

农作物的种植策略

摘要

制定农作物的种植策略, 如何选择种植作物的类型十分重要。本文采用实整数编码种植策略, 建立了以单季种植耕地利润最大为第一层, 双季种植耕地利润最大为第二层的双层优化模型, 并且基于概率分布预估亩产量、销售价格等不确定性因素, 使用遗传算法进行求解, 最终给出了 2024-2030 年最优的农作物种植策略。

针对问题一, 本问首先对数据进行探索, 确定大棚在一季内至多分成两块 0.3 亩地分别种植作物,D1-D6 水浇地作为双季耕地,D7 与 D8 作为单季耕地。而后对数据进行预处理, 基于 2023 年实际种植的每种作物的斤数估计未来的预期销售量。将原始问题的 0-1 整数编码转化为**实整数 (RI) 编码**建立了滞销方案和降价出售方案的农作物最优种植方案的**双层优化模型**, 其中第一层以单季种植耕地利润最大为目标, 第二层以双季种植耕地利润最大为目标。基于**增强精英保留的遗传算法**首先求解第一层模型得到单季 28 块地的决策方案, 而后求解第二层模型得到 26 块双季耕地的决策方案。以滞销方案为例, 未来 2024 年到 2030 年的最优种植方案带来的净利润 (元) 为:**5862343.25, 4940066.75, 5376999.75, 5612970.25, 5518066.25, 5834346.75, 5551846.75**。

针对问题二, 针对农作物在生产过程中不可避免的自然波动和市场风险, 我们选取了适当的模型和概率分布来模拟这种**不确定性因素**。对于农作物的种植成本, 蔬菜类作物的销售价格与食用菌的销售价格平均每年稳定增加或减少, 我们应用**线性增长模型**计算未来的值。对于农作物的亩产量, 其他农作物的预期销售量, 价格的波动范围为对称区间的, 我们使用**正态分布**来量化不确定性。对于小麦和玉米的预期销售量和食用菌的销售价格波动范围在单侧区间的, 我们采取**对数正态分布**来量化不确定性, 基于上述分布函数计算出不同作物的 2024-2030 年的各种指标的变化结果修正问题一建立的滞销方案下的双层优化模型, 基于遗传算法求解得到最终结果。未来 2024 年到 2030 年的最优种植方案带来的净利润 (元) 为:**6034073.54, 5324687.48, 5832735.09, 6382019.87, 6011731.29, 6386223.45, 6433703.44**。

针对问题三, 各种农作物之间存在一定的可替代性和互补性, 因此本文根据实际经验并查找相关文献确定了哪些农作物可以互补或替代。可替代的农作物之间能够分担预期销售量。对于可互补的农作物, 给出了**联合互补增长率**的计算公式。使用**线性回归**拟合出销售价格、种植成本和预期销售量之间的相关性。基于问题二建立的概率分布函数和本问建立的联合互补增长率与线性回归重新计算未来预期销售量, 亩产量, 种植成本等指标作为第一问建立的双层优化模型的参数, 最终基于遗传算法求解得到未来 2024 年到 2030 年的最优种植方案带来的净利润 (元) 为:**7949359.86, 7357715.24, 8859056.78, 9392222.34, 8763917.13, 9668760.45, 10311257.81**。

关键字: 双层优化模型 实整数 (RI) 编码 遗传算法 正态分布 对数正态分布 联合互补增长率

一、 问题重述

1.1 问题背景

根据给定乡村的具体情况，充分利用有限的土地资源，因地制宜地发展有机种植业，更好地推动乡村经济的可持续发展。要求挑选合适的农作物并优化种植策略，从而实现简化田间管理的效果，提高生产效率，并减少不确定因素带来的各种种植风险。

已知某乡村位于华北山区，并且常年温度偏低，因此大多数耕地每年仅能进行一次作物种植。现乡村内有 1201 亩露天耕地，分布于 34 个不同规模的地块，包括平旱地、梯田、山坡地和水浇地四种类型。其中，平旱地、梯田和山坡地适合每年种植一季粮食作物，而水浇地则适合每年种一季水稻或两季蔬菜。此外，乡村还拥有 16 个普通大棚，适合每年种植一季蔬菜和一季食用菌，以及 4 个智慧大棚，适合每年种植两季蔬菜，每个大棚的面积均为 0.6 亩。在同一地块（包括大棚）中，每季可以进行不同作物的组合种植；详细信息见附件 1。

现知同一块土地上不应连续种植相同作物，以避免减产现象。由于豆类作物的根系能够有效改善土壤结构，从而有利于其他作物的生长，从 2023 年起规定每块土地必须在三年内最少种植一次豆类作物。此外，种植方案需要考虑耕作便利性和田间管理；2023 年的种植情况相关统计数据详见附件 2。

1.2 问题重述

现通过建立一定的数学模型，解决下列问题：

问题一：假设各类农作物的未来预期销售量、种植成本、亩产量和销售价格将在 2023 年的水平上保持稳定，并且每季种植的农作物将在当季内完成销售；如果某种作物的季度性总产量超出了对应的预期销售量，超出的部分将会无法正常销售。请根据以下两种情况，分别制定乡村在 2024 年至 2030 年的最优农作物种植方案并将结果分别填入表格。

(1) 超出部分无法销售，导致资源浪费；

(2) 超出部分将会以 2023 年销售价格一半进行降价处理。

问题二：小麦和玉米的预期销售量预计将增长，年均增长率介于 5% 至 10% 之间；其他的农作物的年销售量相较于 2023 上下浮动 5%；受气候等因素的影响，农作物的亩产量每年上下浮动 10%；受市场条件影响，种植成本年均增长 5%；粮食类作物的价格基本不变，而蔬菜类的价格呈上升趋势，年均增长约 5%；食用菌的销售价格总体下滑，年降幅在 1% 至 5% 之间，尤其是羊肚菌的销售价格年降幅为 5%。

通过综合考虑各类农作物在 2024 年至 2030 年期间的预期销售量、亩产量、种植成本和销售价格的不确定性及潜在风险，制定最优种植方案，并将结果填入 result2.xlsx。

问题三：在实际情况下，各类农产品存在相应的可替代性和互补性，同时预期销售量、销售价格和农作物的种植成本之间也存在一定的相关性。要求在问题 2 的基础上，综合考虑这些潜在的相关性因素，制定出乡村 2024 年至 2030 年的最优农作物种植策略；再结合模拟数据进行求解，通过对比分析与问题 2 结果之间的差异。

二、 问题分析

2.1 问题一的分析

首先对数据进行探索,得到例如大棚这些多季耕地在一季内可以分割种植作物,最小单位为 0.3 亩等重要结论。而后对数据进行预处理,分类汇总 2023 年实际种植的每种作物的斤数作为该作物未来的预期销售量。本问对农作物最优种植方案优化模型可以基于 0-1 整数决策变量和实整数 (RI) 编码的目标函数和约束条件;再建立双层优化模型,第一层以单季种植耕地利润最大为目标,第二层以双季种植耕地利润最大为目标。最后基于遗传算法求解上述双层优化模型。

2.2 问题二的分析

针对农作物在生产过程中不可避免的自然波动和市场风险,选取适当的模型和概率分布来模拟和预测农作物的销售价格。对于农作物的种植成本,蔬菜类作物的销售价格与食用菌的销售价格平均每年稳定增加或减少,可以应用线性增长模型计算未来的值;而对于农作物的亩产量,其他农作物的预期销售量,价格的波动范围为对称区间的,可以使用正态分布来量化不确定性;对于小麦和玉米的预期销售量和食用菌的销售价格波动范围在单侧区间的,可以采取对数正态分布来量化不确定性,基于上述分布函数计算出不同作物的 2024-2030 年的各种指标的变化结果,进行修正问题一建立的双层优化模型,再结合遗传算法求解得到最终结果。

2.3 问题三的分析

在现实生活中,各种农作物之间存在一定的可替代性和互补性,根据实际经验并查找相关文献确定了哪些农作物之间可以被替代,哪些农作物之间可以互补。对于可互补的农作物之间能够分担未来预期销售量;对于可替代的农作物,可以对决策编码进行替换。另外,预期销售量与销售价格、种植成本之间也存在一定的相关性,因此可以在问题二的基础上,以销售价格、种植成本为自变量,预期销售量为因变量,对五种作物大类进行线性回归拟合以反映出指标之间的相关性。

三、 模型基本假设

- (1) 通过对 2023 年实际种植情况发现大棚可以在一季内进行分割地块种植,且最小单位都是 0.3 亩,为了方便耕种作业和田间管理,假设每块地的最小分割单位为 0.3 亩
- (2) 假设 D1-D6 在未来一直视为双季种植耕地,D7-D8 在未来一直视为单季种植耕地。
- (3) 每季种植的农作物在当季销售,而普通大棚和智慧大棚的第二季时间范围横跨两年,因此我们假设第二季的销售量与赚的利润算到来年上。
- (4) 在实际生活中,种植农作物超出预售量的部分一般会滞销造成浪费,因此我们假设第二问是基于滞销这种情况计算。

四、 符号说明

符号	符号说明
$x_{ijk}^{(m)}$	m 类型的第 i 块地在第 k 次决策中是否种植第 j 种作物
A	农作物在特定耕地内的种植面积
P	农作物每亩的产量
V	农作物每斤的价格
C	农作物每亩的成本
$lower$	某耕地类型可种植的粮食的最小编号
$upper$	某耕地类型可种植的粮食的最大编号
R_j	农作物的需求量
r	农作物溢出的价格
T	地的类型
$rate$	增长率
Y	预期销售量
E	预期销售价格
R	预期销售价格
f_1, f_2, f_3	预期销售量与销售价格和种植成本的回归模型

注: 其他符号将在文中具体说明.

五、 问题一的模型建立与求解

5.1 问题一的分析

本问首先对数据进行了探索, 得到了例如大棚这些多季耕地在一季内可以分割最小单位为 0.3 亩种植两种作物等重要结论。而后对数据进行预处理, 分类汇总 2023 年实际种植的每种作物的斤数作为该作物未来的预期销售量。本问对农作物最优种植方案优化模型分别建立了基于 0-1 整数编码和实整数 (RI) 编码的目标函数和约束条件, 分析了每种编码的特点。建立了双层优化模型, 第一层以单季种植耕地利润最大为目标, 第二层以双季种植耕地利润最大为目标。最后基于遗传算法求解上述双层优化模型。

5.2 数据准备

5.2.1 原始数据分析

根据附件 1 中的表1, 可以看出平旱地和梯田的总面积较大, 但只能种植一季作物, 这可能导致这两种地块的利用效率相对较低。相比之下, 水浇地面积较小, 但能够种植一季或两季作物, 因此显示出更高的灵活性和生产效率。大棚的总面积较小, 但每年可以种植两季作物, 显示出较强的实际生产能力。其中 D1-D6 水浇地在 2023 年种植的是双季作物, 为了方便耕种作业和田间管理, 我们认为其在未来依旧继续种植双季作物, 将 **D1-D6 归为双季耕地**。D7,D8 水浇地在 2023 年种植的是单季作物, 将 **D7-D8 归为单季耕地**。

对原始附件进行分析, 如表2, 大棚可以在一季种植多种蔬菜, 但是所有的大棚最多只选择两种蔬菜组合, 为了方便耕种作业和田间管理, 我们认为把大棚**每一季分为两个 0.3 亩地**是合

理的, 满足每种作物在单个地块种植的面积不宜太小的要求。

表 1 2023 年农作物种植情况 (汇总)

	平旱地	梯田	山坡地	D1-D6	D7-D8	普通大棚	智慧大棚
块数 (块)	6	14	6	6	2	16	4
总面积 (亩)	385	585	108	67	42	9.6	2.4
种植情况 (单季/双季)	单季	单季	单季	双季	单季	双季	双季

表 2 2023 年农作物种植 (原始附件)

种植地块	作物编号	作物名称	作物类型	种植面积/亩	种植季次
F3	17	豇豆	蔬菜 (豆类)	0.6	第一季
	28	小青菜	蔬菜	0.3	第二季
	30	生菜	蔬菜	0.3	第二季
	19	芸豆	蔬菜 (豆类)	0.6	第一季
F4	34	芹菜	蔬菜	0.3	第二季
	23	菠菜	蔬菜	0.3	第二季

从附件 2 的 2023 统计的相关数据来看, 如表3, 平旱地上的多数作物产量较高, 如玉米 (1000 斤/亩) 和南瓜 (3000 斤/亩), 但成本也较高。梯田的产量普遍低于平旱地, 如玉米的产量为 950 斤/亩, 低于平旱地的 1000 斤/亩, 而成本相同或更低。山坡地产量通常略低于平旱地, 其成本则较高。水浇地的产量较高, 如南瓜产量达到 3000 斤/亩, 但成本也较高。大棚种植的产量通常较高, 但成本也较高。由此可见, 产量与成本之间存在一定的正向关系。通过对比数据我们发现南瓜、红薯和黄瓜为高产作物, 在各个耕地上的产量都较高; 空心菜、羊肚菌和白灵菇是高成本作物, 其中羊肚菌成本高, 但产量相对低, 因此销售价格也非常高。

表 3 2023 年统计的相关数据

序号	作物编号	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)
1	1	黄豆	平旱地	单季	400	400	2.504.00
2	2	黑豆	平旱地	单季	500	400	6.508.50
3	3	红豆	平旱地	单季	400	350	7.509.00
4	4	绿豆	平旱地	单季	350	350	6.008.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
106	33	黄心菜	智慧大棚	第二季	5400	2750	4.806.00
107	34	芹菜	智慧大棚	第二季	6000	1200	3.805.80

表 4 2023 年的农作物种植情况

种植地块	作物编号	作物名称	作物类型	种植面积/亩	种植季次
A1	6	小麦	粮食	80	单季
A2	7	玉米	粮食	55	单季
A3	7	玉米	粮食	35	单季
A4	1	黄豆	粮食（豆类）	72	单季
A5	4	绿豆	粮食（豆类）	68	单季
⋮	⋮	⋮	⋮	⋮	⋮
	19	芸豆	蔬菜（豆类）	0.6	第一季
F4	34	芹菜	蔬菜	0.3	第二季
	23	菠菜	蔬菜	0.3	第二季

5.2.2 数据预处理

问题一假定各种农作物未来的预期销售量相对于 2023 年保持稳定。因此我们可以根据表4,得到每个种植耕地所种的作物和该作物的种植面积(亩)。结合表3得到对应作物的亩产量(斤),从而计算出每个种植耕地所种的作物产量,再对作物进行分类汇总,最终得到 2023 年种植的每种作物的产量,作为估计的销售量,如表5所示。

