中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生
数学建模竞赛

学　　校	江苏科技大学	
参赛队号	23102890004	
队员姓名	1.张子俨	
	2.周文	
	3.何铮	

中国研究生创新实践系列大赛
“华为杯”第二十届中国研究生
数学建模竞赛

题　目： 大规模创新类竞赛评审方案研究

摘　要：

创新类竞赛的特点是没有标准答案，需要评审专家根据命题人（组）提出的评审框架（建议）独立评审。同时，对同一份作品，不同评委的评分可能存在较大差异。极差大的问题更为突出。本文通过构建**基于回避制度的最优 “交叉分发”方案、最大匹配度优化模型、动态加权的标准分模型**等一系列相关模型，给出科学合理的评分排序策略，保证大规模创新类竞赛评审方案的公正性、公平性和科学性。

针对问题一，首先从“不同评审专家所给成绩之间的可比性”中**可比性**入手，通过**不同专家评审的作品集合之间交集**之间的关系。定义了**可比性的两个重要因素：所有交集元素个数总和与所有交集集合数**。所有交集元素个数总和表示不同专家评审专家之间的交流次数，以及所有交集集合数表示交集中的对比人数。引入构建**专家和作品的评阅** 0-1 **矩阵** $\boldsymbol{P}$，经过公式推算得到**所有交集元素个数总和的数学表达式为** $\boldsymbol{PP^T}$ **的 1-范数**和**所有交集集合数的数学表达式为** $\boldsymbol{PP^T}$ **的 0-范数**。进而得到“增加不同评审专家所给成绩之间的可比性”就是要构建**所有交集元素个数总和与所有交集集合数的双目标最优的“交叉分发”模型**。在本文中，通过严格的理论推导发现其构建**所有交集元素个数总和** $\boldsymbol{PP^T}$ **的 1-范数为恒定值 75000**。为此，最优的“交叉分发”模型**简化**为求 $\boldsymbol{PP^T}$ **的 0-范数最优的单目标优化模型**。进一步考虑**专家和归属学校或者归属地区的回避措施**，即添加了新的约束条件，得到**基于回避制度的最优“交叉分发”模型**。

针对问题二，根据题目要求，对附件 1 数据中专家评审的作品集与作品全集的对比性挖掘分析。引入**双样本 t 检验**，结果发现**专家评审的作品集与评审作品全集不属于同一分布，具有差异性**，为此需要寻找新的评审方案。考虑到现有评阅方案的局限性，本文引入了**正态化优化标准分模型、归一化优化标准分模型、组合优化标准分模型**三种评审方案。为了对比原始标准分方案和设计的三种方案，通过**匹配获奖论文的信息**，提出了**匹配函数**和**匹配度**的概念。结果显示三种方案对一等奖、二等奖、三等奖的匹配度分别为（**24.14%，57.89%，3.57%**）、（**27.59%，69.92%，23.21%**）、（**32.14%，76.78%，**

19.64%），明显优于原始标准分方案（**25%，55%，3%**）。对比结果发现，**组合优化标准分模型表现最佳**，为此本文后续部分采用该模型作为新的标准分计算模型。

针对问题三，根据题目所给的模拟数据 2.1 和 2.2。对比了两阶段的成绩整体的变化和两阶段极差整体的变化，数据显示：**第一阶段成绩相比较第二阶段的成绩差距被放大、第二阶段的极差整体于第一阶段极差整体提高了 1 个数量级**。数据表明**第二阶段是扩大极差的阶段**。分析两阶段评审方案相比不分阶段评审方案，**两阶段评审方案在匹配度优于不分阶段评审方案，而两阶段评审工作复杂度明显高于不分阶段评审**。通过对极差的研究对比发现，**第二阶段后获奖论文极差相比较第一阶段普遍变大，说明极差比获奖论文有高度的相关性**。为此，本文设计了"大极差"调整模型，通过对论文质量的初步分类，进行有策略地提高或者降低其评分的均值。对比结果发现：最高分被极大提高，原有排名发生明显改变，以原表格以数据 2.2 中第 1501-3000 为非高且非低分段作品为例，"大极差"调整模型后其顺序前十为：**2459、1600、1634、2780、2601、2241、2334、2145、2109、1813**。

针对问题四，考虑到评审专家个体的局限性，如知识的偏好、性格的偏好等因素引起的分值偏低或者偏高和不敢评价两个方面。在文中，引入了**两个重要的指标：专家标准分与文章标准分均值的偏移程度**和**专家作品评分集的方差**。进一步将这两个重要指标引入指数函数，就得到组合**权重函数**。同时，注意到**网评和会评专家的比重可以适当地变化**。最后，得到以**指数函数参数与专家比重两个决策变量**，修正后排序**接近于两阶段的最终排序**为目标函数的单目标优化模型。得到最优解，进而得到**动态加权评分排序模型**。以数据 2.2 训练集，动态加权评分排序模型结果显示一等奖、二等奖、三等奖的匹配度分别为（**43.48%，88.87%，8.61%**），明显优于原始排序匹配度（**27.54%，83.67%，4.31%**）。

总体来讲，本文针对创新类竞赛的特点，设计了一系列的相关模型，科学合理的评分排序策略，同时利用附件数据作为训练集，得到最优评分排序模型。对比结果显示，新的方案明显优于原始方案，值得推广。

关键词： 回避制度；最优"交叉分发"模型；双样本 t 检验；匹配度；组合优化标准分模型；极差；动态加权评分排序模型

目录

一、问题重述

1.1 问题背景

大规模创新竞赛评审方案的意义在于确保评审过程的公正性、公平性和科学性。当面对众多参赛作品和评委时，仅依据多位评委评分的总和进行排序是不可行的，会存在以下问题：

由于评委差异，对同一份作品，不同评委的评分可能存在较大差异。这是因为创新类竞赛的作品往往没有标准答案，评审专家需要根据评审框架和个人经验进行独立评审，导致评分差异较大；大规模评审困难：当竞赛规模大，评委的人数众多时，评审过程变得更加复杂和困难。大量作品需要评审，评委之间需要进行协调和统一评标标准，而且评审结果的差异也更加突出；极差问题：当评委人数众多时，极差问题变得更为突出。极差是指评委评分中最高分与最低分之间的差距，差异过大可能导致评审结果不准确或不公正；总分排序的局限性：简单地依据多位评委评分的总和进行排序并不是创新类竞赛评审的好方案。这是因为作品的评审标准往往是多维度的，只依据总分无法全面准确地反映作品的质量和创新性。

因此，在大规模创新类竞赛评审中，需要寻找解决方案来提升评审的公正性、公平性和科学性，减少评委差异和极差问题，确保评审结果准确、客观和权威。这样可以增加参赛者的信心，激发创新能力和创造力，并促进创新竞赛的长期发展和声誉提升。

首先，大规模创新竞赛评审方案应该建立在全面的评审标准和框架基础上，确保评审的科学性。这意味着评审专家应该具备相关领域的专业知识和经验，能够根据命题人（组）提供的评审框架，独立评审作品，并给出符合标准的评分。其次，在评审过程中，应该采用多位评审专家对同一份作品进行评审的方式，以减少评委差异。评审专家之间的交流和讨论也是必要的，这有助于提高评审的一致性和公正性。此外，大规模创新竞赛评审方案可以结合多个评审阶段，如网评、现场评审和答辩。这样可以更充分地考察作品的创新性和可行性，并为评审专家提供更多的评估和参考依据。最重要的是，评审方案应该确保参赛者在评审过程中拥有平等的机会，避免偏见和歧视。这可以通过匿名评审、培训评审专家和建立评审结果校验机制来实现。

目前，各项创新类竞赛都在摸索、调整自己的评审方案。而现有的方案虽然有一定的合理性，但也有其局限性。特别是针对大规模创新类竞赛评审，现有方案偏简单，研究不多。为了建立更加合理、公平的评审方案，我们需要在研究现有方案的基础上提出评审的新方案。

1.2 问题提出

问题一：最优的“交叉分发方案”建立

在每个评审阶段，作品通常都是随机分发的，每份作品需要多位评委独立评审要求我们建立数学模型以确定最优的“交叉分发方案”。如何确定最优的"交叉分发"方案，以增加评审专家之间的可比性？如何确保每份作品都由5位专家独立评审？如何定义和评估"交叉分发"方案的指标？

问题二：改进标准分评分模型

在大规模创新类竞赛评审中，如何**设计新的评审方案**，以**解决标准分评审方案的假设不成立的问题**？如何分析每位专家、每份作品原始成绩和调整后成绩的分布特点，并进行排序比较不同方案的优劣？如何设计新的标准分计算模型，以改进**评审结果的可信度**？

问题三：初评阶段处理“大极差”模型

如何分析两阶段评审方案相比不分阶段评审方案的优劣？如何**建立“极差”模型**，以发掘创新论文并处理第一评审阶段中的“大极差”作品？

问题四：未来评审方案的改进建议

如何建立一个**完整的评审模型来评估“创新类”竞赛的作品**？如何利用所给数据来求解评审模型？如何改进现行的评审方案并提出具体建议？未来还需要收集哪些数据来改进评审方案？

二、模型假设

假设 1：每个专家评阅相互独立，互不影响；

假设 2：各评阅专家分配评阅的作品是均匀的；

假设 3：每个专家的学术水平相同；

假设 4：论文是随机发放的。

三、符号说明

符号	含义
P	评审专家与参赛队伍构成的 0-1 矩阵
P_{ij}	第 i 个专家是否评阅第 j 个作品
λ	幂值参数
D	依赖关系字典
f	匹配函数
$y_i^{(\lambda)}$	正态化
X_k	标准分
α	决策变量
Δu_i	偏移程度
S_i^2	专家对作品 i 评分方差
Z_i	修正的检验统计量

四、问题一：基于单连通的最优“交叉分发”模型

4.1 问题一分析

本题要求我们建立一个最优的“交叉分发”方案，而这个方案的关键就是要满足可比性。在每个评审阶段，作品一般使随机分发且由多个评委独立评审。一方面，不同专家评审的作品集合之间应有交集，有的交集大了，则必然有交集小，这种可比性就弱。因此，建立的方案要使每个作品之间专家评审的交集最大。另一方面，只满足交集最大这条件还不够，若几个专家之间评审的作品相互独立，这些作品之间的可比性也不强或者说没有可比性。为此我们引入单联通区域，专家评阅的这些作品是单联通的。

因此，最终建立的模型要是使得：**在单联通区域内专家评阅的作品交集元素最多且关联交集个数最多**，问题一的求解流程图如图 4-1 所示。

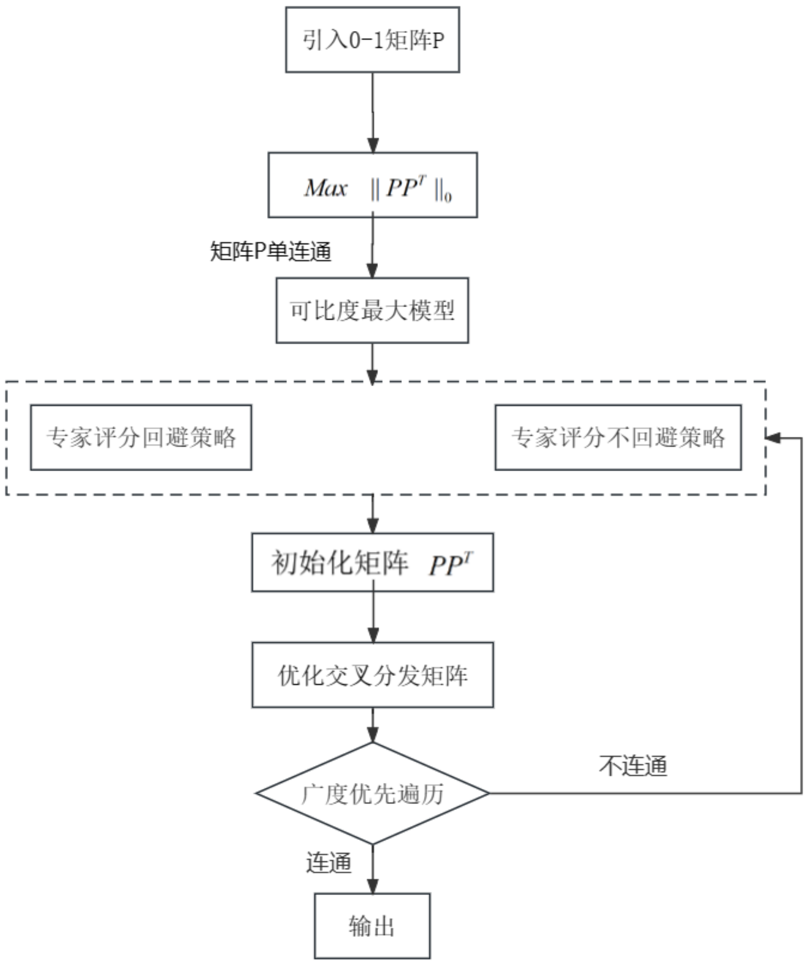

图 4-1 问题一流程图

4.2 模型建立

为保证不同专家评审的作品集合之间具有可比性，不同专家评审的作品之间应有交集。但仅仅满足有交集不够，如图 4-2 所示，专家 1、2、3 虽然有交集，但与专家 4、5、6 不连通。这两个集合组之间处于相互独立的状态，那这两个集合组则没有可比性，而

