

# 农作物种植策略

## 摘要

针对问题，我们采用数学规划进行求解。首先，对于干旱地，梯田，山坡，我们采用的是整数规划，因为此些地块面积多，无需考虑混种。而对于水浇地和大棚，我们采用了线性规划的方法，用每种植物的百分比来代表数值。因为不同土地上的种植情况是独立的，所以我们对不同土地分别使用了不同的决策变量来方便求解。

**关键词：** 线性规划 整数规划 求解器应用

## 一、问题背景与重述

### 1.1 问题背景

在当今快速发展的社会中，乡村经济的可持续发展面临诸多挑战，尤其是在耕地资源日益紧缺的背景下。为了最大程度地发挥有限的土地产能，乡村地区亟需因地制宜地发展有机种植产业。不同的地理和气候条件对作物的生长和经济收益有着直接的影响，因此，选择适宜的农作物及其优化种植策略成为提升乡村经济的重要举措。通过事先制定合理的种植计划，能够有效促进田间管理，降低劳动成本，提高生产效率。通过科学的管理和选择，可在一定程度上减少由天气变化、病虫害等不确定因素引发的种植风险，从而保障农户的经济收益。

### 1.2 问题重述

基于避免重茬种植、每三年种一次豆类植物，每种作物的种植面积不宜过小、季节限制、土地限制的约束条件，针对以下两种情况设计两种最优作物种植方案，方案包括每块地应种植的作物及其对应的面积。情况一：产出的超过预期销售量的作物浪费，卖不出去也不能带来收益。情况二：产出的超出部分按照正常价格的 50% 计算销售额，参与收益计算

## 二、问题分析

根据 2023 年每种作物在不同地块上的种植面积以及该地块上的亩产量求出销售预期量，另外，对于每种作物的销售价格会出现区间，为了简化数学模型，我们取每种作物销售价格的中间值进行计算求解。整理好数据后，我们利用混合规划方法将所有地块和作物拆成两部分分别进行求解。首先，介于前三种地块类型面积较大，适合每个地块一次性全部分配一种作物的方式，所以我们可以对粮食作物与平旱地、梯田、

山坡的规划问题进行整数规划求解；而后面水浇地、普通大棚与智慧大棚，由于面积较小，适合多种作物混种，也便于管理，可以利用线性规划进行求解，将两部分分别求解后得到的结果整合一起即可

### 三、模型假设

1. 假设销售单价中间值能够较好地衡量作物的实际销售价格
2. 假设产出的作物都在较好的环境条件下贮藏，在降价出售的过程及之后很长一段时间保持着良好的品质。
3. 假设 2023 年的所有种植的农作物都被销售出去，下一年每种作物的预期销售量 2023 年的每一种作物的总亩产量保持一致。
4. 假设每种作物生产出来都不会由于运输、环境变化而损耗。

### 四、符号说明

| 符号    | 说明           |
|-------|--------------|
| $D_j$ | 第 j 种植物的种植成本 |
| $Q_j$ | 第 j 种植物的预期销量 |
| $K_j$ | 第 j 种植物的售价   |
| $P_j$ | 第 j 种植物的亩产量  |
| $T_i$ | 第 i 块地的亩数    |

### 五、模型的建立与求解

#### 5.1 对于干旱地，梯田，山坡地耕地类型的问题模型

##### 5.1.1 决策变量的建立

决策变量

$$a_{ijk} \quad (i \in \{1, 2, \dots, 26\}, j \in \{1, 2, \dots, 15\}, k \in \{1, 2, \dots, 7\})$$

其中 i 代表耕地类型（从小到大为干旱地，梯田，山坡地），j 代表农作物类型（前 5 种为豆类，与提供数据相同），k 代表年份（从 2024 到 2030，设 2023 为序号 0）。

##### 5.1.2 模型建立

问题要求我们在地块类型约束、作物类型约束、季节约束、禁止重茬种植约束、豆类作物种植频率（每三年种一次豆类）约束之下给出 2024 年-2030 年每一块地的最优种植方案。我们根据作物利润最大化原则以及根据问题给出的两种情况分别确定目

