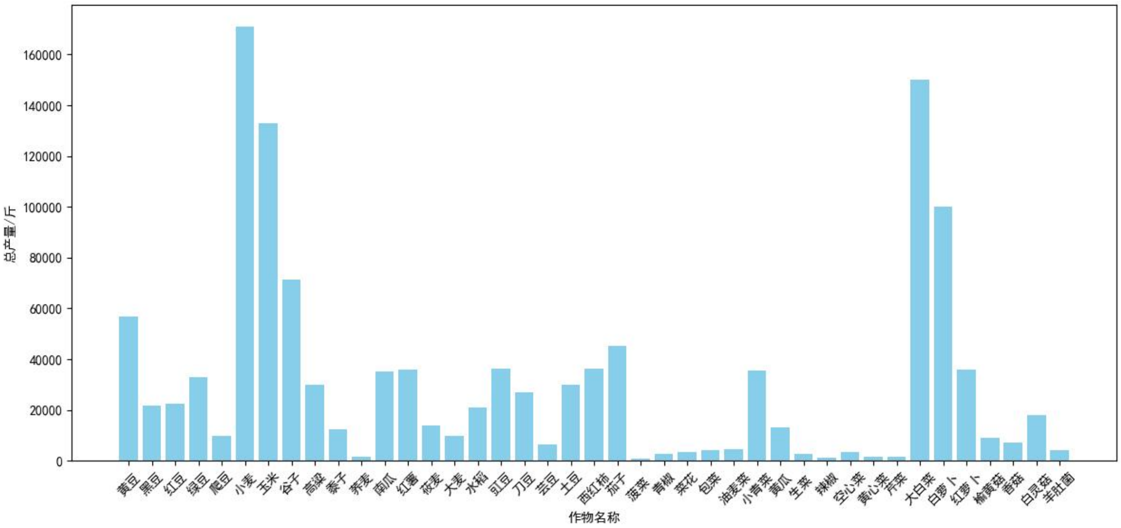


图 2 农作物总产量柱形图

在这张农作物总产量柱形图中，可以清晰地看到不同农作物的产量存在着显著的差异。其中，小麦和玉米等粮食作物以其高产量的特点占据了图表的上部，显示出这些作物在华北山区乡村农业中的重要地位。它们的产量最多，不仅满足了当地居民的基本粮食需求，还可能是该地区农业经济的主要支柱。

相比之下，菜花、菠菜等蔬菜类作物的产量则明显较少，位于图表的下部。这是由于蔬菜类作物对生长环境、气候条件以及种植技术有着更为严格的要求[12]，因此在产量上难以与粮食作物相媲美。

不同农作物总产量的这种显著差异，既反映了华北山区乡村农业的自然条件特点，也体现了农业种植结构的现状[13]。小麦、玉米等粮食作物的高产优势，使得该地区在保障粮食安全方面具有较强的能力；而蔬菜类作物虽然产量较少，但仍是农业发展中不可或缺的一部分。这些特征不仅符合华北山区乡村农业的发展现状，还为后续的作物种植规划提供了重要的参考依据。通过深入分析不同作物的产量特点，可以更加科学地制定种植计划，优化农业资源配置，提高农业生产效益。

在深入分析原始数据的过程中，我们发现种植面积的变动似乎会伴随作物总产量的相应变化。为了进一步验证并明确这种潜在的关系，我们采取了可视化手段，绘制了作物总产量与种植面积之间的散点图。该图直观展现了两者之间的分布模式，具体图示如下：

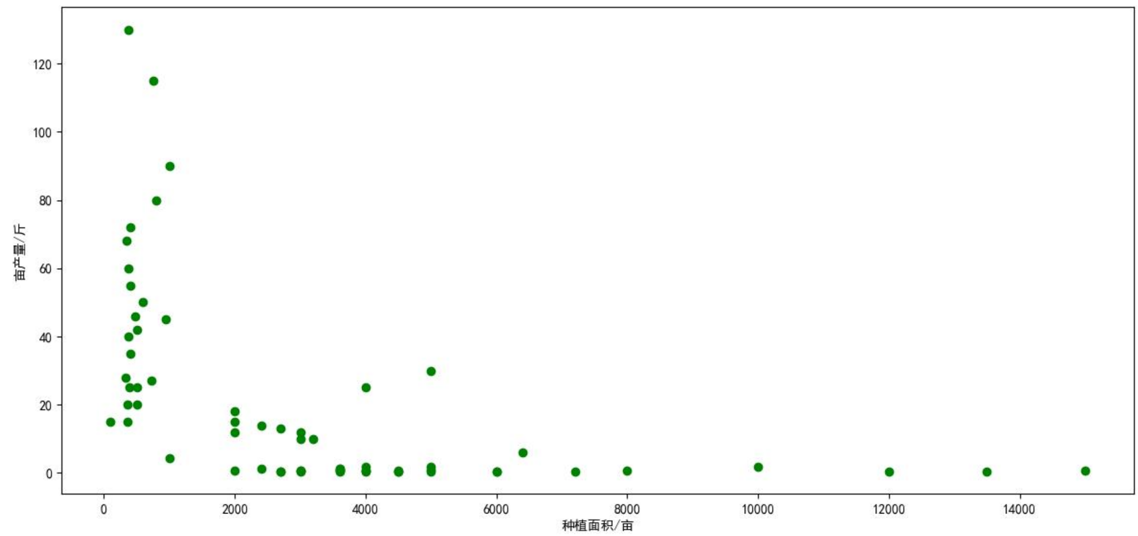


图 3 作物总产量散点图

观察并分析这张作物总产量散点图，我们发现散点图中的数据点分布较为散乱，没有明显的线性趋势或集群模式，这表明在当前情况下，种植面积与作物产量之间并不存在直接的、显著的线性关系。这一发现与通常人们可能持有的“种植面积越大，产量越高”的直观印象有所不同，说明农作物产量受多种因素影响，而不仅仅是种植面积的单一变量。

考虑到农作物产量的复杂性，它受到土壤质量、气候条件、种植技术、作物品种、管理水平以及市场需求等多种因素的共同影响[19]，因此尽管散点图没有显示出种植面积与产量之间的明确关系，但这并不意味着种植面积对产量没有影响，而是说这种影响可能不是简单的线性关系，而是更为复杂和多变的。

为了深入剖析作物的经济效益，我们依托详尽的数据统计与精确的计算，绘制了农作物总利润的直观柱形图。该图表不仅清晰地展示了不同作物在经济效益上的具体表现，

还便于我们直观对比与分析，如下图所示：

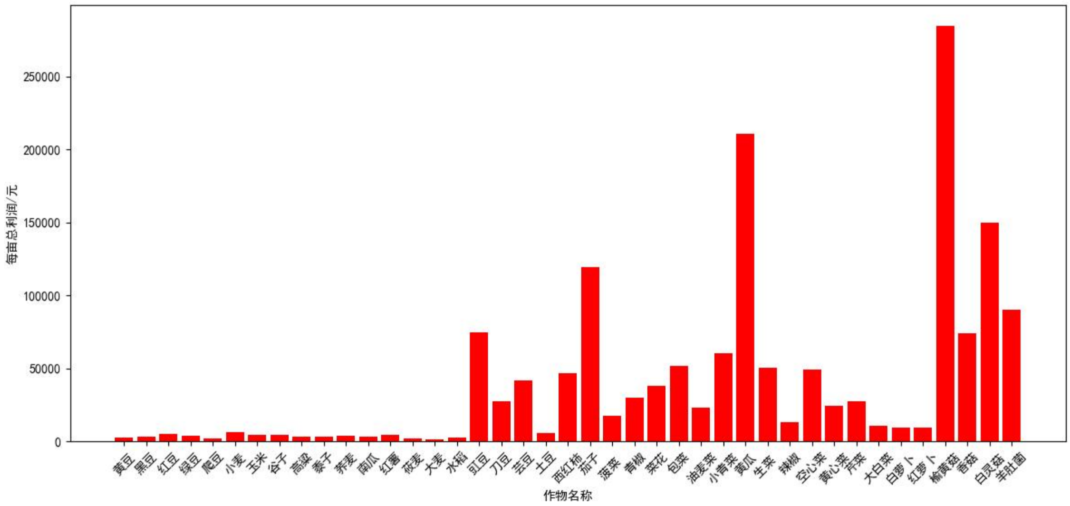


图 4 农作物总利润柱形图

我们发现，不同农作物每亩利润（每亩售价减去每亩成本）之间存在着较大差异。每亩利润越大，说明农作物的经济效益越高。如上图中的榆黄菇、黄瓜、白灵菇等，其每亩总利润均超过 150000 元，都属于经济效益高的作物。我们进一步进行作物的分类，得到不同作物类型的销售单价平均值箱线图，如下图所示：

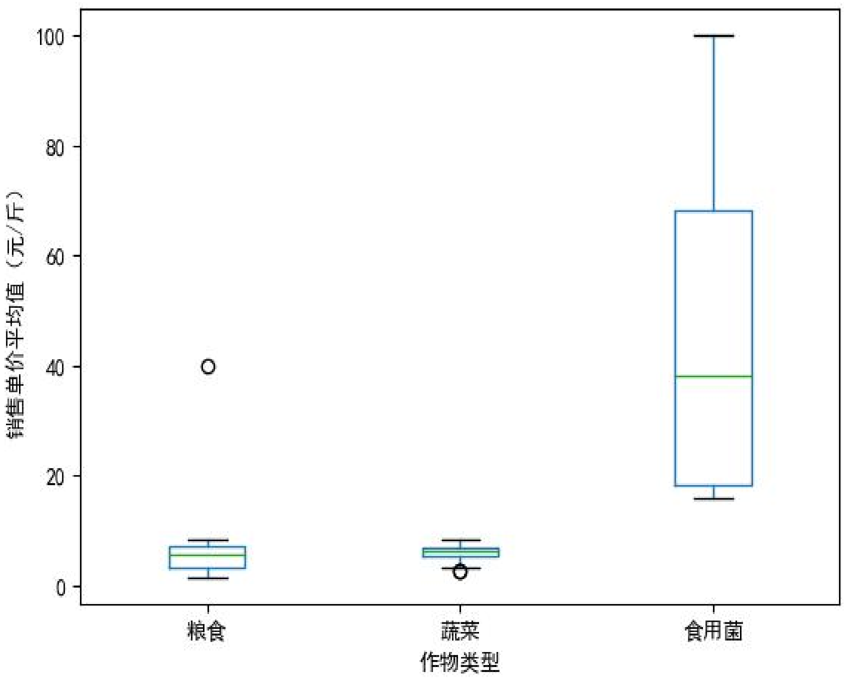


图 5 不同作物类型的销售单价平均值箱线图

根据这张不同作物类型的销售单价平均值箱线图，我们可以明确看到三种作物类型

——粮食、蔬菜和食用菌的销售单价分布情况。图中显示，食用菌类作物的销售单价明显高于粮食类和蔬菜类，这一观察结果与图 4 结论相吻合。如图，食用菌类作物的销售单价平均值不仅位于箱线图的上部，而且其四分位数范围（即箱子的高度）也相对较小，表明其销售单价较为稳定且普遍偏高。相比之下，粮食类和蔬菜类的销售单价则分布在较低的水平，且可能存在一定的波动性。基于这一市场数据，我们可以推断，在不超出或者略微超出市场预期销售的情况下，种植食用菌类作物将有望带来更高的经济效益。因此，这些农作物应鼓励多多种植，以充分利用其较高的市场价值和稳定的需求前景。

