

2020 年“数维杯”大学生 数学建模竞赛论文

题 目 以家庭为单位的小规模多品种绿色农畜产品的 经营模式开发研究

摘 要

在全球化疫情蔓延和国民生活水平稳步提高的大形势下，居民对于绿色农畜产品的需求可能会迎来新的爆发期，在此背景下，问题 1 要求收集不同地区的主要农产品相关时间序列数据，并分析不同农产品预期人力成本以及收入的相关性；问题 2 和问题 3 分别要求给出大规模和小规模的农产品种植计划；问题 4 要求给出农牧渔混合经营的种植计划。最后，问题 5 要求给出农民的合作种植计划。针对于问题 1，收集数据之后采用皮尔森相关系数来获得人力成本与收入的相关性；对于问题 2 和问题 3 采用数学规划的方法得到问题的最优解。针对于问题 4，采用复杂网络理论建模，给出一个比较合适的定性规划；针对于问题 5，采取博弈论的方法求解。

针对于问题 1，对于第 1 小问，首先划分种植区域，将种植区域大致划分为六部分，搜集每个地区的数据后绘制统计图得到主要农作物并结合实际分析结果的合理性。对于第 2 小问，通过可视化个别产物的人力成本与收入的数据进行观察，发现数据之间有明显的线性关系，之后使用皮尔森相关系数对数据之间的线性相关程度进行定量分析。

针对问题 2，利用“能值理论”将“保证土地资源循环利用”这一抽象概念数学化，之后利用相应的数学规划模型进行求解。

针对问题 3，由于题目中各个概念均容易用数学概念描述，因此直接建立线性规划模型进行求解。

针对问题 4，由于资源的局限性，采用复杂网络进行分析，并给出本题的定性描述。

针对问题 5，采用博弈论的方法，给出合作养殖的策略。

关键字： 皮尔森相关系数; 能值理论; 线性规划; 复杂网络模型; 博弈论

目 录

1 问题重述	3
1.1 问题背景	3
1.2 需要解决的问题	3
1.2.1 问题一	3
1.2.2 问题二	3
1.2.3 问题三	3
1.2.4 问题四	3
1.2.5 问题五	3
2 问题分析	4
2.1 问题 1 分析	4
2.2 问题 2 分析	4
2.3 问题 3 分析	4
2.4 问题 4 分析	4
3 模型的假设	5
4 符号说明	5
5 问题一	5
5.1 各地区主要农作物的分析	5
5.2 相关性定性分析	7
5.3 相关性的定量分析	7
5.3.1 皮尔森相关系数	7
6 问题二	10
6.1 本题模型假设	10
6.2 本题符号说明	10
6.3 能值理论简介	10
6.3.1 能值理论的含义	10
6.4 资源循环利用系统的综合评价指标建立	11
6.4.1 指标体系的建立	11
6.4.2 指标建立依据与意义	12
6.4.3 指标体系与计算公式	12

6.4.4 数据的来源与处理	13
6.4.5 层次分析法计算权重	13
6.4.6 优化模型的建立	15
7 问题三	16
7.1 本题模型假设	16
7.2 本题符号说明	16
7.3 线性规划模型的建立	16
7.4 模型的求解	17
8 问题四	18
8.1 我国农牧渔混合经营现状分析	18
8.2 农牧渔复合生态系统结构的建立	18
8.2.1 总体结构	18
8.2.2 各部分结构的确立	19
9 问题五	20
9.1 相关定义	20
9.2 本题模型假设	21
9.3 模型的建立与求解	21
10 模型评价	23
10.1 模型的优点	23
10.2 模型的不足之处	23
参考文献	24
附 录	25

1 问题重述

1.1 问题背景

在全球化疫情蔓延和国民生活水平稳步提高大形势下，居民对于绿色农畜产品的需求可能会迎来新的爆发期。由于传统大规模种植或养殖模式存在着大量缺陷。同时国家提倡乡村振兴策略及解决各地区发展不充分不协调的问题的迫切需求。这些均预示着以家庭为单位或小规模的个性化种植与养殖计划有望成为国家新一轮的经济增长点。这种模式较为适合我国这一具有庞大人口规模及城镇化比率较高的国家加以推广。未来农村牧区的经营模式将会逐步走向两个极端，一类是大规模的种植与养殖模式，而另一类是较小规模的精细化和绿色化的生产模式。

1.2 需要解决的问题

1.2.1 问题一

请您通过网络平台搜集适合不同地区种植的农产品（包括粮食、蔬菜及水果等）产量及价格相关时间序列数据，并对不同农产品预期的人力成本及收入展开相关分析讨论。

1.2.2 问题二

请为具有充足土地资源的农民们提供合理的大规模农产品种植计划，这一计划不仅能够保障农民们的稳定收入，同时也能最大限度的保证土地资源的循环利用。

1.2.3 问题三

针对缺乏土地资源的农民们提供合理的小规模种植计划，这一计划能够兼顾农民基本收入的同时尽量减少水资源的过度使用。

1.2.4 问题四

如果在农业基础之上同时考虑牧业与渔业相关经营，您能否提出更为合理的种植与养殖计划。

1.2.5 问题五

一个家庭经营多种农畜产品往往消耗过多的人力成本，您能否提供一个几个家庭合作后的最优养殖计划。

2 问题分析

2.1 问题 1 分析

对于问题 1.1，首先根据我国关于因地制宜的推进农业现代化的相关改革和措施，将中国农业分为六个农业地区 [1]，其中包括：黄河中下游地区，东北温湿地区，西北干旱地区，长江中下游地区，华南湿热地区，西南湿热地区。然后我们在国家统计局以及全国各省市统计局官网搜集到以上六大农业地区的农产品（包括粮食，蔬菜及水果）产量数据，利用 *Python* 将数据可视化后得到各个地区的主要农产品，并与实际情况进行比较分析结果的合理性。

对于问题 1.2，本题要求分析不同农产品预期的人力成本及收入之间的相关性。我们在《全国农产品成本收益资料汇编》中提取了 2002 年到 2019 年之间每年全国各地主要农作物（包括粮食，蔬菜及水果）的总产值和人力成本数据（以每亩为单位）。将数据录入 *Excel* 绘制各个人力成本及收入的折线图，观察大致的相关关系。然后皮尔森相关系数模型定量分析两者的相关关系。

2.2 问题 2 分析

对于问题 2，初步对本题进行分析可以知道，这是一个数学规划问题，但是由于问题中的“最大限度保证土地资源循环利用”这一概念过于抽象，难以直接用数学语言进行描述，因此本题先建立综合评价体系，给出“土地资源循环利用”的定量描述方式，最后利用数学规划的模型对问题进行求解。

2.3 问题 3 分析

对于问题 3，对本题进行分析可以知道。这是一个数学规划问题，并且题目中的各个概念都容易利用数学语言进行表示，因此针对本题直接建立线性规划模型，并收集内蒙古地区的相关数据，带入模型进行求解，最后将模型求得的解与实际情况进行比较，得出模型的合理性。

2.4 问题 4 分析

对于问题 4，由于条件的局限性，如果一个个收集各种数据则将消耗大量的时间和精力，因此针对本题建立了理想的复杂网络模型，并在本文给出了该问题的定性规划。

3 模型的假设

- 1 假设网络数据是真实可靠的;
- 2 部分地区作物数据缺失，在本文中认为该地区作物数量过少，无法记录，因此缺失的数据都记为 0;
- 3 其它的假设将在下面每一问问题中单独罗列。

4 符号说明

由于记号较多，因此在每一问中单独罗列该问的符号说明。

5 问题一

5.1 各地区主要农作物的分析

根据我国关于因地制宜的推进农业现代化的相关改革和措施，将中国农业分为六个农业地区 [1]，其中包括：黄河中下游地区，东北温湿地区，西北干旱地区，长江中下游地区，华南湿热地区，西南湿热地区。

