

模型建⽴

1. 问题分析

。该问题是⼀个典型的优化决策问题 ，我们需要决策每块地每年每季种植什么作物 ，以及确定种 植⾯积。

⽬标函数是收益最⼤化 ，收益=销售额-成本=种植亩数 × （亩产 × 价格 - 成本） 约束条件有以下的业务约束：

▪ 种植⾯积约束：每块地上种植作物的⾯积总和不能超过地块的⾯积。

▪ ⽔浇地种植⽔稻约束：⼀旦种植了⽔稻 ，那么就不能种植其他作物 ，并且第⼆季也不能种植 其他作物

▪ 不能重种的约束：⼀块地不能连续种植两季同⼀种作物

▪ 三年内⾄少种植⼀次⾖类作物约束

▪ 种植不能太分散： 同⼀作物不能种植在太多个地块

▪ 作物在⼀个地块的种植⾯积不能太⼩： 可以⽤绝对值 ，也可以⽤相对值

。第⼀⼩问：基于价格、销量、成本、产能都不随时间（年） 变化的理想情况考虑 。第⼆⼩问：基于价格、销量、成本、产能随时间（年）独⽴变化的理想情况考虑 。第三⼩问：基于价格、销量、成本、产能随时间（年）相关变化的实际情况考虑

2. 模型假设

。2023年的销售量按照地块类型、作物类型和季节三个维度来统计 ，不同地块类型的同⼀种作 物 ，因为价格不同 ，不会算作同⼀种作物来同⼀统计销量。

。2023年没有种植的作物 ，2024年如果种植的话 ，默认其销量是⽆穷⼤ ,或者为其他类型地块该类 作物的平均销量

3. 模型建模

3.1 集合定义

根据附件1的地块类型和作物类型以及季来划分集合 ，为后续描述⽅便

A : 平旱地 ， A = {*Ai* }=1

B : 梯⽥ ， B = {*B*1,*B*2 , ⋯ ,*B*14 }

C : ⼭坡地 ， C = {*C*1, *C*2 , ⋯ , *C*6 }

D : ⽔浇地 ， D = {*D*1,*D*2 , ⋯ ,*D*8 } E : 普通⼤棚 ， E = {*E*1,*E*2 , ⋯ ,*E*16 } F : 智慧⼤棚 ， F = {*F*1,*F*2 , ⋯ ,*F*4 }