标函数，通过寻找目标函数的最大值求出最优种植方案。首先我们构造目标函数，模型的总利润等于所有年份、地块和作物的总实际销售收入减去总种植成本：

设总利润为 TP，则目标函数为

$$\max\{TP\}$$

$$TP = \sum_{k=1}^7 \sum_{j=1}^{15} \sum_{i=1}^{26} \{\min(T_i \cdot a_{ijk} \cdot P_j, Q_j) \cdot K_j - T_i \cdot a_{ijk} \cdot P_j \cdot Dj + Rest\}$$

第一种情况：超额产出视为浪费，不产生价值。则

$$Rest = 0$$

第二种情况：超过预期销售量部分粮食按照 50

$$Rest = T_i \cdot b_{ijnk} \cdot P_j - Q_j$$

### 5.1.3 约束条件

1. 每一亩耕地都有农作物种植

$$\forall i, k \quad \sum_{j=1}^{15} a_{ijk} = 1$$

2. 每 3 年必须种植一次豆类作物

$$\forall j, \forall k \in 1, 2, \dots, 5 \quad \sum_{i=1}^{26} \{a_{ijk} + a_{ijk+1} + a_{ijk+2}\} \geq 1$$

3. 避免作物重茬种植。

$$\forall j, i \quad \forall k \in 1, 2, \dots, 6 \quad a_{ijk} + a_{ijk+1} \leq 1$$

## 5.2 对于水浇地，普通大棚和智慧大棚耕地类型的问题模型

### 5.2.1 决策变量的建立

决策变量

$$b_{ijnk} \quad (i \in \{28, 29, \dots, 54\}, j \in \{16, 17, \dots, 37\}, n \in 1, 2, k \in \{1, 2, \dots, 7\})$$

其中 i 代表耕地类型（从小到大为水浇地，普通大棚和智慧大棚），j 代表农作物类型，n 代表第 n 季，k 代表年份（从 2024 到 2030，设 2023 为序号 0）。

### 5.2.2 中间变量的建立

定义中间 0, 1 变量

$$B_{ijnk} \quad (i \in \{28, 29, \dots, 54\}, j \in \{16, 17, \dots, 37\}, n \in \{1, 2\} k \in \{1, 2, \dots, 7\})$$

$$\begin{cases} B_{ijnk} = 0 \rightarrow \text{在第 } i \text{ 块地, 第 } k \text{ 年, 第 } n \text{ 季没有种植第 } j \text{ 种植物} \\ B_{ijnk} = 1 \rightarrow \text{在第 } i \text{ 块地, 第 } k \text{ 年, 第 } n \text{ 季没有种植第 } j \text{ 种植物 (无论种植多少)} \end{cases}$$

B 与 b 的转换为

$$\begin{cases} B \leq G \cdot b \\ B \geq b \end{cases}$$

其中 G 为一个相对较大的数

### 5.2.3 模型建立

设总利润为 TP, 则目标函数为

$$\max\{TP\}$$

$$TP = \sum_{n=1}^2 \sum_{k=1}^7 \sum_{j=1}^{15} \sum_{i=1}^{26} \{\min(T_i \cdot b_{ijnk} \cdot P_j, Q_j) \cdot K_j - T_i \cdot b_{ijnk} \cdot P_j \cdot Dj + Rest\}$$

第一种情况: 超额产出视为浪费, 不产生价值。则

$$Rest = 0$$

第二种情况: 超过预期销售量部分粮食按照 50

$$Rest = T_i \cdot b_{ijnk} \cdot P_j - Q_j$$

### 5.2.4 约束条件

1. 每一亩耕地都有农作物种植

$$\forall i, k, n \quad \sum_{j=16}^{37} b_{ijnk} = 1$$

2. 水稻要么种两季, 要么种一季, 且需要全种水稻

$$\forall i, k, \quad \begin{cases} B_{i161k} = B_{i162k} \\ B_{i16nk} = b_{i16nk} \quad \text{且} \quad B_{i161k} = 0, (i \in \{35, 36, \dots, 54\}) \end{cases}$$

