

---

# 学生面试问题

西安交通大学

孙志勇 吴飞龙 刘晶晶

---

## 摘要

论文讨论了学生面试问题中的面试组分配方案，从理论上界分析、模型建立、搜索算法设计与评价等几方面进行了深入探讨。

首先基于组合数学中的区组设计的原理，讨论了在老师数  $M$  给定时不同情况下能面试学生数  $N$  的上界；当不考虑文理科老师时，任何两个面试组没有两位或三位老师同时的  $N$  的上界分别为

$$N \leq \left\lfloor \frac{M}{4} \left\lfloor \frac{M-1}{3} \right\rfloor \right\rfloor \text{ 和 } N \leq \left\lfloor \frac{M}{4} \left\lfloor \frac{M-1}{3} \left\lfloor \frac{M-2}{2} \right\rfloor \right\rfloor \right\rfloor$$

当面试组中文理老师各需一半时，此时  $N$  的上界分别为

$$N \leq \left\lfloor \frac{M}{4} \left\lfloor \frac{M}{4} \right\rfloor \right\rfloor \text{ 和 } N \leq \left\lfloor \left\lfloor \frac{M}{4} \right\rfloor \left\lfloor \frac{M}{4} \right\rfloor * (M/2 - 1) \right\rfloor$$

当确定考生数  $N$  时，同理可以由上述不等式确定  $M$  的下界，即在不同要求下至少要聘请的老师数。

接着根据题设要求建立了问题二与问题三的数学模型，引入关联矩阵，将面试组设计要求作为约束条件， $Y_1 \sim Y_4$  为目标函数，从而建立了 0-1 变量的多目标非线性规划模型。

因上述规划模型难以直接求解，本文设计了两种搜索算法：重复数优先算法与全组合删除算法针对问题二和问题三就  $N=379$ ,  $M=24$  得到了分配分配方案，问题二中方案只需一组三元组；基于全组合删除算法得到问题三的方案不含任何三元组。基于  $Y_1 \sim Y_4$  的目标函数将其量化，设计了性能评判指标，两种算法得到的分配方案均有着很好的性能。

最后对面试组的分配方案提出了其他需考虑的因素与可行的建议。

**关键词：**区组设计理论，上界分析，多目标整数规划，近似搜索算法，模型评价

---

# 目录

<b>1</b>	<b>问题重述</b> .....	<b>1</b>
<b>2</b>	<b>问题分析与模型假设</b> .....	<b>1</b>
2.1	问题分析.....	1
2.2	模型假设.....	1
<b>3</b>	<b>基于区组设计理论的学生数 <math>N</math> 上界估计</b> .....	<b>2</b>
3.1	BIBD 理论介绍.....	2
3.2	问题一的两个上界.....	2
3.2.1	无两位老师相同的情况.....	2
3.2.2	无三位老师相同的情况.....	3
3.3	问题三的两个上界.....	3
3.3.1	无两位老师相同的情况.....	3
3.3.2	无三位老师相同的情况.....	4
3.4	上界分析.....	4
<b>4</b>	<b>模型建立</b> .....	<b>5</b>
4.1	问题二模型建立.....	5
4.2	问题三模型建立.....	6
4.3	模型的求解分析.....	7
<b>5</b>	<b>模型求解与算法评价</b> .....	<b>7</b>
5.1	问题二模型求解.....	7
5.1.1	搜索算法描述.....	7
5.1.2	分配方案与模型评价.....	8
5.2	问题三模型求解.....	9
5.2.1	搜索算法设计.....	9
5.2.2	分配方案与模型评价.....	9
<b>6</b>	<b>面试组分配方案的其他建议</b> .....	<b>11</b>
<b>7</b>	<b>参考文献</b> .....	<b>12</b>
<b>8</b>	<b>附录</b> .....	<b>13</b>
8.1	分配方案表格.....	13
8.2	程序代码.....	22

---

# 1 问题重述

高校自主招生是高考改革中的一项新生事物，现在仍处于探索阶段。某高校拟在全面衡量考生的高中学习成绩及综合表现后再采用专家面试的方式决定录取与否。该校在今年自主招生中，经过初选合格进入面试的考生有  $N$  人，拟聘请老师  $M$  人。每位学生要分别接受 4 位老师（简称该学生的“面试组”）的单独面试。面试时，各位老师独立地对考生提问并根据其回答问题的情况给出评分。由于这是一项主观性很强的评价工作，老师的专业可能不同，他们的提问内容、提问方式以及评分习惯也会有较大差异，因此面试同一位考生的“面试组”的具体组成不同会对录取结果产生一定影响。为了保证面试工作的公平性，组织者提出如下要求：

- Y1. 每位老师面试的学生数量应尽量均衡；
- Y2. 面试不同考生的“面试组”成员不能完全相同；
- Y3. 两个考生的“面试组”中有两位或三位老师相同的情形尽量少的；
- Y4. 被任意两位老师面试的两个学生集合中出现相同学生的人数尽量少的。

请回答如下问题：

问题一：设考生数  $N$  已知，在满足 Y2 条件下，说明聘请老师数  $M$  至少分别应为多大，才能做到任两位学生的“面试组”都没有两位以及三位面试老师相同的情形。

问题二：请根据 Y1~Y4 的要求建立学生与面试老师之间合理的分配模型，并就  $N=379$ ， $M=24$  的情形给出具体的分配方案（每位老师面试哪些学生）及该方案满足 Y1~Y4 这些要求的情况。

问题三：假设面试老师中理科与文科的老师各占一半，并且要求每位学生接受两位文科与两位理科老师的面试，请在此假设下分别回答问题一与问题二。

问题四：请讨论考生与面试老师之间分配的均匀性和面试公平性的关系。为了保证面试的公平性，除了组织者提出的要求外，你们认为还有哪些重要因素需要考虑，试给出新的分配方案或建议。

## 2 问题分析与模型假设

### 2.1 问题分析

本问题可将其抽象成老师数的组合分配与学生数（即面试组合）的匹配问题。组合数学中有平衡不完全区组设计（BIBD）理论，可应用该理论的思想求解在固定老师数之下满足不同情况（比如无二元组或无三元组）下的面试组数上界，并根据此上界公式反推需要的老师数。

在具体分配时，根据题目的要求我们提出了优化模型；但因无具体的最优寻优算法可用，我们提出了几种近似最优搜索算法并给出了  $N=379$ ， $M=24$  下的分配情况，并给出了算法与模型评价。

### 2.2 模型假设

我们提出如下的模型假设：

- (1) 所有参与面试的老师均认为是一样的而不作区别；
- (2) 所有参加面试的考生均认为是一样的，同样不作区别；

- (3) 不考虑分配方案形成的时间冲突,也不考虑分配方案中老师与考生的时间冲突;
- (4) 不同分配方案下的老师组合不影响对学生的评分与录取。
- (5) 面试的先后次序对最终录取结果不影响。
- (6) 问题三中“理科老师与文科老师各占一半”是严格的一半,即老师总人数是偶数。

### 3 基于区组设计理论的学生数 N 上界估计

#### 3.1 BIBD 理论介绍

平衡不完全区组设计 BIBD (balance incomplete block design) 是组合数学中的一种区组设计理论。设  $X = \{x_1, x_2, \dots, x_v\}$  为试验对象, 数目为  $v$  (此处即老师数量), 设计的区组数目为  $b$  (此处即需面试的学生数量), 每组有  $k$  个元素, 且每个试验对象在  $b$  组中正好出现  $r$  次, 任意一对属于  $X$  的元素在  $b$  组中正好同时出现  $\lambda$  次。可以证明: 满足该设计的区组必然满足

$$\begin{cases} bk = vr \\ r(k-1) = \lambda(v-1) \end{cases} \quad (1)$$

其中证明过程可见文献[1]。

按题目要求分析知, 在一个满足条件的面试组合中, 若任意一对老师的组合 (或三个老师的组合) 都在面试组中出现且只出现一次, 则该分配组合方案是 BIBD 组。我们将运用公式(1)的证明原理, 推出在固定老师数  $M$  下问题一与问题三的解析解, 即可面试学生数  $N$  的上界公式, 由此可推出所需老师数  $M$  的下界。

#### 3.2 问题一的两个上界

##### 3.2.1 无两位老师相同的情况

设聘请老师数为  $M$ , 按题设 Y2 条件必须满足。若在任何两个学生的“面试组”都没有两位老师相同, 此时存在可以面试的学生数的一个上界。

当任意两老师的组合都在“面试组”中出现一次, 则此时可将老师数目最大化, 此时老师的人数将最少。换句话说若老师人数  $M$  固定, 则可面试的学生数  $N$  将达到最大。按照公式(1)有

$$\begin{cases} 4N = Mr \\ r(4-1) = (M-1) \end{cases} \quad (2)$$

即有

$$N = \frac{M(M-1)}{12} \quad (3)$$

当然对于某些特定的  $N$ , 不一定做到使任意两老师组合都在面试组中出现 (即若全部出现需满足 BIBD 理论中的同余条件), 因此(3)式只是给出了  $N$  的一个上界, 即

$$N \leq \frac{M(M-1)}{12} \quad (4)$$

从另一个思路分析：任取一老师  $x_i$ ，任一个含  $x_i$  的老师对  $\{x_i, x_j\} (i \neq j)$  都需在面试组中出现且只能出现一次，即在每个区组中老师  $x_j$  不能相同； $x_j$  有  $M-1$  种选择，一共可分成