表 5 根据 2023 年实际数据估计的每种作物的预期销售量

作物名称	预计销售量(斤)	作物名称	预计销售量(斤)	作物名称	预计销售量(斤)	作物名称	预计销售量(斤)	作物名称	预计销售量(斤)
刀豆	26880	小麦	170840	白灵菇	18000	羊肚菌	4200	芹菜	3780
包菜	5280	榆黄菇	9000	白萝卜	100000	芸豆	8400	茄子	47760
南瓜	35100	水稻	21000	空心菜	6900	菠菜	1890	荞麦	1500
土豆	30000	油麦菜	4500	红萝卜	36000	西红柿	37110	莜麦	14000
大白菜	150000	爬豆	9875	红薯	36000	谷子	71400	菜花	4680
大麦	10000	玉米	132750	红豆	22400	豇豆	38400	青椒	3510
小青菜	36680	生菜	4350	绿豆	33040	辣椒	1200	香菇	7200
高粱	30000	黄瓜	17550	黍子	12500	黄心菜	3420	黄豆	57000
黑豆	21850								

因为附件给的销售单价(元/斤)是区间类型数据,因此我们需要对其进行处理,拆分得到该作物的最高,最低和平均单价,在后文主要以平均单价进行参与计算,如表6所示。

表 6 计算平均单价

作物名称	地块类型	种植季次	销售单价	最低单价	最高单价	平均单价
黄豆	平旱地	单季	2.50-4.00	2.5	4.0	3.25
黑豆	平旱地	单季	6.50-8.50	6.5	8.5	7.50
红豆	平旱地	单季	7.50-9.00	7.5	9.0	8.25
绿豆	平旱地	单季	6.00-8.00	6.0	8.0	7.00

5.3 农作物最优种植方案优化模型的建立

5.3.1 决策变量

引入 0-1 变量 $x_{ijk}^{(m)}$ 表示 m 类型的第 i 块地在第 k 次决策中是否种植第 j 种作物, m 的具体含义如下表。

表 7 决策变量中 m 的意义

	m	i, j	m 的意义
单季种植耕地	1	$i = 1, 2, \dots, 6, j = 1, 2, \dots, 15$	平旱地
	2	$i = 1, 2, \dots, 14, j = 1, 2, \dots, 15$	梯田
	3	$i = 1, 2, \dots, 6, j = 1, 2, \dots, 15$	山坡地
	4	$i = 1, 2, j = 1$	水浇地
双季种植耕地	5	$i = 1, 2, \dots, 16, j = 1, 2, \dots, 17$	普通大棚第一季
	6	$i = 1, 2, \dots, 4, j = 1, 2, \dots, 17$	智慧大棚第一季
	7	$i = 1, 2, \dots, 16, j = 1, 2, \dots, 4$	普通大棚第二季
	8	$i = 1, 2, \dots, 4, j = 1, 2, \dots, 17$	智慧大棚第二季
	9	$i = 1, 2, \dots, 6, j = 1, 2, \dots, 17$	水浇地第一季
	10	$i = 1, 2, \dots, 6, j = 1, 2, \dots, 3$	水浇地第二季

5.3.2 约束条件

约束条件一: 每种作物在同一地块 (含大棚) 都不能连续重茬种植

$$x_{ijk}^{(m)} \neq x_{ij(k+1)}^{(m)}, m = 1, 2, 3, 4 \quad (1)$$

由于平旱地、梯田、山坡地和水浇地这四种单季种植耕地的每次决策都在固定的一年内进行, 因此第 k 次决策可以直接表示为从 2023 年开始的第 k 年。

$$\begin{cases} x_{ijk}^{(7)} \neq x_{ij(k+1)}^{(5)}, m = 5, 6, \dots, 10 \\ x_{ijk}^{(8)} \neq x_{ij(k+1)}^{(6)}, m = 5, 6, \dots, 10 \\ x_{ijk}^{(9)} \neq x_{ij(k+1)}^{(10)}, m = 5, 6, \dots, 10 \end{cases} \quad (2)$$

普通大棚和智慧大棚这两种双季种植耕地需要将当年的第二季决策方案与来年的第一季决策方案进行比较, 以避免连续种植约束。因此, k 代表第 k 次决策, 而不能直接表示年份。

- 普通大棚的第一季一般安排在每年的 5 月至 9 月之间, 第二季则从 9 月持续到次年 4 月;
- 智慧大棚的第一季一般安排在每年的 3 月至 7 月之间, 第二季则从 8 月持续到次年的 2 月。

约束条件二: 要求每个耕地 (含大棚) 的所有土地三年内至少种植一次豆类作物

$$x_{ijk}^{(m)} + x_{ij(k+1)}^{(m)} + x_{ij(k+2)}^{(m)} \geq 1, m = 1, 2, 3, 4, i, k = 1, 2, \dots, j \in G \quad (3)$$

其中 G 是表示豆类作物的编号集合, 我们将连续三年内至少种植一次豆类作物转化为上述不等式的约束, 从而确保存在有某个决策变量取值为 1。

$$\sum_{T=0}^5 x_{ij(k+T)}^{(m)} \geq 1, m = 5, 6, \dots, 10, i, k = 1, 2, \dots, j \in G \quad (4)$$

对于双季种植地连续六次决策 (每次决策一季) 可以包含三年的种植情况。

5.3.3 目标函数

$$\max \sum_{m=1}^4 \sum_k \sum_i x_{ijk}^{(m)} \cdot \left(\underbrace{A_{ij} \cdot P_{ij} \cdot V_{ij}}_{\text{售价}} - \underbrace{C_{ij} \cdot A_{ij}}_{\text{成本}} \right) + \sum_{m=5}^{10} \sum_k \sum_i x_{ijk}^{(m)} \cdot \left(\underbrace{A_{ij} \cdot P_{ij} \cdot V_{ij}}_{\text{售价}} - \underbrace{C_{ij} \cdot A_{ij}}_{\text{成本}} \right) \quad (5)$$

$x_{ijk}^{(m)}$ 表示是否种植该作物； A_{ij} 表示该作物在该地产的种植面积； P_{ij} 表示每亩的产量； V_{ij} 表示每斤的价格； C_{ij} 表示该作物每亩的成本。根据这些变量，可以计算出售价和总种植成本。售价减去种植成本即为利润，对所有作物以及 2024 至 2030 年的总利润进行求和，并最大化总利润。

其中， A 、 P 、 V 和 C 都为常量。

5.3.4 模型汇总

$$\begin{aligned} \max & \sum_{m=1}^4 \sum_k \sum_i x_{ijk}^{(m)} \cdot \left(\underbrace{A_{ij} \cdot P_{ij} \cdot V_{ij}}_{\text{售价}} - \underbrace{C_{ij} \cdot A_{ij}}_{\text{成本}} \right) + \sum_{m=5}^{10} \sum_k \sum_i x_{ijk}^{(m)} \cdot \left(\underbrace{A_{ij} \cdot P_{ij} \cdot V_{ij}}_{\text{售价}} - \underbrace{C_{ij} \cdot A_{ij}}_{\text{成本}} \right) \\ \text{s.t.} & \begin{cases} x_{ijk}^{(m)} \neq x_{ij(k+1)}^{(m)}, m = 1, 2, 3, 4 \\ x_{ijk}^{(7)} \neq x_{ij(k+1)}^{(5)}, m = 5, 6, \dots, 10 \\ x_{ijk}^{(8)} \neq x_{ij(k+1)}^{(6)}, m = 5, 6, \dots, 10 \\ x_{ijk}^{(9)} \neq x_{ij(k+1)}^{(10)}, m = 5, 6, \dots, 10 \\ x_{ijk}^{(m)} + x_{ij(k+1)}^{(m)} + x_{ij(k+2)}^{(m)} \geq 1, m = 1, 2, 3, 4, i, k = 1, 2, \dots, j \in G \\ \sum_{T=0}^5 x_{ij(k+T)}^{(m)} \geq 1, m = 5, 6, \dots, 10, i, k = 1, 2, \dots, j \in G \end{cases} \end{aligned} \quad (6)$$

对于上述建立的农作物最优种植方案原始优化模型，一定可以求解出最优解，但是其决策空间庞大，仅以 A1-A6 为代表的平旱地一年的可能情况都有 $\underbrace{15 \times \dots \times 15}_6 = 1.1390 \times 10^7$ ，计算难度大，因此必须要设计一种特定的编码方式简化 **0-1 整数变量**的决策空间并且设计好的求解算法以较低的计算复杂度，使我们能够以一定的启发式规则逼近上述原始模型的最优解。

5.4 基于实整编码 (RealInteger Encoding) 的决策变量

通过实际 2023 年的实际种植情况发现，**D7 和 D8 两个水浇地**为单季地块并且只能种植水稻，而其他种植耕地类型不能种植水稻，因此不会与其他地块产生水稻需求量的冲突，且每年决策方案固定，无需进行编码参与优化，为决策常量。

单季种植耕地的个体编码方式

本文基于 RI 编码定义的单季种植耕地的个体编码规则如下集合所示：

$$X = \{x_i | x_i \in (lower, upper), x_i \in Z, i = 1, 2, \dots, 26\} \quad (7)$$

其中 $lower$ 为该种地形可种植的粮食最小编号， $upper$ 为该种地形可种植的粮食最大编号。如表8所示，这三种地形的 $lower$ 取值都为 1， $upper$ 取值都为 15。 $x_1 \sim x_{26}$ 分别表示 A1, A2, ...

, A6, B1, B2, ..., B14, C1, C2, C3, ..., C6 这 26 块单季种植耕地对应的选择种植的作物编号。

表 8 平旱地, 梯田, 山坡地可种植作物类型

作物名称	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
编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

以 2023 年 A1-C6 的选择种植方案为例, 我们可以将染色体编码为:

$$\{6, 7, 7, 1, 4, 8, 6, 2, 3, 4, 5, 8, 6, 8, 9, 10, 1, 7, 14, 15, 11, 12, 1, 13, 6, 3\} \quad (8)$$

其中, 第一个位置对应的 6 表示 A1 选择种植小麦作物, 末尾的 3 表示 C6 选择种植红豆作物。

多季种植耕地的个体编码方式

对于多季种植耕地的个体编码规则较为复杂。首先每一块地需要分别决策第一季与第二季, 并且在一个季中可以分割种植耕地的面积, 例如 F2 种植耕地在 2023 年的第一季种植菜花 0.3 亩, 种植包菜 0.3 亩。因此我们可以基于前文定义的规则, 用四种决策编码分别表示季与分割的每小块地种植的作物类型, 而后进行拼接。

$$\begin{aligned}
X^{(1)} &= \{x_i | x_i \in (lower_i^{(1)}, upper_i^{(1)}), x_i \in Z, i = 1, 2, \dots, 20\} \\
X^{(2)} &= \{x_i | x_i \in (lower_i^{(2)}, upper_i^{(2)}), x_i \in Z, i = 1, 2, \dots, 20\} \\
X^{(3)} &= \{x_i | x_i \in (lower_i^{(3)}, upper_i^{(3)}), x_i \in Z, i = 1, 2, \dots, 20\} \\
X^{(4)} &= \{x_i | x_i \in (lower_i^{(4)}, upper_i^{(4)}), x_i \in Z, i = 1, 2, \dots, 20\} \\
X^{(5)} &= \{x_i | x_i \in (lower_i^{(5)}, upper_i^{(5)}), x_i \in Z, i = 1, 2, \dots, 6\} \\
X^{(6)} &= \{x_i | x_i \in (lower_i^{(6)}, upper_i^{(6)}), x_i \in Z, i = 1, 2, \dots, 6\} \\
X &= X^{(1)} + X^{(2)} + X^{(3)} + X^{(4)} + X^{(5)} + X^{(6)}
\end{aligned} \quad (9)$$

此处的 + 号表示**集合拼接**操作。

- $X^{(1)}$ 是 20 块大棚第一季前 0.3 亩地的种植作物方案
- $X^{(2)}$ 是 20 块大棚第一季后 0.3 亩地的种植作物方案
- $X^{(3)}$ 是 20 块大棚第二季前 0.3 亩地的种植作物方案
- $X^{(4)}$ 是 20 块大棚第二季后 0.3 亩地的种植作物方案
- $X^{(5)}$ 是 6 块水浇地第一季前 0.3 亩地的种植作物方案
- $X^{(6)}$ 是 6 块水浇地第二季后 0.3 亩地的种植作物方案
- $X_{1 \sim 16}^{(1 \sim 4)}$ 表示 E1-E16 普通大棚的种植作物, $X_{17 \sim 20}^{(1 \sim 4)}$ 表示 F1-F4 智慧大棚的种植作物。

结合表9和表10**确定决策变量取值的上下界**:

- $lower_i^{(1)} = lower_i^{(2)} = lower_i^{(5)} = 17$, 都为 17 对应的豇豆.
- $upper_i^{(1)} = upper_i^{(2)} = upper_i^{(5)} = 34$, 都为 34 对应的芹菜.
- $lower_i^{(3)} = lower_i^{(4)} = 38, i = 1, 2, \dots, 16$, E1-E16 普通大棚都为 38 对应的榆黄菇.
- $upper_i^{(3)} = upper_i^{(4)} = 41, i = 1, 2, \dots, 16$, E1-E16 普通大棚都为 41 对应的羊肚菌.
- $lower_i^{(3)} = lower_i^{(4)} = 17, i = 17, \dots, 20$, F1-F4 智慧大棚都为 17 对应的豇豆.

- $upper_i^{(3)} = upper_i^{(4)} = 34, i = 17, \dots, 20$, F1-F4 智慧大棚都为 34 对应的芹菜.
- $lower_i^{(6)} = 35, i = 1, 2, \dots, 6$, D1-D6 水浇地都为 35 对应的大白菜.
- $upper_i^{(6)} = 37, i = 1, 2, \dots, 6$, D1-D6 水浇地都为 37 对应的红萝卜.