G : 粮⻝（ ⾖类） ， G = {1, 2, ⋯ , 5} （作物编号 ，也可以⽤作物名称）

H : 粮⻝ ， H = {6, 7, ⋯ , 15}

M : 粮⻝⽔浇地 ， M = {16}

N : 蔬菜⾖类 ， N = {17, 18, 19}

O : 蔬菜⽔浇地第⼀季 ， O = {20, 21, ⋯ , 34}

P : 蔬菜⽔浇地第⼆季 ， P = {35, 36, 37}

Q : ⻝⽤菌普通⼤棚第⼆季 ， Q = {38, 39,40,41} K : 季 ， K = {1, 2} = K1 ∪ K2

Y ：年 ， Y = {2024, 2025, ⋯ , 2030} I = A ∪ B ∪ C ∪ D ∪ E ∪ F

J = G ∪ H ∪ M ∪ N∪ O ∪ P ∪ Q

决策变量的下标集合属于稀疏集合 ，根据附件1来组合

平旱地 ，梯⽥和⼭坡地第⼀季种植粮⻝⾖类和粮⻝ ，下同 V1 = (A ∪ B ∪ C) × (G ∪ H) × K1 ,

V2 = D × M × K1

V3 = (D ∪ E ∪ F) × (N ∪ O) × K1 V4 = F × (N ∪ O) × K2

V5 = D × P × K2 V6 = E × Q × K2 

地块季节集合 ，表达⾯积约束 W1 = I × K1

W2 = (D ∪ E ∪ F) × K2 W = W1 ∪ W2

R = {*A*,*B* , *C* ,*D*,*E*,*F* }

3.2 输⼊参数

*ai* : 地块 *i* 的⾯积 ， ∀*i* ∈ I

*θ* : 同⼀个作物最多种植的地块数

*δ* : 作物种植⾯积占总地⾯积的最⼩⽐例

*pI* ,*j* ,*k* ,*t* : 地块类型 *I* 作物 *j* 季节 *k* 年 *t* 的价格, *I* ∈ R *cI* ,*j* ,*k* ,*t* : 地块类型 *I* 作物 *j* 季节 *k* 年 *t* 的成本, *I* ∈ R *qI* ,*j* ,*k* ,*t* : 地块类型 *I* 作物 *j* 季节 *k* 年 *t* 的销量, *I* ∈ R *rI* ,*j* ,*k* ,*t* : 地块类型 *I* 作物 *j* 季节 *k* 年 *t* 的亩产, *I* ∈ R

3.3 决策变量

*xi*,*j* ,*k* ,*t* : 种植亩数 ，连续变量 ， ∀(*i*,*j* , *k*) ∈ V ,*t* ∈ Y

*yi*,*j* ,*k* ,*t* : 是否种植 ，0-1变量 ， ∀(*i*,*j* , *k*) ∈ V ,*t* ∈ Y ∪ {2023}

*λI* ,*j* ,*k* ,*t* : 0-1变量 ，线性化⽬标函数 ， ∀*I* ∈ R,*j* ∈ J, *k* ∈ K,*t* ∈ Y

3.4 约束条件

i. ⾯积约束

1. 种植⾯积约束

 *xi*,*j* ,*k* ,*t* ≤ *ai* , ∀(*i*, *k*) ∈ W ,*t* ∈ Y

2. 种植⾯积不能太⼩

*xi*,*j* ,*k* ,*t* ≥ *aiyi*,*j* ,*k* ,*t δ* , ∀(*i*,*j* , *k*) ∈ V ,*t* ∈ Y

(1)

(2)

3. 不种植 ，种植⾯积为0

*xi*,*j* ,*k* ,*t* ≤ *aiyi*,*j* ,*k* ,*t* , ∀(*i*,*j* , *k*) ∈ V ,*t* ∈ Y

ii. ⽔浇地种植⽔稻约束

1. 如果⽔浇地种植⽔稻 ，则该地块第⼀季不能种植其他作物

*yi*,*j* ,*k* ,*t* + *yi*,*τ* ,*k* ,*t* ≤ 1, ∀*i* ∈ D ,*j* ∈ M, *k* ∈ K1 ,*t* ∈ Y , *τ* ∈ N ∪ O

2. 如果⽔浇地种植⽔稻 ，则该地块第⼆季不能种植其他作物

*yi*,*j* ,*k* ,*t* + *yi*,*τ* ,*α*,*t* ≤ 1, ∀*i* ∈ D ,*j* ∈ M, *k* ∈ K1 ,*t* ∈ Y , *τ* ∈ P , *α* ∈ K2

iii. 作物不能重茬种植 ，否则减产约束

1. 粮⻝不能重茬种植

*yi*,*j* ,*k* ,*t* + *yi*,*j* ,*k* ,*t*+1 ≤ 1, ∀(*i*,*j* , *k*) ∈ V1 ∪ V2 ,*t* ∈ {2023} ∪ Y/{2030}

2. 智慧⼤棚第⼀季和第⼆季不能重茬

*yi*,*j* ,*k* ,*t* + *yi*,*j* ,*τ* ,*t* ≤ 1, ∀*i* ∈ F,*j* ∈ N ∪ O, *k* ∈ K1 , *τ* ∈ K2 ,*t* ∈ Y

3. 智慧⼤棚第⼆季和来年第⼀季不能重茬

(3)

(4)

(5)

(6)

(7)

