2021 第六届“数维杯”大学生
数学建模竞赛论文

题　　目　　**运动会优化比赛模式探索**

摘　要

本文主要探讨运动会比赛中各学院人数以及男女比例、体育特长生分配模型，通过设置公平指数来打分最终得出最优方案。

对于问题一，首先设置一个标准，此标准分为两部分，一部分是全校人数和组数的比得出平均人数，另一部分是全校整体的男女比例，根据这两个标准设置公平指数，根据贪心算法，我们在每个决策点作出在当时看来最佳的选择，即总是遵循我们设置的公平指数的打分规则，做出局部最优的选择，得到合适公平的组队方案。

对于问题二，将单个学院分两组，保证甲乙两组的男女生人数分别都接近男女生人数的平均数。由于每个学院的男女生基数不一样，针对每个学院提出一套专属的评分标准，并根据每个学院男女生分组后与平均人数的最大差值设置扣分标准，男女生扣分权重各占 50%，最终计算出每一个学院分组方案的公平指数。

对于问题三，当体育特长生跨院参加比赛时，首先通过 Excel 算出每个学院的学生总数减去体育特长生的人数和每个学院占整个学校所有人数减去体育特长生人数的比，根据此比例来具体分配 73 名体育特长生；当特长生不能参加比赛时，首先根据各学院男女人数比例减去对应的体育特长生人数，每个学院的男女生人数设置为矩阵每一行的元素，通过熵值法计算出每个学院的加权积分方案，即每个学院的团体得分算出之后需要乘以每个学院对应的加权积分。

对于问题四，通过计算机仿真模拟或理论推导来证明上述四种优化比赛模式，建立相应的变量关系，利用计算机程序模拟出不同组队模式下的情况，来比较各个方式的得分情况，从而选择最优的组队分配方式。

关键词：　贪心算法　公平指数　熵值法　仿真模拟

目录

一、问题重述

已知某校运动会的积分规则为：第一名得 9 分，第二名至第八名获得 7 至 1 分。各学院男生和女生累积得分最终构成团体得分。题中给出了某大学 20 个学院，104 个专业，共计 28523 名学生的分布情况数据，结合相应的数据完成下面的问题。

问题一：若在比赛中允许学院合并后共同参加比赛，需要提出一个分组数量不低于 12 个的、各组人员总数和男女性别比较为均衡的优化分配模型，并讨论分组方案的公平指数。

问题二：若对各个学院进行甲组和乙组分类进行比赛，提出最优的甲乙分组方案，并讨论这一分组方案的公平指数；

问题三：如果特长生可以跨学院参加比赛，提出尽量保障各学院比赛实力更为均衡的方案。如果特长生不参加比赛您能否提出一个对各学院相对公平的加权积分方案？

问题四：能否通过计算机仿真模拟或理论推导来证明上述四种优化比赛模式最优。

二、问题分析

2.2 问题 1 的分析

根据给出的数据，要对学院进行合并再进行组合，得到一个男女性别比较均衡的分配模型，首先设置一个标准，此标准分为两部分，一部分是全校人数和组数的比得出平均人数，另一部分是全校整体的男女比例，根据这两个标准设置公平指数，根据贪心算法，我们在每个决策点作出在当时看来最佳的选择，即总是遵循我们设置的公平指数的打分规则，做出局部最优的选择，得到合适公平的组队方案。

2.2 问题 2 的分析

针对单个学院分两组，保证甲乙两组的男女生人数分别都接近男女生人数的平均数。由于每个学院的男女生基数不一样，针对每个学院提出一套专属的评分标准，并根据每个学院男女生分组后与平均人数的最大差值设置扣分标准，男女生扣分权重各占50%，最终计算出每一个学院分组方案的公平指数。

2.3 问题 3 的分析

当体育特长生跨院参加比赛时，首先通过 Excel 算出每个学院的学生总数减去体育特长生的人数和每个学院占整个学校所有人数减去体育特长生人数的比，根据此比例来

具体分配 73 名体育特长生；当特长生不能参加比赛时，首先根据各学院男女人数比例减去对应的体育特长生人数，每个学院的男女生人数设置为矩阵每一行的元素，通过熵值法计算出每个学院的加权积分方案，即每个学院的团体得分算出之后需要乘以每个学院对应的加权积分。

2.4 问题 4 的分析

针对此问题，采用程序仿真模拟的方式进行验证。假设参加运动会的有 n 个学院，学校编号为 1……n，比赛分成 m 个男子项目，和 w 个女子项目。项目编号为男子 1……m，女子 m+1……m+w。不同的项目取前八名积分，且前八名的积分分别为：9、7、6、5、4、3、2、1（m<=20，n<=20）等，通过程序模拟来比较四种模式的优劣。