此外，虽然粮食类和蔬菜类作物的销售单价相对较低，但它们在农业生产中仍然占据着重要的地位，因为它们通常是人们日常饮食中的基本组成部分。因此，在鼓励种植食用菌类作物的同时，也需要保持粮食和蔬菜生产的稳定性，以满足社会的多样化需求。

5.1.2 优化模型分析

在对原始数据进行简要分析处理之后，我们要开始利用优化模型构建相应的农作物种植策略。优化方法的出发点是系统思维，其基本思路是在一定的约束条件下，保证各方面资源的合理分配，最大限度地提升系统某一性能或系统整体性能，最终达到最理想结果。这种模型的目标是最大化或最小化一个或多个目标函数，这些目标函数通常代表了某种性能指标，如成本、时间、效率、利润等[14]。优化模型广泛应用于工程[15]、经济学[16]、运筹学[17]、管理科学[18] 等领域。在本题中，选择合适的农作物，合理分配到不同地块种植，在满足约束条件的情况下实现利润最大化即为最理想结果。

优化模型的基本组成部分包括：目标函数（Objective Function）——定义优化过程中希望最大化（如利润、产量）或最小化（如成本、风险）的量；决策变量（Decision Variables）——模型中的未知数，代表需要做出决策的量，优化问题的目标就是为这些变量找到最优的值；约束条件（Constraints）——限制决策变量的数学表达式，代表了问题的限制因素，如资源限制、物理限制、政策限制等；优化类型（Type of Optimization）——确定是进行最大化还是最小化；模型类型（Type of Model）——根据问题的性质，优化模型可以是线性的、非线性的、整数的、动态的等。

常见的优化模型包括：线性规划、整数规划、非线性规划、动态规划、多目标优化、随机规划等。

数学规划的一般模型为：

$$\begin{aligned} &Min(Max)\ \ z=\mathrm{f}(\mathrm{x}),\mathrm{x}=(\mathrm{x}_1,\mathrm{x}_2,\cdots\mathrm{x}_n)^{\mathrm{T}} \\ &\mathrm{s.t}\ \ \mathrm{g}_i(\mathrm{x})\le 0,\mathrm{i}=1,2,\cdots \mathrm{m} \end{aligned} \tag{1}$$

其中，x 为决策变量，f(x)为目标函数，$g_i(x)\le 0$ 为约束条件。

本题我们定义决策变量 x_{ijt} ,其中：

$$
\begin{aligned}
&i \in \{1,\cdots M\}\text{：表示第i块地；}\\
&j \in \{1,\cdots N\}\text{：表示第j种农作物，}N=41;\\
&t \in \{1,2\}\text{：表示第}t\text{季的农作物；}\\
&y \in \{0,1\cdots 7\}:\text{表示第y年的农作物；}
\end{aligned}
\tag{2}
$$

该决策变量表示为第 y 年里第 i 块地块上第 t 季种植 j 种农作物的亩数。

定义的目标函数即利润最大化表示：

$$
Max = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{t=1}^{2}\sum_{y=1}^{7}(p^{y}{}_{i,j,t}\cdot v^{y}{}_{i,j,t}\cdot x^{y}{}_{i,j,t} - c^{y}{}_{i,j,t}\cdot x^{y}{}_{i,j,t}) \tag{3}
$$

其中，$p^{y}_{i,j,t}$ 为第 y 年里第 i 块地块上第 t 季农作物 j 的售价，$v^{y}_{i,j,t}$ 为农作物 j 的亩产量，$c^{y}_{i,j,t}$为在地块 i 上种植农作物 j 的每亩成本。

为了计算清晰，我们把地块类型定义为 D_i，

$$
D_{\rm i} \in \{\text{平旱地、梯田、山坡、水浇地、普通大棚、智慧大棚}\} \tag{4}
$$

把农作物分类成粮食（豆类）、粮食、水稻、蔬菜（豆类）、普通蔬菜、水浇地第二季蔬菜、食用菌类、豆类七种，分别对应 F1，F2，…F7。结合题目中所给的条件约束整理出约束方程：

（1）平旱地、梯田、山坡地只能每年种植一季类粮食作物：

$$
x_{i,j,1}^{y} > 0,\ x_{i,j,2}^{y} = 0,\quad (D_i \in \{\text{平旱地、梯田、山坡}\},\ \mathrm{y} \in \mathrm{F1}+\mathrm{F2}) \tag{5}
$$

（2）水浇地只能种植一季水稻或两季蔬菜：

$$
\begin{aligned}
&\underset{j\in F3}{x}{}^{y}_{i,j,2} = 0,\ (D_i \in \{\text{水浇地}\},\ \underset{j\in F3}{x}{}^{y}_{i,j,1} > 0)\\
&\underset{j\in F6}{x}{}^{y}_{i,j,2} > 0,\ (D_i \in \{\text{水浇地}\},\ \underset{j\in F4+F5}{x}{}^{y}_{i,j,1} > 0)
\end{aligned}
\tag{6}
$$

（3）普通大棚只能种植一季蔬菜和一季食用菌：

$$
\underset{j\in F4+F5}{x}{}^{y}_{i,j,1} > 0,\ \underset{j\in F7}{x}{}^{y}_{i,j,2} > 0\ \ (D_i \in \{\text{普通大棚}\}) \tag{7}
$$

（4）智慧大棚只能种植两季蔬菜：

$$
\sum_{j\in F_3} x_{ij1} = 1,\ \sum_{j\in F_3} x_{ij2} = 1,\ D_i \in \{\text{智慧大棚}\} \tag{8}
$$

（5）每个地块至少每三年种植一次豆类作物：

$$
\sum_{\substack{y=y\\ j\in F1+F3}}^{y=y+3} x_{i,j,1}^{y} + \sum_{\substack{y=y\\ j\in F1+F3}}^{y=y+3} x_{i,j,2}^{y} > 0 \tag{9}
$$

（6）每个地块不能重茬：

$$(x_{i,j,1}^{y}+x_{i,j,2}^{y})\cdot(x_{i,j,1}^{y+1}+x_{i,j,2}^{y+1})=0 \tag{10}$$

（7）同一种农作物种植在同一种类的地块中（除大棚）小于或等于 3 个地块区域：

$$x_{i,j,1}^{y}+x_{i,j,2}^{y}\leq 3,（\text{Di}\notin\{\text{普通大棚、智慧大棚}\}） \tag{11}$$

（9）同一种农作物在一个地块上的种植面积在 20%到 100%之间：

$$0.2\leq\frac{x_{i,j,t}^{y}}{v_{i,j,t}^{y}}\leq 1 \tag{12}$$

最终结合约束条件方程和超额部分销售系数 k 得到关于总利润最大化的目标函数方程：

$$0.2\leq\frac{x_{i,j,t}^{y}}{v_{i,j,t}^{y}}\leq 1 \tag{13}$$

其中，A 表示各年的农产品销售量也就是 23 年的具体值，

$$0.2\leq\frac{x_{i,j,t}^{y}}{v_{i,j,t}^{y}}\leq 1 \tag{14}$$

表示取关于每年销售量与生产量的较小值，以此得到销售额。

5.1.3 超过部分滞销时作物的种植策略

假设在农作物生产和销售过程中，若某一季度的总产量超出其预期销售量，则超出部分面临滞销问题，进而引发资源浪费。针对这一情境，我们设定一个关键系数 k，它代表超出预期销售量部分农作物的售价占正常售价的比例。当农作物总产量确实超过预期销售量，导致部分农产品无法及时售出时，该比例系数 k 自动调整为 0，意味着超出部分的销售收益为零，即不产生额外收入。

基于上述分析，我们可以对总利润的目标函数进行相应的调整。具体而言，该目标函数需要能够反映在正常销售量内按正常价格计算收益，而对于超出预期销售量的部分，则因 k 值为 0 而不计入总利润计算之中。因此对应总利润的目标函数可以变化为：

$$0.2\leq\frac{x_{i,j,t}^{y}}{v_{i,j,t}^{y}}\leq 1 \tag{15}$$

在利用 Python 中的 PuLP 库，结合 CBC 求解器的分支定界与割平面技术进行深入计算后，我们成功获取了详尽的农作物种植方案，即得出 result1_1.xlsx 文件。

然而，在审查该表格数据时，我们注意到部分季度与地块上出现了“空种”的情况，即这些区域未被安排种植任何农作物。为了更精准地识别并处理这些潜在的资源配置不

合理问题，我们进一步进行了数据筛选，将这些“空种”的季度与地块信息提炼出来，形成了更为聚焦的表格，得到下表：

表 1 空置地块及对应季次

种植季次	种植地块	合计
第一季	D2	0
第一季	D6	0
第一季	F1	0
第一季	F2	0
第一季	F3	0
第一季	F4	0
第二季	D2	0
第二季	D6	0
第二季	F1	0
第二季	F2	0
第二季	F3	0
第二季	F4	0

在深入分析表格数据后，我们发现：空置地块主要为智慧大棚，其次则是水浇地。这一现象是为了应对总产量超出预期销售量所引发的滞销问题（k=0 时），即为了规避因过剩生产而导致的成本增加与资源浪费，模型策略性地采取了产量调控措施。具体而言，模型倾向于在某些特定地块减少或取消种植计划，尤其是那些成本效益比相对较低的地块，如智慧大棚，以确保总产量与市场需求紧密匹配，从而有效避免滞销风险。

此外，从农业可持续发展的角度出发，保持土壤健康与生态平衡同样重要。因此，除了基于经济因素的考量外，模型还可能考虑到土壤肥力的维护与病虫害防控的需求，从而在特定地块实施轮作或休耕制度。这些措施虽然在特定时期内会导致地块空置，但长远来看，它们对于维护土壤质量、减少化学投入、以及促进生物多样性具有不可替代的作用。

5.1.4 超过部分半价出售时作物的种植策略

假设在农作物生产与销售的周期性循环中，若总产量超出了事先设定的预期销售量，则超过部分按 2023 年销售价格的一半进行降价出售，即 k 在此情境下被设定为 0.5。