*yi*,*j* ,*k* ,*t* + *yi*,*j* ,*τ* ,*t*+1 ≤ 1, ∀*i* ∈ F,*j* ∈ N ∪ O, *k* ∈ K2 , *τ* ∈ K1 ,*t* ∈ {2023} ∪ Y/{2030}(8

iv. 三年内⾄少种植⼀次⾖⼦

1. 平旱地、梯⽥和⼭坡地三年⾄少种⼀次⾖类

Σ *yi*,*j* ,*k* ,*τ* ≥ 1, ∀*i* ∈ A ∪ B ∪ C, *k* ∈ K1 ,*t* ∈ {2023} ∪ Y/{2029, 2030} (9)

2. ⽔浇地、普通⼤棚三年⾄少种⼀次⾖类

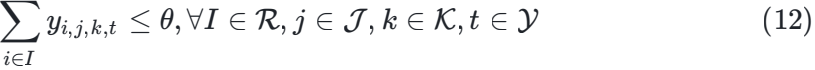
Σ *yi*,*j* ,*k* ,*τ* ≥ 1, ∀*i* ∈ D ∪ E , *k* ∈ K1 ,*t* ∈ {2023} ∪ Y/{2029, 2030} (10)

3. 智慧⼤棚三年⾄少种⼀次⾖类

Σ *yi*,*j* ,*k* ,*τ* ≥ 1, ∀*i* ∈ F,*t* ∈ {2023} ∪ Y/{2029, 2030}

(11)

v. 种植地不能太分散 ，基于假设 ，我们认为每种不同地块上的同⼀种作物不能太分散



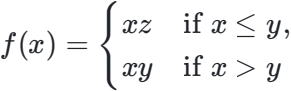
3.5 ⽬标函数

利润最⼤化

max ,*t*,*j*  *f* (*rI* ,*j* ,*k* ,*t*  *xi*,*j* ,*k* ,*t*,*qI* ,*j* ,*k* ,*t*,*pI* ,*j* ,*k* ,*t* ) −  *xi*,*j* ,*k* ,*tcI* ,*j* ,*k* (13)

可以看到每种⼟地类型每年每个季节每个作物的销售额是产量 ，销量和价格的函数 *f*

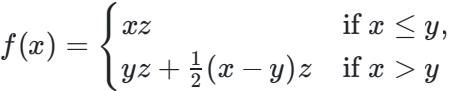
在此局部⽤ *x* 表⽰产量 ，⽤ *y* 表⽰销量 ，⽤ *z* 表⽰价格 ，则根据题⽬ ，有如下两种情况： . 对于滞销的情形

 (14)

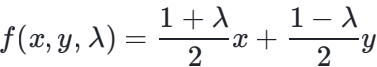
将 *z* 提出 ，引⼊0-1变量 *λ* ,将公式线性化如下

*f* (*x*,*y* , *λ*) = (*λx* + (1 − *λ*)*y*)*z* (15)

. 对于售价减半的情形

 (16)

将 *z* 提出 ，引⼊0-1变量 *λ* ,将公式线性化如下

 (17)

. 需要对 *λ* 进⾏约束 ，满⾜⼀下指⽰性约束(求解器添加约束)

*λ* = 1 ⇒ *x* ≤ *y* (18)

*λ* = 0 ⇒ *x* ≥ *y* (19)

. 或者转为⼤M约束

*x* − *y* ≤ *M*(1 − *λ*) (20)

*y* − *x* ≤ *Mλ* (21)

4. 模型求解

代码如下：

1 from itertools import product

2 import gurobipy as gp

3 import pandas as pd

4 from openpyxl import load\_workbook

5 import os

6 import random

7 from datetime import datetime

8

9 current\_time = datetime.now()

10 timestamp = current\_time.timestamp()

11 random.seed(int(timestamp))

12

13 *# 0. 读⼊⽂件*

14 data\_1 = pd.read\_excel('附件1.xlsx', sheet\_name="乡村的现有耕地")

