

农作物的种植策略 [初稿]

0.1 问题背景

根据乡村的实际情况，充分利用有限的耕地资源，因地制宜，发展有机种植产业，对乡村经济的可持续发展具有重要的现实意义。选择适宜的农作物，优化种植策略，有利于方便田间管理，提高生产效益，减少各种不确定因素可能造成的种植风险。某乡村地处华北山区，常年温度偏低，大多数耕地每年只能种植一季农作物。该乡村现有露天耕地 1201 亩，分散为 34 个大小不同的地块，包括平旱地、梯田、山坡地和水浇地 4 种类型。平旱地、梯田和山坡地适宜每年种植一季粮食类作物；水浇地适宜每年种植一季水稻或两季蔬菜。该乡村另有 16 个普通大棚和 4 个智慧大棚，每个大棚耕地面积为 0.6 亩。普通大棚适宜每年种植一季蔬菜和一季食用菌，智慧大棚适宜每年种植两季蔬菜。同一地块（含大棚）每季可以合种不同的作物。详见附件 1。根据农作物的生长规律，每种作物在同一地块（含大棚）都不能连续重茬种植，否则会减产；因含有豆类作物根菌的土壤有利于其他作物生长，从 2023 年开始要求每个地块（含大棚）的所有土地三年内至少种植一次豆类作物。同时，种植方案应考虑到方便耕种作业和田间管理，譬如：每种作物每季的种植地不能太分散，每种作物在单个地块（含大棚）种植的面积不宜太小，等等。2023 年的农作物种植和相关统计数据见附件 2。

0.2 问题要求

假定各种农作物未来的预期销售量、种植成本、亩产量和销售价格相对于 2023 年保持稳定，每季种植的农作物在当季销售。如果某种作物每季的总产量超过相应的预期销售量，超过部分不能正常销售。请针对以下两种情况，分别给出该乡村 2024-2030 年农作物的最优种植方案，将结果分别填入 *result1_1.xlsx* 和 *result1_2.xlsx* 中（模板文件见附件 3）。(1) 超过部分滞销，造成浪费；

一、模型假设

为简化问题，本文做出以下假设：

- 每种作物在单个地块（含大棚）种植的面积最小可接受面积为 2023 年最小面积的三分之一
- 2024-2030 年一直保持 2023 年的预售量
- 没有预售量的作物预售量为 2023 年预售量的最大值

二、符号说明

符号	说明	单位/范围
$x_{i,j,k,t}$	表示第 t 年第 k 季度在 j 地块种植 i 作物的面积	亩
$y_{i,j,k,t}$	表示第 t 年第 k 季度在 j 地块种植 i 作物	-
p_i	表示第 i 作物的销售单价	斤/元
$c_{i,j,k}$	表示第 i 作物的种植成本	斤/元
$q_{i,j,k}$	表示第 i 作物的亩产量	亩/斤
A_j	表示第 j 块地面积	亩
z_i	表示第 i 块种作物	-
SA_i	平旱地对应集合	-
B_i	梯田对应集合	-
C_i	山坡地对应集合	-
D_i	水浇地对应集合	-
E_i	普通大棚对应集合	-
F_i	智慧大棚对应集合	-
M_i	最小地块面积	-
N_i	大数	-
$S_{i,j,k}$	第 k 季度在 j 地块种植 i 作物预售量	-

三、问题一的模型的建立和求解

3.1 模型建立

设表示第 t 年第 k 季度在 j 地块种植 i 作物的面积为 $x_{i,j,k,t}$, $y_{i,j,k,t} = 1$ 表示第 t 年第 k 季度在 j 地块种植 i 作物。

(1-1) 干旱地, 梯田, 山坡地适应种植一季度作物, 即是要求第 j 块地在每年、每个季度都不能超过第 j 块地的面积

$$\sum_{i=z1}^{z15} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \quad (1)$$

(1-2) 只能种植单季度，后面两个季度都不能用

$$\sum_{i=z1}^{z15} y_{i,j,2,t} = 0, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \quad (2)$$

(1-3) 干旱地，梯田，山坡地适应种植一季度作物，至少选一种种

$$\sum_{i=z1}^{z15} y_{i,j,1,t} \geq 1, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \quad (3)$$

(1-4) 1-15 号作物不能种在其他地块

$$\sum_{i=z1}^{z15} y_{i,j,k,t} = 0, j \in S \setminus SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \quad (4)$$