$\left\lfloor \frac{M-1}{3} \right\rfloor$  组（除 3 取整是对应于每个区组里除了  $x_i$  之外还需要 3 位老师，且各老师不能相

同，也不能少于 3 个老师；符号  $\lfloor \cdot \rfloor$  表示向下取整），即每个老师可组成  $\left\lfloor \frac{M-1}{3} \right\rfloor$  个区组，

故  $M$  个老师最多可组成的区组（即可面试的学生）为

$$N \leq \left\lfloor \frac{M}{4} \left\lfloor \frac{M-1}{3} \right\rfloor \right\rfloor \quad (5)$$

其中的  $M/4$  是因为对于每个老师在 4 元素的区组里面都重复计数了 4 次。

### 3.2.2 无三位老师相同的情况

要使“面试组”中没有三位老师相同的情况，则任意三个老师组成的三元对在整个区组集合中只能出现一次，同时为了保证所需老师数最少，即老师数固定时能面试的学生数最多，则需使每一个三元对都在整个区组集合中出现一次。

考虑老师  $x_i$  以及  $x_i$  在内的三元对  $\{x_i, x_j, x_k\} (i \neq j \neq k)$ ，除  $x_i$  外的二元对  $\{x_j, x_k\} (j \neq k)$  需尽量出现一次且最多出现一次，按照 3.2.1 节的分析过程，同样可求出其能组成的最多子区组数目，只不过此时的  $k' = k - 1 = 3$ （子区组除  $x_i$  固定外需 3 个元素），老师的数量  $M' = M - 1$ （ $x_i$  除外），故最多子区组数目为

$$\hat{N} \leq \left\lfloor \frac{M-1}{3} \left\lfloor \frac{M-2}{2} \right\rfloor \right\rfloor \quad (6)$$

即每个老师可组合到  $\hat{N}$  个区组中（即可面试  $\hat{N}$  个学生），故此时  $M$  个老师能组成的区组数量为

$$N \leq \left\lfloor \frac{M}{4} \left\lfloor \frac{M-1}{3} \left\lfloor \frac{M-2}{2} \right\rfloor \right\rfloor \right\rfloor \quad (7)$$

## 3.3 问题三的两个上界

若面试老师中理科与文科的老师各占一半，设老师数量为  $M$ ，则文科老师与理科老师各为  $M/2$ ；同样按照上述的步骤分析。

### 3.3.1 无两位老师相同的情况

考虑文科老师  $x_i$ ，与其相组合的一个区组还需两位理科老师与另一位文科老师，且理

科老师不能相同（否则不能满足无两位老师相同的条件）； $M/2$  位理科老师可以组成  $\left\lfloor \frac{M/2}{2} \right\rfloor$  个元素均不相同的子区组。同理，在文科老师内也能组成  $\left\lfloor \frac{M/2}{2} \right\rfloor$  个元素均不相同的子区组，按乘法原理，故可面试学生数  $N$  的上界为

$$N \leq \left\lfloor \frac{M/2}{2} \left\lfloor \frac{M/2}{2} \right\rfloor \right\rfloor = \left\lfloor \frac{M}{4} \left\lfloor \frac{M}{4} \right\rfloor \right\rfloor \quad (8)$$

### 3.3.2 无三位老师相同的情况

要使“面试组”中没有三位老师相同的情况并且能组成最多面试组，则任意三个老师组成的三元对在整个区组集合中需出现一次且最多出现一次。此处的分析与 3.2.2 节类似。

因每个面试组各需两位文科老师与两位理科老师，而且任意两文科老师与一理科老师的三元对与任意两理科老师与一文科老师的三元对都只能最多出现一次。不妨考虑两位文科老师组成的二元对  $\{x_i, x_j\} (i \neq j)$ ， $M/2$  位理科老师组成  $\left\lfloor \frac{M/2}{2} \right\rfloor$  个元素均不相同的子区组，

这些子区组都可以与  $\{x_j, x_k\}$  形成一个符合条件的面试组；对于固定的文科老师  $x_i$ ，替换二元对  $\{x_i, x_j\}$  中不同的  $x_j$ ，替换不同的理科老师对以保证与之前的理科老师对不含相同的子区组，可组成  $(M/2 - 1)$  种含  $x_i$  在内的文科老师二元组，即每个老师可面试的学生数最多为

$$\hat{N} \leq \left\lfloor \frac{M/2}{2} \right\rfloor * (M/2 - 1) \quad (9)$$

因此对于所有文科老师可组成的符合条件的面试组合上界为

$$N \leq \left\lfloor \left\lfloor \frac{M/2}{2} \right\rfloor \left\lfloor \frac{M/2}{2} \right\rfloor * (M/2 - 1) \right\rfloor = \left\lfloor \left\lfloor \frac{M}{4} \right\rfloor \left\lfloor \frac{M}{4} \right\rfloor * (M/2 - 1) \right\rfloor \quad (10)$$

其中最左侧的  $\frac{M/2}{2}$  是指  $M/2$  个老师均在 2 元素子区组中重复计算了 2 次，因此需除以 2 以消除重复计算的次数。

## 3.4 上界分析

问题一与问题三分别要求当  $N$  已知时，在不分文理与分文理老师两种情况下，任两位学生的“面试组”都没有两位以及三位老师相同的情形时，所聘请的老师数  $M$  至少为多大，即求老师数的下界。上文分析是在固定  $M$  所求出的  $N$  的上界，同理，可得到当  $N$  固定时，按分析所得出的(5)、(7)、(8)、(10)各式在不同情况下计算出  $M$  的下界，即所需老师的最少数。

按照下界公式计算得到的老师数  $M$  只是一个下界值，因在各式中的向下取整运算的影响，实际所需的老师数可能会更大。也有情况是下界值正好是下确界，即所需的老师数正好是计算得到的  $M$ 。比如对于(5)式，若要使  $N$  正好是上确界，需

$$\begin{cases} (M-1) \equiv 0 \pmod{3} \\ M * \chi \equiv 0 \pmod{4} \end{cases} \quad (11)$$

其中  $\chi = \left\lfloor \frac{M-1}{3} \right\rfloor$  是运算后得到的正整数值。

按上式进一步分析可知 M 需满足

$$\begin{cases} M = 3k_1 + 1 \\ M = 4k_2 \end{cases} \quad (12)$$

其中  $k_1, k_2$  均为非负整数, 可知只有当  $M = 4^n$  时满足上式, 即此时按(5)式计算得到的 N 是上确界。其他式子可类似分析。

问题二中, 在确定了 N 和 M 后, 要求出所需分配方案以尽量满足条件 Y1~Y4, 可按上述式子求出大致的 N 范围后, 可依照模型设计搜索算法得到近似最优的分配策略。

## 4 模型建立

按照 BIBD 理论中的关联矩阵定义, 引入关联矩阵, 设元素为 0-1 变量  $x_{i,j}$ , 且定义为

$$x_{i,j} = \begin{cases} 0, \text{教师 } i \text{ 不面试学生 } j \\ 1, \text{教师 } i \text{ 面试学生 } j \end{cases} \quad (13)$$

其中,  $i = 1, 2, \dots, M; j = 1, 2, \dots, N$ 。下面按照题设及要求分别建立问题二与问题三三模型。

### 4.1 问题二模型建立

题设中有 Y1~Y4 共四个指标, 其中 Y2 要求是硬性要求, 必须满足, 可将其当成约束条件, 其他指标都是目标函数, 据此可建立多目标优化模型。

$$\min \begin{cases} Y_1 = \sqrt{\sum_{i=1}^M (r_i - \frac{1}{M} \sum_{i=1}^M r_i)^2} / \sum_{i=1}^M r_i \\ Y_{31} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Theta_{ij} (2) \\ Y_{32} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Theta_{ij} (3) \\ Y_4 = \max(\sum_{k=1}^N x_{ik} \times x_{jk}), i, j = 1, 2, \dots, M; i \neq j \end{cases} \quad (14)$$

$Y_1$  即为目标函数 Y1, 其中  $r_i$  为各个老师面试学生的数目, 即



$$r_i = \sum_{j=1}^N x_{ij} \quad (15)$$

$Y_{31}$ 与 $Y_{32}$ 分别是目标 Y3 中的函数，即任两个考生的“面试组”中有两位或三位老师相同的情形尽量的小。 $\Theta_{ij}(t)$ 是定义的符号函数，在此处为

$$\Theta_{ij}(t) = f\left(\sum_{k=1}^M x_{ki} \times x_{kj} - t\right) = \begin{cases} 1, & \text{if } \sum_{k=1}^M x_{ki} \times x_{kj} = t \\ 0, & \text{else} \end{cases} \quad (16)$$