15 data\_2 = pd.read\_excel('附件2.xlsx', sheet\_name="2023年的农作物种植情 况").fillna(method='ffill') *# 处理合并单元格的问题*

16 data\_3 = pd.read\_excel('附件2.xlsx', sheet\_name="2023年统计的相关数据")

17 trans\_dict = {"A": "平旱地 ", "B": "梯⽥ ", "C": "⼭坡地 ", "D": "⽔浇地 ", "E": "普通 ⼤棚 ", "F": "智慧⼤棚"}

18

19 *# 1. 输⼊参数* 20 *# 1.1 销量*

21 q\_0 = {(data\_2["种植地块"][i], data\_2["作物编号"][i],

22 2 if data\_2["种植季次"][i] == "第⼆季 " else 1):

23 data\_2["种植⾯积/亩"][i]

24 for i in range(len(data\_2))}

25 q = {}

26 for (i, j, k), v in q\_0.items():

27 if trans\_dict[i[0]] in q:

28 q[trans\_dict[i[0]], j, k] += v

29 else:

30 q[trans\_dict[i[0]], j, k] = v 31

32 data\_4 = {(data\_3["地块类型"][i].strip(), data\_3["作物编号"][i], 33 2 if data\_3["种植季次"][i] == "第⼆季 " else 1):

34 [data\_3["亩产量/⽄"][i], data\_3["种植成本/(元/亩)"][i], data\_3["销售

单价/(元/⽄)"][i]]

35 for i in range(len(data\_3))}

36 *# 1.2 亩产，* *成本和价格*

37 \_, r, c, p = gp.multidict(data\_4)

38 pl = {}

39 pu = {}

40 for k, v in p.items():

41 try:

42 str1 = v.split('-')

43 pl[k] = float(str1[0])

44 pu[k] = float(str1[1])

45 except:

46 pass 47 *# 1.3 地块⾯积*

48 a = {data\_1["地块名称"][i]: data\_1["地块⾯积/亩"][i]

49 for i in range(len(data\_1))} 50

51 *# 2. 定义集合* 52 *# 2.1 作物*

53 LDR = ["黄⾖ ", "⿊⾖ ", "红⾖ ", "绿⾖ ", "爬⾖"] *# 粮⻝（⾖类）*

54 LD = list(range(1, 6))

55 LR = ["⼩麦 ", "⽟⽶ ", "⾕⼦ ", "⾼粱 ", "⿉⼦ ", "荞麦 ", "南⽠ ", "红薯 ", "莜麦 ", "⼤ 麦"] *# 粮⻝*

56 L = list(range(6, 16))

57 LSR = ["⽔稻"] *# 粮⻝⽔浇地*

58 LS = list(range(16, 17))

59 SDR = ["豇⾖ ", "⼑⾖ ", "芸⾖"] *# 蔬菜（⾖类）*

60 SD = list(range(17, 20))

61 S\_1R = ["⼟⾖ ", "西红柿 ", "茄⼦ ", "菠菜 ", "⻘椒 ", "菜花 ", "包菜 ", "油麦菜 ", "⼩⻘菜 ",

62 "黄⽠ ", "⽣菜 ", "辣椒 ", "空⼼菜 ", "黄⼼菜 ", "芹菜"]