(2-1) 若水浇地种植单季度水稻有

$$x_{z16,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \quad (5)$$

同理有

$$y_{z16,j,2,t} = 0, j \in D_1 - D_8 \quad (6)$$

水稻只能种在水浇地

$$y_{z16,j,2,t} = 0, j \in S \setminus D_1 - D_8 \quad (7)$$

(2-1) 若水浇地种植多季度作物。第一季可以种植除去 (大白菜，白萝卜和红萝卜) 外的蔬菜

$$\sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \quad (8)$$

由于只能种植单季度，故

$$\sum_{i=z17}^{z34} y_{i,j,2,t} = 0, j \in D_1 - D_8 \quad (9)$$

$$\sum_{i=z17}^{z34} y_{i,j,1,t} \geq 1, j \in D_1 - D_8 \quad (10)$$

17-34 号作物不能种干旱地，梯田，山坡地

$$\sum_{i=z17}^{z34} y_{i,j,k,t} = 0, j \in SA_1 - SA_6 \cup B_1 - B_{14} \cup C_1 - C_6 \quad (11)$$

第二季可以种植大白菜，白萝卜和红萝卜中的一种

$$\sum_{i=z35}^{z37} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \quad (12)$$

由于只能种植第二季度，故第一季度种植种类为 0

$$\sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in D_1 - D_8 \quad (13)$$

又因为种植第二季度，种植大白菜，白萝卜和红萝卜中的一种

$$\sum_{i=z35}^{z37} y_{i,j,2,t} = 1, j \in D_1 - D_8 \quad (14)$$

(2-3) 水浇地要么种植水稻要么种植两季作物

$$y_{z16,j,1,t} + \frac{\sum_{i=z17}^{z34} y_{i,j,1,t} + \sum_{i=z35}^{z37} y_{i,j,1,t}}{2} = 1, j \in D_1 - D_8 \quad (15)$$

35-37 号作物不能种在第一季

$$\sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in S \quad (16)$$

除水浇地外所有第二季不能种植 35-37 号作物

$$\sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in S \setminus D_1 - D_8 \quad (17)$$

(3) 智慧大棚每年都可种植两季蔬菜（大白菜、白萝卜和红萝卜除外）。

$$\sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in F_1 - F_4, k \in 1, 2 \quad (18)$$

$$\sum_{i=z17}^{z34} y_{i,j,k,t} \geq 1, j \in F_1 - F_4, k \in 1, 2 \quad (19)$$

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

$$\sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in E_1 - E_{15} \quad (20)$$

由于只能种植单季度，故

$$\sum_{i=z17}^{z34} y_{i,j,2,t} = 0, j \in E_1 - E_{15} \quad (21)$$

$$\sum_{i=z17}^{z34} y_{i,j,1,t} \geq 1, j \in E_1 - E_{15} \quad (22)$$

第二季只能种植食用菌, 但不限于一种

$$\sum_{i=z38}^{z41} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in E_1 - E_{15} \quad (23)$$

同理有:

$$\sum_{i=z38}^{z41} y_{i,j,2,t} \geq 1, j \in E_1 - E_{15} \quad (24)$$

食用菌一季度不能种植

$$\sum_{i=z38}^{z41} y_{i,j,1,t} = 0, j \in E_1 - E_{15} \quad (25)$$

38-41 号作物第二季只能种在普通大棚

$$\sum_{i=z38}^{z41} y_{i,j,2,t} = 0, j \in S \setminus E_1 - E_{15} \quad (26)$$

(5) 每种作物在同一地块 (含大棚) 都不能连续重茬种植 (5-1) 连续两个季度不能重复种植同一种作物

$$y_{i,j,1,t} + y_{i,j,2,t} \leq 1 \quad (27)$$

(5-2) 连续两个年不能重复种植同一种作物

$$y_{i,j,k,t} + y_{i,j,k,t+1} \leq 1 \quad (28)$$

(6) 每种作物在单个地块 (含大棚) 种植的面积不宜太小

$$x_{i,j,k,t} * y_{i,j,k,t} + N * (1 - y_{i,j,k,t}) \geq M \quad (29)$$

(7) 每块地的产量不能超过预售量, 超过部分应该舍去

$$S_{i,j,k} * y_{i,j,k,t} + N * (1 - y_{i,j,k,t}) \geq x_{i,j,k,t} * d_{i,j,k} \quad (30)$$

(8) 每隔 3 年至少种植一次豆类 (8-1) 对于豆类作物 1-5 只能种植在干旱地, 梯田, 山坡地单季度

$$\sum_{i=z1}^{z5} y_{i,j,1,t} + y_{i,j,1,t+1} + y_{i,j,1,t+2} \geq 1, j \in SA_1 - SA_6 \cup B_1 - B_{14} \cup C_1 - C_6 \quad (31)$$