基于这一降价机制，我们需要对总利润的目标函数进行相应的调整。新的目标函数将能够捕捉到在正常销售量范围内以全价销售带来的利润，同时，对于超出预期销售量的部分，则以半价（即 k=0.5）计入总利润计算，即对应总利润的目标函数为：

$$0.2 \leq \frac{x_{i,j,t}^{y}}{v_{i,j,t}^{y}} \leq 1 \tag{16}$$

在深入探讨求解过程时，我们同样借助了分支定界与割平面技术，精确得出了具体的种植方案，详尽数据请参见附件 result1_2.xlsx。

值得注意的是，在此分析框架下，我们意外发现并未出现地块闲置未种的情况。进一步对比两种情境下的目标函数设定与约束条件，我们推测这背后可能隐藏着软件计算过程中优化目标细微却关键的差异。尽管两者的终极追求均聚焦于总利润的最大化，但在处理总产量超出预期销售额、进而引发部分产品滞销的情境时，局部优化策略显现出明显分歧。具体而言，当面临超产风险时，一种策略倾向于最大化净利润，同时精打细算，力求减少任何形式的资源浪费。若某地块种植作物的预期收益无法覆盖成本，或市场趋于饱和导致超产作物难以消化，模型则可能主动选择让这些地块保持空置，以此规避潜在的负收益，展现出一种更为审慎的风险规避态度。这种策略将未种植地块视为一种灵活的风险缓冲带，有效应对市场需求波动带来的不确定性；然而，若考虑将超额部分以折扣价（如半价）销售，局部优化目标则可能转变为更倾向于提升总销售额，即便这意味着部分产品需以较低价格成交。在此情境下，模型倾向于更加积极地利用每一寸可用土地，因为即便是半价销售，也能确保每块土地产生正向的经济贡献，从而在一定程度上抵消了市场不确定性带来的负面影响。由此可以看出，不同情境下的优化策略反映了模型在平衡利润最大化与风险控制之间的微妙权衡，体现了数学建模在复杂农业决策中的灵活性与深度洞察力。

5.2 问题 2——考虑不确定性和潜在的种植风险下分析农作物最优种植方案

5.2.1 探究销售量、亩产量、种植成本和销售价格的不确定性

在自然界与社会现象中，诸多随机变量，如销售量、成本及销售价格等经济指标，常近似遵循正态分布规律。此分布以其独特的对称性、可加性等数学特性，在统计分析及建模领域占据重要地位。中心极限定理指出，当样本规模足够庞大时，无论原始数据分布形态如何，其均值分布均趋近于正态分布。鉴于华北乡村经济活动的复杂性，若销

售量、亩产量、种植成本及销售价格等变量受多元独立因素共同作用，其分布形态极可能呈现正态分布特性。

正态分布的概率密度函数如下：

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{17}$$

其中，μ 为分布的均值，表示数据的中心位置；σ 为分布的标准差，表示数据的离散程度，σ^2 为方差，时标准差的平方。由此也可知，正态分布是关于均值对称的，即左侧和右侧的形状是镜像对称的。

基于题目设定，小麦与玉米的未来销售量预计将以 5%至 10%的年均增长率持续增长，而其他农作物的销售量则预计围绕 2023 年水平，每年呈现±5%的浮动。农作物亩产量则受气候等不确定因素影响，年际间波动可达±10%。与此同时，种植成本呈现稳定上升趋势，年均增长率约为 5%。

在价格方面，粮食类作物价格保持相对稳定，而蔬菜类作物则展现出积极的增长态势，年均价格上涨约 5%。相比之下，食用菌市场表现出稳中有降的特点，价格年均降幅在 1%至 5%之间，其中羊肚菌价格尤为显著，每年均以 5%的幅度下降。

为了准确反映这些动态变化，并为后续建模提供坚实的数据基础，我们依据题目给出的变化范围区间，计算并确定了各关键因素的增长率均值，并整理成表呈现，如下表所示：

表 2 各农作物各因素的增长率均值 μ

<table>
<tr><th>农作物种类</th><th>预期销售量</th><th>亩产量</th><th>种植成本</th><th>销售价格</th></tr>
<tr><td>小麦、玉米</td><td>0.05</td><td rowspan="6">0.2</td><td rowspan="6">0.05</td><td rowspan="3">0</td></tr>
<tr><td>其他农作物</td><td rowspan="5">0.1（固定于 23 年）</td></tr>
<tr><td>粮食作物</td></tr>
<tr><td>蔬菜类作物</td><td>0.05</td></tr>
<tr><td>食用菌</td><td>0.04</td></tr>
<tr><td>羊肚菌</td><td>0.05</td></tr>
</table>

参考《预期利润、农业政策调整对中国农产品供给的影响》[20]等研究，我们依据不同农作物的独特属性选取了相应的 σ^2 进行计算分析，得出下图正态分布曲线：

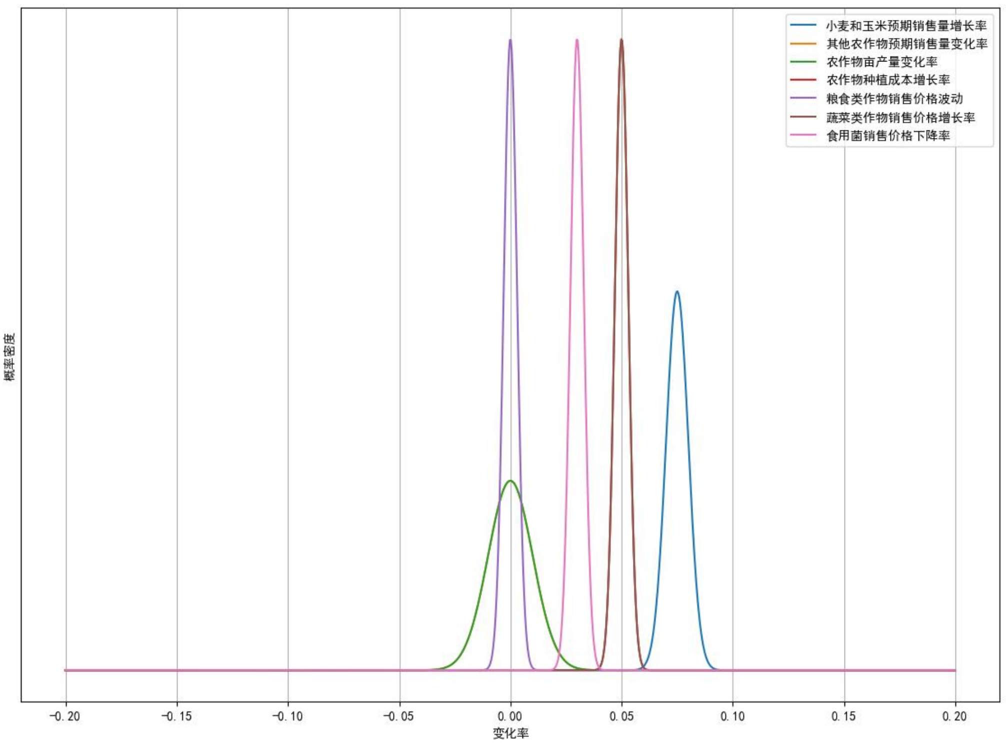


图 6 正态分布图

在确定适用的概率分布模型后，我们需将这些模型融入原先设定的目标函数及约束条件中进行 运算。在这里我们统一拟定在问题一中“当某一农作物的季度总产量超出其预期销售量时，超出部分将仅以半价进行销售”的环境下进行分析求解。

基于这一拟定市场环境，我们对于最大化总利润的目标函数重新进行设定：

$$Max=\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{t=1}^{2}\sum_{y=1}^{7}(p^{y}{}_{i,j,t}\cdot\min(v^{y}{}_{i,j,t}\cdot x^{y}{}_{i,j,t},A)-c^{y}{}_{i,j,t}\cdot x^{y}{}_{i,j,t}+0.5\cdot p^{y}{}_{i,j,t}\cdot|v^{y}{}_{i,j,t}\cdot x^{y}{}_{i,j,t}-A|) \quad (18)$$

在此情境下，除了问题一已明确界定的约束条件之外，我们进一步考虑了作物销售过程中的多个关键变量，包括预计销售量、销售价格、种植成本以及亩产量，这些在第一问中被视为静态值的因素，在现实中往往呈现出动态波动的特性。为了更贴近实际并增强模型的准确性，我们决定采用正态分布来模拟这些变量的动态变化。具体而言，通过引入正态分布模型，我们能够捕捉并量化这些变量在不同情境下的随机性和不确定性，从而使我们的数学建模更加全面、灵活且贴近实际情况。

5.2.2 探究潜在的种植风险

在探讨农作物种植策略时，我们不仅要直面其内在的不确定性，还需精准量化这些不确定性如何加剧策略风险。为此，方差作为关键统计工具被引入分析框架。方差深刻揭示了随机变量或数据集的离散特性，通过计算各数据点偏离平均值的平方的平均值，

有效刻画了波动幅度与风险水平。这一量化手段，不仅加深了我们对种植策略风险的理解，也为制定更为稳健、适应性强的策略提供了科学依据。

对于包含 n 个观测值 $x_1, x_2, \ldots\ldots, x_n$ 的样本集，方差 σ^2 的公式为：

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{19}$$

为有效缓解种植过程中的潜在风险，我们对原有单一利润最大化目标函数进行了拓展，构建了多目标优化框架。此框架不仅追求总利润的最大化，还创新性地融入了种植风险的量化考量，以方差作为风险度量的标准。鉴于总利润年度间虽存波动但总体呈增长态势，我们将稳定增长视为风险低的表现，通过计算各方案总利润一阶差分后的方差，并选取方差较小者，从而识别出风险较低且能实现稳定增长的种植策略。

构成新的关于种植风险目标函数：

$$Min = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}({x^y}_{i,j,t} - \mu)^2 \tag{20}$$

在处理复杂多变的多目标规划问题（MOP）时，我们面临的是一系列相互交织且往往相互冲突的目标函数，其核心挑战在于寻求一组解决方案，这些方案能在各目标间实现精妙平衡，达到所谓的 Pareto 最优状态，即无法在不损害至少一个目标利益的前提下，改进其他任何目标。为解决这一难题，线性加权法作为一种高效实用的策略脱颖而出，它通过创新的手段将原本多维度的目标函数体系融合为一个统一的标量目标函数，极大地简化了问题的复杂度。该方法的精髓在于，它允许决策者根据各目标的重要性程度，为其分配相应的权重系数，以反映其在总目标中的相对重要性，从而有效促进了多目标规划问题向 Pareto 最优解的逼近。