表 9 水浇地, 普通大棚, 智慧大棚第一季可种植作物类型

作物名称	豇豆	刀豆	芸豆	土豆	西红柿	茄子	菠菜	青椒	菜花
编号	17	18	19	20	21	22	23	24	25
作物名称	包菜	油麦菜	小青菜	黄瓜	生菜	辣椒	空心菜	黄心菜	芹菜
编号	26	27	28	29	30	31	32	33	34

表 10 普通大棚第二季可种植作物类型

作物名称	榆黄菇	香菇	白灵菇	羊肚菌
编号	38	39	40	41

以 2023 年 E1-F4 这二十个 (普通/智慧) 大棚的选择种植方案为例, 我们的染色体编码为:

$$\begin{aligned}
 & \underbrace{\{18, 24, 25, 26, 28, 27, 19, 19, 18, 17, 17, 22, 21, 29, 30, 31, 32, 25, 17, 19\}}_{\text{第一季第一块 } 0.3 \text{ 亩地的决策方案}} \\
 & + \underbrace{\{18, 24, 25, 26, 28, 27, 19, 19, 18, 17, 17, 22, 21, 29, 27, 31, 33, 26, 17, 19\}}_{\text{第一季第二块 } 0.3 \text{ 亩地的决策方案}} \\
 & + \underbrace{\{38, 38, 38, 39, 39, 39, 40, 40, 40, 41, 41, 41, 41, 41, 41, 24, 22, 28, 34\}}_{\text{第二季第一块 } 0.3 \text{ 亩地的决策方案}} \\
 & + \underbrace{\{38, 38, 38, 39, 39, 39, 40, 40, 40, 41, 41, 41, 41, 41, 41, 21, 29, 30, 23\}}_{\text{第二季第二块 } 0.3 \text{ 亩地的决策方案}} \\
 & + \underbrace{\{20, 28, 21, 22, 17, 18\}}_{\text{水浇地第一季的决策方案}} + \underbrace{\{36, 35, 35, 35, 36, 37\}}_{\text{水浇地第二季的决策方案}}
 \end{aligned} \tag{10}$$

如表11, 为 E1 与 E2 两个普通大棚和 F1-F3 三个智慧大棚在 2023 年的种植情况。E1 第一季种植 0.6 亩的刀豆, 可以理解为第一季 E1 的第一块和第二块 0.3 亩地的决策取值都为 18, 对应染色体编码的第 1 和第 21 处位置。F3 第二季种植 0.3 亩小青菜, 0.3 亩生菜, 可以理解为第二季 F3 的第一块 0.3 亩地决策取值为 28, 对应染色体编码的第 59 处位置, 第二块 0.3 亩地的决策取值为 30, 对应染色体编码的第 79 处位置。

表 11 2023 年普通大棚和智慧大棚种植情况

地块	作物编号	作物名称	种植面积	季次	地块	作物编号	作物名称	种植面积	季次
E1	18	刀豆	0.6	第一季	F2	25	菜花	0.3	第一季
	38	榆黄菇	0.6	第二季		26	包菜	0.3	第一季
E2	24	青椒	0.6	第一季		22	茄子	0.3	第二季
	38	榆黄菇	0.6	第二季		29	黄瓜	0.3	第二季
F1	32	空心菜	0.3	第一季	F3	17	豇豆	0.6	第一季
	33	黄心菜	0.3	第一季		28	小青菜	0.3	第二季
	24	青椒	0.3	第二季		30	生菜	0.3	第二季
	21	西红柿	0.3	第二季					

5.5 双层优化模型的建立

5.5.1 第一层: 单季种植耕地的农作物最优种植方案优化模型

基于 RI 编码的目标函数

优化的目标是最大化在第 k 次决策中, 根据农作物的最优种植方案所获得的利润:

$$\max \left(\sum_{i=1}^{26} \left(\underbrace{A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}}}_{\text{种植该作物的平均售价}} - \underbrace{C_{T_i, x_i^{(k)}} \times A_i}_{\text{种植成本}} \right) - \underbrace{\sum_{j=1}^{15} r_j^{(k)}}_{\text{溢出价格}} \right) \quad (11)$$

其中 A_i 表示第 i 块地的面积 (亩); T_i 表示第 i 块地的类型 (平旱地、梯田、山坡地), 其中用 0、1、2 分别表示这三种类型; C_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的单位成本 (元/亩), 取决于地块类型 T_i 和作物编号 x_i ; P_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的亩产量 (斤/亩); E_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的单位售价 (元/斤); R_j 表示作物 j 的需求量即未来销售量 (斤)。 $r_j^{(k)}$ 表示第 k 次决策第 j 种作物溢出的价格; $x_i^k \in X^k$ 为前文为二十六块单季种植耕地定义的决策编码; k 表示第几次决策, 对于单季种植耕地第 k 次决策就是第 k 年决策, 如 $X^{(0)}$ 表示的就是第 0 年, 即 2023 年的实际种植情况, 对应前文的 (8)。

对于超出部分导致滞销和浪费的情况, 溢出价格 r_j 计算公式如下所示:

$$r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

其中 R_j 表示第 j 种作物的预期销售量, 取值如表5所示; $\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}, x_i = j$ 表示的是第 j 种作物在 26 块单季种植耕地的总种植斤数, 用它减去其当年的预期销售量 (斤), 如果差值大于 0 说明有溢出, 小于 0 说明无溢出。

针对超出部分按 2023 年售价的半价进行处理的情况, 溢出价格 r_j 计算公式如下所示:

$$r_j^{(k)} = \begin{cases} 0.5 \times \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

基于 RI 编码的约束条件

约束条件一: 每种作物不能在相同地块（包括大棚）上连续重复种植。

$$x_i^{(k)} \neq x_i^{(k+1)}, i = 1, 2, 3, \dots, 26 \quad (14)$$

反映在决策变量上, 即第 k 和第 $k+1$ 得到的决策编码对应位置元素不同。

约束条件二: 要求每块地在三年时间里, 各个土地上至少要种植豆类作物一次。

$$\exists x_i^{(k)}, x_i^{(k+1)}, x_i^{(k+2)} \in \{1, 2, 3, 4, 5\} \quad (15)$$

反映在决策变量上, 即第 $k, k+1, k+2$ 得到的决策编码第 i 个位置元素需要存在 15 的值。

第一层优化模型汇总

超出部分导致滞销和浪费的情况:

$$\begin{aligned} \max & \left(\sum_{i=1}^{26} \left(\underbrace{A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}}}_{\text{种植该作物的平均售价}} - \underbrace{C_{T_i, x_i^{(k)}} \times A_i}_{\text{种植成本}} \right) - \underbrace{\sum_{j=1}^{15} r_j^{(k)}}_{\text{溢出价格}} \right) \\ \text{s.t.} & \begin{cases} r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_i^{(k+1)}, i = 1, 2, 3, \dots, 26 \\ \exists x_i^{(k)}, x_i^{(k+1)}, x_i^{(k+2)} \in \{1, 2, 3, 4, 5\} \end{cases} \end{aligned} \quad (16)$$

超出部分按 2023 年售价的半价进行处理的情况:

$$\begin{aligned} \max & \left(\sum_{i=1}^{26} \left(\underbrace{A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}}}_{\text{种植该作物的平均售价}} - \underbrace{C_{T_i, x_i^{(k)}} \times A_i}_{\text{种植成本}} \right) - \underbrace{\sum_{j=1}^{15} r_j^{(k)}}_{\text{溢出价格}} \right) \\ \text{s.t.} & \begin{cases} r_j^{(k)} = \begin{cases} 0.5 \times \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_i^{(k+1)}, i = 1, 2, 3, \dots, 26 \\ \exists x_i^{(k)}, x_i^{(k+1)}, x_i^{(k+2)} \in \{1, 2, 3, 4, 5\} \end{cases} \end{aligned} \quad (17)$$

5.5.2 第二层: 双季种植耕地的农作物最优种植方案优化模型

基于 RI 编码的目标函数

优化的目标是最大化在第 k 次决策中所确定的农作物最优种植方案的利润:

$$\begin{aligned} \max \left(\underbrace{\sum_{i \in [1,40] \cup [81,86]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right)}_{\text{第一季种植当年收获}} - \sum_{j=1}^{15} r_j^{(k)} \right) \\ + \left(\underbrace{\sum_{i \in [41,80] \cup [87,92]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right)}_{\text{第二季种植来年收获}} - \sum_{j=1}^{15} r_j^{(k+1)} \right) \end{aligned} \quad (18)$$

其中 A_i 表示第 i 块地的面积 (亩); T_i 表示第 i 块地的类型 (普通大棚, 智慧大棚, 水浇地), 其中用 0, 1, 2 分别表示这三种类型; C_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的单位成本 (元/亩), 取决于地块类型 T_i 和作物编号 x_i ; P_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的单位产量 (斤/亩); E_{T_i, x_i} 表示在第 i 块地上种植作物 x_i 的单位售价 (元/斤); R_j 表示作物 j 的需求量 (斤);

其中 $x_i^{(k)}$ 是前文为双季种植耕地定义的决策编码 [10], 由于双季作物第二季种植作物只能在来年收获售卖, 因此第二季应该考虑的是下一年的需求量 $r_j^{(k+1)}$, 因为平旱地, 梯田和山坡地所种的作物编号为 1-16, 双季种植耕地所种的作物编号为 17-41, 彼此不冲突, 所以**第一层优化模型的计算结果 (单季种植的作物数量) 不会影响 17-41 编号的作物的实际需求量。**

对于超出部分导致滞销和浪费的情况, 溢出价格 r_j 计算公式如下所示:

$$r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

其中 R_j 表示第 j 种作物的预期销售量, 取值如表5所示; $\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}, x_i = j$ 表示的是 j 种作物在二十六块双季种植耕地的总种植斤数, 用减去其当年的预期销售量 (斤数), 如果差值大于 0 说明有溢出, 小于 0 说明无溢出。

对于超过部分按 2023 年售价的半价处理的情况, 溢出价格 $r_j^{(k)}$ 计算公式如下所示:

$$r_j^{(k)} = \begin{cases} 0.5 \times \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

基于 RI 编码的约束条件

约束条件一: 每种作物不能在相同地块 (包括大棚) 上连续重复种植

1. 当年第二季不能和当年第一季种植作物相同

$$x_i^{(k)} \neq x_{i+40}^{(k)}, i = 1, 2, 3, \dots, 40 \quad (21)$$

反映在决策变量上, 即第 k 次的决策编码 i 与 $i+40$ 位置元素不同。

2. 当年第一季不能和前年第二季种植作物相同

$$x_{i+40}^{(k-1)} \neq x_i^{(k)}, i = 1, 2, 3, \dots, 40 \quad (22)$$

反映在决策变量上, 即第 k 次的决策编码的 i 位置与第 $k-1$ 次的决策编码的 $i+40$ 位置元素不同。

约束条件二: 要求每块地在三年时间里, 各个土地上至少要种植豆类作物一次

$$\exists x_i^{(Step)}, x_{i+20}^{(Step)}, x_{i+40}^{(Step)}, x_{i+60}^{(Step)} \in \{17, 18, 19\}, Step = 1, 2, 3 \quad (23)$$

上述四个变量表示两季两块 0.3 亩地种植的作物编号。

通过对表12的分析, 如果 2023 年 5 月之后该耕地 (编号 17、18、19) 没有种植豆类作物, 则在 2025 年 9 月至 2026 年 4 月期间必须种植豆类作物。因此, 最极限的决策步长为 3。

表 12 普通大棚的 k 与时间的映射

第 k 次决策	两季时间范围	涵盖的年份
k=1	2023 年 5 月 2024 年 4 月	2023,2024
k=2	2024 年 5 月 2025 年 4 月	2024,2025
k=3	2025 年 5 月 2026 年 4 月	2025,2026

第二层优化模型汇总

超出部分导致滞销和浪费的情况:

$$\begin{aligned}
 & \max \left(\underbrace{\sum_{i \in [1,40] \cup [81,86]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right)}_{\text{第一季种植当年收获}} - \sum_{j=1}^{15} r_j^{(k)} \right) \\
 & + \underbrace{\left(\sum_{i \in [41,80] \cup [87,92]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right) - \sum_{j=1}^{15} r_j^{(k+1)} \right)}_{\text{第二季种植来年收获}} \\
 & s.t. \begin{cases} r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_{i+40}^{(k)}, i = 1, 2, 3, \dots, 40 \\ x_{i+40}^{(k-1)} \neq x_i^{(k)}, i = 1, 2, 3, \dots, 40 \\ \exists x_i^{(Step)}, x_{i+20}^{(Step)}, x_{i+40}^{(Step)}, x_{i+60}^{(Step)} \in \{17, 18, 19\}, Step = 1, 2, 3 \end{cases}
 \end{aligned} \quad (24)$$

超出部分按 2023 年售价的半价进行处理的情况:

$$\begin{aligned}
& \max \left(\underbrace{\sum_{i \in [1,40] \cup [81,86]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right)}_{\text{第一季种植当年收获}} - \sum_{j=1}^{15} r_j^{(k)} \right) \\
& + \underbrace{\left(\sum_{i \in [41,80] \cup [87,92]} \left(A_i \cdot P_{T_i, x_i^{(k)}} \cdot E_{T_i, x_i^{(k)}} - C_{T_i, x_i^{(k)}} \times A_i \right) - \sum_{j=1}^{15} r_j^{(k+1)} \right)}_{\text{第二季种植来年收获}} \\
& \quad \quad \quad (25)
\end{aligned}$$

$$s.t. \begin{cases} r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} - R_j \right) \cdot E_{T_i, x_i^{(k)}}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_{i+40}^{(k)}, i = 1, 2, 3, \dots, 40 \\ x_{i+40}^{(k-1)} \neq x_i^{(k)}, i = 1, 2, 3, \dots, 40 \\ \exists x_i^{(Step)}, x_{i+20}^{(Step)}, x_{i+40}^{(Step)}, x_{i+60}^{(Step)} \in \{17, 18, 19\}, Step = 1, 2, 3 \end{cases}$$

5.6 基于遗传算法求解双层优化模型

遗传算法 (Genetic Algorithm, GA) 是一种基于自然选择和遗传学原理的优化算法。它通过模拟生物进化过程中的选择、交叉、变异等机制, 逐步进化得到问题的最优解。遗传算法是一种全局搜索算法, 适用于复杂、多维和非线性问题的优化。

5.6.1 染色体交叉变异选择方式

单点交叉 (Single Point Crossover), 两点交叉 (Two Point Crossover), 均匀交叉 (Uniform Crossover) 适用于整数编码且不会破坏整数解的特性。

Mutbga 是 (Breeder Genetic Algorithm) 中的一种变异算子。该变异算子通过自适应调整变异幅度, 平衡了全局搜索和局部优化的需求, 防止算法陷入局部最优。

锦标赛选择 (Tournament Selection) 适合整数编码, 通过竞争选择优质个体, 适应度高的个体有更高的几率进入下一代, 避免了轮盘赌选择中由于适应度差异过大而导致的早熟收敛问题。