63 S\_1 = list(range(20, 35))

64 S\_2R = ["⼤⽩菜 ", "⽩萝 ⼘ ", "红萝 ⼘"]

65 S\_2 = list(range(35, 38))

66 SJR = ["榆黄菇 ", "⾹菇 ", "⽩灵菇 ", "⽺肚菌"]

67 SJ = list(range(38, 42))

68 J = LD + L + LS + SD + S\_1 + S\_2 + SJ

69 J1 = LDR + LR + LSR + SDR + S\_1R + S\_2R + SJR

70 J\_d = dict(zip(J1, J)) 71 *# 2.2 地块*

72 A = ["A" + str(i) for i in range(1, 7)] *# ⼲旱*

73 B = ["B" + str(i) for i in range(1, 15)] *# 梯⽥*

74 C = ["C" + str(i) for i in range(1, 7)] *# ⼭坡地*

75 D = ["D" + str(i) for i in range(1, 9)] *# ⽔浇地*

76 E = ["E" + str(i) for i in range(1, 17)] *# 普通⼤棚*

77 F = ["F" + str(i) for i in range(1, 5)] *# 智慧⼤棚*

78 trans\_dict\_2 = {"平旱地 ": A, "梯⽥ ": B, "⼭坡地 ": C, "⽔浇地 ": D, "普通⼤棚 ": E, "智慧⼤棚 ": F}

79 I = A + B + C + D + E + F 80 *# 2.3 年份*

81 Y = list(range(2024, 2031)) 82 *# 2.4 季节*

83 KR = ["第⼀季 ", "第⼆季"]

84 K = [1, 2]

85 *# 2.5 集合组合*

86 V\_1 = list(product(A + B + C, LD + L, [K[0]], Y))

87 V\_2 = list(product(D, LS, [K[0]], Y))

88 V\_3 = list(product(D + E + F, SD + S\_1, [K[0]], Y))

89 V\_4 = list(product(F, SD + S\_1, [K[1]], Y))

90 V\_5 = list(product(D, S\_2, [K[1]], Y))

91 V\_6 = list(product(E, SJ, [K[1]], Y))

92 V = V\_1 + V\_2 + V\_3 + V\_4 + V\_5 + V\_6 93

94 W1 = product(A + B + C + D + E + F, [K[0]], Y)

95 W2 = product(D + E + F, [K[1]], Y)

96 W = list(W1) + list(W2) 97

98 II = list(trans\_dict.values()) 99

100 *# 3. 处理参数*

101

102 *# 3.1 销量处理*

103 sale\_data = {}

104 sale\_data\_2023 = {i: q[i] \* r[i]

105 for i in q}

106 for idx, t in enumerate(Y):

107 *# i: ⼟地类型，* *j: 作物, k: 季节*

108 *# 统⼀：random\_number = random.uniform(-0.05, 0.05)*

109 for i, j, k in q:

110 random\_number = random.uniform(-0.05, 0.05)

111 if idx == 0:

112 sale\_data[i, j, k, t] = sale\_data\_2023[i, j, k] \* (1 +

random\_number)

113 else:

114 sale\_data[i, j, k, t] = sale\_data[i, j, k, Y[idx - 1]] \* (1 +

random\_number)

115 *# ⼩麦* *和* *⽟⽶* *未来* *的* *预期销售量有增⻓的趋势，平均年增⻓率介于*

116 for i, j, k in q:

117 if j not in [J\_d["⼩麦"], J\_d["⽟⽶"]]:

118 continue

119 random\_number = random.uniform(0.05, 0.1)

120 if idx == 0:

121 sale\_data[i, j, k, t] = sale\_data\_2023[i, j, k] \* (1 +

random\_number)

122 else:

123 sale\_data[i, j, k, t] = sale\_data[i, j, k, Y[idx - 1]] \* (1 +

random\_number)

124

125 *# 3.2 亩产处理*

126 mu\_data = {}

127 mu\_data\_2023 = {i: r[i]

128 for i in r}

129 for idx, t in enumerate(Y):

130 *# i: ⼟地类型，* *j: 作物, k: 季节*

131 *# 统⼀：random\_number = random.uniform(-0.05, 0.05)*

132 for i, j, k in r:

133 random\_number = random.uniform(-0.1, 0.1)

134 if idx == 0:

135 mu\_data[i, j, k, t] = mu\_data\_2023[i, j, k] \* (1 + random\_number)

136 else:

137 mu\_data[i, j, k, t] = mu\_data[i, j, k, Y[idx - 1]] \* (1 +

random\_number)

138

139 *# 3.3 成本处理*

140 ch\_data = {}

141 ch\_data\_2023 = {i: c[i]