三、模型假设

在本文对运动会组队模式优化的问题上，要对问题一到四解答，主要假设了关于人数样本的稳定性较高，要建立公平指数对组队分配的问题进行公平性讨论，假设公平指标，满足全校人数和组数的比得出平均人数，及男女比例两个方面，各占一定的比值，得出的评分满足模型的假设需求。

四、定义与符号说明

符号定义	符号说明
H	组队模式下的方案得分情况
P_i	组队模式下各分组的得分情况
X	分组数
Y	分组总数
y_1	男生人数
y_2	女生人数
n	循环次数

五、模型的建立与求解

5.1 问题 1 的模型建立与求解

若在比赛中允许学院合并后共同参加比赛，需要提出一个分组数量不低于 12 个的、各组人员总数和男女性别比较为均衡的优化分配模型，并讨论分组方案的公平指数[1]。

对于问题一，针对题目给出的所有数据，首先将每个学院各个班的人数加起来得到总人数，再将各个学院的人数加起来得到全校的总人数，最终通过 Excel 算出全校的男女比例，结果如下表所示：

全校总人数及男女比例表

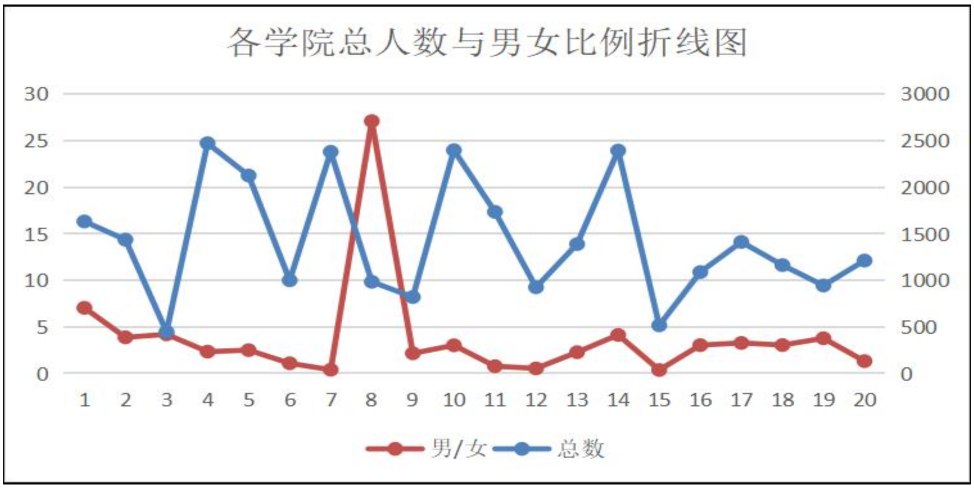
学院	男生	女生	总人数	男/女
A	1428	203	1631	7.034482759
B	1140	295	1435	3.86440678
C	360	86	446	4.186046512
D	1729	742	2471	2.330188679
E	1515	609	2124	2.487684729
F	518	481	999	1.076923077
G	662	1717	2379	0.385556203
H	948	35	983	27.08571429
I	559	260	819	2.15
J	1798	598	2396	3.006688963
K	752	982	1734	0.765784114
L	319	605	924	0.527272727
M	963	425	1388	2.265882353
N	1925	467	2392	4.122055675
O	135	380	515	0.355263158
P	817	270	1087	3.025925926
Q	1080	331	1411	3.262839879
R	874	288	1162	3.034722222
S	745	198	943	3.762626263
T	687	524	1211	1.311068702

根据题目的要求，在比赛中允许学院合并后共同参加比赛，但是分组数量不能低于 12 个、各组人员总数和男女性别比较为均衡。

首先设置一个标准，此标准分为两部分，一部分是全校人数和组数的比得出平均人数，另一部分是全校整体的男女比例，根据这两个标准设置公平指数，各占 50%，通过 Excel 算出来的不同组数的平均人数和男女比例如下表所示：

组数	平均人数
12	2370.833333
13	2188.461538
14	2032.142857
15	1896.666667
16	1778.125
17	1673.529412
18	1580.555556
19	1497.368421
全校男女比例	2

根据总人数和男女比例两列数据画出的折线图如下图所示，可以更直观的看出各学院之间人数的差距和男女比例的均衡状况：



根据这两组数据来给每个学院打分，这里将每个学院的总人数与对应组数算出的平均人数算差值的平均值，得出与平均人数之间相差多少，同时也算出每个学院与全校男女比例的差值的绝对值，根据所有的差值情况，设置每每相差 50 人扣一分，没相差 100 人就扣两分，同时设置男女比例每相差 1 则扣两分。