(8-2) 对于豆类作物 17-19 只能种植在水浇地, 普通 (智慧) 大棚第一季度。

$$\sum_{i=z17}^{z19} y_{i,j,1,t} + y_{i,j,1,t+1} + y_{i,j,1,t+2} \geq 1, j \in D_1 - D_8 \cup E_1 - E_{16} \cup F_1 - F_4 \quad (32)$$

(8-3) 对于豆类作物 17-19 只能种植在智慧大棚第一季度。

$$\sum_{i=z17}^{z19} y_{i,j,2,t} + y_{i,j,2,t+1} + y_{i,j,2,t+2} \geq 1, j \in F_1 - F_4 \quad (33)$$

本文的核心在满足上述条件下收益最大，收益等于总销售额减去总种植成本。但是在实际销售过程中如果某种作物每季的总产量超过相应的预期销售量，超过部分不能正常销售。超过部分滞销，造成浪费；2023 年的预售量等于 2023 年的种植面积 x2023 年的亩产量。故目标函数为：

$$\min \sum_{t=24}^{30} \sum_{i=z1}^{z41} \sum_{j=1}^{53} x_{i,j,k,t} * y_{i,j,k,t} * q_{i,j,k} - x_{i,j,k,t} * y_{i,j,k,t} * c_{i,j,k} \quad (34)$$

综上有如下线性规划模型;

$$\begin{aligned}
\min \quad z = & \sum_{t=24}^{30} \sum_{i=z1}^{z41} \sum_{j=1}^{53} x_{i,j,k,t} * y_{i,j,k,t} * q_{i,j,k} - x_{i,j,k,t} * y_{i,j,k,t} * c_{i,j,k} \\
\text{s.t.} \quad & \left\{ \begin{aligned}
& \sum_{i=z1}^{z15} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \\
& \sum_{i=z1}^{z15} y_{i,j,2,t} = 0, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \\
& \sum_{i=z1}^{z15} y_{i,j,1,t} \geq 1, j \in SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \\
& \sum_{i=z1}^{z15} y_{i,j,k,t} = 0, j \in S \setminus SA_1 - SA_6, B_1 - B_{14}, C_1 - C_6 \\
& x_{z16,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \\
& y_{z16,j,2,t} = 0, j \in D_1 - D_8 \\
& y_{z16,j,2,t} = 0, j \in S \setminus D_1 - D_8 \\
& \sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \\
& \sum_{i=z17}^{z34} y_{i,j,2,t} = 0, j \in D_1 - D_8 \\
& \sum_{i=z17}^{z34} y_{i,j,1,t} \geq 1, j \in D_1 - D_8 \\
& \sum_{i=z17}^{z34} y_{i,j,k,t} = 0, j \in SA_1 - SA_6 \cup B_1 - B_{14} \cup C_1 - C_6 \\
& \sum_{i=z35}^{z37} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in D_1 - D_8 \\
& \sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in D_1 - D_8 \\
& \sum_{i=z35}^{z37} y_{i,j,2,t} = 1, j \in D_1 - D_8 \\
& y_{z16,j,1,t} + \frac{\sum_{i=z17}^{z34} y_{i,j,1,t} + \sum_{i=z35}^{z37} y_{i,j,1,t}}{2} = 1, j \in D_1 - D_8 \\
& \sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in S \\
& \sum_{i=z35}^{z37} y_{i,j,1,t} = 0, j \in S \setminus D_1 - D_8 \\
& \sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in F_1 - F_4, k \in 1, 2 \\
& \sum_{i=z17}^{z34} y_{i,j,k,t} \geq 1, j \in F_1 - F_4, k \in 1, 2 \\
& \sum_{i=z17}^{z34} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in E_1 - E_{15} \\
& \sum_{i=z17}^{z34} y_{i,j,2,t} = 0, j \in E_1 - E_{15} \\
& \sum_{i=z17}^{z34} y_{i,j,1,t} \geq 1, j \in E_1 - E_{15} \\
& \sum_{i=z38}^{z41} x_{i,j,k,t} * y_{i,j,k,t} \leq A_j, j \in E_1 - E_{15} \\
& \sum_{i=z38}^{z41} y_{i,j,2,t} \geq 1, j \in E_1 - E_{15} \\
& \sum_{i=z38}^{z41} y_{i,j,1,t} = 0, j \in E_1 - E_{15} \\
& \sum_{i=z38}^{z41} y_{i,j,2,t} = 0, j \in S \setminus E_1 - E_{15} \\
& y_{i,j,1,t} + y_{i,j,2,t} \leq 1 \\
& y_{i,j,k,t} + y_{i,j,k,t+1} \leq 1 \\
& x_{i,j,k,t} * y_{i,j,k,t} + N * (1 - y_{i,j,k,t}) \geq M \\
& S_{i,j,k} * y_{i,j,k,t} + N * (1 - y_{i,j,k,t}) \geq x_{i,j,k,t} * d_{i,j,k} \\
& \sum_{i=z1}^{z5} y_{i,j,1,t} + y_{i,j,1,t+1} + y_{i,j,1,t+2} \geq 1, j \in SA_1 - SA_6 \cup B_1 - B_{14} \cup C_1 - C_6 \\
& \sum_{i=z17}^{z19} y_{i,j,1,t} + y_{i,j,1,t+1} + y_{i,j,1,t+2} \geq 1, j \in D_1 - D_8 \cup E_1 - E_{16} \cup F_1 - F_4 \\
& \sum_{i=z17}^{z19} y_{i,j,2,t} + y_{i,j,2,t+1} + y_{i,j,2,t+2} \geq 1, j \in F_1 - F_4
\end{aligned} \right.
\end{aligned} \tag{35}$$