5.6.2 增强精英保留的遗传算法步骤及参数选择

Step 1: 根据编码规则初始化 N 个个体的种群。

Step 2: 若满足停止条件则停止, 否则继续执行。

Step 3: 对当前种群进行统计分析, 比如记录其最优个体、平均适应度等等。

Step 4: 独立地从当前种群中选取 N 个母体。

Step 5: 独立地对这 N 个母体进行交叉操作。

Step 6: 独立地对这 N 个交叉后的个体进行变异。

Step 7: 将父代种群和交叉变异得到的种群进行合并, 得到规模为 2N 的种群。

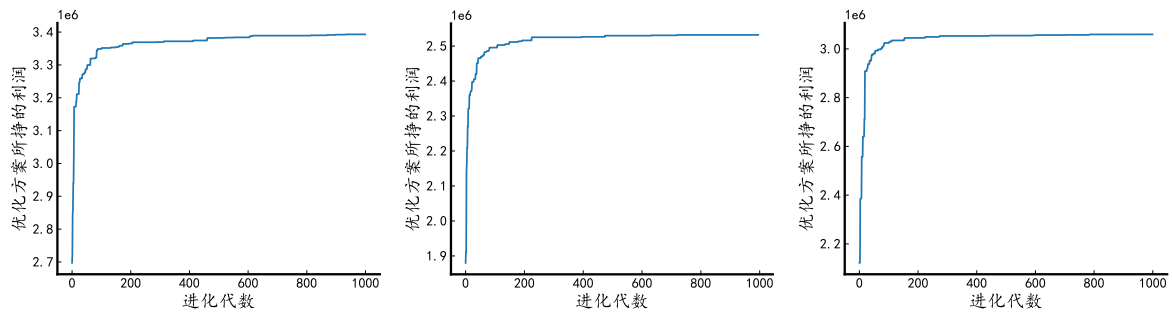
Step 8: 从合并的种群中根据选择算法选择出 N 个个体, 得到新一代种群。

Step 9: 回到 Step 2。

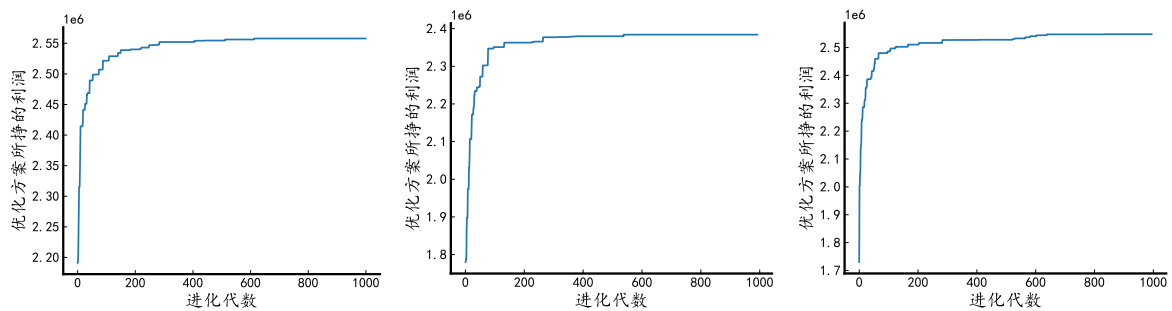
表 13 遗传算法参数选择

参数名称	参数取值
编码方式 (Encoding)	RI
交叉算子 (recOper)	两点交叉
变异算子 (mutOper)	Mutbga
选择算子 (selFunc)	锦标赛选择算子
种群规模 (NIND)	100
最大进化代数 (MAXGEN)	1000
变异概率 (mutOper.Pm)	0.8

5.6.3 求解结果



(滞销)2024 年决策单季种植地收敛图 (滞销)2025 年决策单季种植地收敛图 (滞销)2030 年决策单季种植地收敛图



(降售)2024 年决策单季种植地收敛图 (降售)2025 年决策单季种植地收敛图 (降售)2030 年决策单季种植地收敛图

图 1 遗传算法的损失收敛图

表 14 滞销方案计算出的 54 块地种植总利润

	2024	2025	2026	2027	2028	2029	2030
A1-C6,D7,D8	2469346.25	2408081.25	2463386.25	2508261.25	2490591.25	2492236.25	2492318.75
D1-D6,E1-F4	3392997.00	2531985.50	2913613.50	3104709.00	3027475.00	3342110.50	3059528.00
全部 54 块地	5862343.25	4940066.75	5376999.75	5612970.25	5518066.25	5834346.75	5551846.75

其中 A1-C6,D7,D8 为 28 块单季种植耕地对应的每年的种植利润,D1-D6 与 E1-F4 为 26 块双季种植耕地对应的每年的种植利润,A1-F4 为 54 块地每年种植的利润。对应的每年每季所有种植耕地的种植方案放于附件之中。

表 15 降价出售方案计算出的 54 块地种植总利润

	2024	2025	2026	2027	2028	2029	2030
A1-C6,D7,D8	2676340.00	2502494.38	2674230.63	2669833.75	2501781.88	2668515.00	2666401.25
D1-D6,E1-F4	5185547.00	3692315.00	4999289.00	5014715.00	3941165.00	4603226.00	4930577.00
全部 54 块地	7861887.00	6194809.38	7673519.63	7684548.75	6442946.88	7271741.00	7596978.25

六、 问题二模型的建立与求解

6.1 问题二的分析

针对农作物在生产过程中不可避免的自然波动和市场风险,我们选取了适当的模型和概率分布来模拟和预测农作物的销售价格。对于农作物的种植成本,蔬菜类作物的销售价格与食用菌的销售价格平均每年稳定增加或减少,我们应用**线性增长模型**计算未来的值。而对于农作物的亩产量,其他农作物的预期销售量,价格的波动范围为对称区间的,我们使用**正态分布**来量化不确定性。对于小麦和玉米的预期销售量和食用菌的销售价格波动范围在单侧区间的,我们采取**对数正态分布**来量化不确定性,基于上述分布函数计算出不同作物的 2024-2030 年的各种指标的变化结果进行修正问题一建立的双层优化模型,基于遗传算法求解得到最终结果。

6.2 计算稳增长的农作物预期种植成本和蔬菜、羊肚菌的预期销售价格

因受市场条件影响,农作物的种植成本平均每年增长 5% 左右,因此简单取线性增长趋势即可。

$$C^{(k)} = 1.05 \cdot C^{(k-1)} \quad (26)$$

计算结果如表16所示,完整数据放于附件之中。

表 16 农作物的预期种植成本 (元/亩)

作物名称	地块类型	种植季次	2023	2024	2025	2026	2027	2028	2029	2030
黄豆	平旱地	单季	400	420	441	463.05	486.2	510.51	536.04	562.84
黑豆	平旱地	单季	400	420	441	463.05	486.2	510.51	536.04	562.84
红豆	平旱地	单季	350	367.5	385.88	405.17	425.43	446.7	469.04	492.49
绿豆	平旱地	单季	350	367.5	385.88	405.17	425.43	446.7	469.04	492.49
爬豆	平旱地	单季	350	367.5	385.88	405.17	425.43	446.7	469.04	492.49

蔬菜类作物的销售价格有增长的趋势,平均每年增长 5% 左右:

$$E^{(k)} = 1.05 \cdot E^{(k-1)} \quad (27)$$

计算结果如表17所示,完整数据放于附件之中。

表 17 蔬菜的预期销售价格 (元/斤)

作物名称	2023 年	2024 年	2025 年	2026 年	2027 年	2028 年	2029 年	2030 年
刀豆	6.75	7.09	7.44	7.81	8.2	8.61	9.04	9.49
芸豆	6.5	6.82	7.16	7.52	7.9	8.3	8.72	9.16
土豆	3.75	3.94	4.14	4.35	4.57	4.8	5.04	5.29
西红柿	6.25	6.56	6.89	7.23	7.59	7.97	8.37	8.79

羊肚菌的销售价格每年下降幅度为 5%：

$$E^{(k)} = 0.95 \cdot E^{(k-1)} \quad (28)$$

其中 $E^{(k)}$ 为第 k 年羊肚菌的销售价格。计算结果如表18所示, 完整数据放于附件之中。

表 18 羊肚菌的预期销售价格 (元/斤)

作物名称	2023 年	2024 年	2025 年	2026 年	2027 年	2028 年	2029 年	2030 年
羊肚菌	100	95	90.25	85.74	81.45	77.38	73.51	69.83

6.3 基于正态分布估计其他农作物的预期销售量和农作物的亩产量

假设其他农作物预期销售量的增长率和农作物的亩产量服从正态分布：

$$rate^{(k)} \sim N(\mu, (\sigma)^2) \quad (29)$$

其他农作物未来每年的预期销售量相对于 2023 年大约有 $\pm 5\%$ 的变化增长率, 因此我们认为增长率满足均值为 0, 标准差为 0.05 的正态分布：

$$\mu = 0, \sigma = 0.05 \quad (30)$$

根据 $k - 1$ 年的预期销售量和第 k 年的增长率, 可以计算出第 k 年的预期销售量：

$$Y^{(k)} = Y^{(k-1)} \cdot (1 + rate^{(k)}), rate^{(k)} \sim N(0, (0.05)^2) \quad (31)$$

部分计算结果如表19所示, 完整数据放于附件之中。

表 19 基于正态分布估计的部分农作物的预期销售量 (斤)

作物名称	2023 年	2024 年	2025 年	2026 年	2027 年	2028 年	2029 年	2030 年
刀豆	26880	27033.86	28172.56	29159.38	31276.55	32847.71	33485.37	35510.81
包菜	5280	5360.98	5159.53	5307.35	4735.69	4638.26	4173.35	4316.66
南瓜	35100	36351.34	38374.69	38214.1	40810.07	40825.99	38381.47	36695.16
土豆	30000	28962.75	28970.44	30658.33	30498.73	33164.85	35563.38	35006.65

农作物的亩产量受到气候等因素的影响, 每年大约有 $\pm 10\%$ 的变化率, 因此我们认为增长率满足均值为 0, 标准差为 0.10 的正态分布：

$$\mu = 0, \sigma = 0.10 \quad (32)$$

根据 $k - 1$ 年农作物的亩产量和第 k 年的变化率, 可以计算出第 k 年预期的农作物的亩产量：

$$P^{(k)} = P^{(k-1)} \cdot (1 + rate^{(k)}), rate^{(k)} \sim N(0, (0.10)^2) \quad (33)$$

部分计算结果如表20所示, 完整数据放于附件之中。

表 20 基于正态分布估计的部分农作物的预期亩产量 (斤/亩)

作物名称	地块类型	种植季次	2023	2024	2025	2026	2027	2028	2029	2030
黄豆	平旱地	单季	400	420.1	348.63	306.33	322.53	331.98	328.84	338.3
黑豆	平旱地	单季	500	444.68	442.18	391.11	413.34	362.82	358.17	311.59
红豆	平旱地	单季	400	436.74	501.67	467.57	536.12	494.68	434.4	453.4
绿豆	平旱地	单季	350	337.9	280.06	309.9	292.59	353.71	372.02	360.05
爬豆	平旱地	单季	415	509.86	512.58	437.59	475.42	455.74	502.5	603.9

6.4 基于对数正态分布估计小麦和玉米的预期销售量与食用菌的销售价格

假设小麦和玉米的销售量增长率与食用菌的销售价格下降率服从对数正态分布:

$$\ln(rate^{(k)}) \sim N(\mu, \sigma^2) \tag{34}$$

其中 $rate^{(k)}$ 为第 k 年的增长率, 取 \ln 服从正态分布。

给定小麦和玉米未来的预期销售量的平均年增长率介于 5% 到 10% 之间, 令

$$\begin{cases} l = \ln(a), a = 0.05 \\ u = \ln(b), b = 0.10 \end{cases} \tag{35}$$

基于 95% 的置信度, 计算出增长率服从的对数正态分布的均值和方差:

$$\begin{cases} \mu = \frac{l+u}{2} \\ \sigma = \frac{ul}{2z} \end{cases} \tag{36}$$

其中 $z = \Phi^{-1}(\frac{1+CL}{2})$, $CL = 0.95$ 为置信度, Φ^{-1} 为标准正态分布的逆累积分布函数。根据 t 年的预期销售量和第 t 年的增长率, 可以计算出第 k 年的预期销售量:

$$Y^{(k)} = Y^{(k-1)} \cdot (1 + rate^{(k)}) \tag{37}$$

小麦和玉米的预期销售量计算结果如表21所示:

表 21 基于对数正态分布估计的小麦和玉米的预期销售量 (斤)

作物名称	2023 年	2024 年	2025 年	2026 年	2027 年	2028 年	2029 年	2030 年
小麦	170840	181525.9	195008.7	205769.2	218471.4	236550	259265.4	280449.9
玉米	132750	139854.9	150197.1	160986.7	173532.3	186185.3	199993.2	209616.8

食用菌的销售价格每年下降率介于 1% 到 5% 之间, 令

$$\begin{cases} l = \ln(a), a = 0.01 \\ u = \ln(b), b = 0.05 \end{cases} \tag{38}$$

同理可以计算出第 t 年的预期销售价格:

$$E^{(k)} = E^{(k-1)} \cdot (1 + rate^{(k)}) \tag{39}$$

食用菌 (不含羊肚菌) 的预期销售价格计算结果如表22所示:

表 22 基于对数正态分布估计的食用菌的预期销售价格

作物名称	2023 年	2024 年	2025 年	2026 年	2027 年	2028 年	2029 年	2030 年
榆黄菇	57.5	56.23	54.6	53.76	52.94	51.71	50.79	50.11
香菇	19	18.63	18.06	17.81	17.23	16.98	16.29	15.97
白灵菇	16	15.7	15.43	15.13	14.83	14.65	14.37	14.1

6.5 数据可视化

根据经验, 小麦和玉米未来的预期销售量有增长的趋势, 平均年增长率介于 5% 至 10% 之间, 其他农作物未来每年的预期销售量相对于 2023 年大约有 $\pm 5\%$ 的变化。农作物的亩产量往往会受气候等因素的影响, 每年会有 $\pm 10\%$ 的变化

由蔬菜类作物的销售价格年均增长率在 5% 左右, 蔬菜类作物的售价随着年份的增加也呈现出正增长, 豇豆的预期售价是所有蔬菜中最高的, 白萝卜的预期售价最低, 且低于 4 元,

大多蔬菜的预期售价都集中在 [4,8]。相比于蔬菜的总体增加，食用菌的售价总体呈现下降趋势，羊肚菌的预期售价是所有食用菌当中最高的，也是价格下降最快的，预计在 2030 后售价降低量减少，而榆黄菇、白灵菇和香菇的预测售价的下降趋势较为平缓，其中白灵菇和香菇的预期售价较为相近。