从网络收集搜集到以上六大农业地区的农产品（包括粮食，蔬菜及水果）产量以及价格相关时间序列数据，综合近三年的农作物产量数据，分别统计出六大农业区部分常见农作物的比例关系，如下图所示：

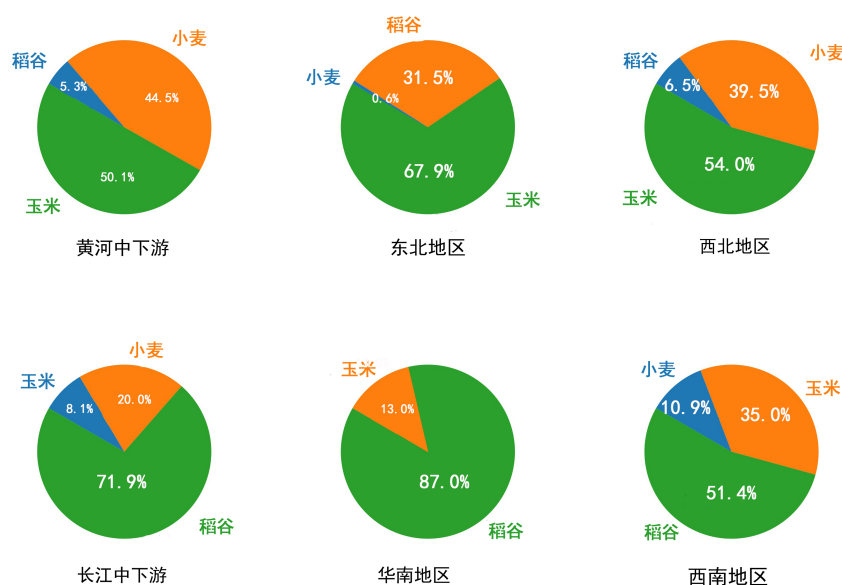


图 5-1 六大地区粮食占比

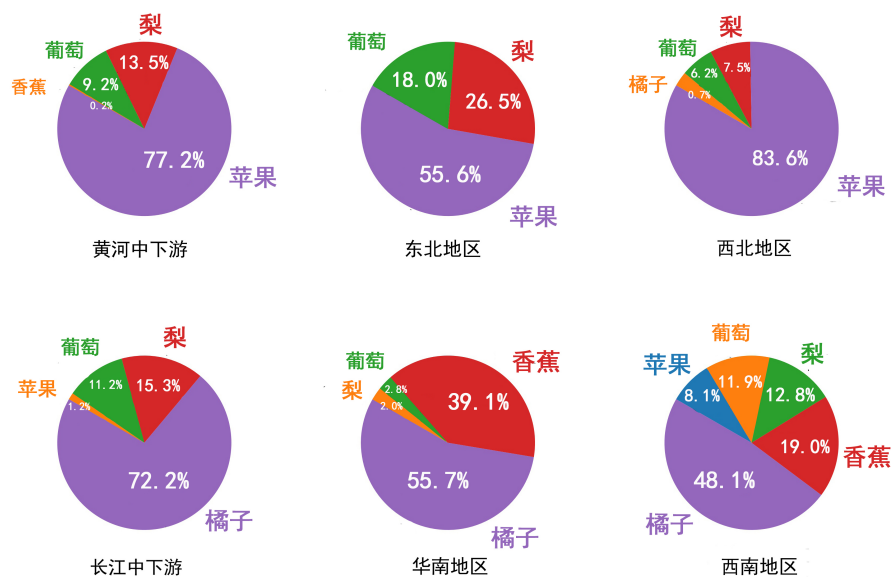


图 5-2 六大地区水果占比

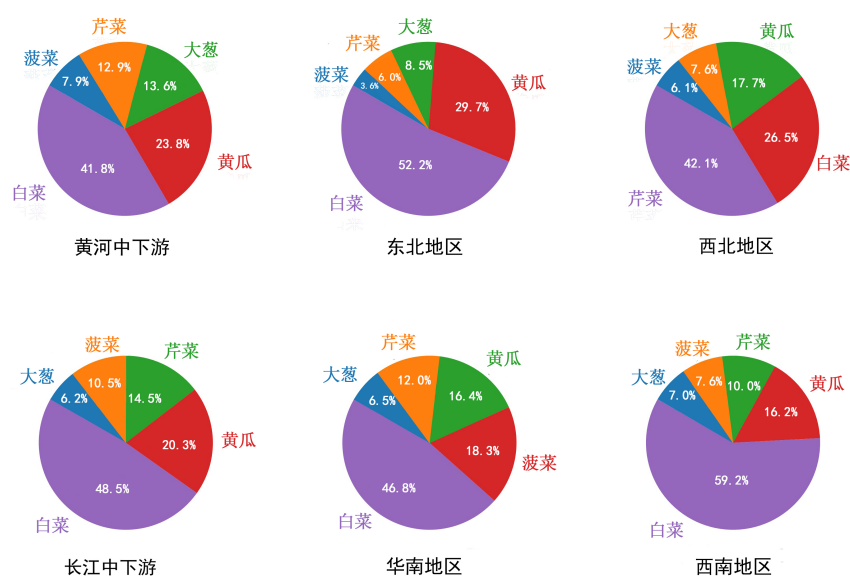


图 5-3 六大地区蔬菜占比

从以上面图5-1，图5-2，图5-3，可以看出以下几点：

对于粮食作物，玉米作物主要分布在东北地区，西北地区，黄河中下游地区，并且产量均超过这三个地区主要粮食总产量的一半。小麦作物主要分布在黄河中下游地区，西北地区以及长江中下游地区，并且在华南地区几乎没有小麦的种植。稻谷作物主要分布在长江中下游，华南地区，西北地区，以及西南地区，其中稻谷产量在华南地区三种粮食总产量中的占比达到 87.0%

对于水果作物，苹果主要分布在黄河中下游，东北地区 and 西北地区，另外在华南地区几乎没有苹果的种植。华南地区，西南地区以及长江中下游地区的水果总产量中橘子都占有很大的比例。葡萄和梨在全国六个地区都有分布，其中在华南地区水果总产量中占比最小，只有 2%。香蕉的种植地域比较集中，主要在华南和西南地区。

对于蔬菜作物，白菜在全国六个地区都有相当的分布，说明白菜的种植对地域要求不大，且人们对其需求量也大。菠菜大葱，黄瓜在全国六个地区也都有适量分布且比值较为平均。芹菜同样分布在全国六个地区，但是比重各不相同，其中芹菜在西南地区蔬菜产量中所占比值较大。

5.2 相关性定性分析

对不同农产品预期的人力成本及收入的相关性分析。使用 *Matlab* 绘制折线图 (如图5-4)，对部分产品两种变量之间的关系有了一个大致的了解。

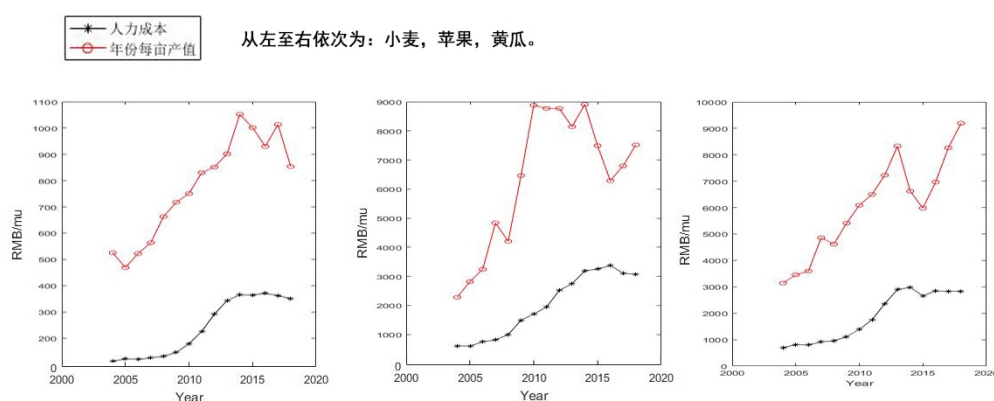


图 5-4 部分农作物人力成本与收入的折线图

根据以上折线图，我们可以大致判断出农作物的人力成本和收入具有一定的线性相关性。因此可以采用皮尔森相关系数法定量分析人力成本和收入的相关性。

5.3 相关性的定量分析

5.3.1 皮尔森相关系数

皮尔森相关系数是一种线性相关系数，是最常用的一种相关系数。记为 r ，用来反映两个变量 X 和 Y 的线性相关程度， r 值介于-1 到 1 之间， $r \geq 0$ 时表示