推广开来，若还存在许多个类似的集合组，那最后的评审得分的整体可比性就会很弱。因此，我们要将这样独立的集合组连接起来，构建**单联通区域**。

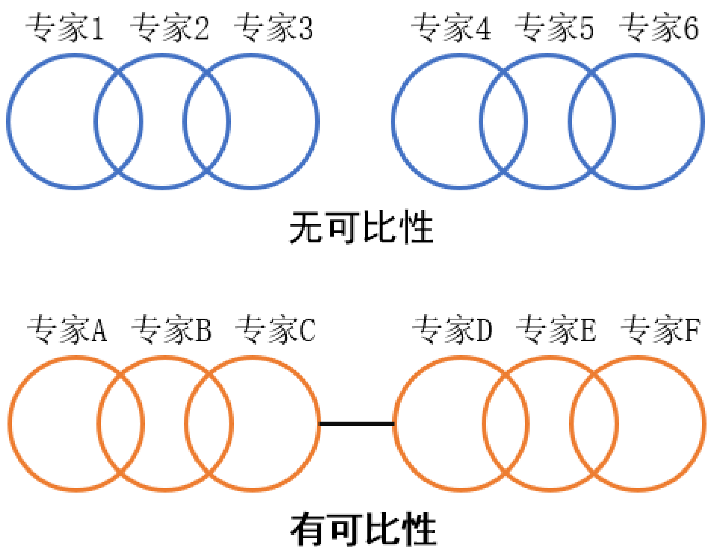

图 4-2 专家评审集合有连通与无连通对比

(1) **决策变量**

根据题目所设条件共有 3000 支参赛队与 125 位评审专家，为此引入决策变量 P_{ij},

$$\text{其中，} P_{ij} = \begin{cases} 1, \text{第i个专家评阅第j个作品} \\ 0, \text{否则} \end{cases} i=1,2,\cdots,125 \quad j=1,2,\cdots,3000$$

因此，可以构建评审专家与参赛队伍的 0-1 矩阵 $P=\left(P_{ij}\right)_{125\times3000}$。图 4-3 给出了具体的矩阵交叉方案。

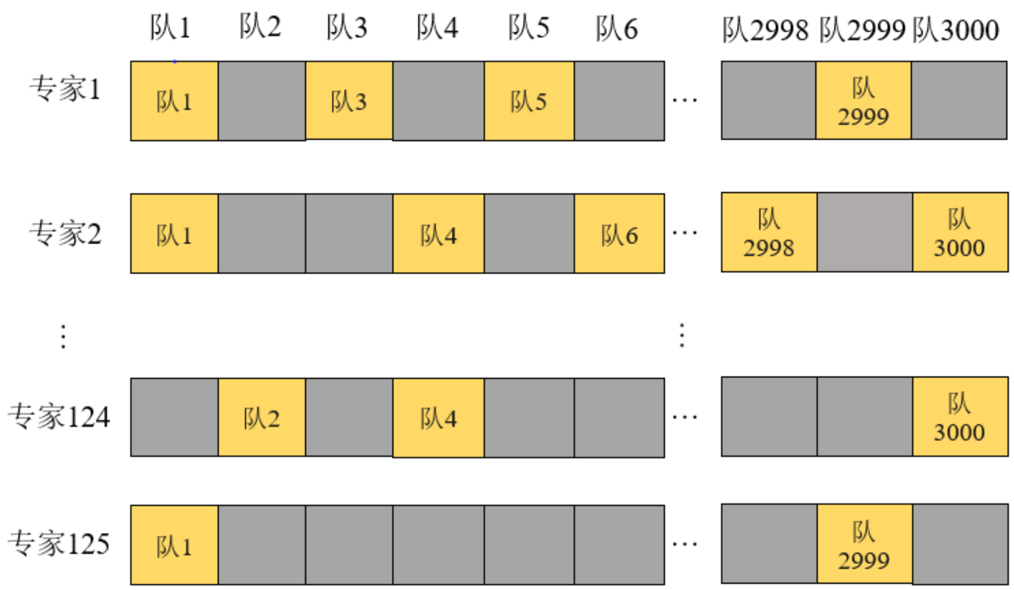

图 4-3 矩阵 P 的示例图

在实际的竞赛评审中，评审专家主要来自不同的高校。为了保证评审的公平性和公正性，一般会采取回避制度，即评审专家不能评阅自己本学校的作品。因此，引入专家所属学校指标集I_s和作品所属学校指标集J_s。若I_s与J_s的 S 相等，则表示专家与作品来自同一学校，否则专家与作品来自不同的学校。

(2) 目标函数：

要建立最优的“交叉分发模型”，即增加不同评审专家所给的成绩之间的可比性，就要让专家评审的作品集合之间的交集最大。

专家 i 与专家 j 评审作品的交集个数为：

$$P_{i-}P_{j-}{}^{T} \tag{3-1}$$

所有专家评审的作品的交集矩阵为：

$$PP^{T} \tag{3-2}$$

目标函数 1：根据题目要求**专家评审的作品集合交集原始总和最大**，其形式如下:

$$\max \left\| PP^{T} \right\|_{1} \tag{3-3}$$

目标函数 2：根据题目要求**专家评审的作品集合交集集合个数最多**，其形式如下:

$$\max \left\| PP^{T} \right\|_{0} \tag{3-4}$$

(3) 约束条件：

1) 约束条件 1：每份作品要被 5 位专家评审：

$$\sum_{i} P_{ij} = 5, \quad j = 1, 2, \cdots, 3000$$

2) 约束条件 2：一共有 3000 份作品，且每份作品要有 5 位专家评审，则每位专家评审 120 份作品：

$$\sum_{j} P_{ij} = 120, \quad i = 1, 2, \cdots, 125$$

3) 约束条件 3：要使方案间可比性强，则使 PP^{T} 矩阵是单连通的

综合上述情况，得到**双目标“交叉分发”优化模型：**

$$\max \left\| PP^T \right\|_0$$
$$\max \left\| PP^T \right\|_1$$
$$s.t.\begin{cases} \sum_i P_{ij} = 5, \ \ j = 1, 2, \cdots, 3000 \\ \sum_j P_{ij} = 120 \ \ i = 1, 2, \cdots, 125 \\ PP^T\text{是单联通的} \end{cases} \tag{3-5}$$

求解该双目标优化模型之前，先介绍一些关于本题中范数的一些特性。同时，介绍一些基本的变换规律。

定义 1（**等** $\left\| \boldsymbol{PP^T} \right\|_1$ **值变换**） 如果 $\boldsymbol{P}$ 的第 i 行元素 $\boldsymbol{P_{ij}}$ (=1)与 $\boldsymbol{P_{ik}}$ (=0)交换，同时存在第 t 行 $\boldsymbol{P_{tj}}$ (=0)与 $\boldsymbol{P_{tk}}$ (=1)交换，那么这种交换就称为等 $\left\| \boldsymbol{PP^T} \right\|_1$ 值变换。

显然，等 $\left\| \boldsymbol{PP^T} \right\|_1$ 值变换的过程，保证 $\left\| \boldsymbol{PP^T} \right\|_1$ 的值不变。

定义 2 (**特殊情况矩阵** $\boldsymbol{P}^*$)：把竞赛论文 1 号-120 号分配给 1-5 号专家评委，竞赛论文 121 号-240 号分配给 6-10 号专家评委，以此类推得到特殊的特殊情况矩阵 $\boldsymbol{P}^*$。显然 $\boldsymbol{P}^* \boldsymbol{P}^{*T}$ 为多连通矩阵，不满足对比性原则。

通过计算，可得 $\left\| \boldsymbol{P}^* \boldsymbol{P}^{*T} \right\|_1$ =75000。为此本文得到关于 $\left\| \boldsymbol{PP^T} \right\|_1$ 的一般性规律。

定理 1： 任意符合**约束条件** 1 和**约束条件** 2 的矩阵 $\boldsymbol{P}$，其 $\left\| \boldsymbol{PP^T} \right\|_1$ 值恒定为 75000。

证明过程简单，作如下解释。通过定义 1 和定义 2 的设定，证明过程中只需要证明任何符合**约束条件** 1 和**约束条件** 2 的矩阵，经过等 $\left\| \boldsymbol{PP^T} \right\|_1$ 值变换到特殊情况矩阵 $\boldsymbol{P}^*$。变换过程如下：如果 $\boldsymbol{P}$ 的第 1 行元素 $\boldsymbol{P}_{11}$ 为零，根据**约束条件** 1 和**约束条件** 2，则可以进行 $\left\| \boldsymbol{PP^T} \right\|_1$ 变换使得元素 $\boldsymbol{P}_{11}$ 为 1，同理 $\boldsymbol{P}_{12}$ 也可以同样找到为 1 的等 $\left\| \boldsymbol{PP^T} \right\|_1$ 变换。

综上所述，为此，双目标优化模型可简化为单目标优化。则**最优"交叉分发"模型**为

$$Max \quad \| PP^T \|_0 \tag{3-6}$$

$$s.t.\begin{cases} \sum_i P_{ij} = 5, \ \ j = 1, 2, \cdots, 3000 \\ \sum_j P_{ij} = 120 \ \ i = 1, 2, \cdots, 125 \\ PP^T\text{是单联通的} \end{cases} \tag{3-7}$$

回避制度的思考：

在分派论文的时候一般需要考虑 2 个原则：即**独立随机原则**和**同校同区域回避原则**。独立随机原则自然遵从，这个在编程序时候编写随机分配。而回避原则需要进一步讨论。本文把 J_s **分别表示为相应的同校同区域的论文集合**。那么回避原则就可以写成如下：

回避约束 $$\sum_{j\in J_s} P_{ij} = 0.$$

这时，我们就可以得到**基于回避制度的最优“交叉分发”模型：**

$$Max \quad \| PP^T \|_0 \tag{3-8}$$

$$s.t.\begin{cases} \sum_j P_{ij} = 5, \ \ j = 1, 2, \cdots, 3000; \\ \sum_i P_{ij} = 120 \ \ i = 1, 2, \cdots, 125; \\ P\text{是单联通的}; \\ \sum_{j\in J_s} P_{ij} = 0. \end{cases} \tag{3-9}$$

4.3 模型求解

4.3.1 求解模型

在求解是首先对按照约束条件运动动态规划策略逐行对矩阵 P 进行随机初始化，然后运用贪心策略进行优化：

算法 1 贪心算法优化交叉分配方案

输出：专家参赛队分配矩阵 P

1. 初始化零矩阵 P，行数 125，列数 3000
2. 生成参赛队对受评阅次数的关系字典 D，每一参赛队初始值为 5
3. FOREACH P_i
4. 将剩余次数与全局最大剩余次数相等的参赛队纳入可选列表
5. IF 可选列表<120
6. 从第二大剩余次数的参赛队中补足 120
7. END IF
8. 从可选列表中随机选取 120 支参赛队分配给评审专家 i
9. 被选区的队伍在字典中的相应元素减一

10.计算 $T_0=\sum_{i=0}^{125}\sum_{j=0}^{125} P_i P_j^T$ ，K0=$\| PP^T \|_0$

11. FOREACH P_i
12. 检查第 i 行每一个非零元素将其移动到相邻位置后计算 T*K
13. IFT*K>T0*K0
14. T0*K0=T*K

15. 更新元素位置

16. END

求解完成后根据可比性原则需要对矩阵 P 进行检验，矩阵 PP^T 可看成图结构 G 的邻接矩阵与度矩阵相乘的结果，为保证所有作品的可比性一致，该图应为连通图，可运用广度前向传播算法进行检验，如果该算法可一次遍历所有节点则该图是连通的，可比性检验通过。

4.3.2 计算结果

表 4-1 交叉方案优化结果

	未回避	回避
$\| PP^T \|_0$	12549	12476
$\| PP^T \|_1$	75000	75000

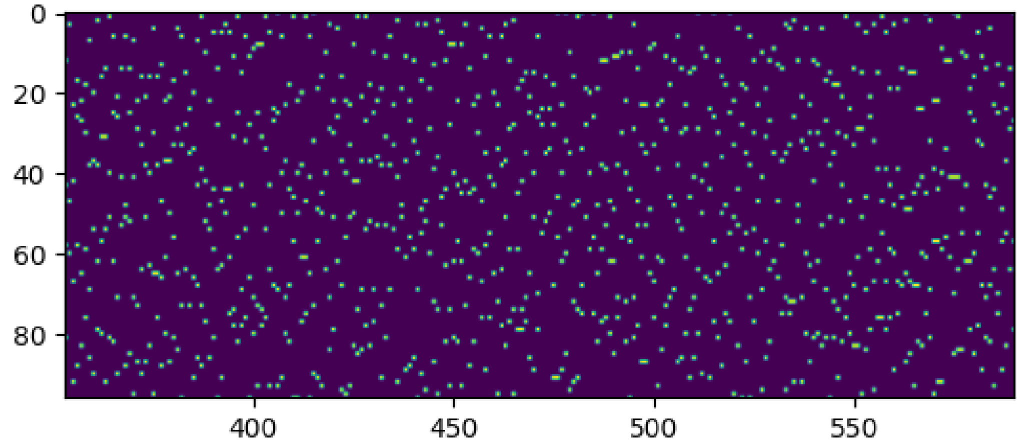

图 4-4 交叉分发方案结果

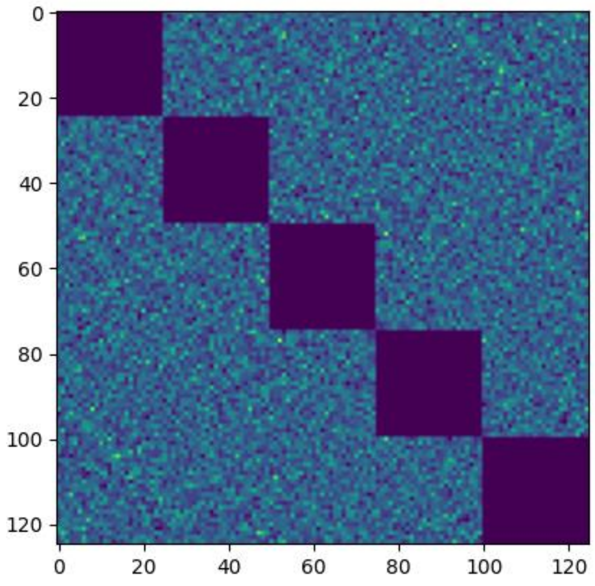

图 4-5 连通性检验

交叉分配方案节选如上图所示，P 矩阵为稀疏矩阵，该分配方案能使得 1 值较为均匀的填充到 P 矩阵中，又能兼顾随机性、公平性原则。下图所示为 P 矩阵的连通性，由于初始化方案优先考虑剩余选择次数较多的参赛队伍，因此未连通区域集中在对角线附近，有违均匀随机分配的原则，但依旧尽最大可能保持了连通性。