3. 假设一块最多四种作物

$$\forall i, k, n \quad \sum_{j=17}^{37} B_{ijnk} \leq 4$$

4. 每 3 年必须种植一次豆类作物

$$\forall j, \forall k \in 1, 2, \dots, 5 \quad \sum_{n=1}^2 \sum_{j=17}^{19} \{b_{ijk} + b_{ijk+1} + b_{ijk+2}\} \geq 1$$

5. 避免作物重茬种植。

$$\forall j, i \quad \forall k \in 1, 2, \dots, 6 \quad \begin{cases} B_{ij1k} + B_{ij2k} \leq 1 \\ B_{ij2k} + B_{ij1k+1} \leq 1 \end{cases}$$

6. 大白菜，白萝卜，红萝卜只能种在水浇地第二季

$$\forall j \in \{35, 36, 37\}, n \in 1, 2 \quad \begin{cases} B_{ij2k} = 0 & (i \in \{27, 28, \dots, 34\}) \\ B_{ijnk} = 0 & (i \in \{35, 36, \dots, 50\}) \end{cases}$$

7. 有些农作物只能种在第一季

$$\forall j \in \{17, 18, \dots, 34\}, i \in \{28, 29, \dots, 50\} \quad B_{ij2k} = 0$$

### 5.3 模型求解与结果

我们运用整数规划和线性规划的方法，通过结合循环、Python 内置函数以及调用开源求解器 Gurobi，将不同的约束条件整合到目标函数中。规划求解中，首先，鉴于前山中地块类型面积较大，适合每个地块一次性全部分配一种作物的方式，所以我们可以对粮食作物与平旱地、梯田、山坡地的规划问题进行整数规划求解；而后面水浇地、普通大棚与智慧大棚，由于面积较小，适合多种作物混种，也便于管理，可以利用线性规划进行求解，将两部分分别求解后得到的结果整合一起即可。

情况一：超出销售预期的部分直接浪费的情况下，两种分组的结果展示。通过导出数据我们可以得知这部分产生的总收益为 15,200,723.75。年度利润每年都在变化，例如，第一年的利润为 2,197,263.75，第三年的利润为 2,271,018.75，这些波动反映了产量、成本和市场价格等因素的影响。此外，通过以下“粮食类型作物与‘平旱地’、‘梯田’、‘山坡地’”的分配方案的 3D 图，我们可以直观看出每年各类作物所分配的种植地块以及其种植面积。

## 六、模型的评价、改进与推广

### 6.1 模型优点

1. 使用求解器进行求解，保证在给定约束下，结果是最优的
2. 给予了一个合理的种植策略去辅助农作物种植

## 6.2 模型缺点

1. 模型不反应价格变动，只是用中间值进行替代
- 2.

## 6.3 模型的改进

- 1.
- 2.
- 3.