变量之间具有一定的正相关性，反之则表示变量之间具有一定的负相关性 r 越接近 1 表示变量之间的正相关性越强， r 越接近 -1 表示变量之间的负相关性越强，当 r 靠近 0 时则可以认为两个变量之间的线性相关性不是特别明显 [2]。皮尔森相关系数的计算公式如下：

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

通常情况下我们通过皮尔森相关系数的取值范围（表5-1）来判断变量的相关强度：

表 5-1 皮尔森相关系数绝对值取值对应相关强度

$ r $	相关程度
0.6-0.8	强相关
0.4-0.6	中等程度相关
0.2-0.4	弱相关
0.0-0.2	极弱相关或者无相关

通过上述公式，可以计算出全国主要农作物的人力成本与收入的相关系数，计算结果如表5-2.

表 5-2 主要农作物人力成本和收入相关系数计算结果

粮食	稻谷	小麦	玉米	大豆	花生	油菜籽
	0.95017	0.93377	0.78913	0.46033	0.61105	0.69952
水果	苹果	香蕉	橘子	梨	葡萄	水蜜桃
	0.74341	0.85488	0.85876	0.86772	0.78665	0.86443
蔬菜	西红柿	黄瓜	芹菜	大葱	白菜	菠菜
	0.94673	0.87895	0.96533	0.93729	0.71536	0.82102

通过主要农作物人力成本和收入相关系数计算结果可以看出除了大豆以外，其他农作物的人力成本与收入都具有较强正相关性。

6 问题二

6.1 本题模型假设

1. 假设该地区的农民自给自足, 即没有雇佣关系, 不产生劳务支出。
2. 假设土地资源能得到完全的利用, 即 $\frac{\text{区域使用资源所占面积}}{\text{区域可利用面积}} = 1$ 。
3. 对于题目中的“保障稳定收入”, 规定总年收益超过 M 时, 称该年收入稳定。

6.2 本题符号说明

表 6-1 问题二符号说明

符号	符号解释
L	系统的总劳动人口。
R_{α}	种植方案中选取的第 α 种可循环利用作物的面积。
N_{β}	种植方案中选取的第 β 种不可循环利用作物的面积。
N	资源的总投入。
o_{α}	单位第 α 种资源预计年收益。
o_{β}	单位第 β 种资源预计年收益。
r_{α}	种植单位面积 R_{α} 作物所需要的资金。
n_{β}	种植单位面积 N_{β} 作物所需要的资金。
$e_{r_{\alpha}}$	单位面积 α 种可循环利用资源等效能值。
$e_{n_{\beta}}$	种植第 β 种不可循环利用资源需等效能值。
u_{α}	第 α 种可再生资源可产出利用物能值。

6.3 能值理论简介

6.3.1 能值理论的含义

能值 (Emergy) 与能量 (Energy) 不同, 创始人 H.T.Odum 对它下的定义为: 一流动或储存的能量所包含的另一种能量的数量, 或理解为: 产品或劳务形成过程

中直接或间接投入应用的一种有效的总能量。任何能量都起源于太阳能，故实际应用以太阳能为基准，用太阳能焦耳来度量不同能量的能值。

能值理论是以能值为基准，把生态系统的物流、能流和价值流转换为同意标准的能值进行评价分析，从而可以定量反应各种系统的结构、功能及经济生态效率。

将不同的能量或物质具有不同的形态、性质转换为同一单位，这个转换率就叫太阳能转换率，是单位物质或能量所含有太阳能值。其表达式为：

$$A \text{ 物质的太阳能转换率} = \frac{\text{应用的太阳能 } J}{1JA \text{ 物质的能量}}$$

由上式可知， X 单位的物质 A 具有的能量值 = X 单位 $\times A$ 物质的能量转换率。

能值转换率是从生态学系统、热力学引申而来的衡量不同类比能量的尺度，这与系统的能量等级密切相连 [3]。

6.4 资源循环利用系统的综合评价指标建立

6.4.1 指标体系的建立



图 6-1 生态系统评价指标体系

图 6-1 中各二级指标的意义如下：

- E_{11} —— 该指标反应经济产出能力；
- E_{12} —— 该指标反应经成本投入度；
- E_{21} —— 该指标反应生态链产生的环境效益；
- E_{31} —— 该指标反应人们的生活水平；
- E_{32} —— 该指标反应开发程度与人们的生活水平。

6.4.2 指标建立依据与意义

通过能值理论的方法介绍，我们已经得知评价一个生态经济系统的能值产出率、废弃物能值负载率等环境、经济和社会方面的生态效率，故在此基础上建立指标体系，结合能值指标的效率指标进行归类分层。

6.4.3 指标体系与计算公式

图 6-1 中各评价指标的计算公式见表6-2, 表6-3。

表 6-2 评价指标的计算公式

评价指标名称	计算公式
E11 经济总产出与资源总投入的比值	$E11 = O/N$
E12 劳务投入与经济总产出比	$E12 = M/O$
E21 循环利用废弃物占总资源比	$E21 = C/U$
E31 区域总产出与区域总人口比	$E31 = O/P$
E32 区域使用资源与区域可利用面积比	$E32 = U/A$

表 6-3 原始数据

原始数据	基准变量
循环利用的废气、废水和固废	循环利用废弃物的能值 (C)
使用的不可再生资源的能值	不可再生资源能值 (N)
不可更新资源能值及循环利用能值	总使用能值 ($U = N+C$)
已开发和记忆开发潜力的土地	区域建设可利用的土地面积 (A)
预计的总产出能值	经济总产出能值 (O)
劳务输出	成本投入 (M)

指标计算中需要用到的公式包括四个：

$$\text{经济生态效率计算公式: } E_1 = \sum_{j=1}^2 f_{1j} w_{1j}$$

$$\text{环境生态效率计算公式: } E_2 = f_{21} w_{21}$$

$$\text{社会经济系统生态效率公式: } E_3 = \sum_{j=1}^2 f_{3j} w_{3j}$$

$$\text{系统生态效率计算公式: } E = \sum_{i=1}^3 E_i w_i$$

其中: $f_{ij} = \frac{x_{ij}}{x_{ij}^0}$, 表示实际值 x_{ij} 与标准值 x_{ij}^0 的比;

w_i ——表示第 i 评价准则权重;

w_{ij} ——表示第 i 评价准则第 j 项评价指标的权重;

E_i ——表示第 i 评价准则的生态效率值;

E ——表示整个生态系统的效率值, 只产品或服务的价值与原材料的消耗、能源使用和废物排放等环境影响的比较。

6.4.4 数据的来源与处理

模型中的数据主要是能值输入和输出的物质、能源和资金, 二是这些物质、能源和资金的能值转化率, 三是由这些基础数据按照指标计算公式得来, 四是指标的权重。前两者通过第一问的工作容易获得, 继而得出第三类数据, 关于权重的获取, 常设计实验通过问卷调查之后使用层次分析法对数据进行一致性检验和校正后得出, 由于条件的限制, 评价的判断矩阵由人为给出。

6.4.5 层次分析法计算权重

①. 判断矩阵的填写:

由于建模条件的限制, 人为给出如下判断矩阵来描述各个指标的重要程度:

表 6-4 评价指标权重分析的判断矩阵

	E_{11}	E_{12}	E_{21}	E_{31}	E_{32}
E_{11}	1	5	8	1/3	1/10
E_{12}	1/5	1	2	1/10	1/10
E_{11}	1/8	1/2	1	1/10	1/10
E_{11}	3	10	10	1	2
E_{11}	2	10	10	1/2	1

②. 一致性检验:

层次分析法在进行权重计算之前需要对所构造的判断矩阵进行一致性检验，上述判断矩阵的一致性检验结果见表6-5。

表 6-5 一致性检验结果

CI	RI	CR
0.0325	1.1200	0.0290

由于一致性比例 $CR = 0.0290 < 0.1$ 因此一致性检验通过。

③. 权重计算:

为了保证结果的准确性, 本题分别采用算数平均法、几何平均法、特征值法分别计算权重并取平均值得到最终权重。

三种方法计算权重结果见图6-2



图 6-2 三种方法计算权重

从而得到最终的权重向量为: [0.180, 0.041, 0.029, 0.443, 0.307]。

6.4.6 优化模型的建立

基于上述分析以及模型假设，可以建立本题的优化模型如下：

$$\begin{aligned}
 \max \quad & E \\
 \text{s.t.} \quad & R_{\alpha} \geq 0 \\
 & N_{\beta} \geq 0 \\
 & \sum_{\alpha} R_{\alpha} o_{r_{\alpha}} + \sum_{\beta} N_{\alpha} o_{n_{\beta}} \geq M
 \end{aligned}$$

将前面所设变量带入计算 E 由于模型假设，从而 E_{32} 为常数，得到最终优化模型如下：

$$\begin{aligned}
 \max \quad & \frac{w_1 w_{11} \sum_{\alpha} R_{\alpha} u_{\alpha} e_{r_{\alpha}}}{\sum_{\alpha} R_{\alpha} u_{\alpha} e_{r_{\alpha}} + \sum_{\beta} N_{\beta} e_{n_{\beta}}} + \frac{w_3 w_{31} (\sum_{\alpha} R_{\alpha} o_{r_{\alpha}} + \sum_{\beta} N_{\alpha} o_{n_{\beta}})}{L} \\
 \text{s.t.} \quad & R_{\alpha} \geq 0 \\
 & N_{\beta} \geq 0 \\
 & \sum_{\alpha} R_{\alpha} o_{r_{\alpha}} + \sum_{\beta} N_{\alpha} o_{n_{\beta}} \geq M
 \end{aligned}$$

显然，这是一个凸优化模型，因此能够接受。

7 问题三

7.1 本题模型假设

1. 假设土地资源能得到完全的利用，即 $\frac{\text{区域使用资源所占面积}}{\text{区域可利用面积}} = 1$ 。
2. 假设人均收入为 q 时能够满足每个人的基本需求。

7.2 本题符号说明

表 7-1 问题三符号说明

符号	符号解释
X_i	种植农产品 i 的面积
ω_i	种植单位面积 i 产品耗水量
p_i	种植单位面积农产品 i 的预计收入
c_i	种植单位面积农产品 i 所需要的成本
S	能够使用的土地面积
n	农民总数
m	可选种植产品总数

7.3 线性规划模型的建立

该问题需要的约束如下：①. 面积非负约束：

$$X_i \geq 0$$

①土地面积限制：

$$\sum_{i=1}^m X_i \leq S$$

②保证基本生活需求约束：

$$\sum_{i=1}^m (p_i - c_i) X_i \geq nq$$

从而建立相应的线性规划模型如下:

$$\begin{aligned}
 \min \quad & \sum_{i=1}^m \omega_i X_i \\
 \text{s.t.} \quad & X_i \geq 0 \\
 & \sum_{i=1}^m X_i \leq S \\
 & \sum_{i=1}^m (p_i - c_i) X_i \geq nq
 \end{aligned}$$

其中 w_i 为每单位 i 农产品所需要的水量。则显然此优化问题是一个线性规划问题 [4], 容易求解。

7.4 模型的求解

以内蒙古为例, 为了方便求解, 做出如下合理假设: 设农民人数为 1000, 人均基本需求为 6000 元每人, 土地面积为 3000 亩, 可选农产品为以下 8 种: 小麦、玉米、大豆、苹果、葡萄、西红柿、黄瓜、马铃薯。

由 2018 全国农产品成本收益资料汇编搜集到如下信息:

表 7-2 部分农产品平均每亩的成本、收入、用水量表

	小麦	玉米	大豆	苹果	葡萄	西红柿	黄瓜	马铃薯
成本 (元/亩)	888.2	889.1	265.8	6935.1	6749.3	6228.0	13239.6	1095.8
收入 (元/亩)	1239.0	1029.7	358.9	8539.5	10241.3	9775.3	16024.6	1548.4
水量 (吨/亩)	310.2	7.2	10.3	6.4	8.2	9.1	7.4	16.4

通过 *Python* 对该线性规划求出最优解:

$$x^* = [1.9 \times 10^{-8}, 7.2 \times 10^{-7}, 5.0 \times 10^{-7}, 1.9 \times 10^{-6}, 1.7 \times 10^3, 6.4 \times 10^{-6}, 5.6 \times 10^{-6}, 3.4 \times 10^{-7}]$$

根据计算结果: 对于内蒙古等西北地区种葡萄是明智之选, 这与实际情况相符合, 模型较为合理。

8 问题四

8.1 我国农牧渔混合经营现状分析

猪肉是我国居民肉类食品消费的主要来源，生猪养殖是农产业的主要经济来源之一。我国人均肉类食品消费种猪肉占比达 65% 以上，生猪养殖收入占农村居民纯收入的 8% - 10%。自 1985 年我国开放肉类市场以来，生猪生产得到迅猛发展，然而快速发展的过程却始终伴随着波动，且有幅度加剧和周期变短的趋势。在整个生猪产业链种生猪养殖环节的工作量最大，资金收益却最低，且生猪养殖环节的波动对价格影响显著。生猪生产的每一次波动，都会给养殖户带来巨大的经济损失，也会影响城乡居民的生活质量。因此，分析我国生猪生产的内在因素，在最大程度上稳定生猪生产，是关系到生猪产业发展及居民生活质量的民生问题 [5]。

随着近三十年农业产业结构调整, 农村劳动力转移, 生猪由一家一户分散养殖走向规模化、专业化快速发展轨道。以沼气为纽带的“猪-沼-水稻/果/蔬/棉/猪青饲料”等等养种循环生态农业模式是我国当前的主要模式。

因此本文考虑在农产品种植的基础上加入生猪养殖的规划, 同时利用猪排除的粪便等作为肥料作渔业养殖规划。

8.2 农牧渔复合生态系统结构的建立

8.2.1 总体结构

农牧渔复合生态系统的建立，是以高环境效益、经济效益以及物质的转化效率和补加能的投放效果等为目标，把净化有机污水与水生饲料生产。沼气能源的开发与初级、次级生产等有机结合在一起。总体规划见图8-1。

行排水，此时污水变清水。

四步利用: 有机污水经过三段净化的同时已完成了三步利用。从鱼塘排放的清水引至稻田作灌溉则完成了四步利用。

③. 初级生产结构:

该场初级生产结构的主体部分设置为水稻和水生饲料，此外还可以通过土地工程的建设，形成台田、林地、堤坝等，用于种植其它合适的农作物或者水果、蔬菜等。

④. 物质、能量多层次、分级利用结构:

可在食物链“饲料——猪”间插入鸡，即饲料喂鸡。可计划养适当的肉鸡、蛋鸡获得的鸡粪发酵后按照适当的比例混合于育肥猪精饲料中。

⑤. 鱼塘立体利用结构:

鱼塘分为两部分，一部分设置在猪舍四周作为防疫沟，可实行两层立体利用。上层养鱼，下层放河蚌。其二是新挖掘两个鱼塘，实行三层立体应用。可按生活习惯搭配不同的鱼，使得上、中、下各栖息环境的饵料都得到充分的利用。