142 for i in c}

143 for idx, t in enumerate(Y):

144 *# i: ⼟地类型，* *j: 作物, k: 季节*

145 *# 统⼀：random\_number = random.uniform(-0.05, 0.05)*

146 for i, j, k in c:

147 random\_number = random.uniform(0, 0.05)

148 if idx == 0:

149 ch\_data[i, j, k, t] = ch\_data\_2023[i, j, k] \* (1 + random\_number)

150 else:

151 ch\_data[i, j, k, t] = ch\_data[i, j, k, Y[idx - 1]] \* (1 +

random\_number)

152

153 *# 3.4 价格处理*

154 jia\_data\_l = {}

155 jia\_data\_2023\_l = {i: pl[i]

156 for i in pl}

157 jia\_data\_u = {}

158 jia\_data\_2023\_u = {i: pu[i]

159 for i in pl}

160 for idx, t in enumerate(Y):

161 *# i: ⼟地类型，* *j: 作物, k: 季节*

162 *# 统⼀：random\_number = random.uniform(-0.05, 0.05)*

163 for i, j, k in pl:

164 if j in SD + S\_1 + S\_2:

165 random\_number = random.gauss(0.05, 0.01)

166 if idx == 0:

167 jia\_data\_l[i, j, k, t] = jia\_data\_2023\_l[i, j, k] \* (1 +

random\_number)

168 else:

169 jia\_data\_l[i, j, k, t] = jia\_data\_l[i, j, k, Y[idx - 1]] \* (1

+ random\_number)

170

171 if idx == 0:

172 jia\_data\_u[i, j, k, t] = jia\_data\_2023\_u[i, j, k] \* (1 +

random\_number)

173 else:

174 jia\_data\_u[i, j, k, t] = jia\_data\_u[i, j, k, Y[idx - 1]] \* (1

+ random\_number)

175 if j in SJ:

176 random\_number = random.uniform(0.01, 0.05)

177 if j == J\_d["⽺肚菌"]:

178 random\_number = 0.05

179 if idx == 0:

180 jia\_data\_l[i, j, k, t] = jia\_data\_2023\_l[i, j, k] \* (1 -

random\_number)

181 else:

182 jia\_data\_l[i, j, k, t] = jia\_data\_l[i, j, k, Y[idx - 1]] \* (1

- random\_number)

183

184 if idx == 0:

185 jia\_data\_u[i, j, k, t] = jia\_data\_2023\_u[i, j, k] \* (1 -

random\_number)

186 else:

187 jia\_data\_u[i, j, k, t] = jia\_data\_u[i, j, k, Y[idx - 1]] \* (1

- random\_number)

188 if j in L + LD + LS:

189 jia\_data\_l[i, j, k, t] = jia\_data\_2023\_l[i, j, k]

190 jia\_data\_u[i, j, k, t] = jia\_data\_2023\_u[i, j, k]

191

192 *# 4. 建⽴模型*

193 m = gp.Model("cumcm\_2024") 194

195 *# 4.1 决策变量*

196 x = m.addVars(V, vtype=gp.GRB.CONTINUOUS, lb=0, name="x")

197 y = m.addVars(V, vtype=gp.GRB.BINARY, lb=0, name="y")

198 x\_s = m.addVars(pl, Y, vtype=gp.GRB.CONTINUOUS, lb=0, name="xs")

199 l\_s = m.addVars(pl, Y, vtype=gp.GRB.BINARY, name="ls") 200

201 *# 4.2 约束条件* 202 *# 1. ⾯积约束* 203

204 m.addConstrs((x[i, j, k, t] <= a[i] \* y[i, j, k, t]

205 for i, j, k, t in V), "C11")

206

207 m.addConstrs((x.sum(i, "\*", k, t) <= a[i]

208 for (i, k, t) in W), "C12")