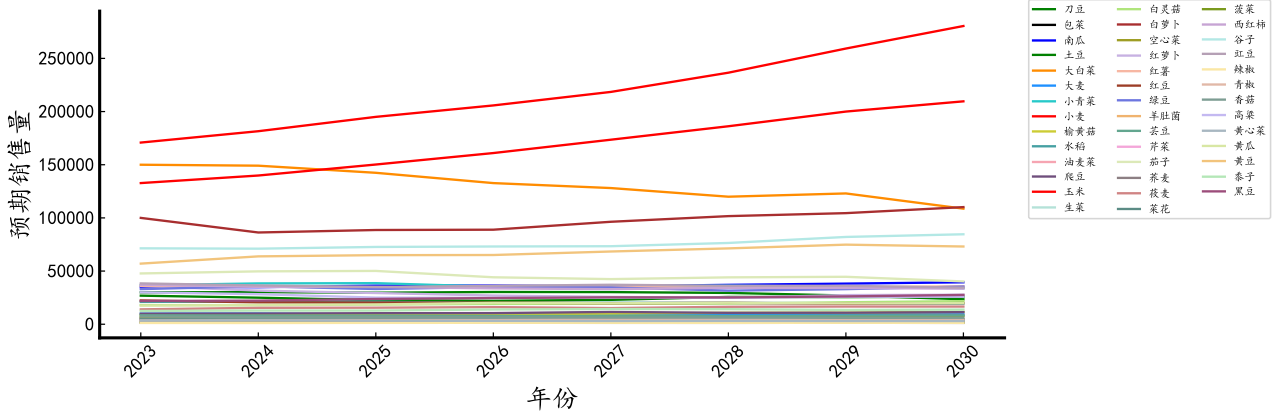


图 2 预测未来的销售量对比图

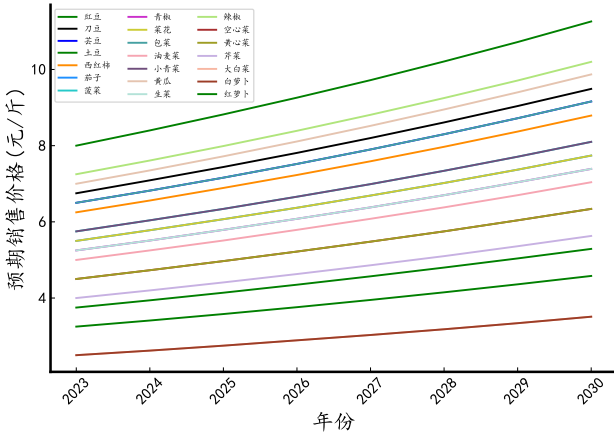


图 3 蔬菜价格趋势图

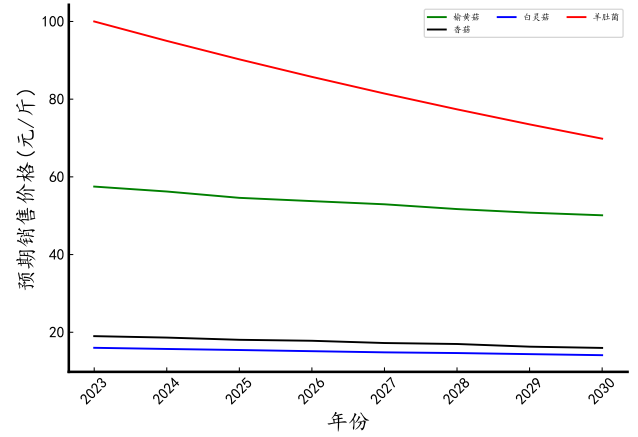


图 4 食用菌价格趋势图

6.6 考虑到不确定性和潜在风险求解农作物最优种植方案优化模型

在此我们只考虑超出部分导致滞销, 造成浪费的情况:

6.6.1 第一层优化模型修正

$$\begin{aligned}
 & \max \left(\sum_{i=1}^{26} \left(\underbrace{A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)}}_{\text{种植该作物的平均售价}} - \underbrace{C_{T_i, x_i^{(k)}}^{(k)} \times A_i}_{\text{种植成本}} \right) - \underbrace{\sum_{j=1}^{15} r_j^{(k)}}_{\text{溢出价格}} \right) \\
 & s.t. \begin{cases} r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} - R_j^{(k)} \right) \cdot E_{T_i, x_i^{(k)}}^{(k)}, & \text{if } \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} > R_j, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_i^{(k+1)}, i = 1, 2, 3, \dots, 26 \\ \exists x_i^{(k)}, x_i^{(k+1)}, x_i^{(k+2)} \in \{1, 2, 3, 4, 5\} \end{cases} \quad (40)
 \end{aligned}$$

在第一问中假定各种农作物未来每一年的预期销售量、种植成本、亩产量和销售价格相对于 2023 年保持稳定, 因此其为常值。在本问中我们采用分布函数量化销售价格的不确定性以及潜在的种植风险, 给出了未来第 k 年的种植成本 $C_{T_i, x_i^{(k)}}^{(k)}$, 取值如表16所示; 未来第 k 年的预期销售量 $R_j^{(k)}$, 取值如表19所示; 未来第 k 年的亩产量 $P_{T_i, x_i^{(k)}}^{(k)}$, 取值如表20所示; 未来第 k 年的销售价格 $E_{T_i, x_i^{(k)}}^{(k)}$, 取值如表22, 表18, 以及原始附件价格未变化的作物。

6.6.2 第二层优化模型修正

$$\begin{aligned}
 & \max \left(\underbrace{\sum_{i \in [1, 40] \cup [81, 86]} \left(A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)} - C_{T_i, x_i^{(k)}}^{(k)} \times A_i \right)}_{\text{第一季种植当年收获}} - \sum_{j=1}^{15} r_j^{(k)} \right) \\
 & + \left(\underbrace{\sum_{i \in [41, 80] \cup [87, 92]} \left(A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)} - C_{T_i, x_i^{(k)}}^{(k)} \times A_i \right)}_{\text{第二季种植来年收获}} - \sum_{j=1}^{15} r_j^{(k+1)} \right) \\
 & s.t. \begin{cases} r_j^{(k)} = \begin{cases} \left(\sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} - R_j^{(k)} \right) \cdot E_{T_i, x_i^{(k)}}^{(k)}, & \sum_{i=1}^{26} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} > R_j^{(k)}, x_i = j \\ 0, & \text{otherwise} \end{cases} \\ x_i^{(k)} \neq x_{i+40}^{(k)}, i = 1, 2, 3, \dots, 40 \\ x_{i+40}^{(k-1)} \neq x_i^{(k)}, i = 1, 2, 3, \dots, 40 \\ \exists x_i^{(Step)}, x_{i+20}^{(Step)}, x_{i+40}^{(Step)}, x_{i+60}^{(Step)} \in \{17, 18, 19\}, Step = 1, 2, 3 \end{cases}
 \end{aligned} \tag{41}$$

6.6.3 基于增强精英保留的遗传算法求解结果

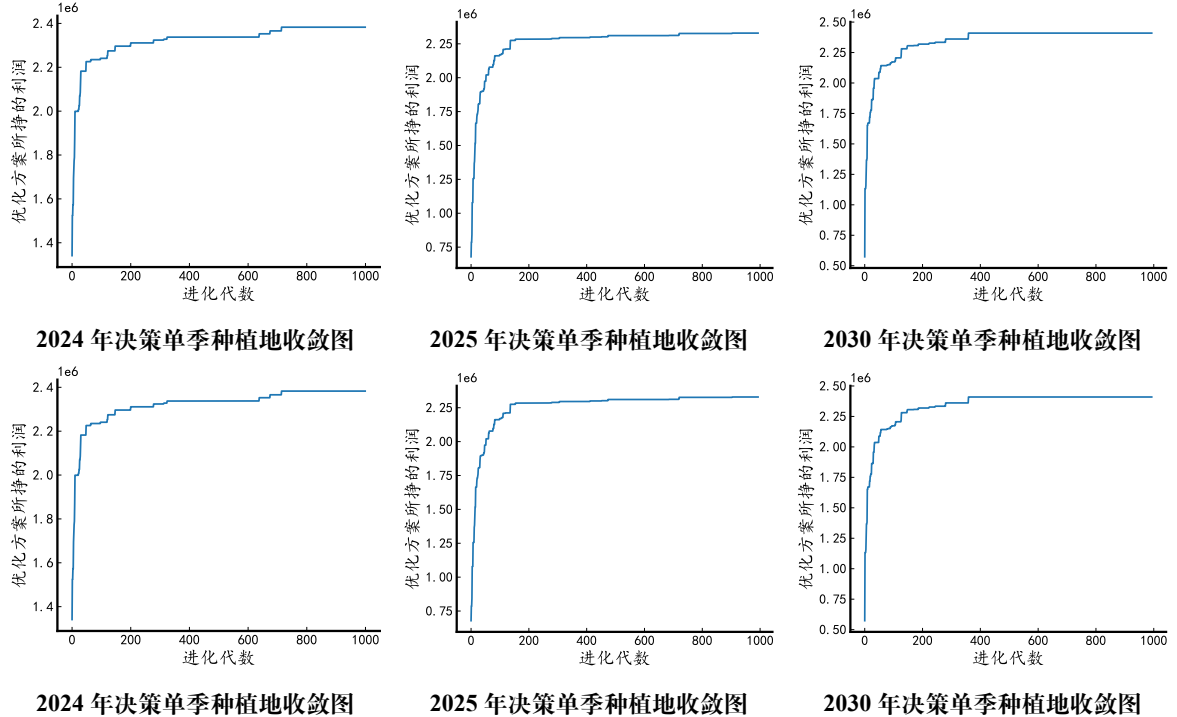


图 5 遗传算法的损失收敛图

表 23 考虑到不确定性和潜在风险计算出的 54 块地种植总利润

	2024	2025	2026	2027	2028	2029	2030
A1-C6,D7,D8	2501250.84	2447533.01	2466730.69	2460657.47	2335763.93	2538046.42	2527964.68
D1-D6,E1-F4	3532822.70	2877154.47	3366004.39	3921362.40	3675967.36	3848177.03	3905738.76
全部 54 块地	6034073.54	5324687.48	5832735.09	6382019.87	6011731.29	6386223.45	6433703.44

如表23所示, 其中 A1-C6,D7,D8 为 28 块单季种植耕地对应的每年的种植利润,D1-D6 与 E1-F4 为 26 块双季种植耕地对应的每年的种植利润,A1-F4 为 54 块地每年种植的利润, 完整决策方案放于附件之中。

七、 问题三模型的建立与求解

7.1 问题三的分析

在现实生活中, 各种农作物之间可能存在一定的可替代性和互补性, 因此我们根据实际经验和查找相关文献确定了哪些农作物之间可以替代, 哪些农作物之间可以互补。对于可以替代的农作物之间可以分担未来预期销售量 R_j 。对于农作物之间可以互补的计算联合互补增长率。预期销售量与销售价格、种植成本之间也存在一定的相关性。因此我们在问题二的基础上, 以销售价格、种植成本为自变量, 预期销售量为因变量, 对五种作物大类进行线性回归拟合以反映出指标之间的相关性。

7.2 基于销售角度的农作物之间的可替代性和互补性

农作物之间的可替代性指的是在市场销售过程中, 某一农作物因供应量减少或价格上涨, 导致另一农作物的销量增加。例如, 黄豆、黑豆、红豆这些粮食豆类都含有丰富的纤维和蛋白质。如果黄豆价格上涨, 消费者可能会转而购买黑豆或红豆, 从而使这两种豆类的销量增加。因此, 粮食豆类之间存在一定的可替代性。对于榆黄菇、香菇、白灵菇和羊肚菌这些食用菌, 虽然它们在营养价值和食用方法上较为相似, 但羊肚菌价格较高, 因此在价格波动时, 消费者可能更倾向于选择其他更便宜的食用菌, 从而体现出羊肚菌的替代性较高。通过分析这些农作物之间的可替代性, 我们可以更好地掌握市场规律, 制定合理的销售策略, 从而获得较高收益。

农作物之间的互补性指的是在市场销售过程中, 某一农作物的销量因另一类农作物的销量增加而增加, 减少而减少。例如, 米饭的销量增加时, 可能会带动蔬菜的销量增加, 因为消费者通常会将主食和蔬菜一起购买。通过挖掘农作物之间的互补性, 有助于制定相应的销售策略, 从而提高整体市场收益。附件给定了粮食 (豆类), 粮食, 蔬菜 (豆类), 蔬菜, 食用

菌, 这五类农作物类, 其编号集合如下:

$$\begin{aligned} G_1 &= \{1, 2, 3, 4, 5\}, \\ G_2 &= \{6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}, \\ G_3 &= \{17, 18, 19\}, \\ G_4 &= \{20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37\}, \\ G_5 &= \{38, 39, 40, 41\}. \end{aligned} \quad (42)$$

我们认为同一类内的作物具有可替代性, 不同类的作物具有互补性, 在前文中我们是对每一个作物判断总产量是否溢出, 而考虑到作物的互补性, 可以将作物个体考虑溢出转化为集合考虑溢出问题, 令整个集合实际总产量减去整个集合的需求量计算溢出:

$$r^{(k)} = \sum_{z=1}^5 \left(\underbrace{\sum_{x_i^{(k)} \in G_z} A_i \cdot P_{T_i, x_i^{(k)}}}_{\text{种植 } G_z \text{ 集合作物的总量}} - \underbrace{\sum_{j \in G_k} R_j^{(k)}}_{\text{集合作物总需求量}} \right) \cdot E_{G_z} \quad (43)$$

其中 E_{G_z} 是每大类作物的平均销售单价。

对于互补性, 我们基于问题二建立的每个作物独立增长率, 使用连乘量化作物之间/类别之间的联合互补增长率:

$$\begin{cases} R_1^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} \geq 0 \\ R_2^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} < 0 \end{cases} \quad (44)$$

其中 $rate_{1-5}^{(k)}$ 为五个农作物类内作物的平均增长率。 R_1 为正向互补增长率, R_2 为负向互补增长率。以预期销售量的增长率来说, 假设 $rate_1^{(k)} = 0.05, rate_2^{(k)} = -0.05, rate_3^{(k)} = 0.05, rate_4^{(k)} = -0.05, rate_5^{(k)} = -0.05$, 因此可以计算出正向互补增长率 $R_1^{(k)} = (1 + R_1^{(k)}) \cdot (1 + R_3^{(k)}) = 1.05 \cdot 1.05 = 1.1025$ 作为 G_1, G_3 新的增长率。负向互补增长率 $R_2^{(k)} = (1 + R_2^{(k)}) \cdot (1 + R_4^{(k)}) \cdot (1 + R_5^{(k)}) = 0.95 \cdot 0.95 \cdot 0.95 = 0.857$ 作为 G_2, G_4, G_5 新的增长率。