4.4 模型小结

针对问题一，在单连通区域内构建可比度最大模型，通过变换专家评分矩阵将双目标优化问题简化为 0-1 规划的单目标优化问题。在模型建立的过程中分别考虑了专家评分回避和未回避的策略；在模型求解过程中，首先采用基于动态规划算法初始化交叉分发矩阵，然后采用贪心算法优化交叉分发方案，最后使用广度优先遍历检验图是否连通。未考虑回避策略时，$\|PP^T\|_0=12549$；考虑回避策略时，$\|PP^T\|_0=12476$。

五、问题二：标准分的赋权排序模型

5.1 问题二分析

在大规模创新类竞赛评审中，由于每位评审专家只看到作品集合的很小部分，因此标准分评审方案的假设可能不成立，需要探索新的评审方案。

评审指标之间的相关性、评审指标的分布特点以及评审指标的权重分配等方面的不确定性和主观性。因此，在设计评审方案时，需要充分考虑实际情况，合理选择评审指标和权重分配的方法，以确保评审方案的有效性和准确性。

按照题意我们先分析每位专家、每份作品原始成绩、调整之后（如取标准分）成绩的分布特点，考察其统计描述特点如均值和方差。在改进标准分模型时，建立一个排序匹配函数，将我们提出的方法得到的分数与第二次评审的标准分匹配。将不同奖项赋不同的权值得分，若改进后的排序与原排序相同，则加上相应匹配成功对于奖项的权值得分，最后求其权值得分总和。将各方案的匹配得分作比较，选取匹配得分大的方案作为最终的标准分的优化方案。如图 5-1 所示。

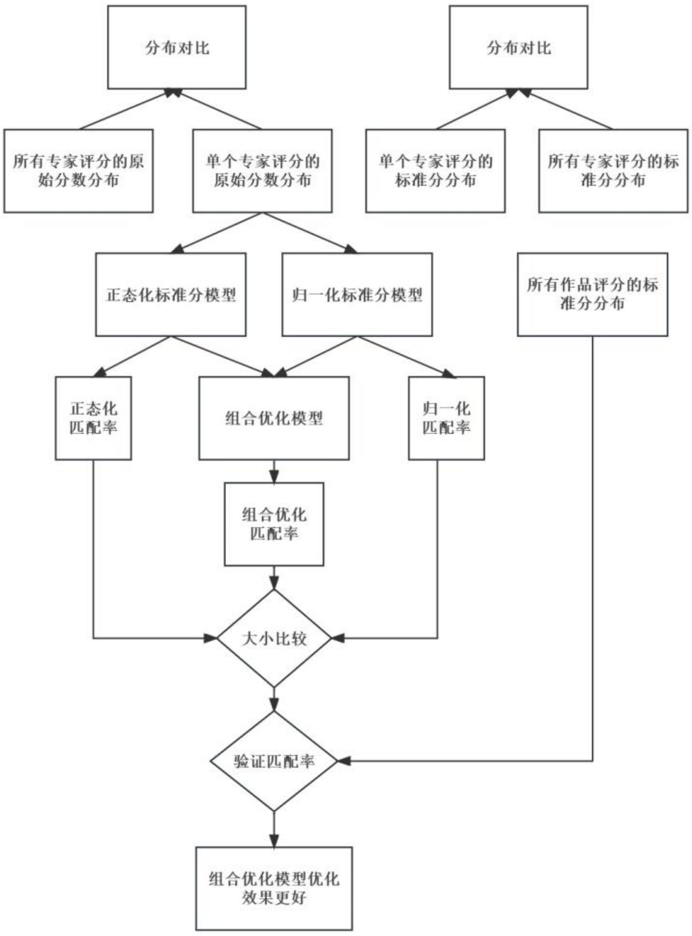

图 5-1 问题一流程图

5.2 数据统计

因为专家和作品的数据大，考察比较专家与作品的成绩分布我们选择抽取典型案例研究。对于专家打分成绩的分布，我们专家打分成绩的原始分和标准分与总体的原始分和标准分的分布对比，对其进行描述性统计分析。类似地，对于作品的分布，也对其进行描述性统计分析。

5.3 *t* 检验：数据分布情况检验

两阶段优劣对比

在大规模创新类竞赛中，赛题紧贴前沿评审难度大，专家数量又收到限制，这种情况下作品的分数很容易受到评审专家偏好的影响。这时便需引入二阶段评审来弥补一阶段的不足。在本小节中运用**双样本 t 检验**将第一阶段与第二阶段的评分成绩与最终成绩做对比，可构造统计量：

$$t=\frac{\overline{X_1}-\overline{X_2}}{\sqrt{\frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}(\frac{1}{n_1}+\frac{1}{n_2})}} \tag{4-1}$$

分别将第一阶段、第二阶段与最终成绩排序，将不同阶段的排序同最终成绩做双样本 t 检验

表 5-1 双样本 *t* 检验结果

	第一阶段	第二阶段
P	0.8561	0.9627

从表 5-1 中可以看出，第一阶段与第二阶段成绩同最终成绩均不存在显著性差异，但第一阶段同最终成绩的差异较第二阶段更大。

5.4 专家打分的数据

1) 专家评分原始分数分布与标准分对比（专家抽样样本与总样本的对比）

该表格有 97 位评委，若将这 97 位评委一一分析的话篇幅有限，我们通过上面的处理将 97 位评委分离出来，然后从中选取 4 位评委。当然我们选取的 4 位评委是具有典型代表性的，对其评分的成绩进行汇总。

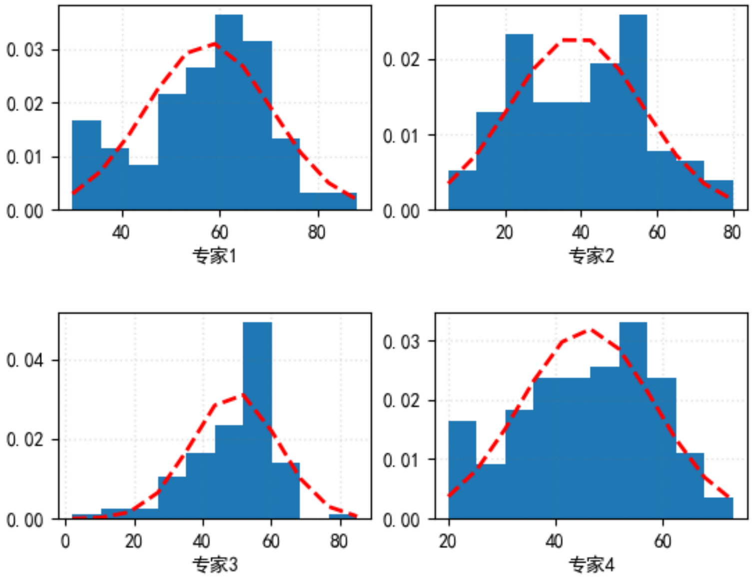

图 5-2 4 位专家评分成绩的概率密度

专家 1 所给分数的分布呈右偏峰型，其主要聚集在 60 分左右，数据向 60 分以上倾斜；而专家 2 的打分呈现出明显的双峰型，在 20 和 60 分周围存在明显的峰值，该专家对于给出 20 和 60 分的概率非常大；专家 3 的打分均有极强的聚集特性，其分布为左偏峰型，该专家对于作品的给分在[50,60]之间；专家 4 评分分数分布左偏，分数分布较为均衡，与原始分数总体分布情况相似。

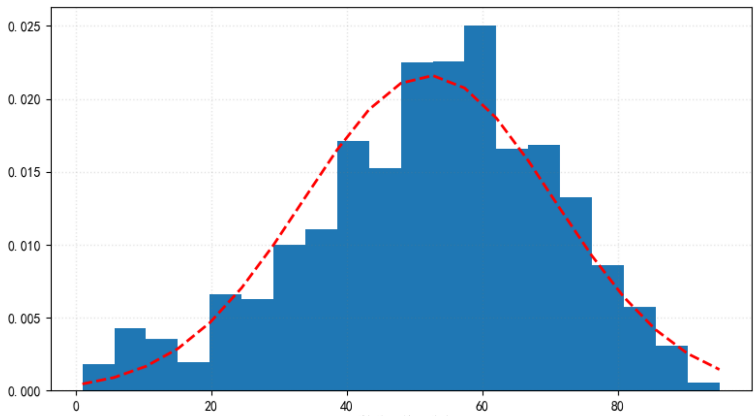

图 5-3 原始分数的总体分布

为了进一步研究四位专家的评分分布与总体评分分布的情况，将四位专家评分的原始分数与原始分数总体分布的均值、方差、偏度、峰度作对比，如表 5-2 所示。

从表中可知，四位专家评分的原始分数均值均大于原始分数总体评分均值，说明该四位专家评分总体偏高；四位专家评分的原始分数方差都比总体方差小，说明四位专家评分的原始分数较为稳定；原始分数总体分布左偏，大部分作品评分在均值以下，专家 1 与专家 2 评分的原始分数则大部分在均值之上；根据峰度值可知，四位专家评分的原始分数分布均没有原始分数总体分布紧密。根据 t 检验的值，**专家评分的原始分数与总体评分的原始分数的分布不是同一分布。**

表 5-2 专家评分与总体评分的原始分数分析

	均值	方差	偏度	峰度	t 检验值
专家 1	75.2	32.96	0.41	2.53	0.57
专家 2	76.6	61.84	0.30	1.26	0
专家 3	69.6	101.84	-0.22	1.84	0
专家 4	75.8	178.96	-0.37	1.18	0
总体	52.13	340.79	-0.40	2.76	

2) 专家评分标准分分布与总体标准分对比（专家抽样样本与总样本的对比）

采取上述四位专家的标准分，对他们对各自评阅作品标准分的分布情况进行分析（如图 5–4 所示），并与总体标准分分布情况（如图 5-5 所示）作对比。

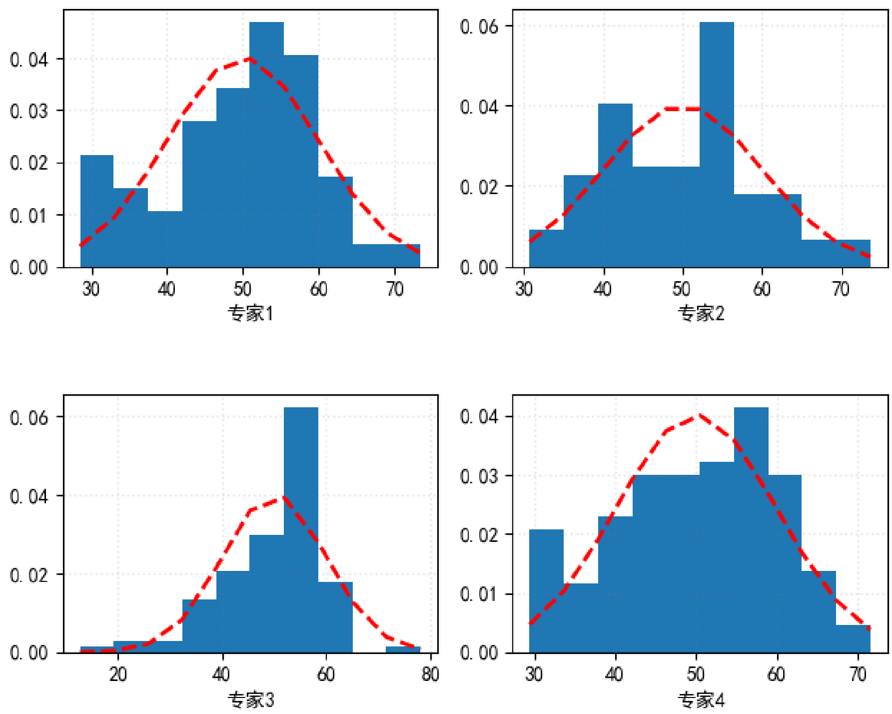

图 5-4 四位专家评分的标准分分布

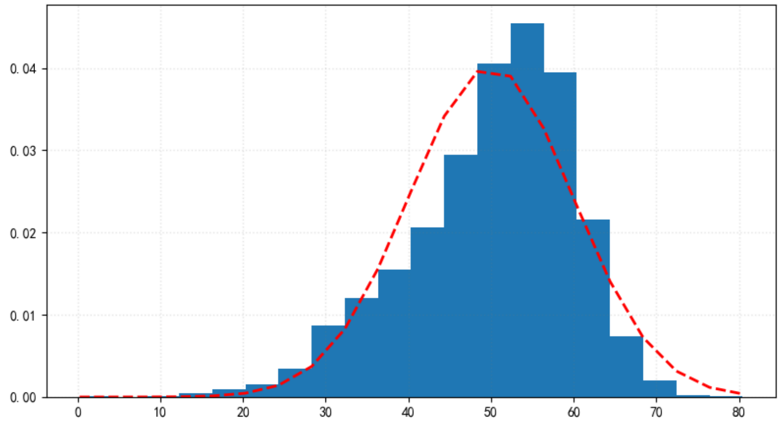

图 5-5 标准分总体分布

专家 1 评分的标准分的分布右偏，与标准分总体分布偏向相反；专家 2 评分的标准分的分布左偏，与标准分总体分布相比，数据分布较为均衡；专家 3 评分的标准分的分布右偏，其数据分布与标准分总体分布贴近，数据分布紧密；专家 4 评分的标准分的分布左偏，与标准分总体分布相反，分数分布较为均衡。