3.2 求解结果

将上述优化问题用 grobi 求解得到如下结果: 获益: 69650964.215 元【注意具体求解结果见附件】

附录 A 文件列表

文件名	功能描述
processData.py	数据预处理程序代码
problem1.py	问题一程序代码

附录 B 代码

processData.py

```
1  # -*- coding: utf-8 -*-
2  import pickle
3  import numpy as np
4  import pandas as pd
5  # 读入原始数据
6  data1=pd.read_excel('../附件1.xlsx',sheet_name='乡村的现有耕地
   ');
7  data2=pd.read_excel('../附件1.xlsx',sheet_name='乡村种植的农作
   物');
8  data3=pd.read_excel('../附件2.xlsx',sheet_name='2023年的农作物
   种植情况');
9  datax=data1.loc[:,['地块类型','地块名称']];
10 datay=data2.loc[:,['作物名称','作物类型']];
11 datay=datay.iloc[0:41,:];
12 data3=pd.merge(datax,data3,left_on='地块名称',right_on='地块名
   称',how='inner');
13 data4=pd.read_excel('../附件2.xlsx',sheet_name='2023年统计的相
   关数据');
14 ## 定义基本参数
15 N=41;#粮食种类
16 M=100000;
17 J=2;#季度数
18 block_counts = {
19     '平旱地': 6,
20     '梯田': 14,
21     '山坡地': 6,
```

```

22     '水浇地': 8,
23     '普通大棚': 16,
24     '智慧大棚': 4
25 }
26 # 地块数
27 S = sum(block_counts.values())
28 block_types = list(block_counts.keys())
29
30 block_sets = {
31     block_type: set(range(sum(block_counts[bt] for bt in
32                             block_types[:i]), sum(block_counts[bt] for bt in block_types
33                             [:i+1])))
34     for i, block_type in enumerate(block_types)
35 }
36 #价格
37 c=np.zeros((N,S,J));
38 #成本
39 q=np.zeros((N,S,J));
40 #需求
41 d=np.zeros((N,S,J));
42 dmand=M*np.ones((N,S,J));
43 ##定义地块类型集合
44 #计算价格和成本表格
45 #41种作物
46 #(2) 智慧大棚第一季可种植的蔬菜作物及其亩产量、种植成本和销售价格均与普通大棚相同，表中省略。
47 val=pd.DataFrame();
48 val2=pd.DataFrame();
49 data4=data4.iloc[0:107,:];
50 def process_price(x):
51     dat=x['销售单价/(元/斤)'];
52     if not isinstance(dat,str):
53         dat=dat.astype(str);
54         dat=dat.str.split('-');
55         datx=dat.tolist();