⑥. “稻——萍——鱼”共生生态结构:

生物间有着共生互惠的关系，水稻为生长的绿萍创造了一个适宜生长的荫湿环境，为鱼提供了舒适的栖息场所。反过来，绿萍的覆盖抑制了杂草的生长，而且绿萍能够通过共生的鱼腥藻固氮，其残体腐解后又为水稻提供氮素等成份[6]。

9 问题五

9.1 相关定义

①博弈: 在一定条件下，遵守一定的规则，一个或几个拥有绝对理性思维的人或团队，从各自允许选择的行为或策略进行选择并加以实施，并从中各自取得相应结果或收益的过程。

②纳什均衡: 经济学定义: 所谓纳什均衡，指的是参与人的这样一种策略组合，在该策略组合上，任何参与人单独改变策略都不会得到好处。换句话说，如果在一个策略组合上，当所有其他人都不改变策略时，没有人会改变自己的策略，则该策略组合就是一个纳什均衡。

数学定义: 在博弈 $G = \{S_1, \dots, S_n : u_1, \dots, u_n\}$ 中，如果由各个博弈方的各一个策略组成的某个策略组合 (s_1^*, \dots, s_n^*) 中，任一博弈方 i 的策略 s_i^* ，都是对

其余博弈方策略的组合 $(s_1^*, \dots, s_{i-1}^*, s_{i+1}^*, \dots, s_n^*)$ 的最佳对策, 也即:

$$u_i(s_1^*, \dots, s_{i-1}^*, s_i^*, s_{i+1}^*, \dots, s_n^*) \geq u_i(s_1^*, \dots, s_{i-1}^*, s_{ij}^*, s_{i+1}^*, \dots, s_n^*) \quad \forall s_{ij} \in S_i$$

则称 (s_1^*, \dots, s_n^*) 为 G 的一个纳什均衡。

经济学定义从字面上还是相对比较好理解的; 这里稍微解释一下数学定义, 博弈论也称 *GameTheory*, 一场博弈用 G 表示, S_i 表示博弈方 i 的策略, u_i 表示收益。因此, 纳什均衡的意思是: 任何一方采取的策略都是对其余所有方采取策略组合下的最佳对策; 当所有其他人都不改变策略时, 为了让自己的收益最大, 任何一方都不会 (或者无法) 改变自己的策略, 这个时候的策略组合就是一个纳什均衡。

③合作博弈与非合作博弈

在博弈模型中, 根据全体局中人的支付总和是否为零, 分为零和博弈与非零和博弈。根据局中人是否合作, 博弈可分为合作博弈与非合作博弈。人们现在谈到博弈论, 一般都是指非合作博弈。合作博弈与非合作博弈之间的主要区别在于博弈方的行为相互作用时, 博弈方能否达成一个具有约束力的协议。如果能, 就是合作博弈, 否则, 就是非合作博弈。例如两个寡头企业, 如果它们之间能达成一个协议, 联合起来获取最大化垄断利润, 并且各自按这个协议生产, 就是合作博弈。它们面临的问题就是如何分享合作带来的剩余。非合作博弈强调的是个人理性、个人最优决策, 其结果往往是低效率的甚至是无效率的。而合作博弈强调的是团体理性、效率、公正、公平, 合作博弈实际上就是一种“双赢”策略, 它通常能获得较高的效率或效益 [7]。

9.2 本题模型假设

1. 每个农民都是理性的人;
2. 每个农民都是为了最大化个人的收益;
3. 处于条件限制, 本题仅考虑两个家庭之间的合作。

9.3 模型的建立与求解

农民 i 的利益最大化问题可通过求解下列优化问题得到:

$$\begin{aligned} \max_x \quad & p(x_i + x_{-i})x_i - H_i(x_i) \\ \text{s.t.} \quad & 0 \leq x_i \leq c_i \end{aligned}$$

其中 $H_i(\cdot)$, $i = 1, 2$ 是生产成本, $p(q)$ 为在总需求为 q 的情况下随机反需求函数, c_i 为生产能力。

则上述优化问题的 KKT 条件为:

$$0 \leq \begin{pmatrix} x_i \\ \mu_i \end{pmatrix} \perp \begin{pmatrix} -p(x_i + x_{-i}) - x_i p'_q(x_i + x_{-i}) + H'_i(x_i) + \mu_i \\ c_i - x_i \end{pmatrix} \geq 0$$

其中 $\mu_i, \quad i = 1, 2$ 为 $x_i \leq c_i$ 约束下的 $Lagrange$ 乘子。

则两个家庭的博弈问题为:

$$0 \leq \begin{pmatrix} x_1 \\ x_2 \\ \mu_1 \\ \mu_2 \end{pmatrix} \perp \begin{pmatrix} -p(x_1 + x_2) - x_1 p'_q(x_1 + x_2) + H'_1(x_1) + \mu_1 \\ -p(x_2 + x_1) - x_2 p'_q(x_2 + x_1) + H'_2(x_2) + \mu_2 \\ c_1 - x_1 \\ c_2 - x_2 \end{pmatrix} \geq 0$$

等价于

$$-\begin{pmatrix} -p(x_1 + x_2) - x_1 p'_q(x_1 + x_2) + H'_1(x_1) + \mu_1 \\ c_1 - x_1 \\ -p(x_2 + x_1) - x_2 p'_q(x_2 + x_1) + H'_2(x_2) + \mu_2 \\ sc_2 - x_2 \end{pmatrix} \in N_{R_+^4}((x_1, \mu_1, x_2, \mu_2)^T)$$

其中 N 为法锥。

设 $p(q) = a - q, H_i(x_i) = bx_i, c_i = +\infty, a > b$, 则农民 i 可通过求解下列问题获得最大利润:

$$\begin{aligned} \max_x \quad & (a - x_i - x_{-i})x_i - bx_i \\ \text{s.t.} \quad & x_i \geq 0 \end{aligned}$$

两个家庭的博弈问题可写成下列变分不等式的形式:

$$0 \leq x_1 \perp 2x_1 + x_2 + b - a \geq 0$$

$$0 \leq x_2 \perp 2x_2 + x_1 + b - a \geq 0$$

该问题的最优解为 $x_1^* = x_2^* = \frac{a-b}{3}$ 。

10 模型评价

10.1 模型的优点

1. 针对模型 1: 该问题采用皮尔森给相关系数法给出相关性分析, 该方法比较成熟, 结果的可信度能够接受。
2. 针对模型 2: 该问题在对“最大限度保证资源循环利用”这一概念的量化描述方面具有一定的创造性。
3. 针对模型 3: 该问题建立线性规划模型求解, 线性规划的求解方法成熟, 且模型简便, 并且举例对模型进行分析, 模型可信度较大。
4. 针对模型 4: 该问题结合相关的专业只是给出了农牧渔混合经营的定性分析理论, 具有一定的参考价值。
5. 针对模型 5: 该问题使用博弈论模型进行求解, 该方法比较成熟, 求解结果比较合理。

10.2 模型的不足之处

1. 针对模型 1: 在建模过程种, 许多数据难以获取, 因此在相关性分析的定性描述中能够可视化观察的数据比较少。
2. 针对模型 2: 未给出具体的例子, 在这一点以后多加改进。
3. 针对模型 3: 模型虽然简单, 但是也肯能出现过于理想的问题, 在实际运用中还应该对模型进行适当的修正。
4. 针对模型 4: 由于条件的限制, 为给出太多数据进行定量分析, 在实际的运用中应当因地制宜, 结合实际情况作进一步的分析。
5. 针对模型 5: 由于时间的限制, 未给出一般情况下多家用户的求解, 在实际运用中还可以对模型作进一步的推广。

参考文献

- [1] 赵松乔, 中国农业 (种植业) 的历史发展和地理分布 [J], 7-9, 2015.
- [2] 韩晟, 韩坚舟, 赵璇, 等, 距离权重改进的 Pearson 相关系数及应用 [J], 2-5, 2019.
- [3] 毛静, 构建资源 LMS 循环利用模式研究 [D], 27-36, 2011.
- [4] 姜启源, 谢金星, 数学建模案例选集, 高等教育出版社 [M], 2006.
- [5] 丁雄, 生态农业产业链系统协调与管理策略研究 [D], 33-34, 2014.
- [6] 卢政民, 西安生态养殖试验场农牧渔复合生态系统的结构与功能 [J], 11-15, 1988.
- [7] 张良桥, 进化稳定均衡与纳什均衡——兼谈进化博弈理论的发展 [J], 103-111, 2011(03).