209

210 *# 2. ⽔稻每年⼀季节或者两季节蔬菜*

211

212 m.addConstrs((y[i, LS[0], K[0], t] + y[i, j, K[0], t] <= 1

213 for i in D

214 for j in S\_1 + SD

215 for t in Y), "C21")

216

217

218 m.addConstrs((y[i, LS[0], K[0], t] + y[i, j, K[1], t] <= 1

219 for i in D

220 for j in S\_2

221 for t in Y), "C22")

222

223

224 *# 3. 重种约束*

225 m.addConstrs((y[i, j, K[0], Y[t]] + y[i, j, K[0], Y[t + 1]] <= 1

226 for i in A + B + C

227 for j in LD + L

228 for t in range(len(Y) - 1)), "C31")

229

230 m.addConstrs((y[i, j, K[0], Y[t]] + y[i, j, K[0], Y[t + 1]] <= 1

231 for i in D

232 for j in LS

233 for t in range(len(Y) - 1)), "C32")

234

235

236 m.addConstrs((y[i, j, K[0], Y[t]] + y[i, j, K[1], Y[t]] <= 1

237 for i in F

238 for j in SD + S\_1

239 for t in range(len(Y))), "C33")

240

241 m.addConstrs((y[i, j, K[1], Y[t]] + y[i, j, K[0], Y[t + 1]] <= 1

242 for i in F

243 for j in SD + S\_1

244 for t in range(len(Y) - 1)), "C34")

245

246

247 *# 4. 所有⼟地三年内⾄少种植—次⾖类作物*

248 m.addConstrs((gp.quicksum(y[i, j, K[0], Y[tau]]

249 for j in LD

250 for tau in range(t, t + 3)) >= 1

251 for i in A + B + C

252 for t in range(0, len(Y) - 2)), "C41")

253

254 m.addConstrs((gp.quicksum(y[i, j, K[0], Y[tau]]

255 for j in SD

256 for tau in range(t, t + 3)) >= 1

257 for i in D + E

258 for t in range(0, len(Y) - 2)), "C42")

259

260 m.addConstrs((gp.quicksum(y[i, j, k, Y[tau]]

261 for k in K

262 for j in SD

263 for tau in range(t, t + 3)) >= 1

264 for i in F

265 for t in range(0, len(Y) - 2)), "C43")

266

267 *# 5. 种植地不能太分散*

268 theta = 5

269 for idx, I\_ in enumerate(list(trans\_dict\_2.values())):

270 m.addConstrs((gp.quicksum(y.get((i, j, k, t), 0)

271 for i in I\_) <= theta

272 for j in J

273 for k in K

274 for t in Y), "C5"+str(idx))

275

276

277 *#*

278 *# 6. 单个地块* *（含⼤棚* *种植* *的* *⾯积不宜太⼩，等等*

279 delta = 1 / 3

280 m.addConstrs((x[i, j, k, t] >= a[i] \* y[i, j, k, t] \* delta

281 for i, j, k, t in V), "C6")

282

283

284 m.addConstrs((x\_s[I0, j, k, t] == l\_s[I0, j, k, t] \* x.sum(trans\_dict\_2[I0],

j, k, t) \* mu\_data[I0, j, k, t] +

285 (1 - l\_s[I0, j, k, t]) \* sale\_data.get((I0, j, k, t), 1e10)

286 for I0, j, k, t in x\_s), "C71")

287 m.addConstrs(((l\_s[I0, j, k, t] == 1) >> (

288 x.sum(trans\_dict\_2[I0], j, k, t) \* mu\_data[I0, j, k, t] <=

sale\_data.get((I0, j, k, t), 1e10))

289 for I0, j, k, t in x\_s), "C72")

290

291 m.addConstrs(((l\_s[I0, j, k, t] == 0) >> (

292 x.sum(trans\_dict\_2[I0], j, k, t) \* mu\_data[I0, j, k, t] >=

sale\_data.get((I0, j, k, t), 1e10))

293 for I0, j, k, t in x\_s), "C73")