```

```

54         dat=np.array(datx);
55     else:
56         dat=dat.split('-');
57         dat=np.array(dat);
58         dat=dat.astype(float);
59     return dat
60 for i in range(0,len(data4)):
61     dat=data4.iloc[i,:];
62     if dat['地块类型']=='智慧大棚':
63         #确定作物
64         cond1=data4['作物编号']==dat['作物编号'];
65         #确定季次
66         cond2=data4['种植季次']=='第一季';
67         #确定地块类型
68         cond3=data4['地块类型']=='智慧大棚';
69         #获取对应行
70         val1=data4.loc[cond1&cond2&cond3,:];
71         if len(val1)==0:
72             cond3=data4['地块类型']=='普通大棚 ';
73             val1=data4.loc[cond1&cond2&cond3,:];
74             datx=process_price(val1);
75             datx=datx.reshape(1,-1);
76             valx=pd.DataFrame(datx,columns=['low','High'])
77             val2=pd.concat([val2,valx], ignore_index=True)
78             val= pd.concat([val,val1], ignore_index=True)
79     else:
80         datx=process_price(dat);
81         datx=datx.reshape(1,-1);
82         valx=pd.DataFrame(datx,columns=['low','High'])
83         val2=pd.concat([val2,valx], ignore_index=True)
84 val['地块类型']='智慧大棚';
85 data4=pd.concat([data4,val]);
86 data4[['low','High']]=val2;
87 for i in range(1,N+1):
88     cond1=data4['作物编号']==i;

```

```

89     datx=data4.loc[cond1,:];
90     i=i-1;
91     for cnt in range(len(datx)):
92         dat=datx.iloc[cnt];
93         crop_season=dat['种植季次'];
94         area_name=dat['地块类型'];
95         for val in block_types:
96             if area_name==val:
97                 for j in block_sets[val]:
98                     if crop_season=='单季' or crop_season=='第
一季':
99                         k=0;
100                     else:
101                         k=1;
102                         c[i,j,k]=(dat['low']+dat['High'])/2;
103                         q[i,j,k]=dat['种植成本/(元/亩)'];
104                         d[i,j,k]=dat['亩产量/斤'];
105     cond1=data3['作物编号']==i;
106     cond2=data4['作物编号']==i;
107     datx=data3.loc[cond1,:];
108     daty=data4.loc[cond2,:];
109     i=i-1;
110     for cnt in range(len(datx)):
111         dat=datx.iloc[cnt];
112         crop_season=dat['种植季次'];
113         area_name=dat['地块类型'];
114         cond1=daty['种植季次']==crop_season;
115         cond2=daty['地块类型']==area_name;
116         datz=daty.loc[cond1&cond2,:];
117         for val in block_types:
118             if area_name==val:
119                 for j in block_sets[val]:
120                     if crop_season=='单季' or crop_season=='第
一季':
121                         k=0;

```

```

122         else:
123             k=1;
124             dmand[i,j,k]=dat['种植面积/亩']*datz['亩产
量/斤'].values[0];
125 with open("c.pkl",'wb') as tf:
126     pickle.dump(c,tf);
127 with open("d.pkl",'wb') as tf:
128     pickle.dump(d,tf);
129 with open("q.pkl",'wb') as tf:
130     pickle.dump(q,tf);
131 with open("dmand.pkl",'wb') as tf:
132     pickle.dump(dmand,tf);

```

problem1.py

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Sep 16 08:59:39 2024
4
5  @author: dell
6  """
7  import numpy as np
8  import pandas as pd
9  import time
10 import pyomo.environ as pyo
11 from pyomo.opt import SolverFactory
12 from itertools import chain
13 import pickle
14 data1=pd.read_excel('../附件1.xlsx',sheet_name='乡村的现有耕地
');
15 with open("c.pkl",'rb') as tf:
16     c=pickle.load(tf);
17 with open("d.pkl",'rb') as tf:
18     d=pickle.load(tf);
19 with open("q.pkl",'rb') as tf:
20     q=pickle.load(tf);

```

```

21 with open("dmand.pkl",'rb') as tf:
22     dmand=pickle.load(tf);
23
24 A=data1['地块面积/亩'].values;
25 min_S=min(A);
26 #最小块的面积
27 min_S=min_S.reshape(1,-1)/3;
28 #获取数据
29 min_S=min_S[0];
30 min_S=min_S[0];
31 #记录时间
32 startTime=time.time();
33 ## 定义基本参数
34 N=41;#粮食种类
35 J=2;#季度数
36 #旱地
37 Dry_N=6;
38 #梯田
39 Terr_N=14;
40 #山坡地
41 Hill_N=6;
42 #水浇地
43 Irri_N=8;
44 #普通大棚
45 Comm_N=16;
46 #智慧大棚
47 Smart_N=4;
48 # 地块数
49 S=Dry_N+Terr_N+Hill_N+Irri_N+Comm_N+Smart_N;
50 #粮食集合1数量
51 grain_n1=15;
52 #粮食集合2数量
53 grain_n2=1;
54 #粮食集合3数量
55 grain_n3=18;

```