本题中涉及两个目标函数，分别为

$$Max = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{t=1}^{2}\sum_{y=1}^{7}({p^y}_{i,j,t} \cdot \min({v^y}_{i,j,t} \cdot {x^y}_{i,j,t}, A) - {c^y}_{i,j,t} \cdot {x^y}_{i,j,t} + 0.5 \cdot {p^y}_{i,j,t} \cdot |\, {v^y}_{i,j,t} \cdot {x^y}_{i,j,t} - A|) \tag{18}$$

$$Min = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}({x^y}_{i,j,t} - \mu)^2 \tag{20}$$

在构建数学模型的过程中，我们考量了每个目标函数对于整体优化目标的相对重要性，赋予它们特定的权重值。根据题目理解和分析，我们自行设定为 0.7 和 0.3 来对应最大总利润和最小种植风险。然后将每个目标函数与其相应的权重相乘并相加，构建一个加权合成目标函数。

为了更精准地反映种植风险的稳定性，即方差值越小越优，我们对上述方法进行了优化调整。具体而言，我们采用了一种加减结合的线性加权策略，即先将最大总利润的目标函数值乘以其权重，再从中减去最小种植风险目标函数值（已乘以其权重），所得到的数设定为某种种植策略下的分数。这样，每一种种植策略都能被赋予一个明确的分数，该分数的高低直接反映了策略的合理性与优越性：分数越高，意味着该策略在追求利润最大化的同时，也有效降低了种植风险，从而被认为是更加合理的选择。

同时，考虑到模型所取因素存在一定的动态波动和偶然特征，我们借助蒙特卡洛方法通过多次独立的随机抽样，使用随机数来模拟复杂系统的概率分布，以此得到不同的种植方案。如图为随机 20 次的生成方案比较：

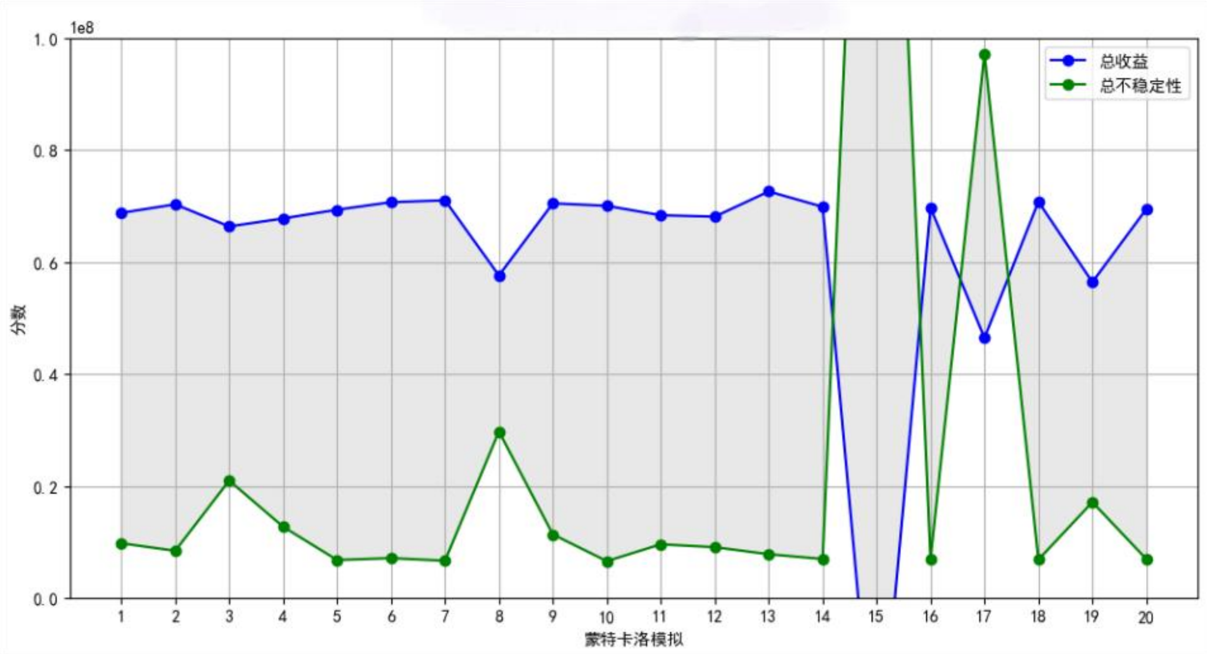


图 7　多次蒙特卡洛模拟的总收益和总稳定性

通过将总收益与总不稳定性作差，我们可以得到所设定的最终得分，即图中间灰色部分。由于存在一定的偶然性，因此会出现类似图中第 15 次的图像，这代表着该随机生成方案由于约束条件的限制，具有极高的低收益性和不稳定性，我们把它剔除即可。

基于上图，我们对数据进行重新整合，可以得到多次蒙特卡洛模拟后的总得分折线图（包含第 15 次），如下图所示：

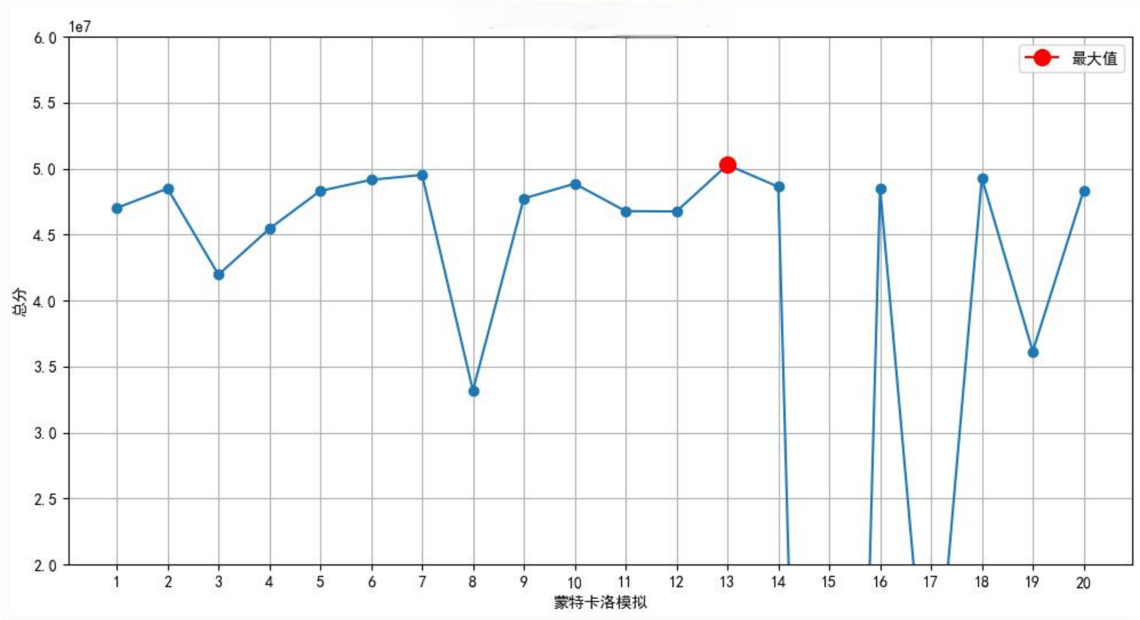


图 8 多次蒙特卡洛模拟总得分折线图

由图 8 可直观地得到在这 20 次模拟中所得到的最优方案为第 13 次模拟生成的方案。

为了进一步确定最优方案，我们利用软件重新进行上千次运算后，最终比较得到最优方案（详细见 result2.xlsx）。 下图为该方案经过评估计算后的总收益和稳定性表现：

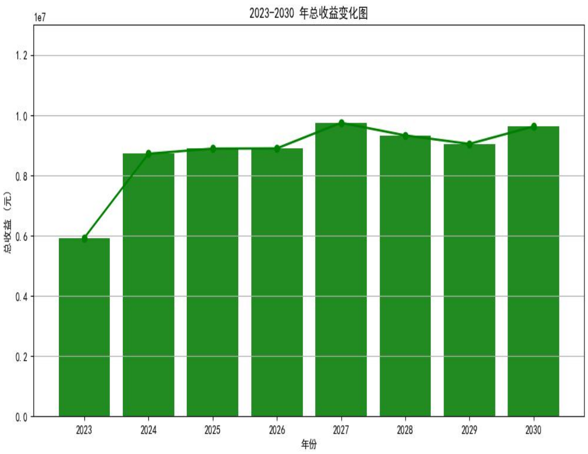


图 9 2023-2030 年总收益变化图

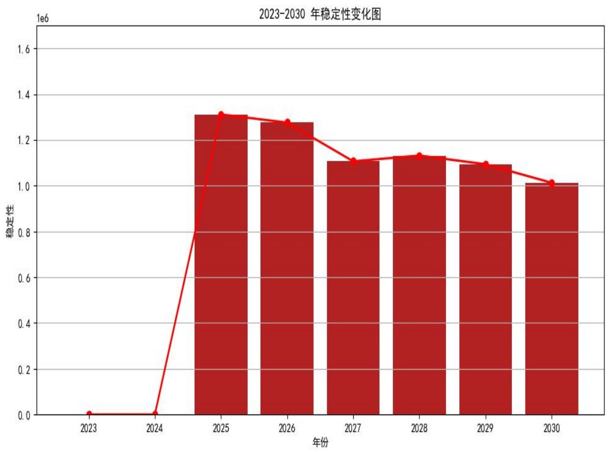


图 10 2023-2030 年稳定性变化图

通过观察图 9，我们可以明显地感受到 2023-2030 年的农作物年产量以一个波动上升的总趋势，符合现实社会生产状况。图 10 表示的是期间总利润变化的稳定性，由于图中数值越小，稳定性越强，那么从 2025 年之后，该稳定性逐渐增强，体现出我们选用模型的有效性。由于 23、24 年间是按照过去的数据进行计算，因此具有很大的误差和偶然性，在左图中表现为 23 年与 24 年间总利润变化大，在右图中表现为这两年的稳定性极高，该情况符合模型自身的特性。