## 附录

附录 1: 支撑材料的文件列表

附录 2: 初始化代码和数据处理代码

```
import os
import openpyxl
import numpy as np

LAND_NUM = 54
CROP_NUM = 41
SEASON_NUM = 2
YEAR_NUM = 8 # 2023–2030

class Land:
    def __init__(self, name: str, type: str, size: float):
        self.name = name # e.g. A1
        self.type = type # e.g. 平旱地
        self.size = size # e.g. 80 (亩)
        self.production = np.zeros((CROP_NUM, SEASON_NUM))
        self.cost = np.zeros((CROP_NUM, SEASON_NUM))
        self.price = np.zeros((CROP_NUM, SEASON_NUM))

class Crop:
    def __init__(self, name: str, type: str):
        self.name = name # e.g. 水稻
        self.type = type # e.g. 粮食 (豆类)
        self.expected_sale = 0 # 预期销量 (斤)

    crop_id_of = {}
    land_id_of = {}
    lands = []
    crops = []

    path = os.path.join(os.getcwd(), 'problem')
    print(path)
    if not os.path.exists(path):
```

```

print("Incorrect_path!")
exit(1)

try:
workbook = openpyxl.load_workbook(os.path.join(path, "附件1_现有
    作物与土地情况.xlsx"))
except FileNotFoundError:
print("Attachment_1_not_found!")
exit(1)

sheet_names = workbook.sheetnames

# 读取第一张工作表
sheet = workbook[sheet_names[0]]
cnt = 0
for row in sheet.iter_rows(min_row=2, values_only=True):
    land_name = row[0].strip()
    land_type = row[1].strip()
    land_size = row[2]
    land_id_of[land_name] = cnt

    land_instance = Land(land_name, land_type, land_size)
    lands.append(land_instance)
    cnt += 1

# for land in lands:
# print(f'Land name: {land.name}, type: {land.type}, size: {land.size},
#     id: {land_id_of[land.name]}')

sheet = workbook[sheet_names[1]]
cnt = 0
for row in sheet.iter_rows(min_row=2, max_row=42, values_only=
    True):
    crop_name = row[1].strip()
    crop_id_of[crop_name] = cnt
    crop_type = row[2].strip()
    crop_instance = Crop(crop_name, crop_type)

```



```

crops.append(crop_instance)
cnt += 1

# for crop in crops:
# print(f'Crop name: {crop.name}, type: {crop.type}, id: {crop_id_of[
#     crop.name]}')
workbook.close()

try:
workbook = openpyxl.load_workbook(os.path.join(path, "附件2_去年
    作物与收成情况.xlsx"))
except FileNotFoundError:
print("Attachment_2_not_found!")
exit(1)

sheet_names = workbook.sheetnames
sheet = workbook[sheet_names[1]] # 作物产量、单价等数据

for row in sheet.iter_rows(min_row=2, max_row=108, values_only=
    True):
crop_name = row[2].strip() # 作物名称
land_type = row[3].strip() # 地块类型
crop_season = row[4] # 种植季次
crop_production = row[5] # 亩产量 斤/亩
crop_cost = row[6] # 种植成本 元/亩
price1, price2 = row[7].split('-') # 销售单价 元/斤

price = (float(price1) + float(price2)) / 2 # 价格暂时取平均数

for land in lands:
if land.type == land_type:
if crop_season == '单季' or crop_season == '第一季':
land.production[crop_id_of[crop_name], 0] = crop_production
land.cost[crop_id_of[crop_name], 0] = crop_cost
land.price[crop_id_of[crop_name], 0] = price
elif crop_season == '第二季':
land.production[crop_id_of[crop_name], 0] = crop_production

```

```

land.cost[crop_id_of[crop_name], 0] = crop_cost
land.price[crop_id_of[crop_name], 0] = price

# 表格中缺了智慧大棚第一季的数据，需要手动填写
format_land = lands[land_id_of['E1']] # 找普通大棚当作模板
for land_id in range(land_id_of['F1'], land_id_of['F4'] + 1): # 智慧大
    棚
    land = lands[land_id]
    for crop_id in range(CROP_NUM):
        land.production[crop_id][0] = format_land.production[crop_id][0]
        land.cost[crop_id][0] = format_land.cost[crop_id][0]
        land.price[crop_id][0] = format_land.price[crop_id][0]

sheet = workbook[sheet_names[0]] # 2023 年种植情况
land_name_buffer = None
for row in sheet.iter_rows(min_row=2, max_row=88, values_only=
    True):
    if row[0] is not None:
        land_name_buffer = row[0]
        crop_name = row[2].strip()
        crop_area = row[4]
        this_crop = crops[crop_id_of[crop_name]]
        this_land = lands[land_id_of[land_name_buffer]]
        season = row[5]
        if season == "单季" or season == "第一季":
            this_crop.expected_sale += crop_area * this_land.production[
                crop_id_of[crop_name], 0]
        elif season == "第二季":
            this_crop.expected_sale += crop_area * this_land.production[
                crop_id_of[crop_name], 1]

workbook.close
# print(crops[crop_id_of["小麦"]].expected_sale)
# print(crops[crop_id_of["空心菜"]].expected_sale)

from gurobipy import GRB, Model