与此同时，将四位专家评分的原始分数与标准分总体分布的均值、方差、偏度、峰度作对比，如表 5-3 所示。

表 5-3 专家评分与总体评分的标准分分析

	均值	方差	偏度	峰度	t 检验值
专家 1	70.01	26.87	0.14	2.05	0.99
专家 2	68.07	7.35	-0.25	1.24	0
专家 3	61.78	13.47	0.53	2.11	0
专家 4	63.05	7.89	-0.16	1.52	0
总体	49.99	98.87	-0.70	3.44	

根据表中内容可以发现，四位专家评分的标准分均值均大于标准分总体评分均值，说明该四位专家评分总体较高；四位专家评分的标准分方差都比总体方差小，说明四位专家评分的标准分较为稳定；标准分总体分布左偏，大部分作品标准分在均值以下，专家 1 与专家 3 评分的标准分则大部分在均值之上；根据峰度值可知，四位专家评分的原始分数分布均没有原始分数总体分布紧密；根据 t 检验可知，专家 1 的 t 检验的值为 0.99，因此可以将专家 1 评分的原始分数与总体评分的原始分数是同一分布的；其余专家则与总体不是同一分布。

5.5 原始标准分方法的局限性

标准分评审作为现有的竞赛评审方案之一，在竞赛的评审中应用比较广。该方法要求对每位评审专家的评分进行标准化，按作品分相加得每件作品的总分，然后以此排序。标准分的计算如下：

某位专家给出的成绩，有 n 个样本 $a_1, a_2, \cdots, a_n$ （不同专家向量不同）

表 5-4 附件中标准差算法

含义	符号
某位专家给出成绩的样本均值	$a_1, a_2, \cdots, a_n$
某位专家给出成绩样本标准差	$S=\sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(a_k-\overline{a})^2}$
标准分	$x_k = 50+10\times\frac{a_k-\overline{a}}{s}$

(1) 标准分不能全面反映选手的实力：标准分是通过比较选手在比赛中获得的总分数和平均分数计算得出的，它只反映了选手在比赛中的平均表现水平，不能全面反映选手的实力。例如，一名选手在比赛中获得了较高的分数，但是这个分数是在相对较弱的对手或者较低难度的比赛环节中获得的，因此标准分可能不能真实反映该选手的实力水平。

(2) 标准分不能反映比赛的难度和选手的特长：标准分是根据选手在比赛中的平均表现水平和比赛环节的难度计算得出的，它不能反映比赛的难度和选手的特长。例如，一个擅长难度较高的动作的选手，在比赛中表现出色，获得了较高的分数，但是由于比赛环节难度较低，标准分可能不能真实反映该选手的特长和实力。

(3) 标准分可能会导致不公平的竞争：由于标准分是根据选手在比赛中的平均表现水平和比赛环节的难度计算得出的，它不能反映选手在比赛中的实际得分和排名情况，因此可能会导致不公平的竞争。例如，一个选手在比赛中获得了较低的分数，但是因为其他选手表现更差，该选手还是能够获得较高的标准分，这种情况可能会对其他选手造成不公平的竞争。

总之，虽然标准分评价具有客观性、可比较性等优点，但它也存在局限性。因此，在比赛评分中，应该考虑使用其他评价方法如加权平均、模糊评价等来克服这些局限性。同时，也应该注意到各种评价方法都有其适用范围和限制条件，要根据实际情况选择合适的评价方法。

5.6 匹配函数

为了比较正态化方案和归一化方案的优劣，我们拟改进的方案求得数分数与原标准分比较。为了量化比较出两个方案的优劣，必须引入参数来计算。

一般认为经多位专家协商一致的获奖论文具有最大的可信度，其第二评审阶段评选出的一等奖作品排序是经专家协商取得一致。引入匹配函数 f，若改进后的方案得到的奖项与原标准分的匹配上，则赋相应的权值。一等奖匹配上等 10 分，二等奖匹配上得 5 分，三等奖匹配上得 2 分。最后得到 f 函数的表达式。

$$\Delta f=\begin{cases}10 & \text{一等奖匹配成功}\\ 5 & \text{二等奖匹配成功}\\ 2 & \text{三等奖匹配成功}\end{cases} \tag{4-2}$$

5.7 方案一：正态化标准分模型

根据题目要求可知，附件 1 中的标准分评审方案存在弊端；从专家评分成绩的分布情况分析中可知，无论是专家的分布数据还是整体的分布数据都是偏正态。

针对该问题，我们采用 Box-Cox 变化，提出一种正态化标准分优化模型，将数据正态化之后再求其标准分，要将偏正态的数据调整为近似正态的数据分布。

Box-Cox 变化作为一种数据正常化的正态变换方法，通过确定一个合适的指数(λ)，将数据转换为正态分布。λ 值指的是所有数据应变换的幂值；本题是基于一个参数的 Box-Cox，其正态化标准分优化模型为：

$$y_i^{(\lambda)}=\begin{cases}\dfrac{y_i^{\lambda}-1}{\lambda} & if\ \lambda\neq 0\\ \ln y_i & if\ \lambda=0\end{cases} \tag{4-3}$$

本题选择某位专家评分的原始分数据，并进行正态化。经过多次调试，当 λ=1.6 时，该专家的原始分数据由偏正态分布调整为最优近似正态分布，如图 5-6 所示。

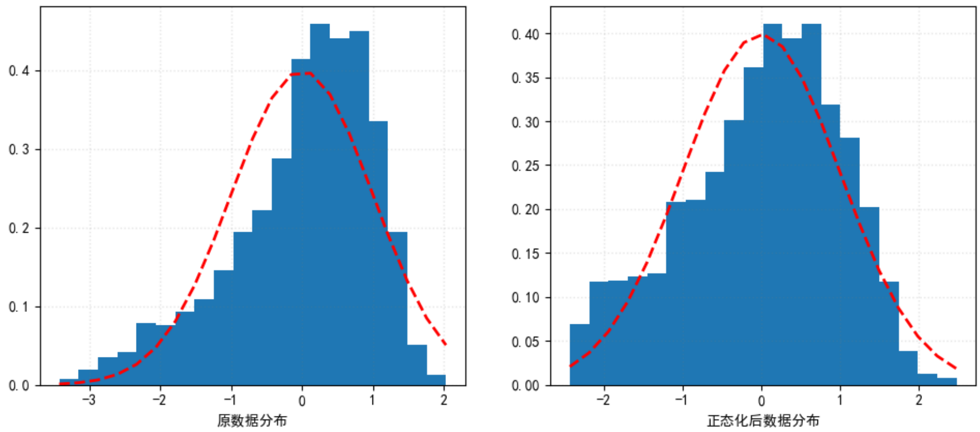

图 5-6 原始数据正态化示例

由上图可知，本题利用的基于 Box-Cox 的正态化标准分优化模型，可以高效地将是偏正态分布数据进一步调整为近似正态分布数据，从而提高数据的准确性，由此证明，本模型具有很高的可靠性。

通过归一化优化的标准分模型计算所有作品得分。因为第二阶段评审的结果是专家一起讨论通过的，其可行度较高，所以我们将归一化优化的得分与第二次的标准分比较。比较之后得到的数据如下。

表 5-5 正态化优化计算分数与二次评分标准分的匹配个数

	一等奖	二等奖	三等奖
匹配个数	6	155	2

按照匹配函数计算得分为：

$$f=6\times10+155\times5+2\times2=839$$

考察不同奖项的匹配程度，我们汇总计算得匹配率得到下表:

表 5-6 正态化优化计算分数与二次评分标准分的匹配个数

	一等奖	二等奖	三等奖
匹配率	0.241379	0.578947	0.035714

使用正态化优化模型输出的匹配率中，二等奖的匹配的成功率接近 60%。但这个模型的匹配效果仍然不好，因为对于最重要、可行度最高的一等奖的匹配率非常低。

5.8 方案二：归一化标准分模型

从标准分的计算公式可以看出，其实质是将每个成绩控制在 50 左右浮动。一个自然而然的思想就是优化 $\frac{a_k-\bar{a}}{s}$ 这部分式子。我们可以将 $\frac{a_k-\bar{a}}{s}$ 以归一化的方法转换成 $\frac{a_k-\min a_k}{\max a_k-\min a_k}$，最终优化得到的标准分计算公式为：

$$x_k=30+60\times\frac{a_k-\min a_k}{\max a_k-\min a_k}\in[30,90] \tag{4-4}$$

通过归一化优化的标准分模型计算所有作品得分。因为第二阶段评审的结果是专家一起讨论通过的，其可行度较高，所以我们将归一化优化的得分与第二次的标准分比较。比较之后得到的数据如下。

表 5-7 归一化优化计算分数与二次评分标准分的匹配成功的个数

	一等奖	二等奖	三等奖
匹配个数	8	186	13

按照匹配函数计算得分为：

$$f=8\times10+186\times5+13\times2=1036$$

比较正态化和归一化方案的匹配得分，显然归一化优化效果比正态化的优化效果好

为了考察不同奖项的匹配程度，我们汇总计算得匹配率得到下表：

表 5-8 归一化优化计算分数与二次评分标准分的匹配个数

	一等奖	二等奖	三等奖
匹配率	27.5862%	69.9248%	23.2143%

使用归一化优化模型输出的匹配率中，二等奖的匹配的成功率较高，接近 70%。但这个模型的匹配效果仍可以进一步改进，因为对于最重要、可行度最高的一等奖的匹配率非常低，还需寻找更优的模型。

5.9 方案三：组合优化融合模型

本题分别抽取一等奖、二等奖、三等奖、未获奖的参赛作品的原始分数，求出该四

个作品的均值、方差、偏度和峰度，由此可以发现获奖等级越高，其均值越大，方差越大，偏度越小，峰度越小，如表 5–9 所示。

表 5–9 参赛作品原始数据抽样

	专家1	专家2	专家3	专家4	专家5	均值	方差	偏度	峰度
一等奖	89.00	59.00	60.00	86.00	85.00	75.80	223.7	0.82	0.52
二等奖	83.00	65.00	74.00	67.00	57.00	69.20	96.2	1.04	1.50
三等奖	68.00	60.00	73.00	63.00	59.00	64.60	34.3	1.03	1.24
未获奖	36.76	41.84	42.70	46.08	42.58	41.99	11.23	1.36	1.95

利用组合原则，融合了上述两个模型，设计了**单目标优化的模型**：

决策变量：α

目标函数：$Max \quad M(\alpha)=\alpha T+(1-\alpha)Q$

约束条件：$0<\alpha<1$

其中 T 是正态化后的标准分，Q 是归一化后的标准分。

本题借鉴投票的方法对该模型进行参数调整，选取**所有的获奖作品的标准分**，作为目标数据，则**参赛作品参数优化模型**：

$$M=0.57*T+0.43*Q$$

则组合优化后的奖项匹配率如表 5-9 所示。

表 5-10 组合优化后各等级奖项的匹配率

奖项	原始匹配率	正态化匹配率	归一化匹配率	组合优化匹配率
一等奖	25%	24.1379%	27.5862%	32.1429%
二等奖	55%	57.8947%	69.9248%	76.779%
三等奖	3%	3.5714%	23.2143%	19.6429%

将上述参赛作品数据代入，其组合优化匹配率不变，从而验证了该组合优化模型的可靠性。

表 5-11 组合优化计算分数与二次评分标准分的匹配成功的个数

	一等奖	二等奖	三等奖
匹配个数	8	206	11

按照匹配函数计算得分为：

$$\boldsymbol{f=8\times10+206\times5+11\times2=1132}$$

比较正态化和归一化方案的匹配得分，显然组合优化的匹配得分比前两种方案高，匹配的效果相对好些。

5.10 模型小结

针对问题要求考察每位专家、每份作品的原始成绩、调整之后（如取标准分）成绩的分布特点。由于数据中的专家有 97 位，参与评审的作品有 2015 个，对于这大量样本我们选择抽取其中一些典型案例考察。在对专家原始评分成绩分布与总体原始分分布、专家评分标准分成绩分布与总体标准分分布作 T 检验之后，他们都不是同一分布。

对于标准分的优化，我们采用正态化优化、归一化优化和组合优化标准分模型。正态化优化是将原来成绩偏正态的分布转化为正态分布以实现评分的优化；归一化是将原来标准分的公式映射到特定区间；组合优化模型则是结合前两种方案的长处来实现标准分的优化。为了对方案进行排序，求得原始数据的匹配率、正态化的匹配率、归一化的匹配率和组合评价的匹配率，求得的结果是组合方案匹配效果最优。

其排序如下：

组合优化方案>归一化方案>正态化方案>原始方案

显然，我们提出的方案相对于原数据都有提升。

六、问题三的模型建立与求解

6.1 问题三分析

在问题三中，本文详细地分析了两阶段的成绩总体变化、极差的分布变化以及不同阶段作品极差的之间的相关性，对比了不同奖项作品分数的极差分布规律，从而进一步探索了极差与创新性之间的联系，可以建立多步极差调整算法，从而提高第一次评审阶段程序化处理“大极差”的准确率。