7.3 基于线性回归拟合预期销售量与销售价格和种植成本之间的相关性

本文采取线性回归以预期销售量为因变量, 销售价格和种植成本为自变量, 分别对蔬菜(豆类), 粮食, 蔬菜, 食用菌, 粮食(豆类)五种作物类型进行拟合, 通过模型和系数显著性检验的有以下三个:

$$\begin{cases} f_1 = 1533.47x_1 + 3.96x_2, \\ f_2 = 1338.02x_1 + 4.19x_2, \\ f_3 = 92.11x_1 + 0.77x_2. \end{cases} \quad (45)$$

其中 x_1, x_2 分别表示销售价格和种植成本, f_1, f_2, f_3 分别代表蔬菜(豆类), 蔬菜, 食用菌的预期销售量。

由表24可知, $p=1.31e^{-7} < 0.05$, 说明线性回归模型在 0.05 的置信度下整体显著。

由表25可知销售价格的系数为 1533.47, 且 P 值为 $0.001 < 0.05$, 表明其在给定置信度下对因变量的具有显著影响; 种植成本的系数为 3.96, P 值为 $0.033 < 0.05$, 说明在给定置信度下变量显著。

表 24 蔬菜（豆类）线性回归模型显著性等指标结果表

指标	值	指标	值
Rsquared:	0.763	LogLikelihood:	260.70
Adj. Rsquared:	0.742	AIC:	525.4
Fstatistic:	35.46	BIC:	527.7
Prob (Fstatistic):	1.31e-7		

表 25 蔬菜（豆类）线性回归模型系数显著性检验结果表

	coef	std err	t	P> t	[0.025	0.975]
销售价格	1533.4718	388.033	3.952	0.001	728.742	2338.202
种植成本	3.9649	1.746	2.271	0.033	0.345	7.585

粮食类作物的销售价格基本稳定, 可以看成常值, 因此难以量化粮食类作物的销售量与销售价格与种植成本之间的相关性, 如表26所示销售价格在 0.05 的置信度下不显著性, 难以量化这种相关性, 因此对于粮食作物依旧独立计算销售量, 销售价格, 种植成本。

表 26 粮食线性回归模型系数显著性检验结果表

	coef	std err	t	P> t	[0.025	0.975]
销售价格	949.9714	1356.085	0.701	0.485	-1745.836	3645.779
种植成本	34.5491	20.333	1.699	0.093	-5.871	74.970

7.3.1 第一层优化模型修正

$$\begin{aligned}
 & \max \left(\sum_{i=1}^{26} \left(\underbrace{A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)}}_{\text{种植该作物的平均售价}} - \underbrace{C_{T_i, x_i^{(k)}}^{(k)} \times A_i}_{\text{种植成本}} \right) - \underbrace{\sum_{j=1}^{15} r_j^{(k)}}_{\text{溢出价格}} \right) \\
 & s.t. \begin{cases} r^{(k)} = \sum_{z=1}^5 \left(\underbrace{\sum_{x_i^{(k)} \in G_z} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)}}_{\text{种植 } G_z \text{ 集合作物的总量}} - \underbrace{\sum_{j \in G_k} R_j^{(k)}}_{\text{集合作物总需求量}} \right) \cdot E_{G_z} \\ \begin{cases} R_1^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} \geq 0 \\ R_2^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} < 0 \end{cases} \\ Y_i^{(k)} = f_1(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_3 \\ Y_i^{(k)} = f_2(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_4 \\ Y_i^{(k)} = f_3(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_5 \\ x_i^{(k)} \neq x_i^{(k+1)}, i = 1, 2, 3, \dots, 26 \\ \exists x_i^{(k)}, x_i^{(k+1)}, x_i^{(k+2)} \in \{1, 2, 3, 4, 5\} \end{cases} \quad (46)
 \end{aligned}$$

在第一问中假定各种农作物未来每一年的预期销售量、种植成本、亩产量和销售价格相对于 2023 年保持稳定, 因此其为常值。在本问中我们采用分布函数量化销售价格的不确定性

以及潜在的种植风险，给出了未来第 k 年的种植成本 $C_{T_i, x_i^{(k)}}^{(k)}$ ，取值如表16所示；未来第 k 年的预期销售量 $R_j^{(k)}$ ，取值如表19所示；未来第 k 年的亩产量 $P_{T_i, x_i^{(k)}}^{(k)}$ ，取值如表20所示；未来第 k 年的销售价格 $E_{T_i, x_i^{(k)}}^{(k)}$ ，取值如表22, 表18, 以及原始附件价格未变化的作物。

7.3.2 第二层优化模型修正

$$\begin{aligned}
 & \max \left(\underbrace{\sum_{i \in [1,40] \cup [81,86]} \left(A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)} - C_{T_i, x_i^{(k)}}^{(k)} \times A_i \right) - r^{(k)}}_{\text{第一季种植当年收获}} \right. \\
 & \quad \left. + \underbrace{\sum_{i \in [41,80] \cup [87,92]} \left(A_i \cdot P_{T_i, x_i^{(k)}}^{(k)} \cdot E_{T_i, x_i^{(k)}}^{(k)} - C_{T_i, x_i^{(k)}}^{(k)} \times A_i \right) - r^{(k+1)}}_{\text{第二季种植来年收获}} \right) \\
 & s.t. \begin{cases} r^{(k)} = \sum_{z=1}^5 \left(\underbrace{\sum_{x_i^{(k)} \in G_z} A_i \cdot P_{T_i, x_i^{(k)}}^{(k)}}_{\text{种植 } G_z \text{ 集合作物的总量}} - \underbrace{\sum_{j \in G_k} R_j^{(k)}}_{\text{集合作物总需求量}} \right) \cdot E_{G_z} \\ \begin{cases} R_1^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} \geq 0 \\ R_2^{(k)} = \prod_z (1 + rate_z^{(k)}), & rate_z^{(k)} < 0 \end{cases} \\ Y_i^{(k)} = f_1(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_3 \\ Y_i^{(k)} = f_2(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_4 \\ Y_i^{(k)} = f_3(E_{T_i, x_i^{(k)}}^{(k)}, C_{T_i, x_i^{(k)}}^{(k)}), i \in G_5 \\ x_i^{(k)} \neq x_{i+40}^{(k)}, i = 1, 2, 3, \dots, 40 \\ x_{i+40}^{(k-1)} \neq x_i^{(k)}, i = 1, 2, 3, \dots, 40 \\ \exists x_i^{(Step)}, x_{i+20}^{(Step)}, x_{i+40}^{(Step)}, x_{i+60}^{(Step)} \in \{17, 18, 19\}, Step = 1, 2, 3 \end{cases} \quad (47)
 \end{aligned}$$

7.3.3 基于增强精英保留的遗传算法求解结果

表 27 滞销方案计算出的 54 块地种植总利润

	2024	2025	2026	2027	2028	2029	2030
A1-C6,D7,D8	4583691.32	4218084.00	5521759.51	5626296.92	5266084.01	5448528.12	6466631.37
D1-D6,E1-F4	3365668.54	3139631.24	3337297.27	3765925.42	3497833.12	4220232.34	3844626.43
全部 54 块地	7949359.86	7357715.24	8859056.78	9392222.34	8763917.13	9668760.45	10311257.81

其中 A1-C6,D7,D8 为 28 块单季种植耕地对应的每年的种植利润,D1-D6 与 E1-F4 为 26 块双季种植耕地对应的每年的种植利润,A1-F4 为 54 块地每年种植的利润。

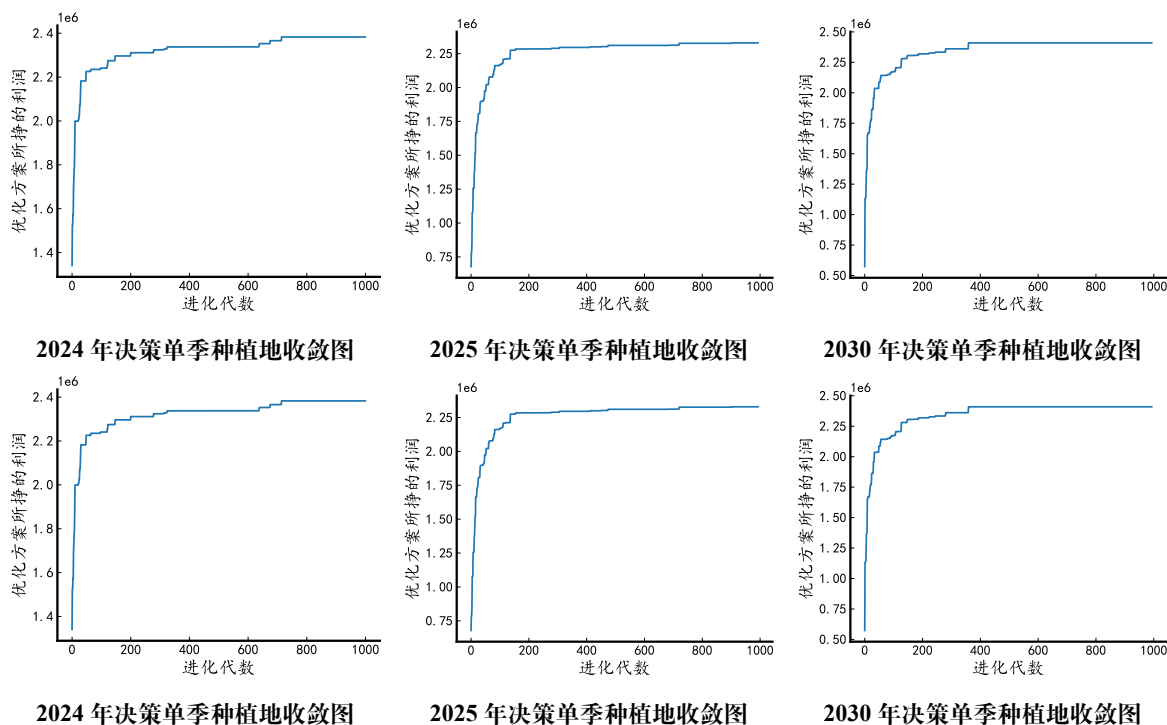


图6 遗传算法的损失收敛图

八、模型的评价与推广

8.1 模型的优点

- (1) 针对问题一，我们选取双层优化模型同时考虑单季耕地和双季耕地的利益最大化，上下层决策可以合理处理不同的目的和约束条件，从而更全面的反映真实情况。
- (2) 采用增强精英保留的遗传算法，可以有效地搜索解决方案空间，提高最终种植决策的质量和效率。
- (3) 针对问题二，选取适当的概率分布（正态分布和对数正态分布）来模拟自然波动和市场风险，使得优化结果更具有真实性
- (4) 针对问题三，通过线性回归建立相关性，有助于理解价格、成本和销售量之间的关系，增强模型的解释力与预测能力。

8.2 模型的缺点

- (1) 我们模型假设未来销售量和种植情况基于过去数据，但具体的市场和环境变化可能影响最终结果的质量。
- (2) 先求解单季耕地再求解双季耕地的顺序可能限制了模型的整体优化效果，特别是在两层决策之间的交互影响未能完全考虑，可能会导致预测的不准确性。
- (3) 概率分布的参数需要准确估计，由于具体的数据量较少，参数估计的质量不一定很好，可能会影响最终结果的可靠性。

参考文献

- [1] 姜启源. 谢金星. 叶俊. 《数学模型（第五版）》[M], 北京：高等教育出版社，2003.

- [2] 司守奎, 孙兆亮. 数学建模算法与应用 [M]. 北京: 国防工业出版社, 2017.
- [3] 葛继科, 邱玉辉, 吴春明, 等. 遗传算法研究综述 [J]. 计算机应用研究, 2008, (10): 2911-2916.
- [4] 黄超. 对数正态分布的参数估计 [J]. 高等数学研究, 2015, 18(4): 4, 20.
DOI:10.3969/j.issn.10081399.2015.04.002.

附录

附录清单

代码清单

文件名称	文件路径
第一问模型 (浪费 _ 决策单季种植地).ipynb	支撑文件\
第一问模型 (浪费 _ 决策双季种植地).ipynb	支撑文件\
第一问模型 (降价出售 _ 决策单季种植地).ipynb	支撑文件\
第一问模型 (降价出售 _ 决策双季种植地).ipynb	支撑文件\
第二问模型 (数据预处理).ipynb	支撑文件\
第二问模型 (决策单季种植地).ipynb	支撑文件\
第二问模型 (决策双季种植地).ipynb	支撑文件\
第三问模型 (数据预处理).ipynb	支撑文件\
第三问模型 (预期销售量与价格, 成本关系).ipynb	支撑文件\
第三问模型 (决策单季种植地).ipynb	支撑文件\
第三问模型 (决策双季种植地).ipynb	支撑文件\

第一问求解结果清单

文件名称	文件路径
总作物的预计销售量.xlsx	支撑文件\第一问求解结果\
浪费 _ 单季地种植的不同作物数量 (斤).xlsx	支撑文件\第一问求解结果\
浪费 _ 54 块地种植总利润.xlsx	支撑文件\第一问求解结果\
浪费 _ 28 个单季种植地所挣利润.xlsx	支撑文件\第一问求解结果\
浪费 _ 26 个双季种植地所挣利润.xlsx	支撑文件\第一问求解结果\
降价出售 _ 单季地种植的不同作物数量 (斤).xlsx	支撑文件\第一问求解结果\
降价出售 _ 54 块地种植总利润.xlsx	支撑文件\第一问求解结果\
降价出售 _ 28 个单季种植地所挣利润.xlsx	支撑文件\第一问求解结果\
降价出售 _ 26 个双季种植地所挣利润.xlsx	支撑文件\第一问求解结果\
第三问模型 (决策单季种植地).ipynb	支撑文件\第一问求解结果\
第三问模型 (决策双季种植地).ipynb	支撑文件\第一问求解结果\
result_1_2.xlsx	支撑文件\第一问求解结果\
result_1_1.xlsx	支撑文件\第一问求解结果\