```

```

SubProblem1 = Model("SubProblem1")

# a_26_15_8 = SubProblem1.addVars(26, 15, 8, vtype=GRB.
    CONTINUOUS, name='a') # 地, 物, 年
a_26_15_8 = SubProblem1.addVars(26, 15, 8, vtype=GRB.BINARY,
    name='a') # 地, 物, 年
max_i, max_j, max_k = 26, 15, 8
bean_id_range = (1, 5) # 0-4 是豆类植物

# # 0.1. 24-30年决策变量上下限, 若使用连续变量需要确定上下界
# for i in range(26):
# for j in range(15):
# for k in range(1, 8):
# a_26_15_8[i, j, k].set(GRB.Attr.LB, 0.0)
# a_26_15_8[i, j, k].set(GRB.Attr.UB, 1.0)

# 0.2. 读入23年的数据
try:
workbook = openpyxl.load_workbook(os.path.join(path, "附件2_去年
    作物与收成情况.xlsx"))
except FileNotFoundError:
print("Attachment_2_not_found!")
exit(1)

sheet_names = workbook.sheetnames
sheet = workbook[sheet_names[0]] # 23年种植情况

for row in sheet.iter_rows(min_row=2, max_row=27, values_only=
    True):
land_name = row[0].strip()
crop_name = row[2].strip()
land_id = land_id_of[land_name]
crop_id = crop_id_of[crop_name]
for j in range(max_j):
SubProblem1.addConstr(a_26_15_8[land_id, j, 0] == (1 if j ==
    crop_id else 0))

```

```
workbook.close()
```

```
# 1. 每块地刚好用满
```

```
constraint_1 = SubProblem1.addConstrs(  
    sum(  
        a_26_15_8[i, j, k]  
        for j in range(max_j) # 所有作物比例之和  
    ) == 1  
    for i in range(max_i) # 对于每块地  
    for k in range(1, max_k) # 对于 24–30 年  
)
```

```
# 2. 豆类作物三年至少种一次
```

```
constraint_2 = SubProblem1.addConstrs(  
    sum(  
        a_26_15_8[i, j, k] + a_26_15_8[i, j, k + 1] + a_26_15_8[i, j, k + 2]  
        # 连续三年之和  
        for j in range(*bean_id_range) # 豆类植物  
    ) >= 1  
    for i in range(max_i) # 对于每块地  
    for k in range(max_k - 2) # 对于 23–28 年  
)
```

```
# 3. 不重茬
```

```
constraint_3 = SubProblem1.addConstrs(  
    a_26_15_8[i, j, k] + a_26_15_8[i, j, k + 1] <= 1  
    for i in range(max_i) # 对于每块地  
    for j in range(max_j) # 对于每种作物  
    for k in range(max_k - 1) # 对于 23–29 年  
)
```

```
# 4. 目标函数
```

```
profit_expr = 0  
for k in range(1, max_k):  
    year = k # 对于每一年  
    for j in range(max_j):
```

```

crop_id = j
crop = crops[crop_id] # 对于每种作物
cost = 0
production = 0
for i in range(max_i): # 对于每块地
    land_id = i
    land = lands[land_id]
    planted_area = a_26_15_8[i, j, k] * land.size
    production += planted_area * land.production[crop_id, 0]
    cost += planted_area * land.cost[crop_id, 0]
    sale_var = SubProblem1.addVar(vtype=GRB.CONTINUOUS, name=f'
        sale_{i}_{j}_{k}')
    # 滞销
    SubProblem1.addConstr(sale_var <= production)
    SubProblem1.addConstr(sale_var <= crop.expected_sale)

income = sale_var * lands[0].price[crop_id, 0]
profit_expr += income - cost

SubProblem1.setObjective(profit_expr, GRB.MAXIMIZE)

SubProblem1.optimize()

for i in range(max_i):
    for j in range(max_j):
        for k in range(max_k):
            print(f'a[{i},{j},{k}]= {a_26_15_8[i,j,k].X}')

```