```

56 #粮食集合4数量
57 grain_n4=3;
58 #粮食集合5数量
59 grain_n5=4;
60 t_max=2030-2023;
61 endTime=time.time();
62 def OptimalProblem():
63     #定义大数
64     Big_M=999999999;
65     ##定义具体模型
66     model=pyo.ConcreteModel();
67     model.name='Planting Strategy'
68     #定义年份
69     t=set(range(1,1+t_max));
70     ##定义粮食集合
71     #粮食集合1数量
72     model.grain1=set(range(1,1+grain_n1));
73     Ns=grain_n1
74     #粮食集合2数量
75     model.grain2=set([Ns+1]);
76     Ns=Ns+1;
77     #粮食集合3数量
78     model.grain3=set(range(Ns+1,Ns+grain_n3+1));
79     Ns=Ns+grain_n3;
80     #粮食集合4数量
81     model.grain4=set(range(Ns+1,Ns+grain_n4+1));
82     Ns=Ns+grain_n4;
83     #粮食集合5数量
84     model.grain5=set(range(Ns+1,Ns+grain_n5+1));
85     #豆类
86     model.grain6=set([1,2,3,4,5]);
87     model.grain7=set([17,18,19]);
88     ##定义地块类型集合
89     #干旱地
90     model.Dry=set(range(1,Dry_N+1));

```

```

91     Ns=Dry_N;
92     #梯田
93     model.Terr=set(range(Ns+1,Ns+Terr_N+1));
94     Ns=Ns+Terr_N;
95     #山坡地
96     model.Hill=set(range(Ns+1,Ns+Hill_N+1));
97     Ns=Ns+Hill_N;
98     #水浇地
99     model.Irri=set(range(Ns+1,Ns+Irri_N+1));
100    Ns=Ns+Irri_N;
101    #普通大棚
102    model.Comm=set(range(Ns+1,Ns+Comm_N+1));
103    Ns=Ns+Comm_N+1;
104    #智慧大棚
105    model.Smart=set(range(Ns,Ns+Smart_N));
106    ##定义决策变量
107    #第k年编号为j的土地种植i号作物的面积为 $x(i,j,k)$  为非负实数
108    model.I=set(range(1,N+1));
109    model.J=set(range(1,S+1));
110    model.K=set(range(1,J+1));
111    model.T=set(range(1,t_max+1));
112    model.x=pyo.Var(model.I,model.J,model.K,model.T,domain=pyo
.NonNegativeReals);
113    #第k年编号为j的土地种植i号作物的为 $y(i,j,k)=1$  为非负实数
114    model.y=pyo.Var(model.I,model.J,model.K,model.T,domain=pyo
.Binary);
115
116    ## 约束条件
117    model.con=pyo.ConstraintList();
118    #单季度约束
119    #def contirans(model):
120    for k in t:
121        ##单季度约束
122        #干旱地
123        for j in model.Dry:

```



```

124         #第一度季度至少种一种
125         model.con.add(sum(model.y[i,j,1,k] for i in model.
grain1)>=1);
126     #梯田
127     for j in model.Terr:
128         #第一度季度至少种一种
129         model.con.add(sum(model.y[i,j,1,k] for i in model
.grain1)>=1);
130     #山坡地
131     for j in model.Hill:
132         #第一度季度至少种一种
133         model.con.add(sum(model.y[i,j,1,k] for i in model
.grain1)>=1);
134     for j in model.J:
135         #1-15号作物不能种在第二季
136         model.con.add(sum(model.y[i,j,2,k] for i in model.
grain1)==0);
137         # i-15无论什么作物都要满足面积约束
138         model.con.add(sum(model.x[i,j,1,k]*model.y[i,j,1,k
] for i in model.grain1)<=A[j-1]);
139         model.con.add(sum(model.x[i,j,2,k]*model.y[i,j,2,k
] for i in model.grain1)<=A[j-1]);
140     for j in model.J-(model.Dry|model.Terr|model.Hill):
141         #1-15号作物不能种在除干旱地，梯田，山坡地外的地
142         model.con.add(sum(model.y[i,j,1,k] for i in model.
grain1)==0);
143
144     #水稻不能种在除水浇地以外的任何地方
145     for j in model.J-model.Irri:
146         model.con.add(sum(model.y[i,j,2,k] for i in model.
grain2)==0);
147         model.con.add(sum(model.y[i,j,1,k] for i in model.
grain2)==0);
148     #物种17-34不能种在干旱地，梯田，山坡地 和
149     for j in model.Dry|model.Terr|model.Hill:

```