$Y_4$ 即目标 Y4，被任意两位老师面试的两个学生集合中出现相同学生的人数需尽量的小。

模型的约束条件只有两个，即每个学生需接受四位老师的面试，且不同学生的面试组成员不能完全相同，即

$$s.t. \begin{cases} \sum_{i=1}^M x_{ij} = 4; j = 1, 2, \dots, N \\ \sum_{i=1}^M (x_{ij} \times x_{ik}) < 4; j, k = 1, 2, \dots, N \\ x_{ij} = 0, 1; i = 1, 2, \dots, M; j = 1, 2, \dots, N \end{cases} \quad (17)$$

## 4.2 问题三模型建立

此时所有老师分成文理科两部分，且文理科各占一半，不妨设文科老师编号从 1, 2, ..., M/2，理科老师标号从 M/2+1, M/2+2, ..., M；因每位学生需接受两位文科与两位理科老师的面试，相当于此处的约束条件改变，而模型的目标函数仍然是 Y1~Y4，与上节分析类似，可建立问题三模型如下。

目标函数

$$\min \begin{cases} Y_1 = \sqrt{\sum_{i=1}^M \left(r_i - \frac{1}{M} \sum_{i=1}^M r_i\right)^2} / \sum_{i=1}^M r_i \\ Y_{31} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Theta_{ij} \quad (2) \\ Y_{32} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Theta_{ij} \quad (3) \\ Y_4 = \max\left(\sum_{k=1}^N x_{ik} \times x_{jk}\right), i, j = 1, 2, \dots, M; i \neq j \end{cases} \quad (18)$$

约束条件

$$s.t. \left\{ \begin{array}{l} \sum_{i=1}^{M/2} x_{ij} = 2; j = 1, 2, \dots, N \\ \sum_{i=M/2+1}^M x_{ij} = 2; j = 1, 2, \dots, N \\ \sum_{i=1}^M (x_{ij} \times x_{ik}) < 4; j, k = 1, 2, \dots, N \\ x_{ij} = 0, 1; i = 1, 2, \dots, M; j = 1, 2, \dots, N \end{array} \right. \quad (19)$$

### 4.3 模型的求解分析

从(14)~(19)的模型表达式可知，这是一个多目标非线性的规划模型，且变量是 0-1 变量，在实际求解时欲得到最优解将非常困难，事实上在  $N=379$ ,  $M=24$  时几乎无法得到最优解；另外多目标函数有可能互相冲突，以致并不能得到同时满足三个最小值目标函数的最优解。这要求我们必须采用其他启发式算法或者近似搜索算法，从而得到一个近似最优解。

下文中我们将采用几种近似搜索算法，针对  $N=379$ ,  $M=24$  给出问题二与问题三的具体分配方案，并将模型的目标函数作为评价函数，从而评判几种搜索算法和分配方案的性质。

## 5 模型求解与算法评价

### 5.1 问题二模型求解

#### 5.1.1 搜索算法描述

按照前文给出的上界公式，可知在无二元组时 24 位老师最多可面试 42 组学生，因此必须存在二元组；当无三元组时可面试 504 名学生。因此时所需面试的学生数目为  $N=379 < 462$  名，可以在无三元组时完成全部面试。但模型中的目标函数除三元组数目尽量少之外，还需考虑到老师面试数量的均衡性与重复学生数尽量少等指标，因此在搜索算法和方案评价中将不仅仅只考虑到三元组数量这一个因素。

对  $M=24$  名老师进行 4 人一组的组合，可知产生的组合自然满足条件 Y2。首先考虑约束条件 1，按下述搜索算法搜索最终方案。

**Step1:** 对  $M=24$  产生每组个数为 4 的全部组合，将组合按行依顺序保存；选取其中的一行存入方案表中；

**Step2:** 从现有的组合中，从第一行开始，依次与方案表中的各行区组进行对比，若元素相同数量小于 2，将其存入方案表中，并将其从所有组合中删除；直至完成组合中的所有行比较；

**Step3:** 从现有的组合中，从第一行开始，依次与方案表中的各行区组进行对比，若元素相同数量小于 3，将其存入方案表中，并将其从所有组合中删除；直至完成组合中的所有行比较；若方案表区组数仍小于  $N$ ；进行 Step4，否则结束循环；

**Step4:** 计算当前方案表中各老师的面试学生数，并从现有的所有组合中，优先选择含有最少学生数量的老师的组合，将其存入方案表中，并将其从所有组合中删除；循环本步骤直到  $N$  达到 379，从而结束搜索。

该搜索算法每次优先考虑与当前区组重复数最少的组合，并将其存入方案表中，不妨称

之为重复数优先搜索算法。

### 5.1.2 分配方案与模型评价

在 MATLAB 中编程完成上述算法，其中具体的老师与学生分配方案见附录表格。

将分配方案表格转换成关联矩阵，可先产生一个全零的  $379 \times 24$  的矩阵  $H(i, j)$ ，然后按照分配表格中的矩阵将  $H(i, j)$  相应元素置一。根据关联矩阵易算得各老师的面试学生数如下。

表 1 各老师面试学生数量对比

老师编号	1	2	3	4	5	6	7	8
学生数量	64	64	64	63	64	63	63	63
老师编号	9	10	11	12	13	14	15	16
学生数量	63	63	63	63	63	63	63	63
老师编号	17	18	19	20	21	22	23	24
学生数量	63	63	63	63	63	63	63	63

从表中可看出，各老师面试学生数量非常均衡，有 20 位老师面试 63 个学生，有 4 位老师面试 64 个学生；事实上从算法搜索过程中可知，当  $N=378$  时正好找到所有老师面试 63 个学生的所有二元组组合，此时只要从剩下的组合里随机抽取 1 个区组即可。此时选取的是 [1 2 3 5] 面试组，正好对应表中相应的老师需多面试 1 位学生。

按题设  $Y_1 \sim Y_4$  分别评判算法和模型的特性，因  $Y_2$  作为约束条件已经满足，将  $Y_1$ 、 $Y_3$ 、 $Y_4$  分别按下述式子量化。

$$\left\{ \begin{array}{l} \delta_1 = Y_1 = \sqrt{\sum_{i=1}^M (r_i - \frac{1}{M} \sum_{i=1}^M r_i)^2} / \sum_{i=1}^M r_i \\ \delta_{31} = Y_{31} / C_N^2 \\ \delta_{32} = Y_{32} / C_N^2 \\ \delta_{41} = \max(S_k) \\ \delta_{42} = \frac{\sum_{k=1}^{C_M} S_k}{C_M^2} \end{array} \right. \quad (20)$$

其中  $\delta_1$  即每个老师面试学生数的均方差，以衡量老师面试学生数量的均衡性； $\delta_{31}$  是有两位老师相同的学生对数量占有所有学生对  $C_N^2$  的比率； $\delta_{32}$  是有三位老师相同的学生对数量占有所有学生对  $C_N^2$  的比率； $\delta_{41}$  是各个老师对之间重复学生数的最大值， $\delta_{42}$  是老师对之间重复学生数的均值，用这两个指标衡量出现相同学生数量尽量少这一目标。

利用 MATLAB 编程求解上述指标，可得出具体值如下表：

表 2 算法与模型指标评价

指标	$\delta_1$	$\delta_{31}$	$\delta_{32}$	$\delta_{41}$	$\delta_{42}$
数值	0.0012	0.1215	5.5842e-005	12	8.2391

可看出搜索算法有着很好的性能，每位老师面试数量非常均衡，面试的集合中出现相同学生人数最大值为 12，平均值为 8.2391，均稳定在很小范围内。另外对于指标 Y3，计算得到的有二位老师相同与三位老师相同的情形所占比例都非常小。事实上，可在所有  $C_N^2$  学生对中，求出没有老师相同、有一位老师相同、有二位和三位老师相同的比例分别，并用直观饼图显示如下。

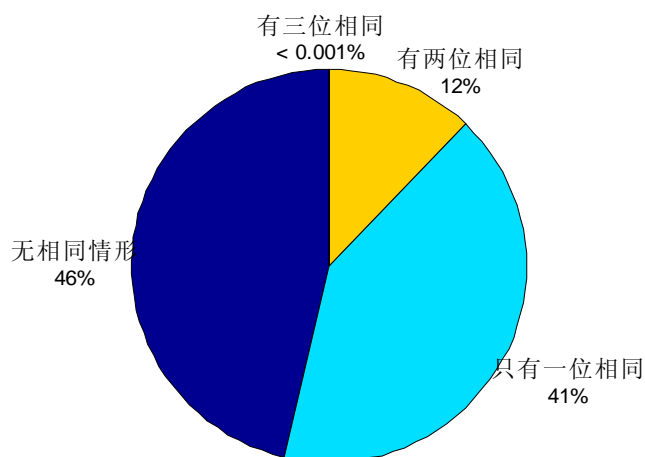


图 1 所有学生面试组对中老师相同情况

## 5.2 问题三模型求解

### 5.2.1 搜索算法设计

按照前文给出的文理科老师分别面试的上界公式，此时每个面试组都需两位文科老师和两位理科老师；可知在无二元组时 24 位老师最多可面试 36 名学生，因此必须存在二元组；当无三元组时可面试 396 名学生。因此此时所需面试的学生数目为  $N=379 < 396$  名，可以在无三元组时完成全部面试。但在搜索算法中可能并不一定搜索出全部二元组合，此时我们将应用 Y1~Y4 的目标函数来评判搜索性能。