294

295 *# 4.3 ⽬标函数*

296 obj = gp.LinExpr()

297 for t in Y:

298 for I0, j, k in pl:

299 if I0 == "平旱地 ":

300 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

301 if I0 == "梯⽥ ":

302 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

303 if I0 == "⼭坡地 ":

304 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

305 if I0 == "⽔浇地 ":

306 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

307 if I0 == "普通⼤棚 ":

308 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

309 if I0 == "智慧⼤棚 ":

310 p\_c = jia\_data\_l[I0, j, k, t] \* 0.75 + 0.25 \* jia\_data\_u[I0, j, k,

t]

311

312 c\_c = ch\_data[I0, j, k, t]

313 current = p\_c \* x\_s[I0, j, k, t] - c\_c \* x.sum(trans\_dict\_2[I0], j, k,

t)

314 obj += current 315

316 m.setObjective(obj, gp.GRB.MAXIMIZE) 317 *# 5. 模型求解*

318 m.write("model.lp")

319 m.setParam(gp.GRB.Param.MIPGap, 0.01)

320 m.setParam(gp.GRB.Param.TimeLimit, 120)

321 m.optimize()

322

323 *# 6. 结果输出*

324 xv = m.getAttr("X", x)

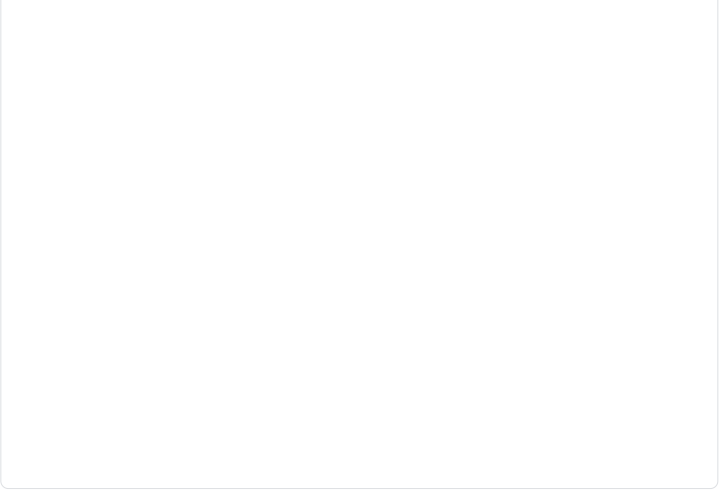
325 yv = m.getAttr("X", y)

326 *# for k, v in xv.items():*

327 *# if v < .001 or k[3] > 2024:*

328 *# continue*

329 *# print(k, ":", v)*



330

331 workbook = load\_workbook(r"附件3\result2.xlsx")

332 for t in Y:

333 data\_r = pd.read\_excel(r"附件3\result2.xlsx", sheet\_name=str(t))

334 *# writer = xlsxwriter.Workbook('existing\_file.xlsx', engine='openpyxl')*

335 worksheet\_writer = workbook[str(t)]

336 for index, row in data\_r.iterrows():

337 r\_data = []

338 for cindex, col\_name in enumerate(data\_r.columns[2:]):

339 cc\_name = col\_name.strip()

340 if (data\_r["地块名"][index], J\_d[cc\_name], K[0] if index <= 53 else K[1], t) in xv:

341 vv = xv[data\_r["地块名"][index], J\_d[cc\_name], K[0] if index <=

53 else K[1], t]

342 if vv > 0.001:

343 worksheet\_writer.cell(index + 2, 3 + cindex, vv) 344

345 file\_path = r"附件3\result2\_v1.xlsx"

346 if os.path.exists(file\_path):

347 os.remove(file\_path)

348 workbook.save(file\_path)

5. 结果分析

见[e result2\_v1.x lsx](https://qakpbhm3yjn.feishu.cn/wiki/Jul4wj8G4ixyD6kzxQvcpgRLnrd?fromScene=spaceOverview)