附 录

Phtyon 源程序

可视化程序

```
province_new = [  
    "北京", "天津", "河北",  
    "山西", "内蒙古", "辽宁",  
    "吉林", "黑龙江", "上海",  
    "江苏", "浙江", "安徽",  
    "福建", "江西", "山东",  
    "河南", "湖北", "湖南",  
    "广东", "广西", "重庆",  
    "四川", "贵州", "云南",  
    "西藏", "陕西", "甘肃",  
    "宁夏", "新疆"  
]  
  
region1 = [3, 4, 25, 15, 14] # 黄河中下游的省份  
region2 = [5, 6, 7] # 东北地区的省份  
region3 = [25, 26, 27, 28 ] # 西北地区的省份  
region4 = [16, 17, 9, 10, 11, 8, 13] # 长江中下游地区的省份  
region5 = [18, 19] # 华南地区的省份  
region6 = [20, 21, 22, 23, 24] # 西南地区的省份  
regions = [region1, region2, region3, region4, region5, region6]  
region_name = ["黄河中下游", "东北地区", "西北地区", "长江中下游",  
    "华南地区", "西南地区"]  
  
path = "data\\第一问数据.xlsx"  
  
unhusked_rice = pd.read_excel(path, sheet_name="稻谷数据")  
wheat = pd.read_excel(path, sheet_name="小麦数据")  
corn = pd.read_excel(path, sheet_name="玉米数据")  
  
unhusked_rice = np.array(unhusked_rice)
```

```

wheat = np.array(wheat)
corn = np.array(corn)

unhusked_rice_avrage = np.mean(unhusked_rice, axis = 1)
wheat_avarage = np.mean(wheat, axis = 1)
corn_avarage = np.mean(corn, axis=1)

plt.rcParams['font.sans-serif']=['SimHei']
plt.figure(figsize=(7, 4))
for i in range(6):
    plt.subplot(2,3,i+1)
    local_unhusked_rice_avrage =
        np.mean(unhusked_rice_avrage[regions[i]])
    local_wheat_avarage = np.mean(wheat_avarage[regions[i]])
    local_corn_avarage = np.mean(corn_avarage[regions[i]])
    prep_data = np.array([
        local_unhusked_rice_avrage,
        local_wheat_avarage,
        local_corn_avarage
    ])
    label = ["稻谷", "小麦", "玉米"]
    index = np.argsort(prep_data)
    plt_data = np.sort(prep_data)
    plt_label = []
    for j in index:
        plt_label.append(label[j])
    plt.pie(x = plt_data, labels=plt_label,
        autopct='%1.1f%%', shadow=False, startangle=150,
        counterclock=False, radius =1 )
    plt.title(region_name[i], y = -0.2)
plt.savefig('data\\fig1.jpg', dpi = 600)
plt.show()

# 水果数据
apple_pro = np.array([
7.1, 5.6, 228.1,

```

```
444.9, 15.5, 240.9,  
4.6, 14.4, 0,  
5.8, 0, 20.0,  
0, 0, 939.5,  
434.5, 1.2, 0,  
0, 0, 0,  
0.4, 65.2, 7.1,  
59.7, 0, 1092.5,  
311.1, 0.4, 44.0,  
144.2  
)  
  
orange_pro = np.array([  
0, 0, 0,  
0, 0, 0,  
0, 0, 9.4,  
3.2, 186.8, 0.8,  
315.4, 404.3, 0,  
4.9, 465.9, 500.9,  
410.3, 682.1, 6.6,  
250.6, 415.7, 25.4,  
88.8, 0, 45.7,  
0.1, 0, 0,  
0  
)  
  
pear_pro = np.array([  
9.1, 6.4, 342.4,  
86.7, 7.9, 116.2,  
3.9, 3.7, 3.2,  
78.0, 38.9, 142.2,  
16.7, 16.8, 103.7,  
121.8, 37.5, 16.6,  
10.4, 35.9, 0,  
26.8, 91.7, 28.0,  
63.6, 0, 105.2,  
21.0, 0, 1.7,
```

```
123.1
```

```
])
```

```
purple_pro = np.array([
    2.7, 11.4, 111.6,
    30.5, 7.5, 70.5,
    6.3, 7.3, 6.4,
    64.9, 80.8, 42.5,
    19.6, 8.0, 109.9,
    70.3, 24.2, 17.6,
    0, 55.5, 0,
    10.2, 37.9, 20.3,
    126.5, 0, 78.6,
    24.9, 0, 13.7,
    249.3
])
```

```
banana_pro = np.array([
    0, 0, 0,
    0, 0, 0,
    0, 0, 0,
    0, 0, 0,
    38.7, 0, 0,
    0, 0, 0,
    395.2, 371.6, 127.2,
    1.2, 4.6, 1.7,
    176.8, 0, 0,
    0, 0, 0,
    0
])
```

```
province = [
    "北京", "天津", "河北",
    "山西", "内蒙古", "辽宁",
    "吉林", "黑龙江", "上海",
    "江苏", "浙江", "安徽",
```

```

"福建", "江西", "山东",
"河南", "湖北", "湖南",
"广东", "广西", "海南",
"重庆", "四川", "贵州",
"云南", "西藏", "陕西",
"甘肃", "青海", "宁夏",
"新疆"
]

plt.rcParams['font.sans-serif']=['SimHei']
plt.figure(figsize=(7, 4))
for i in range(6):
    plt.subplot(2,3,i+1)
        local_apple_avarage = np.mean(apple_pro[regions[i]])
        local_orange_avarage = np.mean(orange_pro[regions[i]])
        local_pear_avarage = np.mean(pear_pro[regions[i]])
        local_purple_avarage = np.mean(purple_pro[regions[i]])
        local_banana_avarage = np.mean(banana_pro[regions[i]])
        prep_data = np.array([
            local_apple_avarage,
            local_orange_avarage,
            local_pear_avarage,
            local_purple_avarage,
            local_banana_avarage
        ])
        label = ["苹果", "橘子", "梨", "葡萄", "香蕉"]
        index = np.argsort(prep_data)
        plt_data = np.sort(prep_data)
        plt_label = []
    for j in index:
        plt_label.append(label[j])
        plt.pie(x = plt_data, labels=plt_label,
            autopct='%1.1f%%', shadow=False, startangle=150,
            counterclock=False, radius =1)
        plt.title(region_name[i], y = -0.2 )
plt.savefig('data\\fig2.jpg', dpi = 600)
plt.show()