此时的一种搜索算法仍沿用上节的算法。得到的结果因篇幅限制不再列出（运行附录中的程序即可得到）。若按上节的搜索算法对文理分科的面试区组进行搜索，此时首先将要产生符合约束的所有组合，即除了上节的约束条件之外，还需加上每个面试组均各有两名文科和两名理科老师这个约束。设文科老师编号 1~12，理科老师编号 13~24，分别进行  $C_{1-12}^2$  和  $C_{13-24}^2$  的全组合，然后对组合进行混合，即分别选出一组理科老师与一组文科老师最终组成符合条件的 4 人面试组。可计算得这种组合共有  $66*66=4356$  组。

另外我们也提出了全组合删除算法进行搜索。考虑到若全部面试的学生数为 396 名，即用完了老师组合中的所有二元组，而且此时所有老师面试学生数量均衡相等，这种分配方案在 396 名学生下将是最优的；此时只有 379 名学生，因与最优的 396 十分接近，可从最优的 396 个面试组合中进行合理删除，得到 379 个面试组的近似最优解。

### 5.2.2 分配方案与模型评价

首先分析上节的搜索算法在本问中的性能。

表 3 文理分科下各老师面试学生数量对比

老师编号	1	2	3	4	5	6	7	8
学生数量	66	63	59	61	64	70	69	56
老师编号	9	10	11	12	13	14	15	16
学生数量	62	64	62	62	69	64	63	62
老师编号	17	18	19	20	21	22	23	24
学生数量	65	59	68	61	61	64	65	57

可看出老师面试学生数有波动，最多的有 70 名，但大部分集中在 61~65 左右。可知文理分科老师面试情况下，这种算法不能取得跟上节一样的十分好的均衡性能。

仍按照(20)的指标量化计算公式，求得各项指标数值如下：

表 4 文理分科时算法与模型指标评价

指标	$\delta_1$	$\delta_{31}$	$\delta_{32}$	$\delta_{41}$	$\delta_{42}$
数值	0.0126	0.1225	0.0038	17	8.2391

对于指标 Y4，此时最多重复数达到 17 名，算法整体搜索性能相对于问题二有所逊色；但整体而言，从上表中的各个指标考察，算法仍然达到了较好的性能。

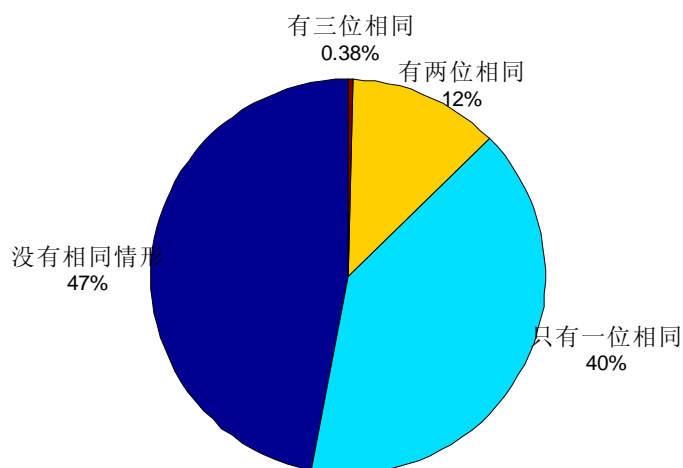


图 2 所有学生面试组对中老师相同情况

若全部面试的学生数为 396 名，即用完了老师组合中的所有二元组，而且此时所有老师面试学生数量均衡相等。

若用全组合删除算法处理本题，即首先产生 396 个符合条件的最多含二元组的全组合，再从中删除 17 组，我们删除的原则是均衡考虑，依次间隔  $396/17 \approx 23$  组则删除一组，最终得到 379 组的分配方案。下面对该算法进行分析。

表 5 全组合删除算法下各老师面试学生数

老师编号	1	2	3	4	5	6	7	8
学生数量	63	62	64	63	64	64	63	63
老师编号	9	10	11	12	13	14	15	16
学生数量	63	62	64	63	64	61	63	62
老师编号	17	18	19	20	21	22	23	24
学生数量	66	64	61	64	63	63	64	63

表 6 全组合删除算法与模型指标评价

指标	$\delta_1$	$\delta_{31}$	$\delta_{32}$	$\delta_{41}$	$\delta_{42}$
数值	0.0034	0.1266	0	11	8.2391

从上述两表可知，每个老师面试学生数十分均衡，大部分集中在 63~64 名，变动方差为 0.0034；而且该分配方案不存在三元组，即任意两个面试组之间没有出现三位老师相同的情形；任两位老师面试的学生集合出现相同学生的最大值仅是 11 人。

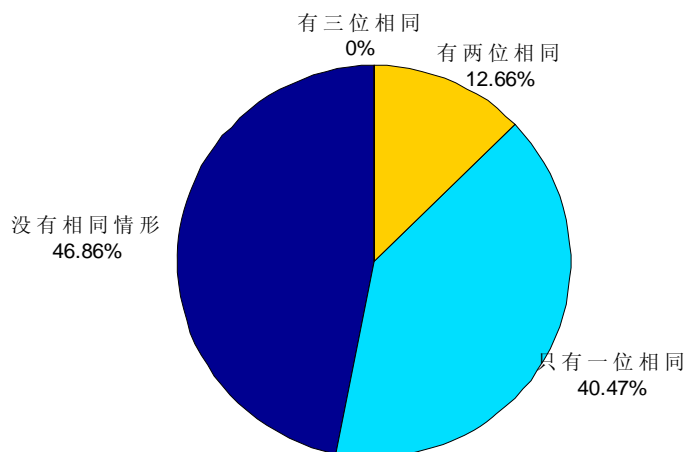


图 3 全删除算法面试组对老师相同情况

可知，全组合删除算法有着极佳的性能。

另外，由全组合删除算法得到的分配方案也适用于问题二中的不区分文理科老师的情形。

## 6 面试组分配方案的其他建议

考虑到现实生活面试存在的各种情况，为了更好地达到面试分配的均匀性与面试的公平性，我们提出了如下的建议：

(1) 实际面试中，各老师的打分都有一定的偏向性，比如有老师偏于打高分而另一些老师偏于打低分，为了保证面试的公平性，应尽量将打高分与打低分老师搭配在一起组成面试组，以尽量消除老师打分因素对最终录取的影响；

(2) 实际中不同专业对男女考生有限制，不同专业老师也可能对男生或女生有偏好，在设计面试方案时应考虑到这些限制和偏好以作调整；

(3) 在面试之前可对面面的老师进行问答和咨询，调查他们对面试组设计的建议和要求，并了解各老师的大体特征；

(4) 在面试之前也可对参与面试的考生进行简要的问答，以了解考生的整体情况与特征；并据此进一步完善整个面试过程。

---

## 7 参考文献

- [1] 卢开澄, 卢华明. 组合数学[M]. 北京: 清华大学出版社, 2006
- [2] 杨振生. 组合数学及其算法[M]. 合肥: 中国科学技术大学出版社, 1997
- [3] 吴孟达. “学生面试问题”评注[J]. 数学的实践与认识, 2007, 37(14): 169-170
- [4] 栗新宇, 赵思齐, 卫佳蕴. 学生面试问题[J]. 数学的实践与认识, 2007, 37(14): 161-167