5.3 问题 3——考虑农作物可替代性、 互补性和预期销售量与销售价格、种植成本之间相关性下分析农作物最优种植方案

5.3.1 探究农作物可替代性和互补性

在现实生活中，农作物之间的替代性是一个普遍且复杂的现象，它深刻影响着生产者与消费者的决策过程。当农作物在市场需求、适宜种植条件或功能用途上展现出相似性时，这种替代效应尤为显著[21]。以小麦和玉米为例，作为粮食类作物的两大支柱，它们不仅共享广泛的种植环境，如平旱地、梯田乃至山坡地，还在市场上形成了紧密的替代关系。一旦小麦市场价格出现波动，特别是当价格下降时，消费者出于成本考虑，生产者则基于收益最大化原则，均可能倾向于选择玉米作为替代，从而调整消费与生产策略。此外，农作物之间的相互作用还体现在生长周期、病虫害防控及市场需求等多个维度上的相互依赖。例如，豆类作物与谷物作物的轮作模式，不仅能够有效改善土壤结构，提升土壤肥力，还是一种可持续的农业实践。将此类农作物间的轮作机制纳入数学建模的考量之中，无论是作为约束条件还是次要目标函数，都能显著提升种植方案的灵活性与优越性，促进农业生产的多元化与可持续性发展。因此，充分考虑农作物间的替代性与相互依赖关系，对于制定科学合理的农业决策具有重要意义。

为了深入研究多种农作物之间的内在联系，我们首先需要依据问题二中获取的关于不同农作物的历史数据，这些数据包括但不限于各农作物的销售量、市场价格、种植成本以及产量等关键指标。

在统计学中，方差用来衡量单个随机变量波动性或离散程度。而当我们需要探讨两个或多个随机变量之间是否存在某种共同变化趋势或相互影响时，协方差则成为了一个不可或缺的度量工具。协方差的计算公式用于量化两个随机变量间的线性相关程度，即它们在同一方向上变动的程度，具体计算公式为：

$$\sigma(x,y)=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) \tag{21}$$

其中，$\bar{x}$ 和 $\bar{y}$ 分别表示两个随机变量所对应的观测样本均值。

通过计算方差与协方差，我们能够构建出协方差矩阵，这一矩阵中每个元素表示两个变量之间的协方差，对角线上的元素直接对应了各个随机变量的方差，反映了各变量自身的离散程度；而非对角线上的元素则代表了任意两个随机变量之间的协方差，揭示了它们之间的线性相关程度。这一矩阵形式化地表示为：

$$\sum=\begin{bmatrix}\sigma(x_1,x_1) & \dots & \sigma(x_1,x_d)\\ \vdots & \ddots & \vdots\\ \sigma(x_d,x_1) & \dots & \sigma(x_d,x_d)\end{bmatrix}\in R^{d\times d} \tag{22}$$

对于两个随机变量 X 和 Y，它们的协方差可以写作：

$$Cov=(X,Y)=E\left[(X-E[X])(Y-E[Y])\right] \tag{23}$$

其中 E[]表示期望值。当我们讨论特征的协方差，就是在讨论这些特征的线性关系：

（1）如果 Cov（X，Y）> 0,那么 X 增加时 Y 也可能增加；

（2）如果 Cov（X，Y）< 0,那么 X 增加时 Y 也可能减少；

（3）如果 Cov（X，Y）= 0,那么 X 和 Y 在统计上是不相关的。

协方差作为衡量两个变量间线性关联强度的关键统计量，可用于反映变量间的相互关系。当两个变量的变动趋势趋于一致，即同时增加或减少时，它们之间的协方差倾向于正值，即两者间存在的正相关关系。反之，若一个变量上升而另一个下降，则协方差可能呈现负值，即二者间存在负相关关系。且协方差矩阵以其对称性为特征，这一特性确保了矩阵中任意两个变量间的协方差值在互换位置时保持不变，无论是从变量 A 视角观察变量 B，还是反之，均得到相同的结果。矩阵的对角线元素直接反映了各自变量内部的离散程度，即方差。因此我们可以说，协方差矩阵不仅精准捕捉了数据集中各变量间的复杂关系，还通过量化线性相关性，为我们提供了深入剖析多维数据集、理解变量间相互作用的强大工具。

考虑到部分作物之间原始比较量级和单位存在较大差异，因此我们将协方差矩阵变换为相关系数矩阵，提供无量纲化的度量标准来衡量这些变量之间的相关性。在转换后的相关系数矩阵中，每一个元素均代表一个相关系数，它表示两个变量之间的相关强度，并且不受变量单位和量级的影响。

协方差矩阵描述了多个随机变量之间的线性关系，但其局限性在于无法直接比较不同量级或单位变量间的相关性。而相关系数矩阵则提供了一种无量纲化的方式来衡量这些变量之间的相关性。相关系数矩阵的每个元素都是相关系数，它表示两个变量之间的相关强度，并且不受变量单位和量级的影响。首先，对于协方差矩阵中的每个变量，我们找到其标准差，也就是协方差矩阵对角线上的元素。然后计算相关系数，对于协方差矩阵中的任意两个变量 X 和 Y，它们的相关系数 ρ_{XY} 可以通过它们的协方差和它们各自的标准差来计算：

$$\rho_{XY}=\frac{Cov(X,Y)}{\sigma_X\sigma_Y} \tag{24}$$

这个公式将协方差无量纲化，得到介于-1 和 1 之间的相关系数。这一过程去除了原始数据中的单位和量级差异，使得相关系数能够纯粹地反映变量间的关联强度。

随后便可构建相关系数矩阵，接近 1 或 -1 的值表示强正相关或强负相关；接近 0 的值表示变量之间的相关性较弱或没有明显的线性关系。

由于题目中所涉及到 41 种不同的农作物，难以同时构建相关系数矩阵，我们再次利用问题一优化模型分析时的想法，把农作物分类成粮食（豆类）、粮食、蔬菜（豆类）、普通蔬菜、水浇地第二季蔬菜、食用菌类、豆类六种（由于水稻自成一类，因此在这里省略掉），分别构建六大类农作物的相关系数矩阵，如下图所示：

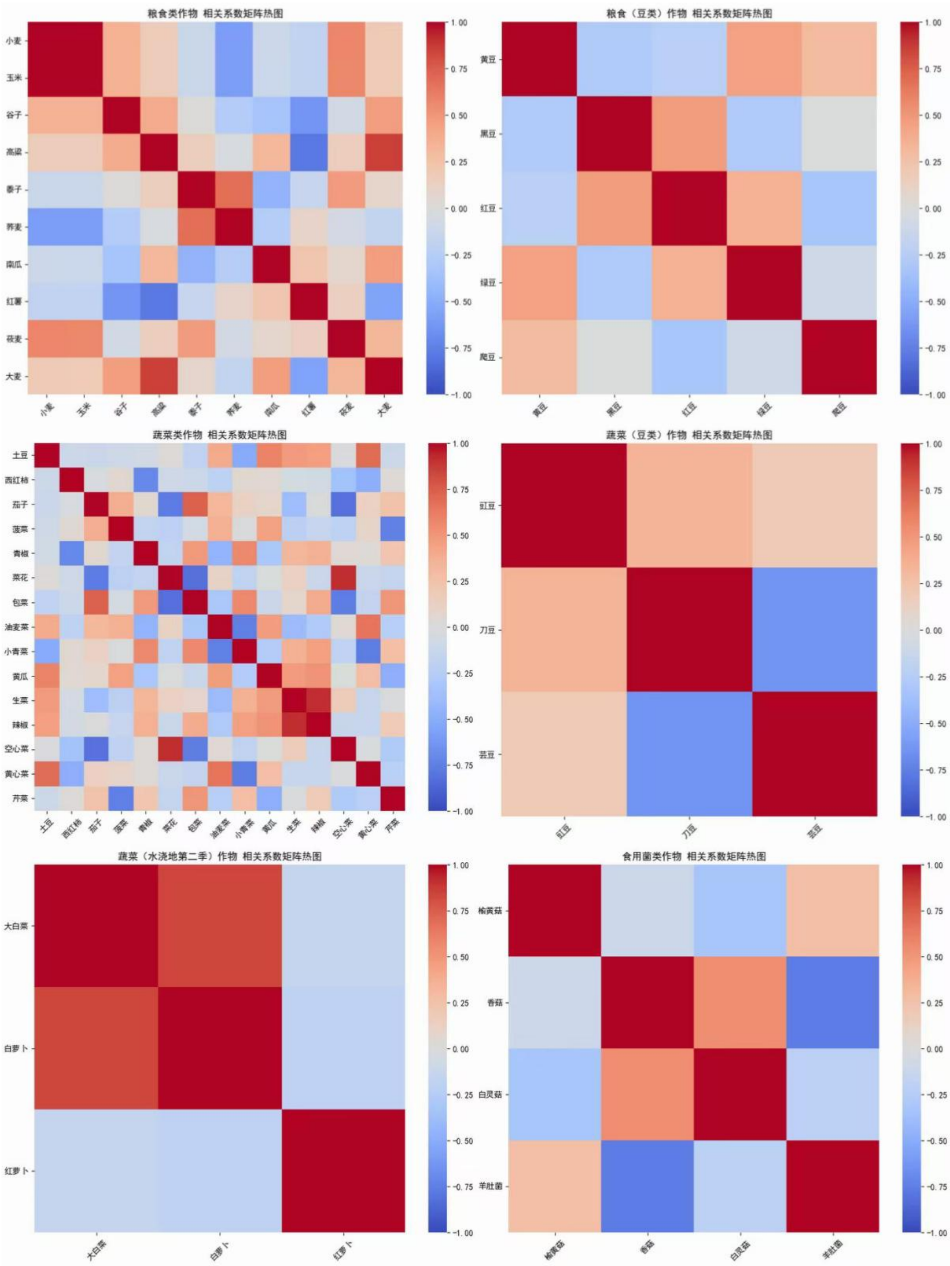


图 11 六大类农作物的相关系数矩阵图

图中红色部分表示的是各农作物之间的正相关关系，也就是互补性，蓝色部分表示的是负相关关系，也就是可替代性。例如，对于粮食类作物来说，小麦和玉米、大麦和高粱具有较强的互补性；荞麦和小麦、荞麦和玉米、红薯和高粱之间具有较强的可替代性。结合实际：小麦和玉米作为主要的粮食作物，在生长周期、养分需求及市场定位上均存在差异，因此它们之间的互补性较强；大麦和高粱在种植区域、用途及市场接受度上也有各自的特色，这种差异使得它们能够相互补充，满足多样化的市场需求；荞麦和小麦、荞麦和玉米在营养成分、口感或用途上存在一定的相似性，使得消费者或市场在某些情况下可以将它们视为替代品；而红薯和高粱之间的可替代性也可能源于它们在某些食品加工或饲料生产中的相似应用。上述事实均与所得相关系数矩阵图结果相符。

5.3.2 探究预期销售量、销售价格、种植成本之间相关性

通常，销售量与价格呈负相关。价格上升可能会导致销售量下降，反之亦然。成本上升可能会推高价格，但市场接受程度和竞争条件也会影响最终价格。产量的波动可能会影响市场价格，尤其是在供应量大幅波动时。