通过 Excel 对数据进行处理，要得到一个组数不低于 12 组的方案，且要对男女的性别比例进行调整考虑，做出最优的选择，通过建立的评判机制来确定是否公平。

将所有学员的分数通过 Excel 公式打出并进行排名，结果如下表所示，以分成 12 组为例：

学院	人数差	人数得分	男女比例差	男女比例差得分	最终得分	排名
A	739	35.22	5.03448275	39.93103448	75.1510344	11
B	935	31.3	1.86440678	46.27118644	77.5711864	9
C	1924	11.52	2.18604651	45.62790698	57.1479069	19
D	101	47.98	0.33018867	49.33962264	97.3196226	2
E	246	45.08	0.48768472	49.02463054	94.1046305	5
F	1371	22.58	0.92307692	48.15384615	70.7338461	14
G	9	49.82	1.61444379	46.77111241	96.5911124	3
H	1387	22.26	25.0857142	-0.171428571	22.0885714	20
I	1551	18.98	0.15	49.7	68.68	15
J	26	49.48	1.00668896	47.98662207	97.4666220	1
K	636	37.28	1.23421588	47.53156823	84.8115682	6
L	1446	21.08	1.47272727	47.05454545	68.1345454	16
M	982	30.36	0.26588235	49.46823529	79.8282352	7
N	22	49.56	2.12205567	45.75588865	95.3158886	4
O	1855	12.9	1.64473684	46.71052632	59.6105263	18
P	1283	24.34	1.02592592	47.94814815	72.2881481	13
Q	959	30.82	1.26283987	47.47432024	78.2943202	8
R	1208	25.84	1.03472222	47.93055556	73.7705555	12
S	1427	21.46	1.76262626	46.47474747	67.9347474	17
T	1159	26.82	0.68893129	48.6221374	75.4421374	10

其中人数得分的 Excel 公式为：=50-人数差/50，男女比例差的 Excel 公式为：=男女比例差*2，最终得分的 Excel 公式为：=人数得分+男女比例差得分。

（具体的 12--19 组数得分及排名情况详见附件）

$$H=\sum_{i=0}^{n}\frac{1}{X}\cdot P_i$$

$$P=0.02\times\left(Y-\frac{28523}{100x}\right)+\frac{2y_1}{y_2}$$

此题是求解最优化问题，在分配时需要经历一系列的步骤，在每个步骤中都面临多项选择，根据贪心算法[2]，我们在每个决策点作出在当时看来最佳的选择，即总是遵循我们设置的公平指数的打分规则，做出局部最优的选择，以推导出全局最优解，追求局

部最优的同时希望达到整体最优的效果。将整个学院分组的问题可细分为多个子结构问题，求出每个子问题的最优解，子问题的局部最优解与贪心选择组合，即可得到原问题的全局最优解。

贪心规律	
1	某个学院如果和另一个有相对需求的学院配对分数并不能提高，则该学院也一定不能提高需求因子更大的其他学院。
2	某个学院可以和分数更少的另一个学院配对且分数很高，则没必要用相对需求指数更大的学院来配对，因为可以保留更合适的学院满足需求因子更大的学院。
3	学院的分数更高则其更容易满足，故优先从分数最低的学院开始尝试，可以的到正确的结果。（因为追求更多的学院分数被提高，所以用一个学院满足分数较低或较高的学院都是一样的）

算法设计	
1	对于分组组数为 12 组，筛选出最终得分最低的前 16 个学院，其中最终得分最低的前八位我们首要考虑将其进行组队来提高其分数，选择剩下分数次低的八个学院进行两两组队。若对于分组组数为 13 组，则筛选最终的分最低的前 14 个学院，得分最低的前 7 个学院和得分次低的后 7 个学院两两组队；对于 14 组则筛选 12 个，前 6 和次 6 两两配对，以此类推直至 19 组。
2	将每个不同分组组数的具体分组列出来并进行打分，根据打分列出所有分数组合的情况并选择最优的分数组合情况。

针对算法设计的步骤 1，通过 Python 可写出代码求出每个组数对应的得分数据，以 12 组为例，结果如下表所示：

学院	R	A	T	B
H	88.21826625	86.51100346	96.44902655	95.64851927
C	77.68879257	85.76629758	85.69575221	86.9664503
O	85.05021672	93.83141869	84.93858407	90.9346856
S	87.02817337	86.8750173	95.50964602	96.19294118
L	89.6574613	90.6183391	94.39513274	98.10454361
I	86.79585139	91.86712803	92.56725664	95.50555781
F	91.95622291	90.01107266	95.60973451	97.56787018
P	87.14037152	80.9015917	97.40070796	91.49955375