第二问求解结果清单

文件名称	文件路径
26 个双季种植地所挣利润.xlsx	支撑文件\第二问求解结果\
28 个单季种植地所挣利润.xlsx	支撑文件\第二问求解结果\
54 块地种植总利润.xlsx	支撑文件\第二问求解结果\
单季地种植的不同作物数量 (斤).xlsx	支撑文件\第二问求解结果\
浪费 _ 26 个双季种植地所挣利润.xlsx	支撑文件\第二问求解结果\
食用菌价格趋势图.pdf	支撑文件\第二问求解结果\
蔬菜价格趋势图.pdf	支撑文件\第二问求解结果\
预测未来的亩产量.xlsx	支撑文件\第二问求解结果\
预测未来的销售价格.xlsx	支撑文件\第二问求解结果\
预测未来的销售量对比图.pdf	支撑文件\第二问求解结果\
预测未来的种植成本.xlsx	支撑文件\第二问求解结果\
result2.xlsx	支撑文件\第二问求解结果\

第三问求解结果清单

文件名称	文件路径
26 个双季种植地所挣利润.xlsx	支撑文件\第三问求解结果\
28 个单季种植地所挣利润.xlsx	支撑文件\第三问求解结果\
54 块地种植总利润.xlsx	支撑文件\第三问求解结果\
result3_ 单季_.xlsx	支撑文件\第三问求解结果\
result3_ 双季.xlsx	支撑文件\第三问求解结果\
result3_ 水浇地.xlsx	支撑文件\第三问求解结果\
单季地种植的不同作物数量 (斤).xlsx	支撑文件\第三问求解结果\
预测未来的亩产量.xlsx	支撑文件\第三问求解结果\
预测未来的销售价格.xlsx	支撑文件\第三问求解结果\
预测未来的销售量对比图.pdf	支撑文件\第三问求解结果\
预测未来的种植成本.xlsx	支撑文件\第三问求解结果\

问题一代码

```

import pandas as pd
import numpy as np
np.set_printoptions(precision=2, suppress=True)
import matplotlib.pyplot as plt
from matplotlib import rcParams
import matplotlib.pyplot as plt

config = {
    "font.family": 'serif',
    "font.size": 12,
    "mathtext.fontset": 'stix',
    "font.serif": ["KaiTi"],
    "axes.unicode_minus": False,
    "xtick.direction": 'in',
    "ytick.direction": 'in',
}
rcParams.update(config)

import random
import time
import networkx as nx
import geatpy as ea
# 单季编码
dict1={"平旱地":["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", "苡麦", "大麦"],
    "梯田":["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", "苡麦", "大麦"],

```

```

    "山坡地":["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"],
}

df_get_type_all = pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
for type_ in ["平旱地", "梯田", "山坡地"]:
    df_get_type = pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
    data_to_append = []
    if type_ in ["平旱地", "梯田", "山坡地"]:
        for i in range(len(dict1[type_])):
            data_to_append.append([type_, i + 1, dict1[type_][i]])
    df_temple=pd.DataFrame(data_to_append, columns=df_get_type_all.columns)
    df_get_type_all = pd.concat([df_get_type_all, df_temple], ignore_index=True)

df1=pd.read_excel("表1.xlsx")
df2=pd.read_excel("表2.xlsx")
df3=pd.read_excel("表3.xlsx")
df4=pd.read_excel("附件2.xlsx")

df4['种植地块'] = df4['种植地块'].fillna(method='ffill')
df4.rename(columns={'种植地块': '地块名称'}, inplace=True)
df2["地块类型"] = df2["地块类型"].str.strip()
df2["作物名称"] = df2["作物名称"].str.strip()
df2[['最低单价', '最高单价']] = df2['销售单价/(元/斤)'].str.split('-', expand=True)
df2['最低单价'] = pd.to_numeric(df2['最低单价'])
df2['最高单价'] = pd.to_numeric(df2['最高单价'])
df2['平均单价'] = (df2['最低单价'] + df2['最高单价']) / 2

df_merged = pd.merge(df4, df1, how='left', on='地块名称')
df_merged.drop("地块面积/亩",axis=1,inplace=True)
df_merged["地块类型"] = df_merged["地块类型"].str.strip()
df_merged["作物名称"] = df_merged["作物名称"].str.strip()
df_final = pd.merge(df_merged, df2[['作物名称', '地块类型', '亩产量/斤']],
                    how='left',
                    on=['作物名称', '地块类型'])
df_final['总产量/斤'] = df_final['种植面积/亩'] * df_final['亩产量/斤']
summary = df_final.groupby('作物名称').agg({
    '总产量/斤': 'sum'
}).reset_index()

#15是平旱地/梯田/山坡地可种植数量
# 1-26 对应的值表示A1-C6(单季地可选择种植的作物数量) 1-6是平旱地A 7-20是梯田B 21-26是山坡地C
#确定编码上界
upper=[]
#地的类型
type_list=[]
a=max( df_get_type_all[df_get_type_all["地块类型"]=="平旱地"]["编码取值"] )
b=max( df_get_type_all[df_get_type_all["地块类型"]=="梯田"]["编码取值"] )

```

```

c=max( df_get_type_all[df_get_type_all["地块类型"]=="山坡地"]["编码取值"] )
# 确定单季作物
for index in range(df1.shape[0]):
    #地块类型
    type_=( df1.loc[index]["地块类型"] )
    type_ = type_.replace(" ", "")
    type_list.append(type_)
    if type_ == "平旱地":
        #地块可种植的作物数量
        upper.append( a )
    elif type_ == "梯田":
        upper.append( b )
    elif type_ == "山坡地":
        upper.append( c )

dikuai_type_list=list( df1["地块类型"] )
area_list=list( df1["地块面积/亩"] )
cost_list=[list( df2[df2["地块类型"]=="平旱地"]["种植成本/(元/亩)"] ),
            list( df2[df2["地块类型"]=="梯田"]["种植成本/(元/亩)"] ),
            list( df2[df2["地块类型"]=="山坡地"]["种植成本/(元/亩)"] ),
            ]
type_list=[list( df2[df2["地块类型"]=="平旱地"]["作物名称"] ),
            list( df2[df2["地块类型"]=="梯田"]["作物名称"] ),
            list( df2[df2["地块类型"]=="山坡地"]["作物名称"] )]

product_list= [list( df2[df2["地块类型"]=="平旱地"]["亩产量/斤"] ),
                list( df2[df2["地块类型"]=="梯田"]["亩产量/斤"] ),
                list( df2[df2["地块类型"]=="山坡地"]["亩产量/斤"] )]

earn_per_list= [list( df2[df2["地块类型"]=="平旱地"]["平均单价"] ),
                 list( df2[df2["地块类型"]=="梯田"]["平均单价"] ),
                 list( df2[df2["地块类型"]=="山坡地"]["平均单价"] )]

require=summary
require_dict = require.set_index('作物名称')['总产量/斤'].to_dict()
crop_price_dict = df2.set_index('作物名称')['平均单价'].to_dict()
flattened_list = [item for sublist in type_list for item in sublist]

# 计算单季总利润
def cal_earn(x):
    cost_all_list=[]
    earn_average_list=[]
    # 种植的作物数量(斤)
    product_zuowu_type_account={i:0 for i in set( flattened_list )}
    for i in range(len(x)):
        #该地类型(平旱地/梯田/山坡地)
        dikuai_type=dikuai_type_list[i]
        #该地面积(亩)

```



```

area=area_list[i]
j=None
if dikuai_type == "平旱地":
    j=0
elif dikuai_type == "梯田":
    j=1
elif dikuai_type == "山坡地":
    j=2

#该编码表示的种植的作物
zuowu_type=type_list[j][x[i]-1]
#该作物种植成本(元/亩)
cost=cost_list[j][x[i]-1]
#该地总种植成本(元)
cost_all=cost*area
cost_all_list.append( cost_all )

#该地亩产量
product=product_list[j][x[i]-1]
#该地总产量 (斤)
product_all=product*area
product_zuowu_type_account[zuowu_type]=product_zuowu_type_account[zuowu_type]+product_all
#平均单价 (元/斤)
average_per=earn_per_list[j][x[i]-1]
#该地总平均售价 (元)
earn_average_list.append( product_all*average_per )

a=np.sum( np.array( earn_average_list ) -np.array( cost_all_list ) )
for key, value in product_zuowu_type_account.items():
    r=require_dict[key]
    exceed=value -r
    danjia=crop_price_dict[key]
    if exceed >0:
        a=a-exceed*danjia*0.5

return a,product_zuowu_type_account

# 计算双季
# 确认双季作物
dict1={"普通大棚第一季":["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"],
    "智慧大棚第一季":["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"],
    "智慧大棚第二季":["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"],
    "普通大棚第二季":["榆黄菇", "香菇", "白灵菇", "羊肚菌"],
    "水浇地第一季":["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"],
        "水浇地第二季":["大白菜", "白萝卜", "红萝卜"]
    }
}

# 确定一二季的编码
# 第一季的编码组合
df_get_type_all = pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
for type_ in ["普通大棚第一季", "智慧大棚第一季", "水浇地第一季"]:
    df_get_type = pd.DataFrame(columns=["地块类型", "编码取值", "第一季种植作物"])
    # 暂存要追加的数据
    data_to_append = []
    for i in range(len(dict1[type_])):
        # 将数据添加到列表中
        data_to_append.append([type_, i + 1, dict1[type_][i]])

    # 将数据列表转换为 DataFrame, 并追加到 df_get_type
    df_temple=pd.DataFrame(data_to_append, columns=df_get_type_all.columns)
    df_get_type_all = pd.concat([df_get_type_all, df_temple], ignore_index=True)

# 第二季的编码组合
df_get_type_all2= pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
for type_ in ["普通大棚第二季", "智慧大棚第二季", "水浇地第二季"]:
    df_get_type = pd.DataFrame(columns=["地块类型", "编码取值", "第二季种植作物"])
    # 暂存要追加的数据
    data_to_append = []
    for i in range(len(dict1[type_])):
        # 将数据添加到列表中
        data_to_append.append([type_, i + 1, dict1[type_][i]])

    # 将数据列表转换为 DataFrame, 并追加到 df_get_type
    df_temple=pd.DataFrame(data_to_append, columns=df_get_type_all2.columns)
    df_get_type_all2= pd.concat([df_get_type_all2, df_temple], ignore_index=True)

# 返回两季种植
# 返回两季种植的作物数量
def cal_product_zuowu_type_account(x):
    cost_all_list=[]
    earn_average_list=[]
    # 第一季种植的作物数量(斤)
    product_zuowu_type_account1={i:0 for i in set( flattened_list )}
    # 第二季种植的作物数量(斤)
    product_zuowu_type_account2={i:0 for i in set( flattened_list )}

    for i in range(len(x)):
        # 该地类型(平旱地/梯田/山坡地)
        dikuai_type=dikuai_type_list[i]
        # 该地面积(亩)
        area=area_list[i]/2
        tmp_x=x[i]

```

```

j=None
if i < 40 :
    if dikuai_type == "普通大棚":
        j=0
        tmp_x-=17
    elif dikuai_type == "智慧大棚":
        j=1
        tmp_x-=17
else:
    if dikuai_type == "普通大棚":
        j=2
        tmp_x-=38
    elif dikuai_type == "智慧大棚":
        j=3
        tmp_x-=17
if(dikuai_type=='水浇地'):
    j=4
    tmp_x-=35

zuowu_type=vegetables_dict[x[i]]
#该地亩产量
product=product_list[j][tmp_x]
#该地总产量（斤）
product_all=product*area
if i in range(0,40) or i in range(80,86) :
    product_zuowu_type_account1[zuowu_type]=product_zuowu_type_account1[zuowu_type]+product_all
else:
    product_zuowu_type_account2[zuowu_type]=product_zuowu_type_account2[zuowu_type]+product_all

return product_zuowu_type_account1,product_zuowu_type_account2

# 计算两季的利润
def cal_earn(x,t=-1):
    cost_all_list=[]
    earn_average_list=[]
    # 种植的作物数量(斤)
    product_zuowu_type_account={i:0 for i in set( flattened_list )}
    for i in range(len(x)):
        #该地类型(平旱地/梯田/山坡地)
        dikuai_type=dikuai_type_list[i]
        #该地面积(亩)
        area=area_list[i]/2
        tmp_x=x[i]
        j=None
        if i < 40 :
            if dikuai_type == "普通大棚":
                j=0

```

```

        tmp_x-=17
    elif dikuai_type == "智慧大棚":
        j=1
        tmp_x-=17
    else:
        if dikuai_type == "普通大棚":
            j=2
            tmp_x-=38
        elif dikuai_type == "智慧大棚":
            j=3
            tmp_x-=17
    if(dikuai_type=='水浇地'):
        j=4
        tmp_x-=35

    zuowu_type=vegetables_dict[x[i]]
    #zuowu_type=type_list[j][x[i]-1]
    #该作物种植成本(元/亩)
    cost=cost_list[j][tmp_x]
    #该地总种植成本(元)
    cost_all=cost*area
    cost_all_list.append( cost_all )

    #该地亩产量
    product=product_list[j][tmp_x]
    #该地总产量 (斤)
    product_all=product*area
    product_zuowu_type_account[zuowu_type]=product_zuowu_type_account[zuowu_type]+product_all
    #平均单价 (元/斤)
    average_per=earn_per_list[j][tmp_x]
    #该地总平均售价 (元)
    earn_average_list.append( product_all*average_per )

a=np.sum( earn_average_list[0:39]+earn_average_list[80:85] )
b=np.sum( cost_all_list[0:39]+cost_all_list[80:85] )
#has_product为去年种的 今年卖的 即2023年第二季种植的数量 在2024年收获
    2024年需求量需要减去这个种植的数量
if t >= 0:
    _,has_product=cal_product_zuowu_type_account(history[t])
else:
    _,has_product=cal_product_zuowu_type_account(history[-1])
#now_product为今年第一季种植的数量
now_product,_=cal_product_zuowu_type_account(x)

#在考虑到2023年第二季收获的种植作物数量后决策2024第一季的实际需求量
require1={key: require_dict[key] - has_product[key] for key in has_product.keys()}
for key, value in now_product.items():
    r=require1[key]