```

```
# 白菜
```

```
cabbage = np.array([
49.98, 74.93, 1907.68,
187.52, 272.16, 787.09,
350.17, 313.81, 14.04,
754.40, 100.97, 227.41,
171.97, 165.67, 1510.45,
904.91, 536.02, 492.49,
324.49, 345.79, 8.94,
89.11, 378.86, 367.03,
343.37, 7.01, 163.10,
187.67, 17.25, 47.26,
186.15
```

```
])
```

```
# 黄瓜
```

```
cucumber = np.array([
25.49, 73.46, 955.19,
108.41, 119.61, 650.62,
96.70, 78.36, 14.75,
280.96, 67.71, 164.31,
71.67, 55.80, 723.21,
689.72, 194.42, 179.34,
131.34, 104.11, 25.94,
48.44, 171.42, 43.87,
34.67, 1.23, 146.25,
98.10, 4.85, 34.66,
37.74
```

```
])
```

```
# 大葱
```

```
onion = np.array([
5.81, 21.98, 317.50,
71.81, 58.74, 121.62,
53.25, 62.74, 4.55,
```

```
72.47, 11.81, 91.08,  
0.00, 14.93, 413.51,  
395.94, 57.48, 40.01,  
65.32, 28.11, 3.15,  
19.04, 57.49, 31.52,  
29.58, 0.41, 58.90,  
38.31, 9.69, 11.12,  
42.96  
])  
  
# 菠菜  
spinach = np.array([  
10.88, 24.36,  
359.22, 14.44, 21.04,  
58.87, 26.35, 15.48,  
5.97, 159.65, 18.78,  
102.16, 36.20, 39.35,  
261.97, 245.53, 113.74,  
58.21, 185.39, 76.89,  
3.91, 18.52, 71.62,  
30.33, 28.96, 0.75,  
47.56, 28.06, 10.22,  
6.48, 25.89  
])  
  
# 芹菜  
celery = np.array([  
25.37, 61.51, 298.07,  
22.62, 97.66, 123.44,  
24.23, 18.64, 14.67,  
235.86, 60.24, 150.97,  
38.71, 55.24, 406.45,  
362.33, 95.21, 70.58,  
124.85, 47.44, 4.61,  
14.49, 119.62, 26.14,  
35.66, 2.05, 107.13,  
481.88, 4.01, 78.19,
```



```
46.00
])

province = [
    "北京", "天津", "河北",
    "山西", "内蒙古", "辽宁",
    "吉林", "黑龙江", "上海",
    "江苏", "浙江", "安徽",
    "福建", "江西", "山东",
    "河南", "湖北", "湖南",
    "广东", "广西", "海南",
    "重庆", "四川", "贵州",
    "云南", "西藏", "陕西",
    "甘肃", "青海", "宁夏",
    "新疆"
]

plt.rcParams['font.sans-serif']=['SimHei']
plt.figure(figsize=(7, 4))
for i in range(6):
    plt.subplot(2,3,i+1)
    local_cabbage_avarage = np.mean(cabbage[regions[i]])
    local_cucumber_avarage = np.mean(cucumber[regions[i]])
    local_onion_avarage = np.mean(onion[regions[i]])
    local_spinach_avarage = np.mean(spinach[regions[i]])
    local_celery_avarage = np.mean(celery[regions[i]])
    prep_data = np.array([
        local_cabbage_avarage,
        local_cucumber_avarage,
        local_onion_avarage,
        local_spinach_avarage,
        local_celery_avarage
    ])
    label = ["白菜", "黄瓜", "大葱", "菠菜", "芹菜"]
    index = np.argsort(prepare_data)
    plt_data = np.sort(prepare_data)
```

```

plt_label = []
for j in index:
    plt_label.append(label[j])
plt.pie(x = plt_data, labels=plt_label,
        autopct='%1.1f%%', shadow=False, startangle=150,
        counterclock=False, radius =1)
plt.title(region_name[i] , y =-0.2)
plt.savefig('data\\fig3.jpg', dpi = 600)
plt.show()

```

层次分析法计算程序

```

import numpy as np

class weightAnalysis:
    def __init__(self,dimensionOfJudgeMatric,judgementMatric):
        self.judgementMatric = np.array(judgementMatric)
        self.dimesionOfJudgeMatric = dimensionOfJudgeMatric
        self.__RIdict = {1:0, 2:0, 3:0.52, 4:0.89, 5:1.12,\
6:1.26, 7:1.36, 8:1.41, 9:1.46, 10:1.49, 11:1.52, 12:1.54}

    def set_dimesion_of_judge_matric(self, dimesion):
        self.dimesionOfJudgeMatric = dimesion

    def set_judgement_matric(self, judgementMatric):
        self.judgementMatric = np.array(judgementMatric)
        #判断是否为正互反矩阵.

    def is_positive_reciprocal(self):
        isValid = True
        for i in range(self.dimesionOfJudgeMatric):
            for j in range(i):
                if self.judgementMatric[i, j] * self.judgementMatric[j, i] != 1:
                    isValid = False
                    break
            return isValid

# 获取正互反矩阵的主特征值

```

```
def get_main_eigen_value(self):
    return np.max(np.linalg.eig(self.judgementMatric)[0].real)

# 获取正互反矩阵的主特征向量
def get_main_eigen_vector(self):
    eigValue,eigVector = np.linalg.eig(self.judgementMatric)
    index = np.argmax( eigValue )
    return eigVector[:, index].real

def getRI(self):
    return self.__RIdict[self.dimesionOfJudgeMatric]

def getCI(self):
    return (self.get_main_eigen_value() - \
            self.dimesionOfJudgeMatric) / (self.dimesionOfJudgeMatric - 1)

def getCR(self):
    return self.getCI()/self.getRI()

def get_weiht_vector1(self):
    sumOfRow = np.sum(self.judgementMatric, axis = 0)
    return np.mean(self.judgementMatric/sumOfRow , axis =1 )

def get_weiht_vector2(self):
    prodOfCol = np.prod(self.judgementMatric , axis = 1)
    return prodOfCol ** (1/self.dimesionOfJudgeMatric) \
    / np.sum(prodOfCol ** (1/self.dimesionOfJudgeMatric))

def get_weiht_vector3(self):
    return self.get_main_eigen_vector() /
           np.sum(self.get_main_eigen_vector())

if __name__ == '__main__':
    key = 1
    while key != 0:
        dimension = int(input("请输入判断矩阵维数:"))
        assert (dimension > 0), "请输入正整数"
```

```

assert (dimension <= 12), "问题维数超出范围,重新输入."
inputMatric = []
print("请输入判断矩阵")
for i in range(dimension):
    row = input().split()
    assert (len(row) == dimension),
        "输入错误,矩阵大小与判断矩阵维数不匹配"
    inputMatric.append([eval(ele) for ele in row])

analysis = weightAnalysis(dimension, inputMatric)
if analysis.is_positive_reciprocal():
    print("判断矩阵: \n", analysis.judgementMatric,
        "是正互反矩阵")
    print("判断矩阵最大特征值: %.4f" %
        (analysis.get_main_eigen_value()))
    print("最大特征值对应的特征向量:",
        analysis.get_main_eigen_vector())
    print("判断矩阵一致性指标 CI = %.4f" % (analysis.getCI()))
    print("平均随机一致性指标 RI = %.4f" % (analysis.getRI()))
    print("一致性比例 CR = %.4f" % (analysis.getCR()))
    if analysis.getCR() >= 0.1:
        print("CR = %.4f >= 0.1,一致性检验不通过,请重新填写问题." %
            (analysis.getCR()))
        continue
    else:
        print("CR = %.4f < 0.1,一致性检验通过,权重计算结果如下:" %
            (analysis.getCR()))
        print("利用算数平均法所得的权重向量为: \n",
            analysis.get_weih_t_vector1())
        print("利用几何平均法所得的权重向量为: \n",
            analysis.get_weih_t_vector2())
        print("利用特征值法所得的权重向量为: \n",
            analysis.get_weih_t_vector3())
else:
    print("判断矩阵: \n", analysis.judgementMatric,
        "\n不是正互反矩阵,重新输入.")