```

150         model.con.add(sum(model.y[i,j,2,k] for i in model.
grain3)==0);
151         model.con.add(sum(model.y[i,j,1,k] for i in model.
grain3)==0);
152         ##水浇地
153
154         for j in model.Irri:
155             #水稻约束
156             model.con.add(sum(model.x[i,j,1,k]*model.y[i,j,1,
k] for i in model.grain2)<=A[j-1]);
157             model.con.add(sum(model.x[i,j,2,k]*model.y[i,j,2,
k] for i in model.grain2)<=A[j-1]);
158             #水浇地只能种植一季
159             model.con.add(sum(model.y[i,j,2,k] for i in model
.grain2)==0);
160             #水稻要么种植要么不种
161             model.con.add(sum(model.y[i,j,1,k] for i in model
.grain2)<=1);
162
163             ##分两季种植
164             #第一季
165             model.con.add(sum(model.x[i,j,1,k] for i in model
.grain3)<=A[j-1]);
166             model.con.add(sum(model.x[i,j,2,k] for i in model
.grain3)<=A[j-1]);
167             model.con.add(sum(model.y[i,j,2,k] for i in model
.grain3)==0);
168             model.con.add(sum(model.y[i,j,1,k] for i in model
.grain3)>=1);
169             #第二季
170             model.con.add(sum(model.x[i,j,2,k] for i in model
.grain4)<=A[j-1]);
171             model.con.add(sum(model.x[i,j,1,k] for i in model
.grain4)<=A[j-1]);
172             model.con.add(sum(model.y[i,j,1,k] for i in model

```

```

        .grain4)==0);
173         #大白菜和红白萝卜选一种
174         model.con.add(sum(model.y[i,j,2,k] for i in model
        .grain4)==1);
175         #要么种植水稻 要么种植两季
176         sum1=sum(model.y[i,j,1,k] for i in model.grain3);
177         sum2=sum(model.y[i,j,2,k] for i in model.grain4);
178         for i in model.grain2:
179             model.con.add(model.y[i,j,1,k]+(sum1+sum2)
        /2<=1);
180         #所有第一季不能种值35-37
181         for j in model.J:
182             model.con.add(sum(model.y[i,j,1,k] for i in model.
        grain4)==0);
183         #除水浇地外所有第二季不能种值35-37
184         for j in model.J-model.Irri:
185             model.con.add(sum(model.y[i,j,2,k] for i in model.
        grain4)==0);
186         #只有普通大棚第一季
187         ##普通大棚
188         for j in model.Comm:
189             #第一季
190             model.con.add(sum(model.x[i,j,1,k] for i in model.
        grain3)<=A[j-1]);
191             model.con.add(sum(model.x[i,j,2,k] for i in model.
        grain3)<=A[j-1]);
192             model.con.add(sum(model.y[i,j,2,k] for i in model.
        grain3)==0);
193             model.con.add(sum(model.y[i,j,1,k] for i in model.
        grain3)>=1);
194
195         #model.con.add(sum(model.y[i,j,k] for i in grain3)
        <=1);
196         #第二季 食用菌中最少一种
197         model.con.add(sum(model.x[i,j,1,k] for i in model.

```

```

grain5)<=A[j-1]);
198         model.con.add(sum(model.x[i,j,2,k] for i in model.
grain5)<=A[j-1]);
199         model.con.add(sum(model.y[i,j,1,k] for i in model.
grain5)==0);
200         model.con.add(sum(model.y[i,j,2,k] for i in model.
grain5)>=1);
201         #所有第一季不能种值38-41
202         for j in model.J:
203             model.con.add(sum(model.y[i,j,1,k] for i in model.
grain5)==0);
204             #除普通大棚外所有第二季不能种值38-41
205             for j in model.J-model.Comm:
206                 model.con.add(sum(model.y[i,j,2,k] for i in model.
grain5)==0);
207             ##智慧大棚
208             for j in model.Smart:
209                 #第一季度
210                 model.con.add(sum(model.x[i,j,1,k] for i in model.
grain3)<=A[j-1]);
211                 #第二季
212                 model.con.add(sum(model.x[i,j,2,k] for i in model.
grain3)<=A[j-1]);
213                 #第一季度
214                 model.con.add(sum(model.y[i,j,1,k] for i in model.
grain3)>=1);
215                 #第二季
216                 model.con.add(sum(model.y[i,j,2,k] for i in model.
grain3)>=1);
217             #每种作物不能重茬种植
218             for i in model.I:
219                 for j in model.J:
220                     #季数不能重茬
221                     model.con.add(model.y[i,j,1,k]+model.y[i,j,2,k
]<=1);

```