```

```

        exceed=value - r
        danjia=crop_price_dict[key]
        if exceed >0:
            a=a-exceed*danjia*0.5

#第二季赚的钱
c=np.sum( earn_average_list[40:79]+earn_average_list[86:91] )
#第二季种植成本
d=np.sum( cost_all_list[40:79]+cost_all_list[80:86] )
#第二季单独减去水浇地超标的数量
for key, value in require_dict.items():
    if key in ["大白菜","白萝卜","红萝卜"]:
        r=require_dict[key]
        exceed=value - r
        danjia=crop_price_dict[key]
        if exceed >0:
            c=c-exceed*danjia*0.5

return a-b,c-d

history=[[6, 7, 7, 1, 4, 8, 6, 2, 3, 4, 5, 8, 6, 8, 9, 10, 1, 7, 14, 15, 11, 12, 1, 13, 6, 3]] #
    第一年的种植情况
max_all_earn=[]
all_cost=[]
class MyProblem(ea.Problem):
    def __init__(self):
        name = 'MyProblem'
        Dim=len( upper )
        M = 1
        maxormins = [-1]
        varTypes = [1]*Dim
        ub = upper
        lb = [1] * Dim #编码从1开始 所以下界为1
        ubin = [1] *Dim #(upper) #
        lbin = [1] *Dim #(lower)
        ea.Problem.__init__(self, name, M, maxormins, Dim, varTypes, lb, ub, lbin, ubin)

    def aimFunc(self, pop):
        x = pop.Phen.copy()
        cv1=[]
        cv2=[]
        ObjV = []
        for i in range(pop.sizes):
            #计算约束值一
            same_values_count = sum(1 for j in range(len(x[i])) if x[i][j] == history[-1][j])
            cv1.append( same_values_count *100)

```

```

        #计算约束二:
        if len(history)>=2:
            history_np = np.array(history[-2:] + [x[i]])
            count_not_in_range = np.sum((history_np < 1) | (history_np > 5), axis=0)
            no_values_count=sum([1 for i in count_not_in_range if i==3])
            cv2.append( no_values_count *100)
        else:
            cv2.append( 0 )
            a,b=cal_earn( x[i] )
            ObjV.append(a)
        pop.ObjV = np.array([ObjV]).reshape(pop.sizes,self.M)
        pop.CV = np.hstack([np.array(cv1).reshape(pop.sizes,1) ,
                             np.array(cv2).reshape(pop.sizes,1) ,
                             ])
for times in range(7):
    problem = MyProblem()
    #设置编码方式
    Encoding = 'RI'
    NIND = 100
    Field = ea.crtfld(Encoding, problem.varTypes, problem.ranges, problem.borders)
    population = ea.Population(Encoding, Field, NIND)
    myAlgorithm = ea.soea_SEGA_templet(problem, population)
    myAlgorithm.MAXGEN = 1000
    myAlgorithm.mutOper.Pm = 0.8
    myAlgorithm.logTras = 0
    myAlgorithm.verbose = False
    myAlgorithm.drawing = 0
    #执行算法
    [BestIndi, population] = myAlgorithm.run()
    #保存损失迭代图
    # BestIndi.save()
    plt.figure(dpi=500,figsize=(5, 4))
    plt.plot( myAlgorithm.trace["f_best"] )
    plt.xlabel('进化代数',fontsize=16)
    plt.ylabel('优化方案所挣的利润',fontsize=16)
    ax = plt.gca()
    ax.spines['top'].set_color('none')
    ax.spines['right'].set_color('none')
    plt.rcParams['xtick.direction'] = 'in'
    plt.rcParams['ytick.direction'] = 'in'
    ax.spines["bottom"].set_linewidth(2)
    ax.spines["left"].set_linewidth(2)
    # plt.title('Loss Iteration Graph')
    plt.savefig(f"第一问求解结果/遗传算法收敛图/浪费_{2024+times}年决策单季种植地收敛图.pdf")
    print(f'算法求解次数: {myAlgorithm.evalsNum}') #即最大进化代数和种群规模的乘积
    print(f'算法执行秒数:{ myAlgorithm.passTime}')
    if BestIndi.sizes != 0:
        print(f'最佳目标函数值为: { BestIndi.ObjV}')

```

```

max_all_earn.append( BestIndi.ObjV[0][0])
print(f'最佳决策变量为: {BestIndi.Phen}')
print(f'最佳决策变量的约束矩阵值为: {BestIndi.CV}')
history.append( list( BestIndi.Phen[0] ) )

else:
    print("未找到最优解")
# 结果按照年份写进excel
with pd.ExcelWriter("第一问求解结果/result1_1_双季水稻地_浪费_.xlsx") as writer:
    for year, j in zip(range(2024, 2031), range(1,8)):
        result=pd.DataFrame(0,columns= [
            "地块名称",
            "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", "苡麦", "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"],index=index_list_doll )
        result['地块名称']=index_list_doll
        for i in range(80,92):
            tmp_area=df1[(df1['地块类型']=='水浇地')&(df1['地块名称']==dic1[i])]['地块面积/亩']
            +\.values[0]
            if(i<=85):
                next_season = f'第一季{dic1[i]}'
                #print(vegetables_dict[i])
                result.loc[next_season,vegetables_dict[history[j][i]]]+=tmp_area
            if(i>85):
                next_season = f'第二季{dic1[i]}'
                #print(vegetables_dict[i])
                result.loc[next_season,vegetables_dict[history[j][i]]]+=tmp_area
        result.fillna(0).to_excel(writer, sheet_name=str(year), index=False)

```

问题二代码

```

import pandas as pd
import numpy as np
np.set_printoptions(precision=2, suppress=True)
import matplotlib.pyplot as plt
from matplotlib import rcParams
import matplotlib.pyplot as plt

config = {
    "font.family": 'serif',
    "font.size": 12,
    "mathtext.fontset": 'stix',
    "font.serif": ["KaiTi"],
    "axes.unicode_minus": False,
    "xtick.direction": 'in',
    "ytick.direction": 'in',
}

```

```

rcParams.update(config)

import random
import time
import networkx as nx
import geatpy as ea

expect_result=pd.DataFrame(columns=["作物名称"]+[str(2023+i)+"年预期销售量" for i in range(8)])
expect_result["作物名称"]= expect["作物名称"]
expect_result["2023年预期销售量"]= expect["总产量/斤"]

for year in range(7):
    for i in range(expect_result.shape[0]):
        zuowu = expect_result.loc[i,"作物名称"]
        if zuowu != "小麦" and zuowu != "玉米":
            # 获取该作物当前年份的预期销售量
            mu = 0
            # 计算标准差
            std = 0.05
            # 生成预期增长率（正态分布）
            growth_rate=np.random.normal(loc=mu, scale=std, size=1)
            y_t_pre=expect_result.loc[i,str(2023 + year) + "年预期销售量"]
            expect_result.loc[i,str(2023+1+ year) + "年预期销售量"] = y_t_pre*(1+growth_rate)

        # 判断作物类型
        elif zuowu in ["小麦", "玉米"]:
            # 给定的置信度
            CL = 0.95
            # 计算 z 值
            z = norm.ppf((1 + CL) / 2)
            # 计算 l 和 u
            a = 0.05
            b = 0.10
            l = np.log(a)
            u = np.log(b)
            # 计算对数正态分布的均值和标准差
            mu_ln = (l + u) / 2
            sigma_ln = (u - l) / (2 * z)
            # 获取当前年份的预期销售量
            y_t_pre=expect_result.loc[i,str(2023 + year) + "年预期销售量"]
            # 生成对数正态分布的增长率 r
            growth_rate = np.exp(np.random.normal(loc=mu_ln, scale=sigma_ln))
            # print( 1+growth_rate )
            # 计算未来的预期销售量
            expect_value = np.round(y_t_pre * (1+growth_rate), 2)
            expect_result.loc[i,str(2023 +1+ year) + "年预期销售量"] = expect_value

# 定义绘图函数
def plot_sales_price(df, crop_type_dict, crop_categories, title, colors, y_label):

```



```

# 设置图形大小
plt.figure(dpi=400, figsize=(7, 5))

count = 0
# 遍历每一行作物并绘制折线
for i in range(df.shape[0]):
    label = df.iloc[i, 0]

    # 检查作物是否属于 crop_categories 中的任意类别
    if any(label in crop_type_dict[category] for category in crop_categories):
        # 设置颜色
        color = 'red' if label == "羊肚菌" else colors[count % 20]
        count += 1

    # 绘制折线图
    plt.plot([2023 + i for i in range(8)], df.iloc[i, 2:],
             label=label,
             color=color)

# 设置标题和标签
plt.xlabel('年份', fontsize=17)
plt.ylabel(y_label, fontsize=17)

# 显示图例
# plt.legend(loc='best', bbox_to_anchor=(1.3, 0.7), ncol=3, fontsize=7)
plt.legend(loc='best', ncol=3, fontsize=7)

# 设置图形的样式
ax = plt.gca()
ax.spines['top'].set_color('none') # 去除顶部边框
ax.spines['right'].set_color('none') # 去除右侧边框
plt.rcParams['xtick.direction'] = 'in' # x轴刻度线向内
plt.rcParams['ytick.direction'] = 'in' # y轴刻度线向内
ax.spines["bottom"].set_linewidth(2) # 底部边框加粗
ax.spines["left"].set_linewidth(2) # 左侧边框加粗

# 旋转 x 轴标签以防止重叠
plt.xticks(rotation=45)

# 调整布局以避免标签重叠
plt.tight_layout()
plt.savefig(f"第二问求解结果/{title}图.pdf", dpi=400, format="pdf", bbox_inches="tight")

# 显示图形
plt.show()
def plot_hist(x, x_lab, y_lab, title="直方图", bin=60, pltx_size=15, plty_size=6, color='blue',
             ecolor='black'):

```

```

# 绘制特征的直方图
fig,ax=plt.subplots(figsize=(plt.x_size, plt.y_size))
sns.histplot(x, kde=False, color=color, bins=30, edgecolor=ecolor, linewidth=1.2, ax=ax)
# 设置x轴和y轴的标签
plt.xlabel(x_lab,fontsize=20)
plt.ylabel(y_lab,fontsize=20)
# 设置x和y刻度值大小
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
# 设置边框大小为2磅
for spine in plt.gca().spines.values():
    spine.set_linewidth(2)

plt.savefig(f"{title}.pdf",dpi=500,format="pdf", bbox_inches='tight')

```

问题三代码

```

import pandas as pd
import numpy as np
np.set_printoptions(precision=2, suppress=True)
import matplotlib.pyplot as plt
from matplotlib import rcParams
import matplotlib.pyplot as plt

config = {
    "font.family": 'serif',
    "font.size": 12,
    "mathtext.fontset": 'stix',
    "font.serif": ["KaiTi"],
    "axes.unicode_minus": False,
    "xtick.direction": 'in',
    "ytick.direction": 'in',
}
rcParams.update(config)

#第一季的编码组合
df_get_type_all = pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
for type_ in ["普通大棚第一季", "智慧大棚第一季", "水浇地第一季"]:
    df_get_type = pd.DataFrame(columns=["地块类型", "编码取值", "第一季种植作物"])
    # 暂存要追加的数据
    data_to_append = []
    for i in range(len(dict1[type_])):
        # 将数据添加到列表中
        data_to_append.append([type_, i + 1, dict1[type_][i]])

    # 将数据列表转换为 DataFrame, 并追加到 df_get_type
    df_temple=pd.DataFrame(data_to_append, columns=df_get_type_all.columns)
    df_get_type_all = pd.concat([df_get_type_all, df_temple], ignore_index=True)

```

```

#第二季的编码组合
df_get_type_all12= pd.DataFrame(columns=["地块类型", "编码取值", "单季种植作物"])
for type_ in ["普通大棚第二季","智慧大棚第二季",'水浇地第二季']:
    df_get_type = pd.DataFrame(columns=["地块类型", "编码取值", "第二季种植作物"])
    # 暂存要追加的数据
    data_to_append = []
    for i in range(len(dict1[type_])):
        # 将数据添加到列表中
        data_to_append.append([type_, i + 1, dict1[type_][i]])

    # 将数据列表转换为 DataFrame, 并追加到 df_get_type
    df_temple=pd.DataFrame(data_to_append, columns=df_get_type_all12.columns)
    df_get_type_all12= pd.concat([df_get_type_all12, df_temple], ignore_index=True)

upper_temp=[]
#地的类型
type_list=[]
a=max( df_get_type_all[df_get_type_all["地块类型"]=="普通大棚第一季"]["编码取值"] )
b=max( df_get_type_all[df_get_type_all["地块类型"]=="智慧大棚第一季"]["编码取值"] )

for index in range(df1.shape[0]):
    #地块类型
    type_=( df1.loc[index]["地块类型"] )
    type_ = type_.replace(" ", "")
    type_list.append(type_)

    if type_ == "普通大棚":
        #地块可种植的作物数量
        upper_temp.append( a )
    elif type_ == "智慧大棚":
        upper_temp.append( b )

#地的类型
type_list=[]
a=max( df_get_type_all12[df_get_type_all12["地块类型"]=="普通大棚第二季"]["编码取值"] )
b=max( df_get_type_all12[df_get_type_all12["地块类型"]=="智慧大棚第二季"]["编码取值"] )

upper_temp2=[]
for index in range(df1.shape[0]):
    #地块类型
    type_=( df1.loc[index]["地块类型"] )
    type_ = type_.replace(" ", "")
    type_list.append(type_)

```

```

if type_ == "普通大棚":
    #地块可种植的作物数量
    upper_temp2.append( a )
elif type_ == "智慧大棚":
    upper_temp2.append( b )

c=max( df_get_type_all[df_get_type_all["地块类型"]=="水浇地第一季"]["编码取值"] )
upper_temp3=[]
for index in range(df1.shape[0]):
    #地块类型
    if(index==32 or index==33):
        continue
    type_=( df1.loc[index]["地块类型"] )
    type_ = type_.replace(" ", "")
    type_list.append(type_)
    if type_ == "水浇地":
        #地块可种植的作物数量
        upper_temp3.append( c )
c=max( df_get_type_all2[df_get_type_all2["地块类型"]=="水浇地第二季"]["编码取值"] )
upper_temp4=[]
for index in range(df1.shape[0]):
    #地块类型
    if(index==32 or index==33):
        continue
    type_=( df1.loc[index]["地块类型"] )
    type_ = type_.replace(" ", "")
    type_list.append(type_)
    if type_ == "水浇地":
        #地块可种植的作物数量
        upper_temp4.append( c )
upper=upper_temp+upper_temp+upper_temp2+upper_temp2+upper_temp3+upper_temp4
len(upper)
# 下面运行geat优化即可得到结果
# 保存结果
with pd.ExcelWriter("第三问求解结果/result3_单季_.xlsx") as writer:
    for year, j in zip(range(2024, 2031), range(1,8)):
        result=pd.DataFrame(columns= [
            "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"],index=list( df1["地块名称"] )[:26] )
        result=result.reset_index()
        result=result.rename(columns={"index":"地块名称"})
        for i in range(len(history[j])):
            result.loc[i,crop_dict[history[j][i]]]=area_list[i]
        result.fillna(0).to_excel(writer, sheet_name=str(year), index=False)

```