6.2 数据统计分析

选取数据二 9329 支参赛队的数据进行分析。首先对数据进行汇总，分别按作品以及专家两个维度进行数据的汇总，得到 9329 支参赛队与 380 为评审专家表格。因为在第一阶段每支参赛队仅由 5 位专家评审，因此矩阵存在稀疏性，稀疏度 98.68%。

6.3 两阶段成绩整体变化和两阶段极差整体变化

为探讨极差与创新性的关系，首先藐视两阶段的成绩整体的变化和两阶段极差整体的变化，对比两阶段评审方案与不分阶段评审方案的结果。探索极差大和创新性强之间的关系。

6.3.1 两阶段成绩整体变化

根据题目要求，描述两阶段成绩的整体变化，需要兼顾到平均值、极差等数据，因此利用箱线图来描述两阶段评审的成绩总体变化，如图 6-1 所示。

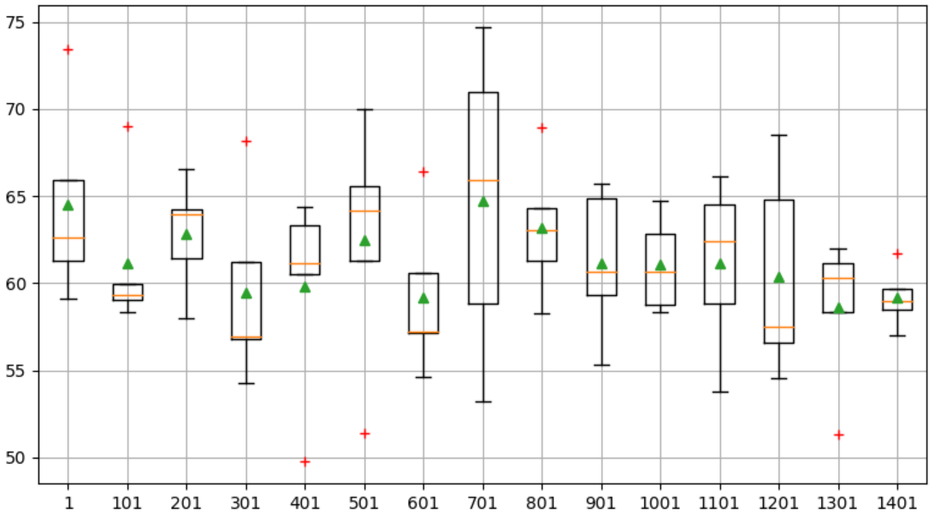

图 6-1 第一阶段评审成绩箱线图

从第一名作品开始，每隔 100 名绘制一个箱线图。由图可知，每个箱子都包含评审分数的平均值、中位数、异常值、最大值、最小值。从异常值来看 1500 个作品如排名一般呈现下降趋势，但均值、最大值、最小值等的下降趋势表现并不明显，作品之间仅存在细微差别。

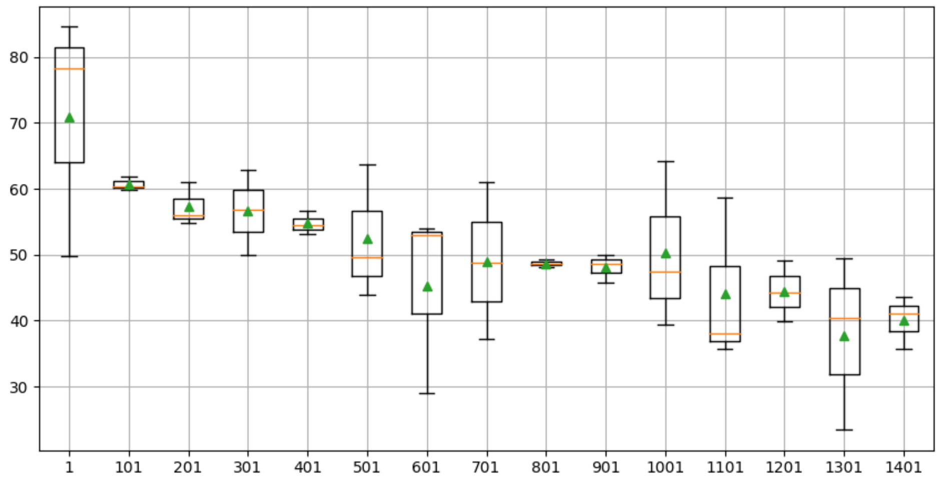

图 6-2 第二阶段评审成绩箱线图

由图 6-2 可以发现，第二阶段评审的成绩呈现了明显的下降趋势，每一个作品分数的离散程度急剧缩小，最大值、均值等下降趋势明显，并且异常值大幅减少。

6.3.2 两阶段极差整体变化

极差可以描述不同专家对一个作品评价差异的大小，也从另一方面反映的作品所存在的争议性。对于创新型竞赛，不同专家较难达成共识，因此极差的存在同时反映了专家偏好与作品争议两个方面。将两阶段前 1500 名的作品的原始分数分别计算极差绘制直方图如图 6-3 所示。

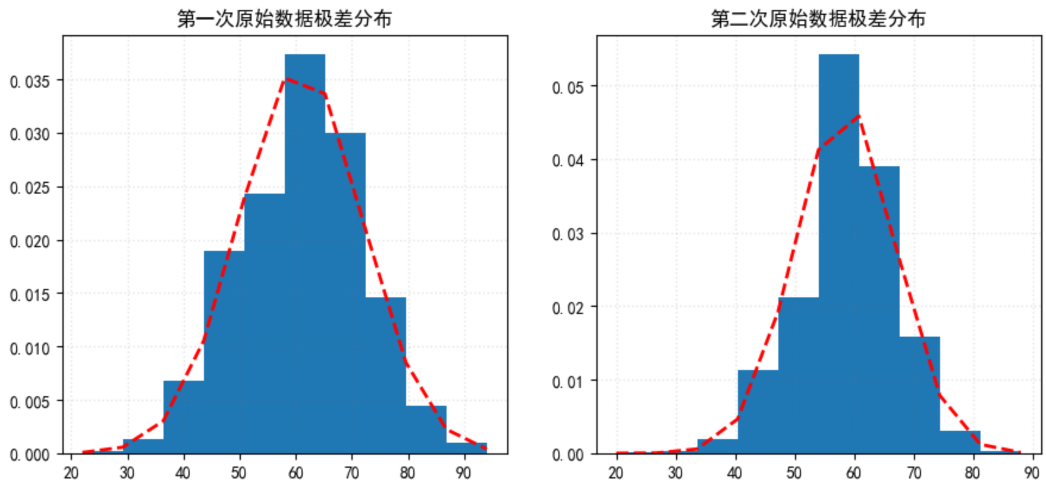

图 6-3 前两次原始数据极差分布图

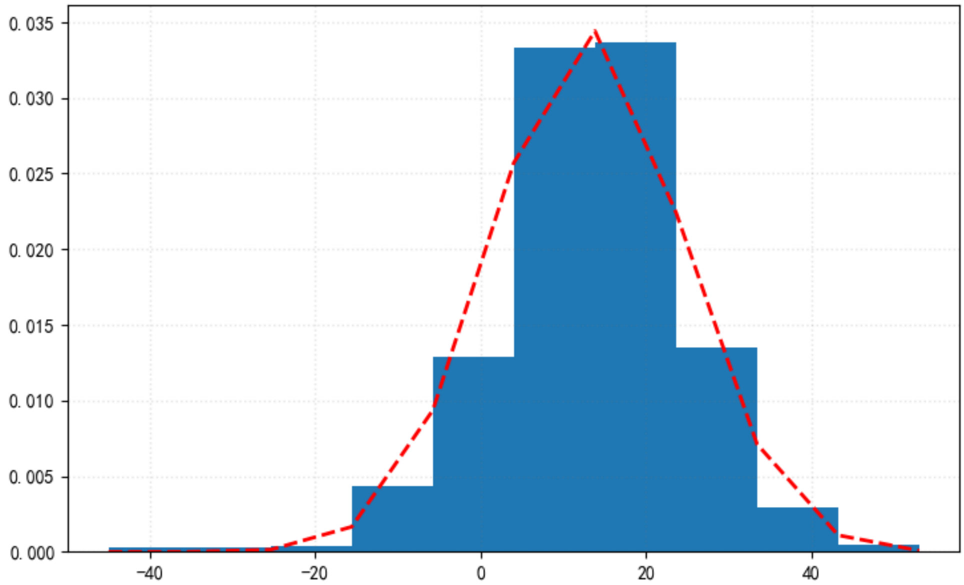

图 6-4 两阶段评审极差对比

从图 6-4 可以看出，两次评审的原始评分极差均值大抵相同，但第一次评审的极差分布范围更大，离散程度更高，而第二阶段评审的极差分布范围较为集中。对每个作品两次评审的极差相减可以看出第一阶段的极差总体上大于第二阶段的极差，说明分两阶段评审可以降低作品评审的争议性。

6.3.3 两阶段评审方案与不分阶段评审方案优劣

两阶段评审可以充分照顾到第一阶段由于专家偏好作品争议等问题产生的不公平，下图奖第一次评审、第二次评审与综合成绩做 Pearson 相关性检验，可以看出第二次评审的成绩更接近综合和成绩，而第一次评审成绩与综合成绩相关性较弱，二次评审对第一次评审做了极大的修正。