为了弄清楚其中的相关性，我们依旧根据题目二所得到的相关数据，分别对七大类农作物进行散点图的绘制和拟合，得到下面两类多元回归模型：

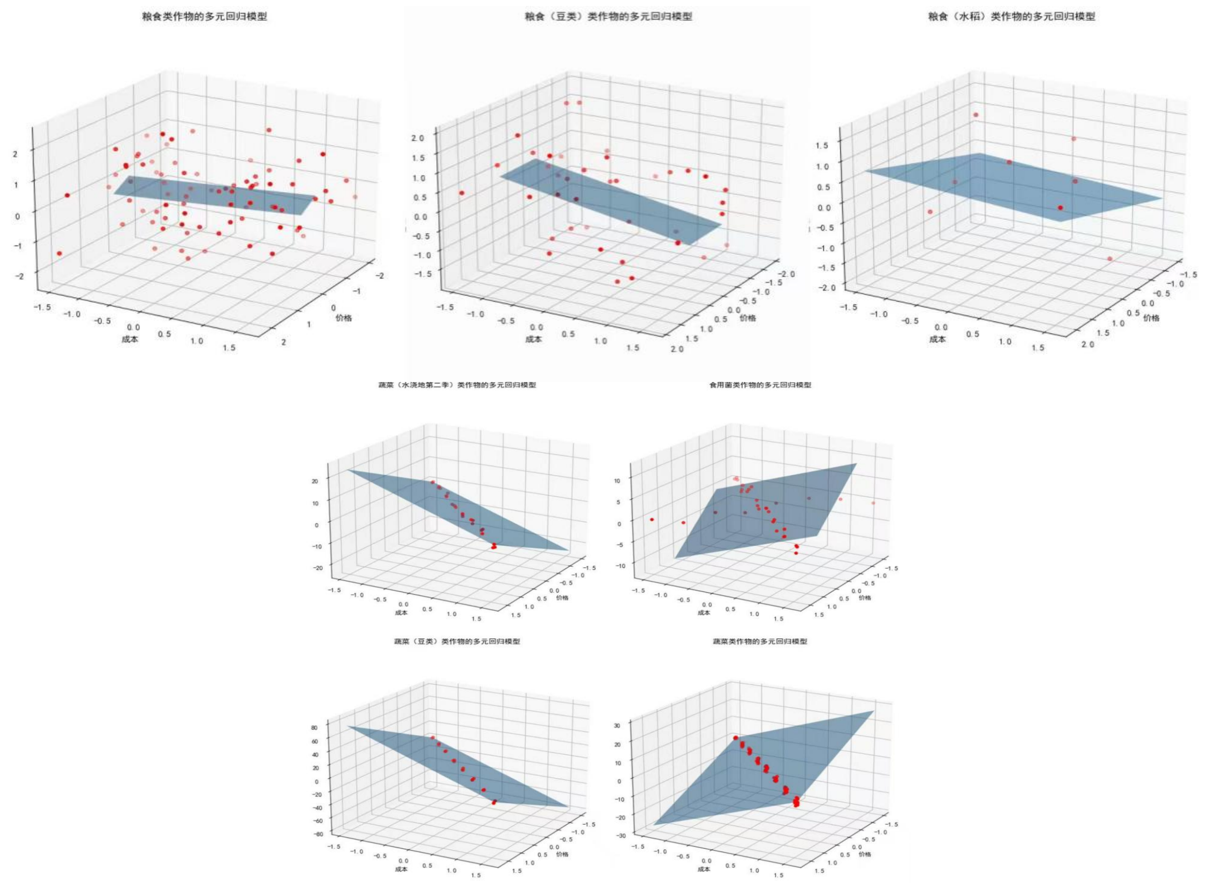


图 12 七大类农作物多元回归模型图

比较这两类模型，我们发现粮食、粮食（豆类）和水稻的预期销售量、销售价格、种植成本之间相关性较差，拟合效果并不明显。但是蔬菜（水浇地二期）、食用菌类、蔬菜（豆类）和普通蔬菜类作物的预期销售量、销售价格、种植成本之间相关性较好，拟合效果明显。这说明不同种类的作物其三者相关性存在较大差异，我们需要分类进行分析运算。这种差异的原因可能与作物的生长周期、市场需求特性、以及生产成本的构成密切相关。水稻和豆类等粮食类作物往往受到较为稳定的政策支持和市场调节机制影响，其价格波动相对较小，且由于种植技术相对成熟，成本变化也较为平稳，因此预期销售量、销售价格与种植成本之间的直接相关性不够明显。而蔬菜类作物，尤其是蔬菜（水浇地二期）、食用菌类、蔬菜（豆类）和普通蔬菜类这些作物通常具有较短的生长周期，对市场反应灵敏，季节性和地域性特征明显。随着消费者偏好的快速变化、气候条件的波动以及物流效率的提升，这些作物的供需状况能够迅速调整，导致销售价格和预期销售量在短期内可能出现较大波动[19]。

5.3.3 综合考虑因素得出农作物最优种植策略与问题二比较分析

在综合考虑农作物可替代性、互补性和预期销售量与销售价格、种植成本之间相关性后，我们再次分析农作物最优种植方案，对于同样对目标优化下利用分支定界和割平面技术进行求解得到具体的方案进行总收益和稳定性的分析，将结果可视化得到柱状折线图与问题二进行比较：

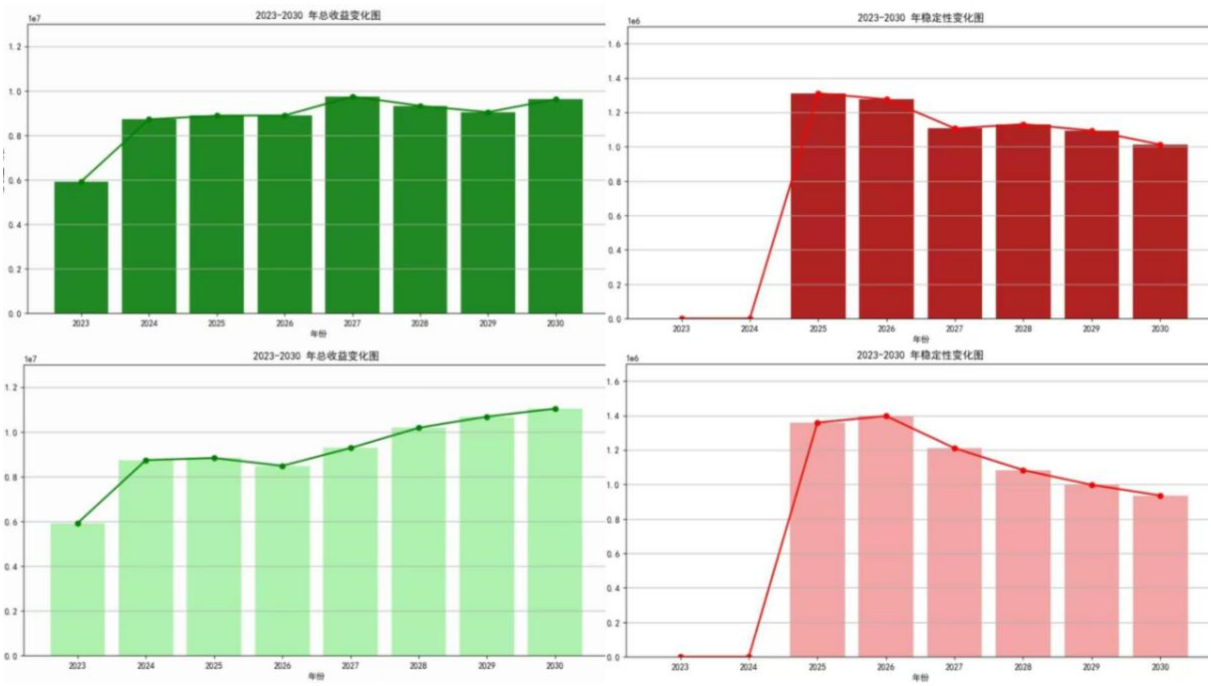


图 13 两种方案总收益与稳定性变化图

其中，上面深色部分的两张图为问题二中最优方案所对应的七年总收益变化图和稳定性变化图，下面浅色部分的两张图则是问题三中综合考虑过后的结果。对比两组图片，

我们发现综合考虑多种因素后，农作物总利润更多，同时每年生产的稳定性也更好。这反映了现实中，通过考虑作物之间的相互作用，可以更合理地分配资源，如土地、水资源和肥料，从而提高整体的种植效率和经济回报。并且，考虑销售量、价格和成本之间的相关性可以帮助预测市场趋势，从而调整种植策略以适应市场需求，增加销售量和利润。

六、模型结果分析

对于问题一，我们主要利用单目标优化模型，根据题目所给条件构建关于最大化农作物总利润的目标函数，并设定出题目所给的约束条件方程。在产量超出预期销售部分滞销和半价出售两种条件下，利用 python 中结合其内置的 CBC 分支定界算法与割平面技术进行求解得到了具体的种植方案。该过程不仅保证了求解的精确性，还有利于高效得出最优化的农作物种植方案，为决策提供了坚实的数学依据。

对于问题二，考虑了销售量、价格、成本和亩产量的不确定性，我们采用了蒙特卡洛模拟和多目标优化方法来评估风险和收益，在不确定性条件下寻找最优的种植策略。模拟结果表明，不确定性因素对种植策略有显著影响，特别是在价格波动较大的情况下。多目标优化模型提供了一系列的 Pareto 最优解，可以根据风险偏好选择合适的种植方案。模型还评估了不同种植方案的稳定性，为长期规划提供了重要参考。

对于问题三，在前两问的基础上，利用相关系数矩阵和相关性分析进一步考虑了农作物之间的可替代性和互补性和销售量、价格和成本之间的相关性。考虑可替代性和互补性后，模型发现某些作物的组合可以带来更高的经济效益和更低的风险。相关性分析揭示了销售量、价格和成本之间的动态关系，为种植策略的调整提供了依据。最终的种植策略不仅考虑了经济效益，还兼顾了生态效益和可持续性，为农业长期发展提供了支持。

七、总结

7.1 模型的优、缺点评价分析

7.1.1 模型的优点

优化模型：利用优化模型来分析农作物的最优种植策略，可以系统地考虑所有相关的决策变量和约束条件，提供一个全面的分析框架。通过数学公式明确地表达目标函数和约

束条件，定量地评估不同种植策略的经济效益，也可以轻松地通过调整模型参数来比较不同种植策略，从而选择最优方案。并且优化模型有助于进行长期规划，通过预测未来的需求和成本变化来制定战略。

利用正态分布考虑不确定性：正态分布是连续的，可以模拟各种可能的值，适合于模拟连续变量如销售量和价格。同时，它具有良好的数学性质，便于进行数学运算和统计分析，有大量的理论和工具支持。