```

222         #年份·不能重茬
223         if k<t_max:
224             model.con.add(model.y[i,j,1,k]+model.y[i,j
,1,k+1]<=1);
225             model.con.add(model.y[i,j,2,k]+model.y[i,j
,2,k+1]<=1);
226         for i in model.I:
227             for j in model.J:
228                 model.con.add(model.x[i,j,1,k]+Big_M*(1-model.
y[i,j,1,k])>=min_S);
229                 model.con.add(model.x[i,j,2,k]+Big_M*(1-model.
y[i,j,2,k])>=min_S);
230                 # 每块地在每一季度 每一年的产量不能超过dmand量
231                 # if x*d>dmands ->y=1
232                 model.con.add(dmand[i-1,j-1,0]+Big_M*(1-model.
y[i,j,1,k])>=model.x[i,j,1,k]*d[i-1,j-1,0]);
233                 model.con.add(dmand[i-1,j-1,1]+Big_M*(1-model.
y[i,j,2,k])>=model.x[i,j,2,k]*d[i-1,j-1,1]);
234                 #每块地在每一季度 每一年只能用一次
235                 #每隔3年种植一次豆类1 2 3 4 5 单季度
236                 for j in model.Dry|model.Terr|model.Hill:
237                     if k+3<=t_max:
238                         model.con.add(sum(model.y[i,j,1,k]+model.y[i,j
,1,k+1]+model.y[i,j,1,k+2] for i in model.grain6)>=1);
239                 #水浇地第一季度普通（智慧大棚）
240                 for j in model.Irri|model.Comm|model.Smart:
241                     if k+3<=t_max:
242                         model.con.add(sum(model.y[i,j,1,k]+model.y[i,j
,1,k+1]+model.y[i,j,1,k+2] for i in model.grain7)>=1);
243                 #智慧大棚第二季度
244                 for j in model.Smart:
245                     if k+3<=t_max:
246                         model.con.add(sum(model.y[i,j,2,k]+model.y[i,j
,2,k+1]+model.y[i,j,2,k+2] for i in model.grain7)>=1);
247

```

```

248         #return model;
249     ## 定义目标函数
250     #c 物种 地块类型 季数
251     val=sum((model.y[i,j,h,k]*model.x[i,j,h,k])*d[i-1,j-1,h
-1]*c[i-1,j-1,h-1]-model.y[i,j,h,k]*model.x[i,j,h,k]*q[i-1,j
-1,h-1] \
252         for k in model.T for i in model.I for j in model.J
for h in model.K);
253     model.obj=pyo.Objective(expr=val,sense=pyo.maximize);
254     solver=pyo.SolverFactory('gurobi',solve_io='python').solve
(model,tee=False);
255     data=np.zeros((N,S,J,t_max))
256     print(pyo.value(model.obj));
257     for k in model.T:
258         for i in model.I:
259             for j in model.J:
260                 for h in model.K:
261                     data[i-1,j-1,h-1,k-1]=pyo.value(
model.x[i,j,h,k]);
262     return data
263 data = OptimalProblem()
264 listx=['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']
265 listy=[[ 'A{}'.format(i) for i in range(1,7)],['B{}'.format(i)
for i in range(1,15)],['C{}'.format(i) for i in range(1,7)
],['D{}'.format(i) for i in range(1,9)],\
266     ['E{}'.format(i) for i in range(1,17)],['F{}'.format(i)
for i in range(1,5)],['D{}'.format(i) for i in range(1,9)
],['E{}'.format(i) for i in range(1,17)],['F{}'.format(i)
for i in range(1,5)]];
267

```

```

268 listy=list(chain(*listy));
269 def save_res(data):
270     for t in range(0,t_max):
271         datax=data[:, :, 0, t];
272         datax=datax.T;
273         dat=pd.DataFrame(datax,columns=listx);
274         datax=data[:, 25:-1, 1, t];
275         datax=datax.T;
276         da2=pd.DataFrame(datax,columns=listx);
277         datay=pd.concat([dat, da2], axis=0);
278         datay['地块']=listy;
279         datay.set_index('地块', inplace=True);
280         with pd.ExcelWriter('result.xlsx', mode='a') as tf:
281             #删除所有表单
282             '''
283             excel_file=pd.ExcelFile('result.xlsx');
284             for x in excel_file.sheet_names:
285                 excel_file.parse(x).to_excel(tf, sheet_name=x,
index=False);
286             tf.save();
287             '''
288         datay.to_excel(tf, sheet_name='{}'.format(2024+t));
289 save_res(data);

```