## 8 附录

### 8.1 分配方案表格

表 7 重复数优先算法对问题二的具体分配方案

编号	面试 老师组合				编号	面试 老师组合				编号	面试 老师组合			
	1	2	3	4		1	2	3	4		1	2	3	4
1	1	2	3	4	128	2	8	20	22	255	9	12	13	16
2	5	6	7	8	129	3	4	5	6	256	9	12	14	15
3	9	10	11	12	130	3	4	7	8	257	9	12	17	20
4	13	14	15	16	131	3	4	9	10	258	9	12	18	19
5	17	18	19	20	132	3	4	11	12	259	9	12	21	24
6	21	22	23	24	133	3	4	13	14	260	9	12	22	23
7	1	5	9	13	134	3	4	15	16	261	9	13	17	21
8	1	6	10	14	135	3	4	17	18	262	9	13	18	22
9	1	7	11	15	136	3	4	19	20	263	9	13	19	23
10	1	8	12	16	137	3	4	21	22	264	9	13	20	24
11	2	5	10	15	138	3	4	23	24	265	9	14	17	22
12	2	6	9	16	139	3	5	9	15	266	9	14	18	21
13	2	7	12	13	140	3	5	10	14	267	9	14	19	24
14	2	8	11	14	141	3	5	12	13	268	9	14	20	23
15	3	5	11	16	142	3	5	17	23	269	9	15	17	23
16	3	6	12	15	143	3	5	18	24	270	9	15	18	24
17	3	7	9	14	144	3	5	19	21	271	9	15	19	21
18	3	8	10	13	145	3	5	20	22	272	9	15	20	22
19	4	5	12	14	146	3	6	9	13	273	9	16	17	24
20	4	6	11	13	147	3	6	10	16	274	9	16	18	23
21	4	7	10	16	148	3	6	11	14	275	9	16	19	22
22	4	8	9	15	149	3	6	17	24	276	9	16	20	21
23	1	2	5	6	150	3	6	18	23	277	10	11	13	16
24	1	2	7	8	151	3	6	19	22	278	10	11	14	15
25	1	2	9	10	152	3	6	20	21	279	10	11	17	20
26	1	2	11	12	153	3	7	10	15	280	10	11	18	19
27	1	2	13	14	154	3	7	11	13	281	10	11	21	24
28	1	2	15	16	155	3	7	12	16	282	10	11	22	23
29	1	2	17	18	156	3	7	17	21	283	10	12	13	15
30	1	2	19	20	157	3	7	18	22	284	10	12	14	16
31	1	2	21	22	158	3	7	19	23	285	10	12	17	19
32	1	2	23	24	159	3	7	20	24	286	10	12	18	20
33	1	3	5	7	160	3	8	9	16	287	10	12	21	23
34	1	3	6	8	161	3	8	11	15	288	10	12	22	24



35	1	3	9	11	162	3	8	12	14	289	10	13	17	22
36	1	3	10	12	163	3	8	17	22	290	10	13	18	21
37	1	3	13	15	164	3	8	18	21	291	10	13	19	24
38	1	3	14	16	165	3	8	19	24	292	10	13	20	23
39	1	3	17	19	166	3	8	20	23	293	10	14	17	21
40	1	3	18	20	167	4	5	9	16	294	10	14	18	22
41	1	3	21	23	168	4	5	10	13	295	10	14	19	23
42	1	3	22	24	169	4	5	11	15	296	10	14	20	24
43	1	4	5	8	170	4	5	17	24	297	10	15	17	24
44	1	4	6	7	171	4	5	18	23	298	10	15	18	23
45	1	4	9	12	172	4	5	19	22	299	10	15	19	22
46	1	4	10	11	173	4	5	20	21	300	10	15	20	21
47	1	4	13	16	174	4	6	9	14	301	10	16	17	23
48	1	4	14	15	175	4	6	10	15	302	10	16	18	24
49	1	4	17	20	176	4	6	12	16	303	10	16	19	21
50	1	4	18	19	177	4	6	17	23	304	10	16	20	22
51	1	4	21	24	178	4	6	18	24	305	11	12	13	14
52	1	4	22	23	179	4	6	19	21	306	11	12	15	16
53	1	5	10	16	180	4	6	20	22	307	11	12	17	18
54	1	5	11	14	181	4	7	9	13	308	11	12	19	20
55	1	5	12	15	182	4	7	11	14	309	11	12	21	22
56	1	5	17	21	183	4	7	12	15	310	11	12	23	24
57	1	5	18	22	184	4	7	17	22	311	11	13	17	23
58	1	5	19	23	185	4	7	18	21	312	11	13	18	24
59	1	5	20	24	186	4	7	19	24	313	11	13	19	21
60	1	6	9	15	187	4	7	20	23	314	11	13	20	22
61	1	6	11	16	188	4	8	10	14	315	11	14	17	24
62	1	6	12	13	189	4	8	11	16	316	11	14	18	23
63	1	6	17	22	190	4	8	12	13	317	11	14	19	22
64	1	6	18	21	191	4	8	17	21	318	11	14	20	21
65	1	6	19	24	192	4	8	18	22	319	11	15	17	21
66	1	6	20	23	193	4	8	19	23	320	11	15	18	22
67	1	7	9	16	194	4	8	20	24	321	11	15	19	23
68	1	7	10	13	195	5	6	9	10	322	11	15	20	24
69	1	7	12	14	196	5	6	11	12	323	11	16	17	22
70	1	7	17	23	197	5	6	13	14	324	11	16	18	21
71	1	7	18	24	198	5	6	15	16	325	11	16	19	24
72	1	7	19	21	199	5	6	17	18	326	11	16	20	23
73	1	7	20	22	200	5	6	19	20	327	12	13	17	24
74	1	8	9	14	201	5	6	21	22	328	12	13	18	23
75	1	8	10	15	202	5	6	23	24	329	12	13	19	22
76	1	8	11	13	203	5	7	9	11	330	12	13	20	21
77	1	8	17	24	204	5	7	10	12	331	12	14	17	23

78	1	8	18	23	205	5	7	13	15	332	12	14	18	24
79	1	8	19	22	206	5	7	14	16	333	12	14	19	21
80	1	8	20	21	207	5	7	17	19	334	12	14	20	22
81	2	3	5	8	208	5	7	18	20	335	12	15	17	22
82	2	3	6	7	209	5	7	21	23	336	12	15	18	21
83	2	3	9	12	210	5	7	22	24	337	12	15	19	24
84	2	3	10	11	211	5	8	9	12	338	12	15	20	23
85	2	3	13	16	212	5	8	10	11	339	12	16	17	21
86	2	3	14	15	213	5	8	13	16	340	12	16	18	22
87	2	3	17	20	214	5	8	14	15	341	12	16	19	23
88	2	3	18	19	215	5	8	17	20	342	12	16	20	24
89	2	3	21	24	216	5	8	18	19	343	13	14	17	18
90	2	3	22	23	217	5	8	21	24	344	13	14	19	20
91	2	4	5	7	218	5	8	22	23	345	13	14	21	22
92	2	4	6	8	219	6	7	9	12	346	13	14	23	24
93	2	4	9	11	220	6	7	10	11	347	13	15	17	19
94	2	4	10	12	221	6	7	13	16	348	13	15	18	20
95	2	4	13	15	222	6	7	14	15	349	13	15	21	23
96	2	4	14	16	223	6	7	17	20	350	13	15	22	24
97	2	4	17	19	224	6	7	18	19	351	13	16	17	20
98	2	4	18	20	225	6	7	21	24	352	13	16	18	19
99	2	4	21	23	226	6	7	22	23	353	13	16	21	24
100	2	4	22	24	227	6	8	9	11	354	13	16	22	23
101	2	5	9	14	228	6	8	10	12	355	14	15	17	20
102	2	5	11	13	229	6	8	13	15	356	14	15	18	19
103	2	5	12	16	230	6	8	14	16	357	14	15	21	24
104	2	5	17	22	231	6	8	17	19	358	14	15	22	23
105	2	5	18	21	232	6	8	18	20	359	14	16	17	19
106	2	5	19	24	233	6	8	21	23	360	14	16	18	20
107	2	5	20	23	234	6	8	22	24	361	14	16	21	23
108	2	6	10	13	235	7	8	9	10	362	14	16	22	24
109	2	6	11	15	236	7	8	11	12	363	15	16	17	18
110	2	6	12	14	237	7	8	13	14	364	15	16	19	20
111	2	6	17	21	238	7	8	15	16	365	15	16	21	22
112	2	6	18	22	239	7	8	17	18	366	15	16	23	24
113	2	6	19	23	240	7	8	19	20	367	17	18	21	22
114	2	6	20	24	241	7	8	21	22	368	17	18	23	24
115	2	7	9	15	242	7	8	23	24	369	17	19	21	23
116	2	7	10	14	243	9	10	13	14	370	17	19	22	24
117	2	7	11	16	244	9	10	15	16	371	17	20	21	24
118	2	7	17	24	245	9	10	17	18	372	17	20	22	23
119	2	7	18	23	246	9	10	19	20	373	18	19	21	24
120	2	7	19	22	247	9	10	21	22	374	18	19	22	23

121	2	7	20	21	248	9	10	23	24	375	18	20	21	23
122	2	8	9	13	249	9	11	13	15	376	18	20	22	24
123	2	8	10	16	250	9	11	14	16	377	19	20	21	22
124	2	8	12	15	251	9	11	17	19	378	19	20	23	24
125	2	8	17	23	252	9	11	18	20	379	1	2	3	5
126	2	8	18	24	253	9	11	21	23					
127	2	8	19	21	254	9	11	22	24					