多目标优化：在第二问和第三问中，利用多目标优化模型可以帮助我们生成一系列 Pareto 最优解，提供了多个可行的解决方案，增加了选择的灵活性。与传统的单目标优化相比，多目标优化模型可以同时考虑多个目标，更全面地反映实际问题的需求。模型可以根据实际情况的变化进行调整，具有较强的适应性，能够适应问题条件的变化。

7.1.2 模型的缺点

优化模型：优化模型的准确性高度依赖于输入数据的质量和完整性，数据不足或不准确可能导致次优解。优化模型通常是静态的，可能无法充分捕捉农业市场的动态变化和实时信息。

利用正态分布考虑不确定性：现实世界的数据可能不完全遵循正态分布，特别是在极端情况下，正态分布可能无法准确反映真实情况。

多目标优化：不同目标之间可能存在冲突，需要通过权衡和折中来寻找最优解。

7.2 模型的改进与推广

7.2.1 模型的改进

针对多目标优化模型来探索农作物种植方案，我们可以在本文的基础上进一步深入研究和扩展。具体而言，我们可以加入敏感性分析，细致探讨在问题 3 中引入的可替代性和互补性参数对最优策略的影响。这种分析有助于我们理解这些参数在不同情境下的作用，以及它们如何共同作用于种植策略的制定。同时，在后续评估过程中，除了考虑已有的相关性因素，我们还应将不同风险因素纳入考量范围。例如，价格波动和成本变化是农业生产中常见的风险因素，它们对最优策略的制定和执行具有重要影响。通过评估这些风险因素，我们可以更全面地了解种植策略在实际应用中的稳健性和适应性。此外，为了更好地应对外部环境的变化，我们可以设计更精确的环境变化检测机制。这一机制应能够实时监测关键环境因素的变化，如气候条件、土壤状况等，以便我们及时调整种植策略，快速响应外部环境的变化。通过这样的机制，我们可以确保种植策略始终与外部环境保持同步，从而提高农作物的产量和质量。

7.2.2 模型的推广

该模型可以进行跨行业应用推广。如渔业、畜牧业等自然资源管理领域同样存在产量、成本、价格及市场需求的不确定性，且不同资源间可能存在替代或互补关系。因此我们可以通过适当调整模型中的参数和约束条件，将本文的模型和方法可以应用于这些行业，从而制定出科学合理的资源利用策略，实现经济效益与生态保护的双重目标。

另一方面，本文模型可以与人工智能、大数据技术等先进技术相结合，构建一个智能化的农作物种植规划决策支持系统。这种系统可以实时收集、分析市场数据、气象信息、作物生长状况等多源数据，自动调整和优化种植规划方案，为决策者提供及时、准确的决策支持，从而推动农业生产的智能化和现代化进程。

参考文献

[1] 吕铭.五常市水稻有机种植模式推广研究[D].西北农林科技大学,2019.

[2] 丁雄.生态农业产业链系统协调与管理策略研究[D].南昌大学,2014.

[3] 田堃.黑龙江省有机种植现状与效益评估[D].东北农业大学,2015.

[4] 李想.河南省农作物虚拟水流动格局及种植结构优化调控[D].华北水利水电大学,2022.DOI:10.27144/d.cnki.ghbsc.2022.000028.

[5] 孙宁,冯利平.利用冬小麦作物生长模型对产量气候风险的评估[J].农业工程学报,2005,(02):106-110.

[6] 郝小宇.基于水资源高效利用的榆林市农业种植结构优化研究[D].西北农林科技大学,2019.

[7] 陈兆波.基于水资源高效利用的塔里木河流域农业种植结构优化研究[D].中国农业科学院,2008.

[8] 成桂红,成长玉,张林,等.浅谈一元线性加权回归在分析化学领域中的研究应用现状[J].中国食品卫生杂志,2023,35(02):303-310.DOI:10.13590/j.cjfh.2023.02.025.

[9] 景湉佳,贾世会,迟晓妮,等.基于线性加权和法的装配线平衡问题求解[J].现代制造工程,2024,(03):8-14+22.DOI:10.16731/j.cnki.1671-3133.2024.03.002.

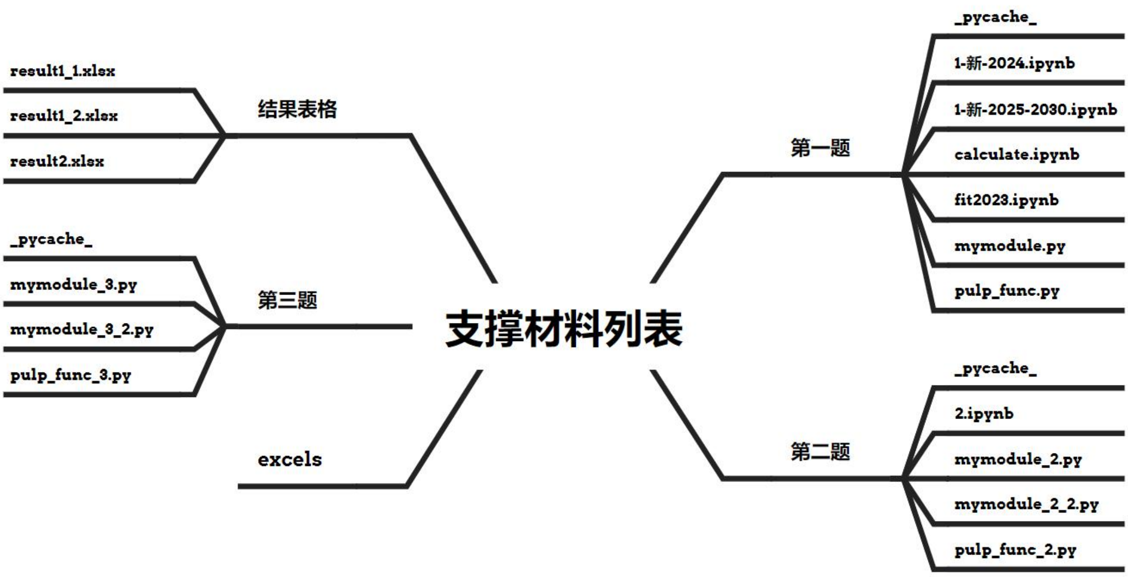
[10] 闫伟男.双碳背景下晋城市经济社会可持续发展研究——基于线性加权模型动态的分析[J].现代营销(下旬刊),2023,(04):107-109.DOI:10.19932/j.cnki.22-1256/F.2023.04.107.

[11] 周广为.支持新能源产业技术创新的财税政策研究[D].江西财经大学,2023.DOI:10.27175/d.cnki.gjxcu.2023.001828.

[12] 李娅.隆阳区蔬菜产业影响因素分析[D].云南农业大学,2023.DOI:10.27458/d.cnki.gynyu.2023.000005.

[13] 段泽丽,刘超群.农业生产与社会整合——基于一个华北村庄旱作梯田的社会研究[J].中国农业大学学报(社会科学版),2022,39(06):82-99.DOI:10.13240/j.cnki.caujsse.2022.06.011.
[14] 张垚.共生视角下分布式可再生能源开发规模及配置优化模型[D].华北电力大学(北京),2023.DOI:10.27140/d.cnki.ghbbu.2023.000037.
[15] 赵磊.卡车与无人机两级配送网络鲁棒优化模型及算法研究[D].吉林大学,2022.DOI:10.27162/d.cnki.gjlin.2022.007765.
[16] 解国玉.基于风险度量的多周期稀疏投资组合优化模型与实证研究[D].南京信息工程大学,2023.DOI:10.27248/d.cnki.gnjqc.2023.001068.
[17] 武旭.冷链物流协同配送多目标路径优化研究[D].兰州交通大学,2023.DOI:10.27205/d.cnki.gltec.2023.001005.
[18] 王冠然.含电动汽车的虚拟电厂运行优化与效益评价研究[D].华北电力大学(北京),2023.DOI:10.27140/d.cnki.ghbbu.2023.000546.
[19] 邓宗兵.中国农业全要素生产率增长及影响因素研究[D].西南大学,2010.
[20] 王晨,王济民.预期利润、农业政策调整对中国农产品供给的影响[J].中国农村经济,2018,(06):101-117.
[21] 赵瑞琴.农业产业集群竞争力研究[D].首都经济贸易大学,2021.DOI:10.27338/d.cnki.gsjmu.2021.000344.

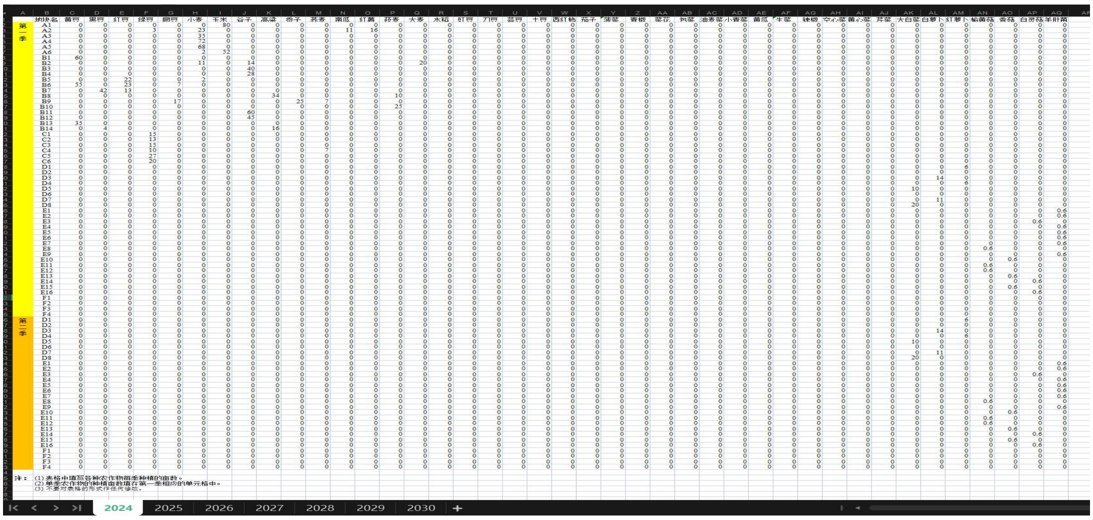
附录 A 支撑材料列表



附录 B 计算结果

（1）第一问计算结果

完整表格请见支撑材料中的 result1_1.xlsx 与 result1_2.xlsx，下图为局部展示：



（2）第二问计算结果

完整表格请见支撑材料中的 result2.xlsx， 下图为局部展示：

（3）第三问计算结果

见论文正文。

附录 C 自编程序代码

第一题

定义对象以存储表格数据

```
class Field:
    def __init__(self, field_name, field_type, field_area, crop_and_season, season, seasonmonth,
planted_crop):
        self.field_name = field_name
        self.field_type = field_type
```

```python
        self.field_area = field_area
        self.crop_and_season = crop_and_season
        self.season = season
        self.seasonmonth = seasonmonth
        self.planted_crop = planted_crop

    def __str__(self):
        return f'Field(planted_crop={self.planted_crop}, field_name={self.field_name}, 
field_type={self.field_type}, field_area={self.field_area}, crop_and_season={self.crop_and_season}, 
season={self.season}, seasonmonth={self.seasonmonth})'

class Crop:
    def __init__(self, crop_id, crop_name, crop_type, planting_fields, crop_price):
        self.crop_id = crop_id
        self.crop_name = crop_name
        self.crop_type = crop_type
        self.planting_fields = planting_fields
        self.crop_price = crop_price

    def __repr__(self):
        return f"Crop(crop_id={self.crop_id}, crop_name={self.crop_name}, crop_type={self.crop_type}, 
planting_fields={self.planting_fields}, crop_price={self.crop_price})"
```