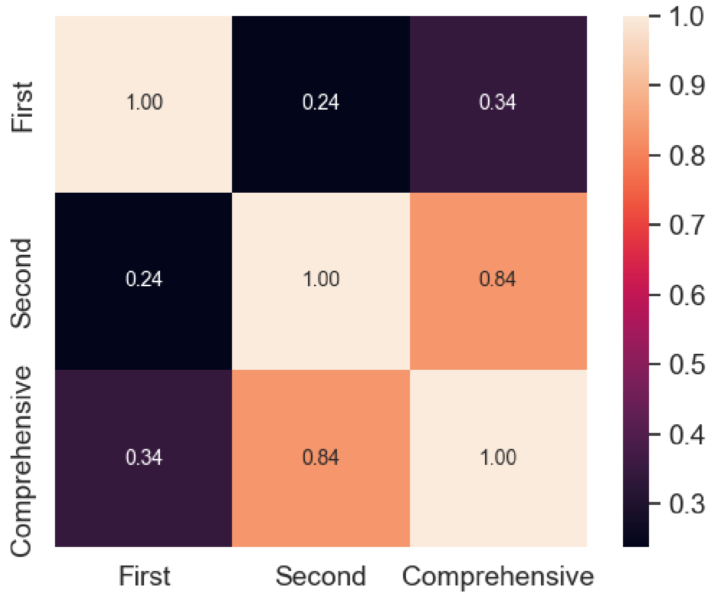

图 6-5 两次评审与综合成绩相关性检验

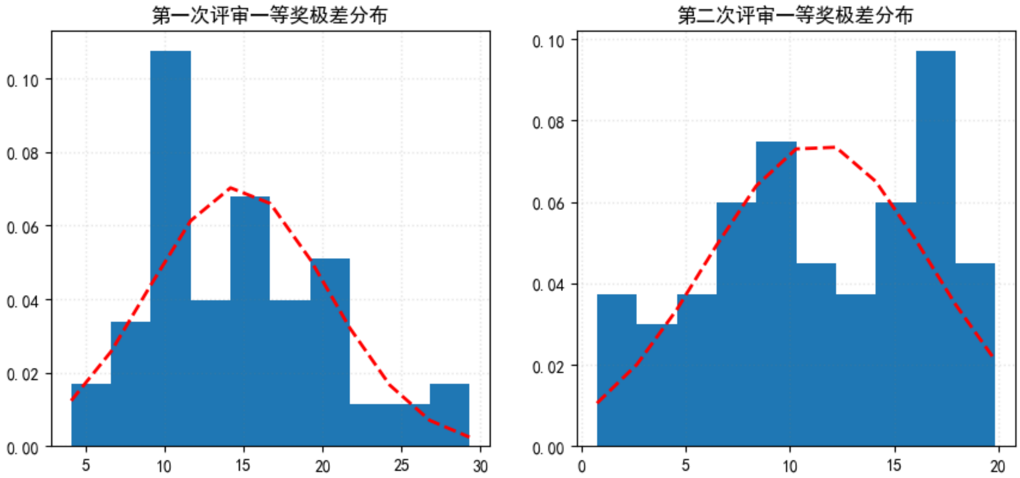

图 6-6 两次评审的一等奖极差分布

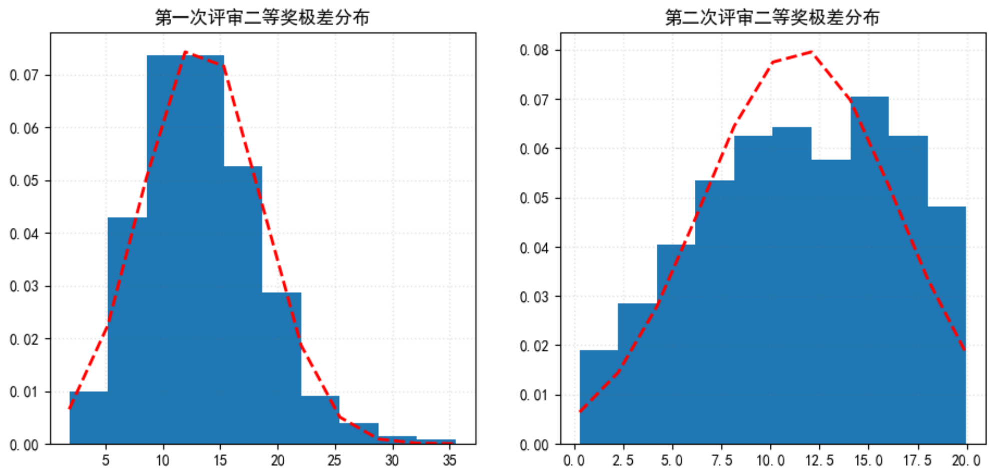

图 6-7 两次评审的二等奖极差分布

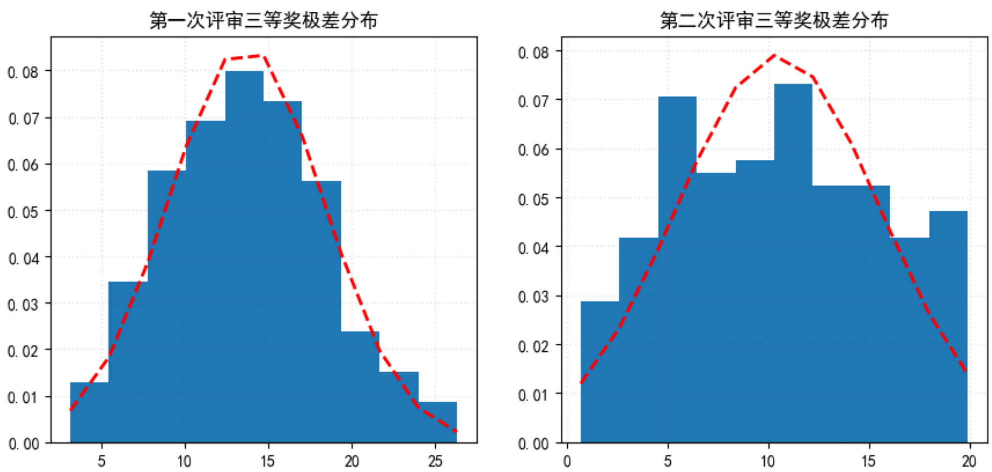

图 6-8 两次评审的三等奖极差分布

按照一、二、三等奖分别对原始成绩的极差分布做比较。不难看出第二次评审一、

二、三等奖的极差范围相比第一次评审都大幅度缩小，分布也更均匀，专家对作品的争议性进一步得到了降低。

6.3.4 两阶段评审极差与创新性的联系

大规模创新类竞赛由于其“创新性”的特点，没有标准答案，对于作品的评价众多专家很难达成一致看法，作品评分的极差在一定程度上可以反映出创新型，按照作品排名罗列其极差序列观察其趋势如下图 6-9 所示：

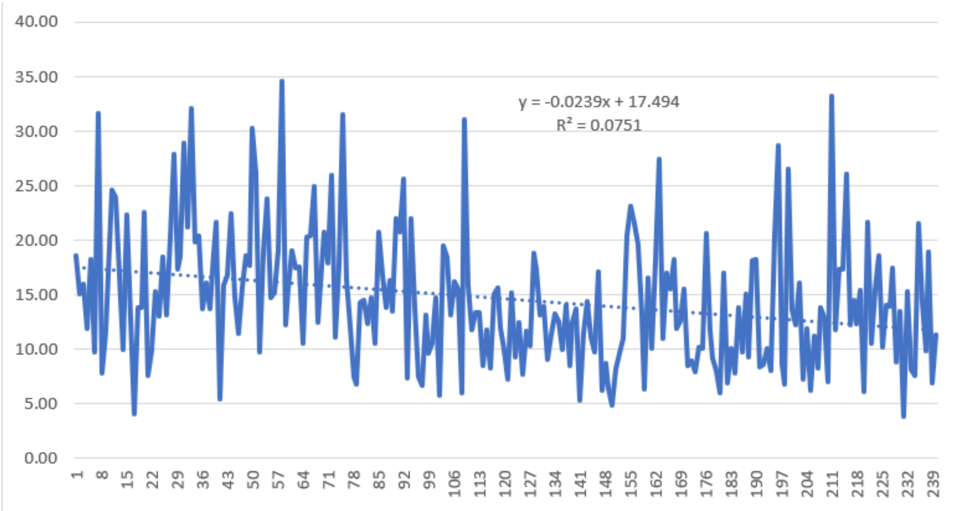

图 6–9 第一阶段评审极差序列

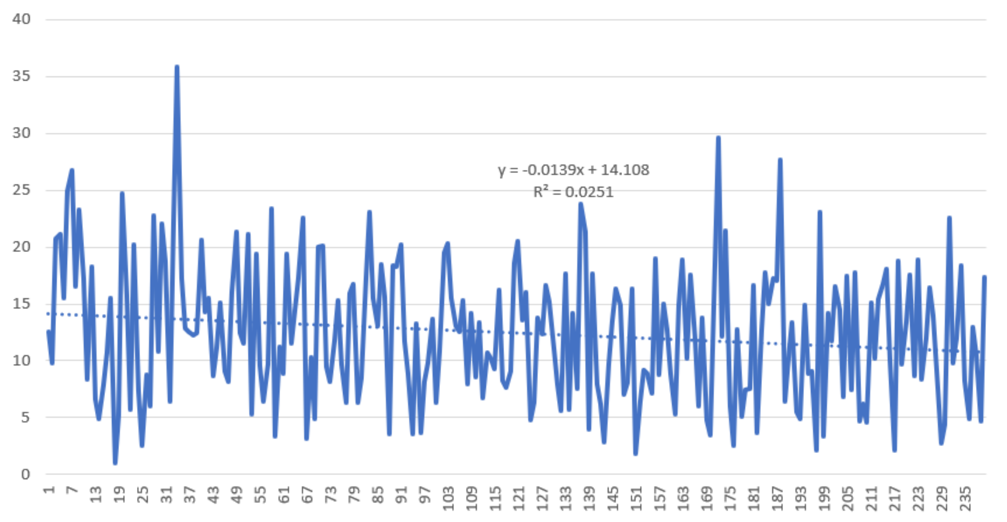

图 6-10 第二阶段评审极差序列

从图 6-9 和 6-10 可知，两个阶段的评分极差对排名呈现了正相关的趋势，表明随着作品排名的提升，创新性有增强的趋势。由于第二次评审针对极差大的作品进行了调整，相关性减弱，但仍旧呈现了正相关。

6.4 新型极差调整模型建立

由于创新类竞赛具备评分极差大的特点，有资质的选手能否获奖很大程度上受到了专家偏好等非自然因素的影响，往往一个离群程度较大的最低分就埋没了优秀作品。有相当一部分徘徊在未获奖中排名靠前的位置。为了发掘这些创新论文采用多步极差调整策略。

通过观察排名居中但不在获奖范围内的数据可以发现，1501 名之后，很多作品的分值突然出现了大量离群点。

表 6-1 排名中间作品表(节选)

排名	分数 1	分数 2	分数 3	分数 4	分数 5
1497	65	67.33	48	59	81
1498	54	61.73	47	74	60
1499	55	60.26	51	66	69
1500	67	62.62	66	48	65
1501	50	51.62	43	40	50
1502	7	62.11	15	25	15
1503	30	64.29	25	16	23
1504	43	58.27	0	32	0
1505	15	56.88	15	20	15
1506	1	49.99	1	15	85

多步极差调整策略是通过多次调整实现降极差的。很多作品的分的离群点不仅有极高异常值，也会有极低值，甚至二者都有，通过多次逐步调整可以充分考虑各种离群点的情况。

首先设置阈值 ε，检查每个作品的极差是否大于阈值 ε，如果大于就需要进行极差调整，如果小于，则其分数不变。然后根据均值与中位数的关系确定调整最大值还是最小值：

$$\begin{gathered}P_{i(N/2)}-\overline{P_i}>0\xrightarrow{adjust}Min(P_i)\\P_{i(N/2)}-\overline{P_i}<0\xrightarrow{adjust}Max(P_i)\end{gathered}\tag{5-1}$$

找到最大离群点 j：

$$j=\arg\max(|P_{ij}-\overline{P_i}|)\tag{5-2}$$

极差调整是通过改变离均值较远点实现的，其更新公式如下：

$$P_{ij}=\frac{Max(P_{ij})+\overline{P_j}}{2}\tag{5-3}$$

$$OR\quad P_{ij}=\frac{Min(P_{ij})+\overline{P_j}}{2}\tag{5-4}$$

代码如下所示：

算法 2 多步极差调整策略

输入：阈值 ε

输出：作品评分矩阵 P

1. **WHILE** TRUE
2. 设置标记 *Flag*=0，用于记录是否极差是否有过调整
3. **FOREACH** P_i 遍历每一个作品
4. **IF** 极差大于阈值
5. Flag=1 本次循环有极差需要进行调整
6. **IF** 中位数>平均数
7. 最小值更新至最小值与均值的算数平均数
8. **ELSE**
9. 最大值更新至最大值与均值的算数平均数
10. **ENDIF**
11. **ENDIF**
12. **END FOREACH**
13. **IF** Flag=0 表示本次循环没有极差做出调整，则终止极差调整
14. **BREAK**
15. **END IF**

多步极差调整策略通过逐步调整极差大的数值，使五个分数的离散程度得到有效缩小。通过优先调整离群点，降低了专家个人偏好对整体分数带来的影响。

为了验证模型的有效性，将更新后的作品分数重新进行排名，可以看出有大量作品的名次进行了调换。靠近 1500 接近获奖者几乎全部淘汰，剩余作品的极差也都保持在稳定区间。

表 6-2 极差调整后的排名情况(节选)

现排名	原排名	总分	极差
1501	**2459**	340	27
1502	**1634**	315.07	26
1503	**1600**	312.37	11.12
1504	**2780**	310.75	21
1505	**2334**	304.27	20
1506	**2241**	300.39	16.88
1507	**2601**	296.82	25
1508	**2145**	295.33	19.03

1509	**1813**	295	20
1510	**2509**	295	18

6.5 模型小结

针对问题三，我们讨论了两阶段的成绩整体的变化和两阶段极差整体的变化，分析两阶段评审方案相比不分阶段评审方案的优劣。对于极差与创新性的关系，我们通过观察其在排名序列上的规律初步得出创新性与极差有一定正相关的结论。为调整未获奖但排名靠前作品的风叔，决定采用多步极差调整策略，同时兼顾极大值和极小值，最终将极差稳定在了 30 以下。

七、问题四：动态加权评分模型

7.1 问题四的分析

在问题四中，本文通过对专家评分的统计量 $Z=\dfrac{u-\bar{u}}{\sigma/n}$ 进行讨论，引入两个部分的专家因素：**专家标准分与作品标准分均值的偏移程度**和**专家作品评分的方差**。进而介入指数模型，建立以**指数参数决策变量**和**匹配率为目标函数**的单目标**优化模型**。

通过所给**历史数据来训练模型**，得到最优情况下的模型参数。同时为了更好的训练模型，需要**更多的历史数据（如历年的评分数据）**，进而提供训练样本提高准确率。

7.2 专家评分的两个重要指标

为了更好的评价学生论文，本文对**专家**的考虑涉及到**两个重要的指标：专家标准分与文章标准分均值的偏移程度**和**专家作品评分集的方差**。

➢ 重要指标 1：专家标准分与文章标准分均值的偏移程度

专家总体上由于个人的性格和对作品的熟练情况或多或少对评分有偏移，为了减轻这类因素对最后评分的影响，本文用专家标准分与文章标准分均值的**偏移程度**来度量每个专家出现的情况。

$$\Delta u_i=\sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(u_{ij}-\bar{u}_i)^2} \tag{6-1}$$

其中，u_{ij} 表示第 i 个专家对第 j 篇作品的标准分，$\bar{u}_i$ 表示第 i 篇标准分的平均分。

➢ 重要指标 2：专家作品评分集的方差

专家**由于**对作品的熟练情况**不足，经常出现不敢打高分和低分的情况**，为了减轻这类因素对最后评分的影响，本文用专家作品评分的方差来度量每个专家出现的情况。**即专家作品评分的方差为：**

$$S_i^2=\frac{1}{n-1}\sum_{j=1}^{n}(u_{ij}-\bar{u})^2 \tag{6-2}$$

其中，$\bar{u}$ 表示专家所有论文标准分的平均分。

7.3 含 Z 检验的指数模型来衡量专家的权重

考虑到两个重要指标专家标准分与文章标准分均值的偏移程度和专家作品评分的方差引入权重指数函数，首先给出修正的 Z 检验统计量：

$$Z_i=\frac{\Delta u_i}{S_i/n} \tag{6-3}$$

同时，通过指数模型进行融合得到广义的权重指数分布模型：

$$w_i = \frac{1}{\kappa} e^{-\frac{Z_i}{\kappa}} \tag{6-4}$$

其中，κ 为指数分布的参数。结合上述公式，本文的**权重指数分布模型为**.

$$W_i = w_i \Big/ \sum_i w_i \tag{6-5}$$

对于**权重** W_i **有如下特性：**

1) **偏移程度** Δu_i **越大，表明该专家偏移过大，其对应的** W_i **越小；**
2) **专家作品评分集的方差** S_i^2 **越大，表明该专家可靠性高，其对应** W_i **越大。**

进一步得到，具体的**论文的成绩修正模型**为：

$$S = W_1 S_1 + W_2 S_2 + W_3 S_3 + W_4 S_4 + W_5 S_5 \tag{6-6}$$

7.4 基于匹配率最大的单目标优化模型

把论文的成绩修正的成绩。为了更好地找到最优模型，本文考虑网**评专家和会评专家的比重**:

网评专家：会评专家= C: 1-C.