表 8 重复数优先算法对问题三的具体分配方案

编号	面试老师组合				编号	面试老师组合				编号	面试老师组合			
1	1	2	13	14	128	3	11	16	22	255	8	10	19	21
2	1	2	15	16	129	3	12	13	6	256	8	10	20	22
3	1	2	17	18	130	3	12	14	8	257	8	11	17	22
4	1	2	19	20	131	3	12	15	10	258	8	11	18	21
5	1	2	21	22	132	3	12	16	12	259	8	11	19	24
6	1	2	23	24	133	4	5	13	14	260	8	11	20	23
7	1	3	13	15	134	4	5	14	16	261	8	12	17	21
8	1	3	14	16	135	4	5	15	18	262	8	12	18	22
9	1	3	17	19	136	4	5	16	20	263	8	12	19	23
10	1	3	18	20	137	4	6	13	22	264	8	12	20	24
11	1	3	21	23	138	4	6	14	24	265	9	10	13	14
12	1	3	22	24	139	4	6	15	15	266	9	10	15	16
13	1	4	13	16	140	4	6	16	14	267	9	10	17	18
14	1	4	14	15	141	4	7	13	13	268	9	10	19	20
15	1	4	17	20	142	4	7	14	23	269	9	10	21	22
16	1	4	18	19	143	4	7	15	24	270	9	10	23	24
17	1	4	21	24	144	4	7	16	21	271	9	11	13	15
18	1	4	22	23	145	4	8	13	22	272	9	11	14	16
19	1	5	13	17	146	4	8	14	13	273	9	11	17	19
20	1	5	14	18	147	4	8	15	16	274	9	11	18	20
21	1	5	15	19	148	4	8	16	14	275	9	11	21	23
22	1	5	16	20	149	4	9	13	24	276	9	11	22	24
23	1	6	13	18	150	4	9	14	23	277	9	12	13	16
24	1	6	14	17	151	4	9	15	22	278	9	12	14	15
25	1	6	15	20	152	4	9	16	21	279	9	12	17	20
26	1	6	16	19	153	4	10	13	15	280	9	12	18	19
27	1	7	13	19	154	4	10	14	13	281	9	12	21	24
28	1	7	14	20	155	4	10	15	16	282	9	12	22	23
29	1	7	15	17	156	4	10	16	21	283	10	11	13	16
30	1	7	16	18	157	4	11	13	22	284	10	11	14	15

31	1	8	13	20	158	4	11	14	23	285	10	11	17	20
32	1	8	14	19	159	4	11	15	24	286	10	11	18	19
33	1	8	15	18	160	4	11	16	16	287	10	11	21	24
34	1	8	16	17	161	4	12	13	15	288	10	11	22	23
35	1	9	13	21	162	4	12	14	14	289	10	12	13	15
36	1	9	14	22	163	4	12	15	22	290	10	12	14	16
37	1	9	15	23	164	4	12	16	21	291	10	12	17	19
38	1	9	16	24	165	5	6	13	24	292	10	12	18	20
39	1	10	13	22	166	5	6	15	23	293	10	12	21	23
40	1	10	14	21	167	5	6	17	16	294	10	12	22	24
41	1	10	15	24	168	5	6	19	13	295	11	12	13	14
42	1	10	16	23	169	5	6	21	15	296	11	12	15	16
43	1	11	13	23	170	5	6	23	24	297	11	12	17	18
44	1	11	14	24	171	5	7	13	23	298	11	12	19	20
45	1	11	15	21	172	5	7	14	22	299	11	12	21	22
46	1	11	16	22	173	5	7	17	21	300	11	12	23	24
47	1	12	13	24	174	5	7	18	14	301	6	11	17	23
48	1	12	14	23	175	5	7	21	15	302	3	8	18	23
49	1	12	15	22	176	5	7	22	16	303	5	11	21	22
50	1	12	16	21	177	5	8	13	23	304	3	7	14	23
51	2	3	13	16	178	5	8	14	24	305	7	8	18	19
52	2	3	14	15	179	5	8	17	21	306	10	11	13	17
53	2	3	17	20	180	5	8	18	22	307	5	10	19	20
54	2	3	18	19	181	5	8	21	13	308	3	8	14	19
55	2	3	21	24	182	5	8	22	14	309	10	11	13	24
56	2	3	22	23	183	5	9	17	15	310	2	12	20	21
57	2	4	13	15	184	5	9	18	22	311	8	12	16	24
58	2	4	14	16	185	5	9	19	21	312	7	10	15	17
59	2	4	17	19	186	5	9	20	24	313	4	7	13	15
60	2	4	18	20	187	5	10	17	23	314	7	8	15	23
61	2	4	21	23	188	5	10	18	14	315	2	11	18	21
62	2	4	22	24	189	5	10	19	16	316	9	11	17	23
63	2	5	13	18	190	5	10	20	13	317	2	6	22	24
64	2	5	14	17	191	5	11	17	21	318	8	11	14	18
65	2	5	15	20	192	5	11	18	22	319	5	9	14	24
66	2	5	16	19	193	5	11	19	23	320	5	10	13	18
67	2	6	13	17	194	5	11	20	24	321	6	12	23	24
68	2	6	14	18	195	5	12	17	10	322	5	6	14	16
69	2	6	15	19	196	5	12	18	12	323	6	11	19	24
70	2	6	16	20	197	5	12	19	14	324	4	9	18	22
71	2	7	13	20	198	5	12	20	16	325	4	12	18	19
72	2	7	14	19	199	6	7	13	18	326	7	12	17	23
73	2	7	15	18	200	6	7	14	20	327	4	12	17	23

74	2	7	16	17	201	6	7	17	22	328	2	11	21	23
75	2	8	13	19	202	6	7	18	24	329	9	10	19	24
76	2	8	14	20	203	6	7	21	11	330	3	7	15	20
77	2	8	15	17	204	6	7	22	12	331	5	6	19	23
78	2	8	16	18	205	6	8	13	15	332	5	11	17	24
79	2	9	13	22	206	6	8	14	16	333	4	5	13	21
80	2	9	14	21	207	6	8	17	19	334	6	8	22	23
81	2	9	15	24	208	6	8	18	20	335	5	11	13	17
82	2	9	16	23	209	6	8	21	23	336	3	5	14	22
83	2	10	13	21	210	6	8	22	24	337	2	9	13	16
84	2	10	14	22	211	6	9	17	12	338	1	7	18	23
85	2	10	15	23	212	6	9	18	11	339	6	11	16	22
86	2	10	16	24	213	6	9	19	16	340	7	8	16	24
87	2	11	13	24	214	6	9	20	15	341	9	11	20	22
88	2	11	14	23	215	6	10	17	20	342	1	5	15	20
89	2	11	15	22	216	6	10	18	19	343	5	9	13	18
90	2	11	16	21	217	6	10	19	24	344	7	11	17	19
91	2	12	13	23	218	6	10	20	23	345	9	11	19	22
92	2	12	14	24	219	6	11	17	12	346	3	6	20	22
93	2	12	15	21	220	6	11	18	11	347	10	11	15	17
94	2	12	16	22	221	6	11	19	16	348	1	6	16	18
95	3	4	13	14	222	6	11	20	15	349	7	8	13	19
96	3	4	15	16	223	6	12	17	20	350	2	3	23	24
97	3	4	17	18	224	6	12	18	19	351	1	3	19	20
98	3	4	19	20	225	6	12	19	24	352	5	12	13	16
99	3	4	21	22	226	6	12	20	23	353	4	11	21	23
100	3	4	23	24	227	7	8	13	11	354	3	7	13	23
101	3	5	13	19	228	7	8	15	12	355	3	7	15	24
102	3	5	14	20	229	7	8	17	15	356	3	6	21	24
103	3	5	15	17	230	7	8	19	16	357	6	9	17	19
104	3	5	16	18	231	7	8	21	19	358	8	12	21	23
105	3	6	13	20	232	7	8	23	20	359	3	9	13	17
106	3	6	14	19	233	7	9	17	23	360	10	12	14	22
107	3	6	15	18	234	7	9	18	24	361	5	11	15	19
108	3	6	16	17	235	7	9	19	10	362	9	10	13	21
109	3	7	13	17	236	7	9	20	12	363	8	11	15	23
110	3	7	14	18	237	7	10	17	14	364	1	8	15	24
111	3	7	15	19	238	7	10	18	16	365	8	12	20	23
112	3	7	16	20	239	7	10	19	18	366	4	5	14	23
113	3	8	13	18	240	7	10	20	20	367	8	11	21	23
114	3	8	14	17	241	7	11	17	22	368	5	12	19	21
115	3	8	15	20	242	7	11	18	24	369	5	6	22	23
116	3	8	16	19	243	7	11	19	14	370	4	8	19	23

117	3	9	13	23	244	7	11	20	16	371	1	6	19	21
118	3	9	14	24	245	7	12	17	18	372	2	10	13	15
119	3	9	15	21	246	7	12	18	20	373	5	6	14	20
120	3	9	16	22	247	7	12	19	22	374	1	2	16	20
121	3	10	13	24	248	7	12	20	24	375	7	8	15	22
122	3	10	14	23	249	8	9	17	15	376	6	9	15	21
123	3	10	15	22	250	8	9	18	16	377	4	10	19	22
124	3	10	16	21	251	8	9	19	19	378	3	6	16	20
125	3	11	13	21	252	8	9	20	20	379	5	6	17	23
126	3	11	14	22	253	8	10	17	23					
127	3	11	15	23	254	8	10	18	24					