初步构建优化模型

```python
def create_variables(new_fields, crops):
    # 创建决策变量
    variables = {}
    binary_variables = {}
    for field in new_fields:
        for crop in crops:
            var_name = f"{field.field_name}_{crop.crop_name}_{field.season}"
            variables[(field.field_name, crop.crop_name, field.season)] = pulp.LpVariable(var_name, 
lowBound=0)
            binary_var_name = f"binary_{var_name}"
            binary_variables[(field.field_name, crop.crop_name, field.season)] = 
pulp.LpVariable(binary_var_name, cat='Binary')
    # 创建实际销售量变量
    actual_sales = {}
    for crop in crops:
        actual_sales[crop.crop_name] = pulp.LpVariable(f"actual_sales_{crop.crop_name}", lowBound=0)
    # 创建超额部分变量
    excess_yield = {}
    for crop in crops:
        excess_yield[crop.crop_name] = pulp.LpVariable(f"excess_yield_{crop.crop_name}", lowBound=0)
    return variables, binary_variables, actual_sales, excess_yield
def define_objective_function(variables, actual_sales, excess_yield, crops, new_fields, k):
    # 收入部分：实际销售量 * 作物价格
```

```python
        revenue = pulp.lpSum([
            actual_sales[crop.crop_name] * crop.crop_price
            for crop in crops
        ])
        # 成本部分：种植成本
        cost = pulp.lpSum([
            variables[(field.field_name, crop.crop_name, field.season)] * get_cost(field, crop)
            for field in new_fields for crop in crops
        ])
        # 超额部分处理：超额部分 * 作物价格 * 系数
        excess_handling = pulp.lpSum([
            excess_yield[crop.crop_name] * crop.crop_price * k
            for crop in crops
        ])
        # 总利润 = 收入 - 成本 + 超额部分处理
        profit = revenue - cost + excess_handling
        return profit

    def add_constraints(prob, variables, binary_variables, actual_sales, excess_yield,
                        crops, new_fields, expected_sales_data, df_last1_result, df_last2_result,
                        min_area_percent, max_plots):
        # 添加约束条件
        # 地块面积限制
        for field in new_fields:
            prob += pulp.lpSum([variables[(field.field_name, crop.crop_name, field.season)] for crop in
crops]) <= field.field_area
        # 作物种植限制
        for field in new_fields:
            for crop in crops:
                if not check_crop_constraints(field, crop):
                    prob += variables[(field.field_name, crop.crop_name, field.season)] == 0
        # 重茬限制
        for field in new_fields:
            for crop in crops:
                if not check_rotation_constraints(field, crop, df_last1_result):
                    prob += variables[(field.field_name, crop.crop_name, field.season)] == 0
        # 豆类作物种植要求
        for field in new_fields:
            # 约束：确保三年内至少有一年种植过豆类作物
            prob += (
                pulp.lpSum([
                    binary_variables[(field.field_name, crop.crop_name, field.season)]
                    for crop in crops if crop.crop_type in ['粮食（豆类）', '蔬菜（豆类）']
                ]) +
                (df_last1_result[(df_last1_result['种植地块'] == field.field_name) &
                                 (df_last1_result['种植季次'] == field.season)]
                .iloc[:, 2:]
```

```
                .apply(lambda row: any(row[crop_name] > 0 for crop_name in row.index if crop_name in
[crop.crop_name for crop in crops if crop.crop_type in ['粮食（豆类）', '蔬菜（豆类）']]), axis=1)
                .any()) +
                (df_last2_result[(df_last2_result['种植地块'] == field.field_name) &
                                (df_last2_result['种植季次'] == field.season)]
                .iloc[:, 2:]
                .apply(lambda row: any(row[crop_name] > 0 for crop_name in row.index if crop_name in
[crop.crop_name for crop in crops if crop.crop_type in ['粮食（豆类）', '蔬菜（豆类）']]), axis=1)
                .any()) >= 1
            )
        # 实际销售量约束
        for crop in crops:
            actual_yield = pulp.lpSum([
                variables[(field.field_name, crop.crop_name, field.season)] * get_yield(field, crop)
                for field in new_fields
            ])
            expected_sales = get_expected_sales(crop, expected_sales_data)
            prob += actual_sales[crop.crop_name] <= actual_yield
            prob += actual_sales[crop.crop_name] <= expected_sales
            # 超额部分约束
            prob += excess_yield[crop.crop_name] == actual_yield - expected_sales # 这里不是>=，应该是==？！
            prob += excess_yield[crop.crop_name] >= 0
        # 作物在单个地块某一季节种植面积不宜太小
        for field in new_fields:
            for crop in crops:
                var = variables[(field.field_name, crop.crop_name, field.season)]
                binary_var = binary_variables[(field.field_name, crop.crop_name, field.season)]
                min_area = min_area_percent * field.field_area
                prob += var >= min_area * binary_var
                prob += var <= field.field_area * binary_var
        # 作物种植不宜太分散
        for crop in crops:
            for field_type in set(field.field_type for field in new_fields):
                if field_type not in ['普通大棚', '智慧大棚']:
                    prob += pulp.lpSum([
                        binary_variables[(field.field_name, crop.crop_name, field.season)]
                        for field in new_fields if field.field_type == field_type
                    ]) <= max_plots
    return prob
```

第二题

更新数据

```
    def update_expected_sales_data(expected_sales_data, expected_sales_data_2023, std_devs):
        # 定义增长率的均值
        wheat_growth_mean = (0.05 + 0.1) / 2
        corn_growth_mean = (0.05 + 0.1) / 2
        others_growth_mean = (-0.05 + 0.05) / 2
```

```python
        # 遍历 expected_sales 中的每一行
        for index, row in expected_sales_data.iterrows():
            crop_name = row['作物名称']
            if crop_name == '小麦':
                growth_rate = np.random.normal(loc=wheat_growth_mean, scale=std_devs[0])
                expected_sales_data.at[index, '预期销售量/斤'] *= (1 + growth_rate)
            elif crop_name == '玉米':
                growth_rate = np.random.normal(loc=corn_growth_mean, scale=std_devs[1])
                expected_sales_data.at[index, '预期销售量/斤'] *= (1 + growth_rate)
            else:
                # 从 expected_sales_2023 中找到对应的行
                row_2023 = expected_sales_data_2023[expected_sales_data_2023['作物名称'] == crop_name]
                if not row_2023.empty:
                    base_sales = row_2023.iloc[0]['预期销售量/斤']
                    growth_rate = np.random.normal(loc=others_growth_mean, scale=std_devs[2])
                    expected_sales_data.at[index, '预期销售量/斤'] = base_sales * (1 + growth_rate)
        return expected_sales_data
    def update_crops(crops, std_devs):
        # 定义增长率的均值
        yield_growth_mean = (-0.1 + 0.1) / 2
        cost_growth_mean = 0.05
        price_growth_mean_vegetables = 0.05
        price_growth_mean_mushrooms = (0.01 + 0.05) / 2
        # 遍历 crops 列表
        for crop in crops:
            for field in crop.planting_fields:
                # 更新亩产量
                yield_growth_rate = np.random.normal(loc=yield_growth_mean, scale=std_devs[3])
                field[2] *= (1 + yield_growth_rate)
                # 更新种植成本
                cost_growth_rate = np.random.normal(loc=cost_growth_mean, scale=std_devs[4])
                field[3] *= (1 + cost_growth_rate)
            # 更新作物价格
            if crop.crop_type in ['粮食', '粮食（豆类）', '粮食（水稻）']:
                price_growth_rate = np.random.normal(loc=0, scale=std_devs[5])
                crop.crop_price *= (1 + price_growth_rate)
            elif crop.crop_type in ['蔬菜', '蔬菜（水二）', '蔬菜（豆类）']:
                price_growth_rate = np.random.normal(loc=price_growth_mean_vegetables, scale=std_devs[6])
                crop.crop_price *= (1 + price_growth_rate)
            elif crop.crop_type == '食用菌':
                if crop.crop_name == '羊肚菌':
                    crop.crop_price *= 0.95  # 下降 5%
                else:
                    price_growth_rate = np.random.normal(loc=price_growth_mean_mushrooms,
scale=std_devs[7])
                    crop.crop_price *= (1 + price_growth_rate)
        return crops
```

多目标优化计算分数

```
def calculate_stability(past_profits):
    # 对往年利润进行一阶差分
    profit_diffs = np.diff(past_profits)
    # 计算标准差
    stability = np.std(profit_diffs)
    return stability
def calculate_score(profits, stability, weights):
    score = profits * weights[0] - stability * weights[1]
    return score
objective_value = optimize_planting_strategy(new_fields, crops, expected_sales_data, df_last1_result, 
df_last2_result, list)
df_template_list.append(update_template(df_template, new_fields))
past_profits.append(objective_value)
profits[year - 2023] = objective_value
stability = calculate_stability(past_profits)
stabilitys[year - 2023] = stability
score_now = calculate_score(objective_value, stability, weights)
```

第三题

添加约束条件

```
# 引入相关性约束。传入总体的相关性矩阵
for i, crop1 in enumerate(crops):
    for j, crop2 in enumerate(crops):
        if i != j:
            prob += actual_sales[crop1.crop_name] * correlation_matrix[i, j] <= 
actual_sales[crop2.crop_name]
# 引入回归关系
for crop in crops:
    price = crop.crop_price
    cost = get_cost(crop)
    # 根据作物类型选择对应的回归参数
    params = regression_params_dict.get(crop.crop_type)
    if params:
        prob += actual_sales[crop.crop_name] == params['intercept'] + params['price_coef'] * price + 
params['cost_coef'] * cost
    prob += actual_sales[crop.crop_name] == params['intercept'] + params['price_coef'] * price + 
params['cost_coef'] * cost
return prob
```