带入匹配度模型，就得到了**基于动态加权评分模型下的匹配率最大的单目标优化模型**

决策变量

κ：指数分布的参数

C: 5 位网评专家占整个评审过程中的比重

目标函数 $$Max \quad f$$

7.5 梯度下降法求解问题三最优化模型

根据问题三预处理所得汇总数据，将作品的评分矩阵还原为作品-专家评分矩阵，抽取进入二阶段评审的队伍 1500 队，可得第一阶段评分矩阵 P_1，第二阶段评分矩阵 P_2。根据式 1.4 生成权重矩阵 W：

$$W = \text{softmax}\left(\frac{1}{\kappa} e^{-\frac{Z_i}{\kappa}}\right) \tag{6-7}$$

则最终成绩 Y：

$$Y \leftarrow C P_1 W + (1-C) P_2 W \tag{6-8}$$

可定义损失函数 Loss:

$$Loss = Y - CP_1W + (1-C)P_2W \tag{6-9}$$

通过梯度下降法逐步求得最优参数 C 和 κ：

$$\kappa \leftarrow \kappa - \gamma \frac{dLoss}{d\kappa} \tag{6-10}$$

$$C \leftarrow C - \gamma \frac{dLoss}{dC} \tag{6-11}$$

7.6 模型结果

为了验证我们模型的优越性，拟将动态加权得到的分数与最终成绩的奖项匹配算出动态加权的匹配率。计算第一阶段的分数与最终成绩的奖项匹配，得到第一阶段的匹配率。然后将这两种方案的匹配率对比分析。首先计算第一阶段的匹配率，其匹配率结果如下。

表 7-1 原始数据第一阶段的分数与最总奖项匹配

	一等奖	二等奖	三等奖
匹配率	27.54%	83.67%	4.31%

可以看出一等奖的匹配率不高，而二等奖的匹配效果不错。通过采用梯度下降法对动态加权模型进行计算，算出动态加权模型的两个重要参数：指数分布的参数 κ 和比重 C。

$$\kappa = 0.86,\ C = 0.3882$$

两个参数已经求出，代入动态加权模型后求得加权的分数，将其与最终成绩匹配得到加权动态模型的匹配率。

表 7-2 优化后的分数与最总奖项匹配

	一等奖	二等奖	三等奖
匹配率	43.48%	88.87%	8.61%

一等奖的重要程度高，很大程度上决定匹配效果的好坏，通过一等奖的匹配率可以明显看出动态加权模型的匹配效果好。

7.7 模型小结

针对问题四，我们建立一个动态加权评分模型，该模型有两个重要参数：指数分布的参数 κ 和比重 C。就是因为这两个参数模型才能够动态地调整评审专家的权重以使得匹配的效果达到最优；求解该模型的策略本文提出了梯度下降法，通过求最终成绩的损失函数的导数，不断逼近全局最小值得到比重 C，进而求得模型结果；为了验证该模型的效果，将其匹配率与第一次评审成绩与最终成绩的匹配率比较，我们的模型的匹配率高于第一次评审成绩与最终成绩的匹配率，模型匹配效果好。

八、模型评价

8.1 模型优点

（1）本文在研究改进标准分模型时，根据模型的渐进性原则，在原有模型的基础上不断地修正。如将总体偏正态转换为正态的正态化标准分优化模型，能够将标准差映射到指定区间研究的归一化标准分优化模型，以及将正态化标准分优化模型和归一化标准分优化模型融合的组合标准分优化模型。为了能够对不同方案进行排序，特地引入匹配函数和匹配度指标，帮助我们评价方案的优劣。特别是，3 种方案的匹配度与第一次评审的匹配度比较，我们提出的 3 种方案匹配效果均有提升。

（2）在第四问中，我们提出了一种动态加权评分模型，该方法能够灵活地通过调节评审专家的比重使得匹配效果不断优化。

8.2 模型缺点

3 种方案虽然与第一次评审的原始成绩与最终成绩的匹配度相比，匹配效果均有的提升，但就其匹配度来看比不是很高，特别时一等奖的匹配度。后续可以深入研究如何使一等奖的匹配成功率尽可能高。

参考文献

[1] 夏亚梅,程渤,陈俊亮,等.基于改进蚁群算法的服务组合优化[J].计算机学报，2012, 35(2): 12.

[2] 黄柳倩.加权平均型综合评价模型在成绩排序中的应用[J].幸福家庭, 2010(13):3.

[3] 尹升华.矿柱稳定性影响因素敏感性正交极差分析[J].煤炭学报,2012(S1):5.

[4] 冯冬青,王非,马雁.遗传算法中选择交叉策略的改进[D].CNKI；WanFang,2008.

[5] 衣建中.客资信息的交叉分发方法及装置：CN201910117125.5[P]. CN109886785A [2023-09-26].

[6] YY Ji, et al. Semiparametric estimation of a Box-Cox transformation model with varying coefficients model. *Science China Mathematics* (2017).

[7] 韩福荣,郝进.质量管理体系有效性综合评价模型[J].北京工业大学学报, 2000, 26(3):5.

[8] 张文泉,张世英.基于熵的决策评价模型及应用[J].系统工程学报, 1995, 10(3):6.

[9] 张立军,袁能文.线性综合评价模型中指标标准化方法的比较与选择[J].统计与信息论坛, 2010, 25(8):6.

[10] 韩兆洲,谢铭杰.上市公司投资价值评价模型及其实证分析[J].中央财经大学学报, 2004(11):5.

[11] 章穗,张梅,迟国泰.基于熵权法的科学技术评价模型及其实证研究[J].管理学报，2010, 7(1):34.

[12] 余胜泉.基于互联网络的远程教学评价模型[J].开放教育研究, 2003(1):5.

[13] 徐晓栋龚玉玲.基于改进层次分析法的竞赛选拔评价模型研究与应用[J].科技创新与生产力, 2022(7):21-23.

[14] 李是良,雷永鹏,苑洪亮,等.学科竞赛对大学生创新能力培养的作用研究——基于云模型的评价方法[J].软件导刊(教育技术), 2014.DOI:CNKI:SUN:RJDJ.0.2014-08-033.

[15] 常志强、吕俊杰、张博、赵文媛.数学建模竞赛创新分析体系构建及实践探索[J].科技视界, 2020(30):3.

附录

附件一

问题一：生成交叉方案

```
import numpy as np
import matplotlib.pyplot as plt
import random
P=np.zeros((125,3000))
#判断是否合法
def islawable(P):
    for i in range(3000):
        if(np.sum(P[:,i])!=5):
            return False
    for i in range(125):
        if(np.sum(P[i]!=120)):
            return False
    return True
#选取 120 个对象
def select_120(dict,s):
    selectable=[]
    max_v=np.max(np.array(list(dict.values()))[:,0])
    for i in range(3000):
        if(dict[i][0]==max_v and dict[i][1]!=s):
            if(i not in selectable):
                selectable.append(i)
    if(len(selectable)>=120):
        return random.sample(set(selectable),120)
    else:
        for i in range(3000):
            if(dict[i][0]==max_v-1 and dict[i][1]!=s ):
                if(i not in selectable):
                    selectable.append(i)
            if(len(selectable)>=120):
                return random.sample(set(selectable),120)
dict={}
dict2={}
dict3={}
for i in range(3000):
    dict[i]=[5,np.random.randint(1,30)]
    dict2[i]=0
for i in range(125):
    dict3[i]=np.random.randint(1,30)
a=list(np.arange(3000))
```

```
for i in range(125):
    print(i)
    p=select_120(dict,dict3[i])
    print(p)
    P[i,p]=1
    for j in p:
        dict[j][0]-=1
        dict2[j]+=1
# np.save('./origin_P1.npy',P)
print(np.count_nonzero(np.matmul(P,P.T)))
print(np.sum(np.matmul(P,P.T)))
print('----------------120 检验-------------------')
for i in range(125):
    if(np.sum(P[i])!=120):
        print(i)
print('----------------5 检验-------------------')
for j in range(3000):
    if(np.sum(P[:,j])!=5):
        print(j,' ',np.sum(P[:,j]))
for i in range(3000):
    if(dict[i][0]!=0):
        print(i,' ',dict[i][0])
P=np.load('./origin_P.npy')
for i in range(3000):
    if(np.sum(P[:,i])!=5):
        print(i)
def is_better(P_nonzero,PP0):
    P=np.zeros((125,3000))
    for i in range(len(P_nonzero)):
        for j in range(5):
            P[P_nonzero[i][j]][i]=1
    PP01=np.count_nonzero(np.matmul(P,P.T))
    if(PP01>PP0):
        return PP01
    return False
def recover_P(P_nonzero):
    P=np.zeros((125,3000))
    for i in range(3000):
        for j in range(5):
            P[P_nonzero[i][j]][i]=1
    return P
P=np.load('./origin_P.npy')
Q=np.matmul(P,P.T)
PP_0=np.count_nonzero(Q)
```

```
T=np.sum(Q)
best_PP_0=np.count_nonzero(Q)
best_T=np.sum(Q)
print(PP_0)
print(T)
P_nonzero=[[] for i in range(3000)]
for j in range(3000):
    for i in range(125):
        if(P[i][j]!=0):
            P_nonzero[j].append(i)
best_P_npnzero=P_nonzero.copy()
tmp=-1
optim_num=0
for i in range(2999):
    print(i,' ','optim_num',optim_num,'PP_0',' ',best_PP_0)
    for j in range(5):
        for k in range(5):
            tmp=P_nonzero[i][j]
            P_nonzero[i][j]=P_nonzero[i+1][k]
            P_nonzero[i+1][k]=tmp
            PP1=is_better(P_nonzero,best_PP_0)
            if(best_PP_0!=False):
                best_P_npnzero=P_nonzero.copy()
                best_PP_0=PP1
                optim_num+=1
            else:
                P_nonzero=best_P_npnzero.copy()
for i in range(3000):
    if(np.sum(P[:,i])!=5):
        print(i,' ',np.sum(P[:,i]))
np.save('./optimer_P',recover_P(best_P_npnzero))
```

附件二

问题二相关代码

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import json
data=pd.read_excel('./data1.xlsx').values
zhuanjia=[]
for i in range(len(data)):
    for j in range(5):
        if(data[i][j*3] not in zhuanjia):
            zhuanjia.append(data[i][j*3])
```

```
dict={}
for i in range(len(zhuanjia)):
	dict[zhuanjia[i]]=i
f=open('./professor_th.json','w')
json.dump(dict,f)
big_matrix=np.load('./big_matrix.npy').T
#归一化
liqun_n=0
#去离群
data=np.zeros(big_matrix.shape)
for i in range(len(big_matrix)):
	line=big_matrix[i]
	line=line[line!=0]
	standvar=np.sqrt(np.var(line))
	avg=np.mean(line)
	for j in range(len(big_matrix[i])):
		if(big_matrix[i][j]!=0):
			if(np.abs(big_matrix[i][j]-avg)<standvar*3):
				data[i][j]=big_matrix[i][j]
			else:
				data[i][j]=big_matrix[i][j]*1.1
				liqun_n+=1

data1=np.zeros(data.shape)
for i in range(data.shape[0]):
	for j in range(data.shape[1]):
		if(data[i][j]!=0):
			data1[i][j]=(data[i][j]-np.min(data[i]))/(np.max(data[i])-np.min(data[i]))

data2=[]
data1=data1.T
for i in range(len(data1)):
	line=data1[i]
	line=line[line!=0]
	if(len(line)<5):
		for j in range(5-len(line)):
			line=np.append(line,0)
	data2.append(line)
pd.DataFrame(data2).to_excel('./guiyi.xlsx')


lam=1.1
data=[]
data=np.zeros(big_matrix.shape)
```

```
for i in range(len(big_matrix)):
    line=big_matrix[i]
    line=line[line!=0]
    standvar=np.sqrt(np.var(line))
    avg=np.mean(line)
    for j in range(len(big_matrix[i])):
        if(big_matrix[i][j]!=0):
            data[i][j]=(big_matrix[i][j]**lam-1)/lam
data2=[]
data1=data.T
for i in range(len(data1)):
    line=data1[i]
    line=line[line!=0]
    if(len(line)<5):
        for j in range(5-len(line)):
            line=np.append(line,0)
    data2.append(line)

pd.DataFrame(data2).to_excel('./zhengtai.xlsx')

f=pd.read_excel('./data1.xlsx')
data=f.values[:,[2,5,8,11,14]]
avg=np.mean(data,axis=1).reshape(-1,1)
standvar=np.var(data,axis=1).reshape(-1,1)+1e-15
data=(data-avg)/standvar

label=f.values[:,[1,4,7,10,13]]
dict=json.load(open('./professor_th.json'))
shape=(len(data),len(dict))
big_matrix=np.zeros(shape)
for i in range(len(data)):
    for j in range(5):
        big_matrix[i][dict[label[i][j]]]=data[i][j]

    data=pd.read_excel('./dea2.xlsx')
    dea=data['dea2'].values
    er=data['er'].values

    plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
    plt.rcParams["axes.unicode_minus"]=False #该语句解决图像中的“-”负号的乱码问题
    prob1=dea
    sigma=np.sqrt(np.var(prob1))
    mu=np.mean(prob1)
    ax =plt.subplot(1,2,1)
```

```
 n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
 y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
 ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
 ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
 plt.xlabel('改进第一次评审方案标准分')

 ax =plt.subplot(1,2,2)
 prob1=er
 sigma=np.sqrt(np.var(prob1))
 mu=np.mean(prob1)
 n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
 y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
 ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
 ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
 plt.xlabel('第二次评审方案标准分')
plt.show()
```

选择语言：