表 9 全组合删除算法对问题三的具体分配方案

编号	面试 老师组合				编号	面试 老师组合				编号	面试 老师组合			
1	1	2	15	16	128	3	5	14	24	255	5	12	14	21
2	1	2	17	18	129	3	5	15	17	256	5	12	15	20
3	1	2	19	20	130	3	5	16	18	257	5	12	17	24
4	1	2	21	22	131	3	5	19	21	258	5	12	18	23
5	1	2	23	24	132	3	5	20	22	259	6	7	13	16
6	1	3	13	15	133	3	6	13	24	260	6	7	14	15
7	1	3	14	16	134	3	6	14	23	261	6	7	17	20
8	1	3	17	19	135	3	6	15	18	262	6	7	18	19
9	1	3	18	20	136	3	6	16	17	263	6	7	21	24
10	1	3	21	23	137	3	6	19	22	264	6	7	22	23
11	1	3	22	24	138	3	6	20	21	265	6	8	13	15
12	1	4	13	16	139	3	7	13	21	266	6	8	14	16
13	1	4	14	15	140	3	7	14	22	267	6	8	17	19
14	1	4	17	20	141	3	7	16	20	268	6	8	18	20
15	1	4	18	19	142	3	7	17	23	269	6	8	21	23
16	1	4	21	24	143	3	7	18	24	270	6	8	22	24
17	1	4	22	23	144	3	8	13	22	271	6	9	13	20
18	1	5	13	17	145	3	8	14	21	272	6	9	14	19
19	1	5	14	18	146	3	8	15	20	273	6	9	15	24
20	1	5	15	21	147	3	8	16	19	274	6	9	16	23
21	1	5	16	22	148	3	8	17	24	275	6	9	17	22
22	1	5	19	23	149	3	8	18	23	276	6	9	18	21
23	1	5	20	24	150	3	9	13	17	277	6	10	13	19
24	1	6	14	17	151	3	9	14	18	278	6	10	14	20
25	1	6	15	22	152	3	9	15	21	279	6	10	15	23

26	1	6	16	21	153	3	9	16	22	280	6	10	17	21
27	1	6	19	24	154	3	9	19	23	281	6	10	18	22
28	1	6	20	23	155	3	9	20	24	282	6	11	13	22
29	1	7	13	19	156	3	10	13	18	283	6	11	14	21
30	1	7	14	20	157	3	10	14	17	284	6	11	15	20
31	1	7	15	23	158	3	10	15	22	285	6	11	16	19
32	1	7	16	24	159	3	10	16	21	286	6	11	17	24
33	1	7	17	21	160	3	10	19	24	287	6	11	18	23
34	1	7	18	22	161	3	10	20	23	288	6	12	13	21
35	1	8	13	20	162	3	11	13	19	289	6	12	14	22
36	1	8	14	19	163	3	11	14	20	290	6	12	15	19
37	1	8	15	24	164	3	11	16	24	291	6	12	16	20
38	1	8	16	23	165	3	11	17	21	292	6	12	17	23
39	1	8	17	22	166	3	11	18	22	293	6	12	18	24
40	1	8	18	21	167	3	12	13	20	294	7	8	13	14
41	1	9	13	21	168	3	12	14	19	295	7	8	15	16
42	1	9	14	22	169	3	12	15	24	296	7	8	17	18
43	1	9	15	19	170	3	12	16	23	297	7	8	19	20
44	1	9	16	20	171	3	12	17	22	298	7	8	21	22
45	1	9	17	23	172	3	12	18	21	299	7	8	23	24
46	1	9	18	24	173	4	5	13	24	300	7	9	13	23
47	1	10	13	22	174	4	5	14	23	301	7	9	14	24
48	1	10	15	20	175	4	5	15	18	302	7	9	15	17
49	1	10	16	19	176	4	5	16	17	303	7	9	16	18
50	1	10	17	24	177	4	5	19	22	304	7	9	20	22
51	1	10	18	23	178	4	5	20	21	305	7	10	13	24
52	1	11	13	23	179	4	6	13	23	306	7	10	14	23
53	1	11	14	24	180	4	6	14	24	307	7	10	15	18
54	1	11	15	17	181	4	6	15	17	308	7	10	16	17
55	1	11	16	18	182	4	6	16	18	309	7	10	19	22
56	1	11	19	21	183	4	6	19	21	310	7	10	20	21
57	1	11	20	22	184	4	6	20	22	311	7	11	13	17
58	1	12	13	24	185	4	7	13	22	312	7	11	14	18
59	1	12	14	23	186	4	7	14	21	313	7	11	15	21
60	1	12	15	18	187	4	7	16	19	314	7	11	16	22
61	1	12	16	17	188	4	7	17	24	315	7	11	19	23
62	1	12	19	22	189	4	7	18	23	316	7	11	20	24
63	1	12	20	21	190	4	8	13	21	317	7	12	13	18
64	2	3	13	16	191	4	8	14	22	318	7	12	14	17
65	2	3	14	15	192	4	8	15	19	319	7	12	15	22
66	2	3	17	20	193	4	8	16	20	320	7	12	16	21
67	2	3	18	19	194	4	8	17	23	321	7	12	19	24
68	2	3	21	24	195	4	8	18	24	322	7	12	20	23

69	2	3	22	23	196	4	9	13	18	323	8	9	13	24
70	2	4	13	15	197	4	9	14	17	324	8	9	14	23
71	2	4	17	19	198	4	9	15	22	325	8	9	15	18
72	2	4	18	20	199	4	9	16	21	326	8	9	16	17
73	2	4	21	23	200	4	9	19	24	327	8	9	20	21
74	2	4	22	24	201	4	9	20	23	328	8	10	13	23
75	2	5	13	18	202	4	10	13	17	329	8	10	14	24
76	2	5	14	17	203	4	10	14	18	330	8	10	15	17
77	2	5	15	22	204	4	10	15	21	331	8	10	16	18
78	2	5	16	21	205	4	10	16	22	332	8	10	19	21
79	2	5	19	24	206	4	10	19	23	333	8	10	20	22
80	2	5	20	23	207	4	10	20	24	334	8	11	13	18
81	2	6	13	17	208	4	11	13	20	335	8	11	14	17
82	2	6	14	18	209	4	11	14	19	336	8	11	15	22
83	2	6	15	21	210	4	11	15	24	337	8	11	16	21
84	2	6	16	22	211	4	11	17	22	338	8	11	19	24
85	2	6	19	23	212	4	11	18	21	339	8	11	20	23
86	2	6	20	24	213	4	12	13	19	340	8	12	13	17
87	2	7	13	20	214	4	12	14	20	341	8	12	14	18
88	2	7	14	19	215	4	12	15	23	342	8	12	15	21
89	2	7	15	24	216	4	12	16	24	343	8	12	16	22
90	2	7	16	23	217	4	12	17	21	344	8	12	19	23
91	2	7	17	22	218	4	12	18	22	345	8	12	20	24
92	2	7	18	21	219	5	6	13	14	346	9	10	13	14
93	2	8	13	19	220	5	6	15	16	347	9	10	15	16
94	2	8	15	23	221	5	6	17	18	348	9	10	17	18
95	2	8	16	24	222	5	6	19	20	349	9	10	19	20
96	2	8	17	21	223	5	6	21	22	350	9	10	23	24
97	2	8	18	22	224	5	6	23	24	351	9	11	13	15
98	2	9	13	22	225	5	7	13	15	352	9	11	14	16
99	2	9	14	21	226	5	7	14	16	353	9	11	17	19
100	2	9	15	20	227	5	7	17	19	354	9	11	18	20
101	2	9	16	19	228	5	7	18	20	355	9	11	21	23
102	2	9	17	24	229	5	7	21	23	356	9	11	22	24
103	2	9	18	23	230	5	7	22	24	357	9	12	13	16
104	2	10	13	21	231	5	8	13	16	358	9	12	14	15
105	2	10	14	22	232	5	8	14	15	359	9	12	17	20
106	2	10	15	19	233	5	8	17	20	360	9	12	18	19
107	2	10	16	20	234	5	8	21	24	361	9	12	21	24
108	2	10	17	23	235	5	8	22	23	362	9	12	22	23
109	2	10	18	24	236	5	9	13	19	363	10	11	13	16
110	2	11	13	24	237	5	9	14	20	364	10	11	14	15
111	2	11	14	23	238	5	9	15	23	365	10	11	17	20



112	2	11	15	18	239	5	9	16	24	366	10	11	18	19
113	2	11	16	17	240	5	9	17	21	367	10	11	21	24
114	2	11	19	22	241	5	9	18	22	368	10	11	22	23
115	2	11	20	21	242	5	10	13	20	369	10	12	13	15
116	2	12	13	23	243	5	10	14	19	370	10	12	14	16
117	2	12	15	17	244	5	10	15	24	371	10	12	17	19
118	2	12	16	18	245	5	10	16	23	372	10	12	18	20
119	2	12	19	21	246	5	10	17	22	373	10	12	21	23
120	2	12	20	22	247	5	10	18	21	374	11	12	13	14
121	3	4	13	14	248	5	11	13	21	375	11	12	15	16
122	3	4	15	16	249	5	11	14	22	376	11	12	17	18
123	3	4	17	18	250	5	11	15	19	377	11	12	19	20
124	3	4	19	20	251	5	11	16	20	378	11	12	21	22
125	3	4	21	22	252	5	11	17	23	379	11	12	23	24
126	3	4	23	24	253	5	11	18	24					
127	3	5	13	23	254	5	12	13	22					