```

```
continue
```

```
key = int(input("是否继续输入？ 输入1继续, 输入0退出程序"))
```

求解线性规划

```
import numpy as np

p1= np.array([1239.00 ,1029.65 ,358.93 ,8539.51 ,10241.36
              ,9775.28 ,16024.58 ,1548.41])
c1= np.array([888.16 ,889.10,265.76 ,6935.07 ,6749.27 ,6228.00
              ,13239.61 ,1095.77])
q=p1-c1
c = np.array([310.20,7.24,10.33,6.42 ,8.20 ,9.13 ,7.41 ,16.43])
A = np.array([np.ones(8, dtype = int),-q])
b = np.array([3000,-6e6])
□
res= optimize.linprog(c,A,b)
print(res)
```

MATLAB 源程序

```

clear;clc;
t=[2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
    2016 2017 2018]';
%t代表年份; 1代表粮食, 2代表水果, 3代表蔬菜;
c1=[111.84 121.34 119.61 124.72 131.19 145.64 178.83 225.68
    291.4 343.78 364.77 364.39 370.99 361.87 350.76
]';
r1=[525.5 468.96 522.46 563.91 663.06 717.51 750.8 830.2 851.73
    901.93 1052.69 1001.71 930.36 1013.74 853.53
]';
c2=[612.18 604.07 752.14 816.88 1001.51 1488.97 1707.2 1944.15
    2519.85 2746.83 3186.83 3253.8 3369.15 3110.96 3065.22
]';
r2=[2283.03 2817.55 3243.29 4837 4203.14 6462.27 8881.18 8772.61
    8772.26 8141.33 8912.32 7490.4 6285.53 6797.22 7518.84
]';
c3=[686.96 812.78 795.04 914.1 950.86 1110.39 1385.3 1752.14
    2359.9 2894.34 2977.94 2661.57 2838.51 2830.14 2835.5
]';
r3=[3139.63 3453.41 3594.84 4857.99 4615.21 5412 6088.03 6499.87
    7228.32 8332.44 6620.43 5978.41 6967.79 8264.99 9193.7
]';
%画散点
subplot(1,3,1)
plot(t,c1,'-*k')
hold on;
plot(t,r1,'-or')
xlabel('Year');
ylabel('RMB/mu');
legend('人力成本','年份每亩产值');
title('小麦')
axis square

subplot(1,3,2)
plot(t,c2,'-*k')

```

```

hold on;
plot(t,r2,'-or');
xlabel('Year');
ylabel('RMB/mu')
legend('人力成本','年份每亩产值');
title('苹果');
axis square

subplot(1,3,3)
plot(t,c3,'-*k')
hold on;
plot(t,r3,'-or')
xlabel('Year');
ylabel('RMB/mu');
legend('人力成本','年份每亩产值');
title('黄瓜')
axis square

%计算相关系数.1代表粮食,2代表水果,3代表蔬菜;
C1=[167.29 179.36 181.14 186.84 202.61 210.73 239.74 301.28
    389.07 449.84 466.55 470.29 460.79 454.74 437.65;
111.84 121.34 119.61 124.72 131.19 145.64 178.83 225.68 291.4
    343.78 364.77 364.39 370.99 361.87 350.76;
140.49 148.38 149.94 159.78 176.98 192.61 235.1 295.49 398.4
    455.37 474.68 468.72 458.1 441.2 433.52;
74.16 101.65 110.41 87.7 88.32 103.53 115.31 136.38 177.5 200.95
    216.73 215.16 218.11 215.85 204.27;
173.4 190.6 200.62 208.46 233.15 254.83 319.61 399.15 533.59
    645.51 678.07 567.94 695.08 693.87 691.58;
140.2 147.06 153.36 167.63 131.54 203.86 257.02 322.91 444.33
    535.82 547.72 303.5 580.08 560.51 548.24;
];
R1=[617.6 561.41 597 663.56 791.24 810.99 801.42 1034.71 1091.03
    1110.6 1135.98 1147.26 1098.87 1127.27 1064.94;
525.5 468.96 522.46 563.91 663.06 717.51 750.8 830.2 851.73
    901.93 1052.69 1001.71 930.36 1013.74 853.53;
510.64 487.82 556.53 650.52 682.67 726.47 872.28 1027.32 1121.9
    1089.56 1145.71 949.54 765.89 850.69 881.48;

```



```
380.11 496.63 478.38 466.96 526.44 485.71 586.35 610.72 706.83
    659.58 641.61 559.62 468.63 537.91 474.29;
767.01 677.33 876.82 1198.44 933.46 1224.35 1285.21 1681.46
    1839.33 1441.7 1487.17 647.89 1684.48 1470.97 1445.16;
373.16 294.78 314.11 521.37 744.17 478.95 509.8 608.79 652.84
    745.86 710.01 561.03 590.22 713.66 724.16;
];

C2=[612.18 604.07 752.14 816.88 1001.51 1488.97 1707.2 1944.15
    2519.85 2746.83 3186.83 3253.8 3369.15 3110.96 3065.22;
1085.11 532.27 810.05 1208.27 1049.87 957.4 914.21 1176.1
    1415.74 1503.51 1552.53 1765.48 1557.37 1691.59 1812.18;
415.88 492.83 1191.42 859.92 684.67 717.04 739.32 946.66 921.16
    1339.66 1545.82 1955.54 1624.84 1809.21 1494.25;
];

R2=[
2283.03 2817.55 3243.29 4837 4203.14 6462.27 8881.18 8772.61
    8772.26 8141.33 8912.32 7490.4 6285.53 6797.22 7518.84;
3306.39 3676.06 2762.85 3607.56 2814.04 3127.2 3699.99 4776.74
    4053.07 4698.83 6474.24 7535.01 5626.91 6366.52 6988.6;
2635.35 3223.51 3932.6 3487.23 1876.88 2738.9 4022.9 3940.4
    3179.09 4444.26 4940.71 4759.28 4528.18 5154.47 4793.44
];

C3=[
1475.77 856.54 925.11 1019.82 1045.35 1250.16 1479.57 1795.26
    2495.98 4241.86 3180.14 2723.78 3950.6 4348 3843.77;
686.96 812.78 795.04 914.1 950.86 1110.39 1385.3 1752.14 2359.9
    2894.34 2977.94 2661.57 2838.51 2830.14 2835.5;
646.45 732.92 802.73 898.98 870.81 965.37 1090.23 1373.42
    2038.87 2344.56 2390.06 1657.77 3664.39 4330.48 3914.5;
631.04 634 670.82 747.54 834.61 939.57 1115.58 1367.18 1760.94
    2237.4 2381.06 1936.48 2567.69 2496.94 2635.12;
356.54 407.37 460.35 482.77 547.44 510.55 695.21 713.53 996.96
    1126.71 1228.47 1439.03 1423.61 1840.26 1719.69;
204.71 274.08 252.98 364.34 335.88 302.85 306.98 391.89 576.48
```

```
        668.79 655.2 572.4 779.69 828.96 853.1;
];

R3=[6595.48 4121.37 4474.22 6023.25 5245.53 6592.16 7314.07
    6613.18 9313.66 12942.82 9228.82 7719.28 9929.74 12333.42
    13108.2;
3139.63 3453.41 3594.84 4857.99 4615.21 5412 6088.03 6499.87
    7228.32 8332.44 6620.43 5978.41 6967.79 8264.99 9193.7;
2988.13 3266.35 2867.39 4491.29 4066.53 4241.67 5652.37 5511.51
    5518.95 7107.37 5601.92 5321.19 9914.46 10624.61 10493.16;
2868.16 2994.17 2955.44 4300.51 3542.1 3787.86 4123.43 4718.94
    5522.74 6058.26 5003.77 5549.57 6585.51 5773.36 6545.3;
1541.85 2238.13 2221.5 2717.91 2485.09 2368.07 4234.81 2757.18
    2366.51 3334.92 2896.95 3364.48 3754.61 3811.46 3679.57;
1141.08 1260.05 1343.4 1821.63 1798.26 1542.43 1581.31 1338.3
    2403.53 2984.92 1829.42 1794.37 2483.52 2288.22 2358.12
];

for i=1:size(C1,1)
A1(i)=corr(C1(i,:),R1(i,:), 'type', 'Pearson');
end
A1

for i=1:size(C2,1)
A2(i)=corr(C2(i,:),R2(i,:), 'type', 'Pearson');
end
A2

for i=1:size(C3,1)
A3(i)=corr(C3(i,:),R3(i,:), 'type', 'Pearson');
end
A3
```