```
        lt.rcParams["font.sans-serif"]=["SimHei"]
        plt.rcParams["axes.unicode_minus"]=False
        data=pd.read_excel('./data1.xlsx').values
        gs = gridspec.GridSpec(2, 6)
        gs.update(wspace=0.8)
        plt.subplot(gs[0, :2])
        plt.hist(data[:,2],bins=30)
        plt.xlabel('专家 1')
        plt.subplot(gs[0, 2:4])
        plt.hist(data[:,5],bins=30)
        plt.xlabel('专家 2')
        plt.subplot(gs[0, 4:6])
        plt.hist(data[:,8],bins=30)
        plt.xlabel('专家 3')
        plt.subplot(gs[1, 1:3])
        plt.hist(data[:,11],bins=30)
        plt.xlabel('专家 4')
        plt.subplot(gs[1, 3:5])
        plt.hist(data[:,14],bins=30)
        plt.xlabel('专家 5')
        plt.subplots_adjust(left=None, bottom=None, right=None, top=None,
                                            wspace=0, hspace=0.5)
        plt.show()
       for i in range(5):
        for j in range(5):
```

```
            if(i==j):
                plt.subplot(5,5,i*5+j+1)
                plt.hist(data[:,i*3+2])
                plt.xticks([])
                plt.yticks([])
                if(j==0):
                    plt.ylabel('专家'+str(i+1))
                if(i==4):
                    plt.xlabel('专家'+str(j+1))
            else:
                plt.subplot(5,5,i*5+j+1)
                plt.scatter(data[:,i*3+2],data[:,j*3+2],s=0.3)
                plt.xticks([])
                plt.yticks([])
                if(j==0):
                    plt.ylabel('专家'+str(i+1))
                if(i==4):
                    plt.xlabel('专家'+str(j+1))
            #print(data[i*3+2])
    plt.show()
    bins=10
    Y=[]
    n=4
    for i in range(n):
        prob1=big_matrix[:,i]
        prob1=prob1[prob1!=0]
        sigma=np.sqrt(np.var(prob1))
        mu=np.mean(prob1)
        ax =plt.subplot(1,1,1)
        n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
        y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
        Y.append(y)

    plt.clf()
    Y=np.array(Y).T
    plt.plot(Y)
    label=['专家'+str(i+1) for i in range(4)]
    plt.legend(label)
    plt.xticks([i for i in range(10)],[(i-5)/5 for i in range(10)])
    plt.show()

    x1=pd.read_excel('./guiyi.xlsx')['归一'].values
    x2=pd.read_excel('./zhengtai.xlsx')['归一'].values
```

```
data=torch.tensor([x1,x2],dtype=torch.float32).T[:352]
y=torch.tensor(pd.read_excel('./professor.xlsx')['                    归                    一
'].values,dtype=torch.float32).reshape(-1,1)
class Moudle(nn.Module):
    def __init__(self):
        super(Moudle, self).__init__()
        self.w=nn.Parameter(torch.randn(1))
    def forward(self,x):
        return self.w*x[:,0]+(1-self.w)*x[:,1]
class Loss(nn.Module):
    def __init__(self):
        super(Loss, self).__init__()
        self.loss=nn.MSELoss()
    def forward(self,x,y):
        return
self.loss(x[:28],y[:28])/self.loss(x[28:295],y[28:295])#+self.loss(x[295:352],y[295:352])
net=Moudle()
loss=Loss()
optimer=torch.optim.Adam(net.parameters(),lr=1e-2)
scheduler = StepLR(optimer, step_size=30, gamma=0.9)
print(data.shape)
print(y.shape)
print(net(data).shape)
for i in range(1000):
    out=net(data)
    l=loss(out,y)
    optimer.zero_grad()
    l.backward()
    optimer.step()
    #scheduler.step()
    print(l.detach().numpy())

pd.DataFrame(out.detach().numpy()).to_excel('./xgbost.xlsx')
print(net.w)
prob1=standardize2
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第一次评审方案标准分')
ax =plt.subplot(1,2,2)
```

```
prob1=prof
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第二次评审方案标准分')
plt.show()

f1=pd.read_excel('./data1.xlsx')
f2=pd.read_excel('./dea_result.xlsx')
f3=pd.read_excel('./latent_factor.xlsx')
professor_2nd=standardize3=f2['原排名'].values
standardize1=np.argsort(-1*np.sum(f1.values[:351,[3,6,9,12,15]],axis=1))

standardize2=np.argsort(-1*np.sum(np.sort(f1.values[:351,[2,5,8,11,14]],axis=1)[:,[1,2,3]],axi
s=1))
standardize3=f2['现排名'].values[:351]
standardize4=f3.values[:351,0]
def spearman(x,y):
    n=len(x)
    return 1-np.sum((x-y)**2)*6/n/(n**2-1)
rate1=spearman(standardize1,professor_2nd)
rate2=spearman(standardize2,professor_2nd)
rate3=spearman(standardize3,professor_2nd)
rate4=spearman(standardize4,professor_2nd)
plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
plt.rcParams["axes.unicode_minus"]=False #该语句解决图像中的“-”负号的乱码问题
plt.bar(['方案一','方案二','方案三','方案四'],[rate1,rate2,rate3,rate4])
plt.show()
print(rate1,rate2,rate3,rate4)
```

附件三

问题三相关代码

```
f=pd.read_excel('./data3.xlsx',sheet_name=1)
data=f.values
zhuanjia=json.load(open('./zhuanjia_num.json'))
big_matrix=np.zeros((len(data),380))
for i in range(len(big_matrix)):
    for j in range(3):
        big_matrix[i][zhuanjia[data[i][4*j]]]=data[i][4*j+1]
print(data[0])
print(big_matrix.shape)
```

```python
np.save('./big_matrix2',big_matrix)

big_matrix=np.load('./big_matrix2.npy')
var=[]
avg=[]
for i in range(big_matrix.shape[1]):
        line=big_matrix[:,i]
        line=line[line!=0]
        print(line)
        var.append(np.var(line))
        avg.append(np.mean(line))
        plt.hist(line)
        plt.show()

print(np.max(var))
print(np.min(var))
print(np.max(avg))
print(np.min(avg))
# plt.hist(var)
# plt.show()
plt.rcParams["font.sans-serif"]=["SimHei"]
plt.rcParams["axes.unicode_minus"]=False
prob1=avg
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
n, bins, patches=ax.hist(prob1, 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('专家总体均值分布')
prob1=var
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,2)
n, bins, patches=ax.hist(prob1, 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('专家总体方差分布')
plt.show()

f=pd.read_excel('./wen3.xlsx')
```

```
plt.rcParams["font.sans-serif"]=["SimHei"]
plt.rcParams["axes.unicode_minus"]=False
prob1=f['一极差']
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第一次评审极差分布')

ax =plt.subplot(1,2,2)
prob1=f['二极差'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第二次评审极差分布')

ax =plt.subplot(1,1,1)
prob1=f['极差差'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('两次评审极差差值分布')
plt.show()

f=pd.read_excel('./wen3.xlsx')
plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
plt.rcParams["axes.unicode_minus"]=False #该语句解决图像中的“-”负号的乱码问题
prob1=f['一方差']
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
```

```
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y  =  ((1/(np.power(2*np.pi,  0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第一次评审方差分布')

ax =plt.subplot(1,2,2)
prob1=f['二方差'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y  =  ((1/(np.power(2*np.pi,  0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第二次评审方差分布')

ax =plt.subplot(1,1,1)
prob1=f['方差差'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y  =  ((1/(np.power(2*np.pi,  0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('两次评审方差差值分布')
plt.show()

f=pd.read_excel('./wen3.xlsx')
plt.rcParams["font.sans-serif"]=["SimHei"]
plt.rcParams["axes.unicode_minus"]=False
prob1=f['一均'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y  =  ((1/(np.power(2*np.pi,  0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第一次评审标准分均值分布')
```

```
ax =plt.subplot(1,2,2)
prob1=f['二均'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('第二次评审标准分均值分布')

ax =plt.subplot(1,1,1)
prob1=f['一二均差'].values
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.xlabel('两次评审标准分均值差分布')
plt.show()
f1=pd.read_excel('./data3.xlsx',sheet_name=0)
f2=pd.read_excel('./data3.xlsx',sheet_name=1).fillna(0)
data1=f1.values
data2=f2.values
for i in range(1500):
    for j in range(3):
        if(data2[i][j*4+3]!=0):
            data2[i][j*4+2]=data2[i][j*4+3]

plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
plt.rcParams["axes.unicode_minus"]=False #该语句解决图像中的“-”负号的乱码
问题

prob1=np.max(data1[:1500,[3,6,9,12,15]],axis=1)-np.min(data1[:1500,[3,6,9,12,15]],axis=1)-
np.max(data2[:,[1,5,9]],axis=1)+np.min(data2[:,[1,5,9]],axis=1)
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,1,1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma,
2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
```

```
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('两次评审极差差值分布')
ax =plt.subplot(1,2,2)
prob1=data2[:,[1,5,9]]
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('第二次原始数据极差分布')
plt.show()
f1=pd.read_excel('./data3.xlsx',sheet_name=0)
f2=pd.read_excel('./data3.xlsx',sheet_name=1).fillna(0)

one1=f1.values[:70,14]
two1=f1.values[70:1301,14]
three1=f1.values[1301:1501,14]

f2data=f2.values
for i in range(len(f2data)):
    if(f2data[i][13]!=0):
        f2data[i][12]=f2data[i][13]

one2=f2data[:70,12]
two2=f2data[70:1301,12]
three2=f2data[1301:1501,12]
plt.rcParams["font.sans-serif"]=["SimHei"]
plt.rcParams["axes.unicode_minus"]=False
prob1=three1
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
ax =plt.subplot(1,2,1)
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('第一次评审三等奖极差分布')
ax =plt.subplot(1,2,2)
prob1=three2
sigma=np.sqrt(np.var(prob1))
mu=np.mean(prob1)
```

```
n, bins, patches=ax.hist(prob1[prob1!=0], 10, density=True, histtype="bar")
y = ((1/(np.power(2*np.pi, 0.5)*sigma))*np.exp(-0.5*np.power((bins-mu)/sigma, 2)))
ax.plot(bins, y, color="#ff0000", ls="--", lw=2)
ax.grid(ls=":", lw=1, color="gray", alpha=0.2)
plt.title('第二次评审三等奖极差分布')
plt.show()
```

附件四

问题四相关代码

```
zhuanjia=json.load(open('./zhuanjia_num.json'))
data3=pd.read_excel('./data3.xlsx',sheet_name=0).values[:1500,[3,6,9,12,15]]
zhuanbian=pd.read_excel('./data3.xlsx',sheet_name=0).values[:1500,[2,4,6,8,10]]
big_matrix1=np.load('./big_matrix.npy')[:1500]
big_matrix2=np.load('./big_matrix2.npy')[:1500]

y=pd.read_excel('./data3.xlsx',sheet_name=1).fillna(0).values[:1500]
y=torch.tensor(np.sum(y[:,[2,6,10]].astype('float32'),axis=1),dtype=torch.float32)


for i in range(len(big_matrix1)):
    for j in range(380):
        if(big_matrix1[i][j]!=0):
            big_matrix1[i][j]=(big_matrix1[i][j]-np.mean(data3[i]))**2
u_u0=np.sqrt(np.sum(big_matrix1,axis=0)/(np.count_nonzero(big_matrix1,axis=0)-1))

np.save('./u_u0',u_u0)
u_u0=np.load('./u_u0.npy')
S=np.sqrt(np.sum((big_matrix1-np.mean(big_matrix1))**2,axis=0)/379)
u_u0=u_u0/S/380
np.save('./Z',u_u0)
Z=np.load('./Z.npy')
class Mouule(nn.Module):
    def __init__(self,Z):
        super(Mouule, self).__init__()
        self.z=torch.tensor(Z,dtype=torch.float32)

self.C=nn.Parameter(torch.tensor(0.3,dtype=torch.float32,requires_grad=True))
        self.k=nn.Parameter(torch.randn(1,dtype=torch.float32,requires_grad=True))
        self.softmax=nn.Softmax()
    def forward(self,x1,x2):
        w=torch.exp(-1*self.z/self.k)/self.k
        w=self.softmax(w)
    return torch.matmul(x1,w)*self.C**2+torch.matmul(x2,w)*(1-self.C*2)
```

```
x1=torch.tensor(big_matrix1,dtype=torch.float32)
x2=torch.tensor(big_matrix2,dtype=torch.float32)

net=Mouule(Z)
optimer=torch.optim.SGD(net.parameters(),lr=1e-1)
scheduler = StepLR(optimer, step_size=10, gamma=0.5)
loss=nn.MSELoss()
for i in range(100):
    out=net(x1,x2)
    l=loss(out,y)
    optimer.zero_grad()
    l.backward(retain_graph=True)
    optimer.step()
    scheduler.step()
    print(i,' ',l.detach().numpy())

print(net.k)
print(net.C)
pd.DataFrame(net(x1,x2).detach().numpy()).to_excel('./k1.xlsx')
    data=pd.read_excel('./data3.xlsx').values[1501:3000,[3,6,9,12,15]]
    data=np.sort(data,axis=1)
    dict={}
    for i in range(1500):
        dict[i]=[]
    data0=1
    while True:
        flag=0
        for i in range(len(data)):
            if(data[i][-1]-data[i][0]>10):
                data0=np.sum(data[i])
                print(i,' ',data[i][-1]-data[i][0])
                if(data[i][2]>np.mean(data[i])):
                    change=(np.mean(data[i])-data[i][0])/2
                    dict[i].append(change)
                    data[i][0]=data[i][0]+change
                else:
                    change=(np.mean(data[i])-data[i][-1])/2
                    dict[i].append(change)
                    data[i][-1]=data[i][-1]+change
                if(data0!=np.sum(data[i])):
                    flag=1
        if(flag==0):
```

```
        break
pd.DataFrame(data).to_excel('./newdata2.xlsx')
```