## 8.2 程序代码

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.生成所有可行组合;
% 2.从组合中寻找没有任一元素（面试老师）相同的方案;
% 3.保存在FangAn[]中，数目保存在FangAnNum中
% author: 孙志勇于 2010-08
clear
clc
juzhen = combntns(1:24,4);
[hang,lie] = size(juzhen);
FangAn = [];

test = randperm(hang);
FangAn(1,:) = juzhen(1,:);
flag = zeros(hang,1);
flag(1) = 1;

NumFan0 = 1;
FangAnNum = 1;
TestFlag0 = 0;

```

---

```

for i = 1:hang
    tmp = juzhen(i,:);

    for j = 1:FangAnNum
        tmp2 = FangAn(j,:);
        TestFlag0 = no_one(tmp,tmp2);
        if TestFlag0
            break;
        end
    end

    if ~TestFlag0
        NumFan0 = NumFan0+1;
        FangAnNum = FangAnNum+1;
        flag(i) = 1;
        FangAn(FangAnNum,:) = tmp;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.从组合中寻找最多有两个元素（面试老师）相同的方案；
% 2.保存在FangAn[]中，数目保存在FangAnNum中
% author: 孙志勇 于 2010-08
NumFan2 = 0;
SameNum = 0;
FindFlag = 1;

for i = 1:hang

    if ~flag(i)
        tmp = juzhen(i,:);
        FindFlag = 1;

        for j = 1:FangAnNum
            tmp2 = FangAn(j,:);
            SameNum = FindSameNum(tmp,tmp2);
            if SameNum>2
                FindFlag = 0;
                break;
            end
        end

        if FindFlag

```

---

```

        NumFan2 = NumFan2+1;
        FangAnNum = FangAnNum+1;
        flag(i) = 1;
        FangAn(FangAnNum,:) = tmp;
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function SameNum = FindSameNum(input1,input2)
% 判断两个向量input1,input2之间元素相同的个数，保存于SameNum中
% 向量input1,input2必须同行同列，否则出错，此处不加考虑
% Author: 孙志勇于 2010-08
lie = 4;
SameNum = 0;

for i = 1:lie
    for j =1:lie
        if input1(i) == input2(j)
            SameNum = SameNum+1;
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function flag = no_one(input1,input2)
% 判断两个向量input1,input2之间是否元素全不相同
% 若全不相同，返回flag = 0; 只要找到一个相同，返回flag = 1
% 向量input1,input2必须同行同列，否则出错，此处不加考虑
% Author: 孙志勇于 2010-08
lie = 4;
flag = 0;

for i = 1:lie
    for j =1:lie
        if input1(i) == input2(j)
            flag = 1;
            break;
        end
    end
end
end

```

---

```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.将方案矩阵转换成 0-1的关联矩阵, 即
% [1 2 3 4] --> [1 1 1 1 0 0...0]形式
% 2.然后计算每个老师面试学生的数量TeacherNum
% Author: 孙志勇于 2010-08
newjuzhen = zeros(FangAnNum,24);
lie = 4;

for i = 1:FangAnNum
    for j = 1:lie
        newjuzhen(i,FangAn(i,j)) = 1;
    end
end

TeacherNum = sum(newjuzhen);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = RepeatStuNum(newjuzhen)
% 寻找任一对老师之间面试学生的重复数, 保存在S中
% author: 孙志勇于 2010-08

[hang,lie] = size(newjuzhen);
S = [];
cishu = 0;

for i = 1:lie
    tmp = newjuzhen(:,i);
    for j = i+1:lie
        tmp2 = newjuzhen(:,j);
        cishu = cishu+1;
        tmp3 = tmp2+tmp;
        tt = find(tmp3 == 2); %即有面试相同的学生的编号
        num = size(tt);
        S(cishu) = num(1); %每一对老师之间的重复学生数
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

```

% 1.生成所有可行组合;
% 2.按依次删除策略求出最终分配方案
% 3.保存在FangAn[]中,数目保存在FangAnNum中
% Author: 孙志勇于 2010-08

clear
clc
juzhen1 = combntns(1:12,2); %理科老师(编号13-24)两两组合
juzhen2 = combntns(13:24,2); %文科老师(编号1-12)两两组合

%求出全部组合
juzhen = [];
[TmpHang, TmpLie] = size(juzhen1);
for i = 1:TmpHang
    for j = 1:TmpHang
        count = (i-1)*TmpHang+j;
        juzhen(count,:) = [juzhen1(i,:),juzhen2(j,:)];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.本代码针对问题三,从组合中寻找最多有两个元素(面试老师)相同的方案;
% 2.保存在FangAn[]中,数目保存在FangAnNum中
% author: 孙志勇于 2010-08
[hang,lie] = size(juzhen);
SameNum = 0;
FangAnNum = 0;
FindFlag = 1;
flag = zeros(hang,1);

FangAn = [];

for i = 1:hang

    if ~flag(i)
        tmp = juzhen(i,:);
        FindFlag = 1;

        for j = 1:FangAnNum
            tmp2 = FangAn(j,:);
            SameNum = FindSameNum(tmp,tmp2);
            if SameNum>2

```

---

```

        FindFlag = 0;
        break;
    end
end

if FindFlag
    FangAnNum = FangAnNum+1;
    flag(i) = 1;
    FangAn(FangAnNum,:) = tmp;
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.本代码针对问题三，加进三元组进方案表中
% 2.保存在FangAn[]中，数目保存在FangAnNum中
% author: 孙志勇于 2010-08

Count = 0;
for i = 1:hang
    if flag(i) == 0
        Count = Count+1;
        juzhenRest(Count,:) = juzhen(i,:);
    end
end

ttmp = randperm(Count);

for i = 1:79
    FangAn(i+300,:) = juzhenRest(ttmp(i),:);
end
FangAnNum = FangAnNum+79;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.按依次删除策略求出最终分配方案
% 2.保存在FangAn[]中，数目保存在FangAnNum中
% Author: 孙志勇于 2010-08
[hang,lie] = size(juzhen);
DelFlag = ones(hang,1);

for i = 1:hang

```

---

```

if ~DelFlag(i)
    i = i+1;
    continue;
end

tmp = juzhen(i,:);

for j = i+1:hang
    if ~DelFlag(j)
        j = j+1;
        continue;
    end

    tmp2 = juzhen(j,:);
    SameNum = FindSameNum(tmp,tmp2);
    if SameNum>2
        DelFlag(j) = 0;
    end
end
end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Num0,Num1,Num2,Num3] = FindNum23(fangan)
% 计算任一学生面试组合中相同老师的个数
% Num1,Num2,Num3分别是有一位老师、有两位老师、有三位老师相同的面试组合对数
% Author: 孙志勇于 2010-08
[hang,lie] = size(fangan);
Num0 = 0;
Num1 = 0;
Num2 = 0;
Num3 = 0;
cishu = 0;
SameNum = 0;

for i = 1:hang
    tmp = fangan(i,:);
    for j = i+1:hang
        cishu = cishu+1;
        tmp2 = fangan(j,:);
        SameNum = FindSameNum(tmp,tmp2);

        if SameNum == 0

```

---

```

        Num0 = Num0+1;
    end

    if SameNum == 1
        Num1 = Num1+1;
    end

    if SameNum == 2
        Num2 = Num2+1;
    end

    if SameNum == 3
        Num3 = Num3+1;
    end

end

end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 模型与方案评估，分别计算几组参数sigma1-sigma4
% Author: 孙志勇 于 2010-08

%指标1: 每位老师的面试学生数量尽量均衡
MeanTeacherNum = mean(TeacherNum);
SumTeacherNum = sum(TeacherNum);
tmp = (TeacherNum-MeanTeacherNum).^2;
tmp1 = sum(tmp);
tmp1 = tmp1^0.5;
sigma1 = tmp1/SumTeacherNum

%指标2: 有两位老师与三位老师相同的学生对数在总学生对数中所占比例
CN2 = nchoosek(FangAnNum,2);
[Num0,Num1,Num2,Num3] = FindNum23(FangAn);
sigma211 = Num1/CN2
sigma21 = Num2/CN2
sigma22 = Num3/CN2

%指标4: 被任意两位老师面试的两个学生集合中出现相同学生的人数尽量少;
%S首先计算任意一对老师之间的重复学生数
S = RepeatStuNum(newjuzhen);

```



---

```
sigma41 = max(S)
sigma42 = mean(S)
```

其余代码见电子稿文件夹。