学院	Q	M	K	E
H	99.41316239	98.66029197	91.55077239	85.11437998
C	89.56051282	88.05956204	94.64429255	95.93174061
O	88.87641026	89.06	94.70371839	92.49030717
S	99.65222222	97.71635036	92.43689679	85.77103527
L	98.28931624	98.75532847	92.15250854	85.62316268
I	96.45213675	96.29620438	94.42065619	88.00302617
F	97.91367521	99.3680292	90.47615858	83.82964733
P	96.45709402	95.09124088	90.16796992	81.87397042

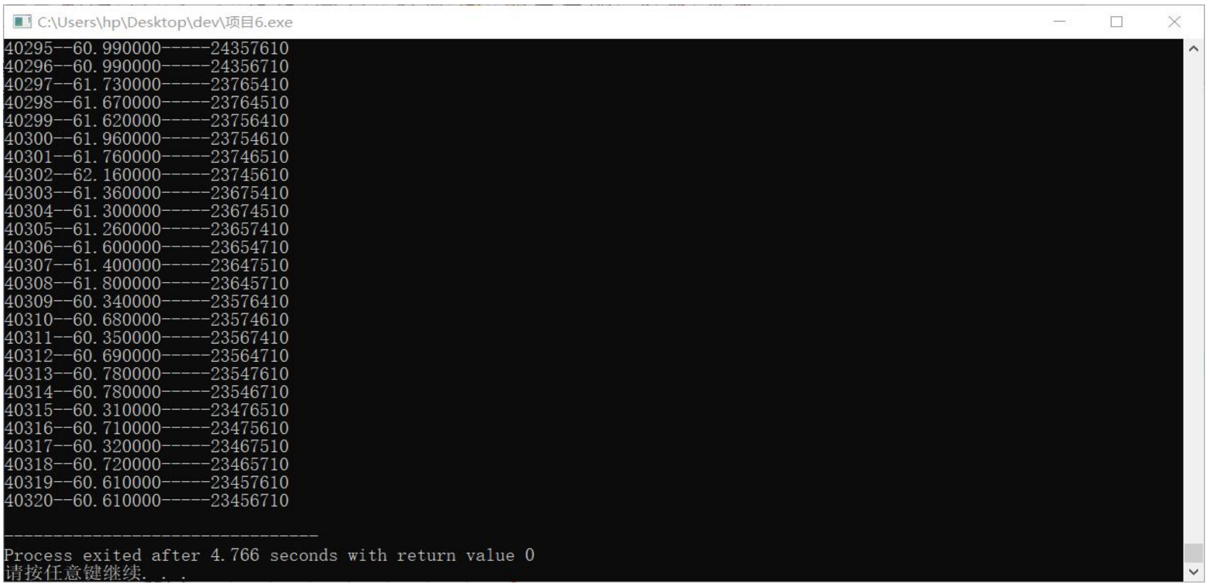
Python 代码如下部份所示：

```
x1=[]#x1 为对应组数的最低前 n 位学院的男生人数，需自行输入进列表
y1=[]#y1 为对应组数的最低前 n 位学院的女生人数，需自行输入进列表
x2=[]#x2 为对应组数的最低次 n 位学院的男生人数，需自行输入进列表
y2=[]#y2 为对应组数的最低次 n 位学院的女生人数，需自行输入进列表
renshu1=[]#此为对应组数的最低前 n 位学院的总人数
renshu2=[]#此为对应组数的最低次 n 位学院的总人数
```

```
nansheng=[]#此为对应组数的男生总人数
nvsheng=[]#此为对应组数的女生总人数
zongrenshu=[]
bili=[]
for i in range(len(x1)):
    a=x1[i]+y1[i]
    renshu1.append(a)
    b=x2[i]+y2[i]
    renshu2.append(b)
for j in renshu1:
    for m in renshu2:
        c=j+m
        zongrenshu.append(c)
n=1
for k in range(0,len(zongrenshu),n):
    zongrenshus=zongrenshu[k:k+n]
    print(zongrenshus)
for i in range(len(x1)):
    d=x1[i]+x2[i]
    nansheng.append(d)
    e=y1[i]+y2[i]
    nvsheng.append(e)
for j in nansheng:
    for m in nvsheng:
        f=j/m
        bili.append(f)
for k in range(0,len(bili),n):
    bilis=bili[k:k+n]
   print(bilis)
```

再根据已选择的学院不能进行二次甚至多次配对，可大大减少最终成绩方案的数量，且经过去重，通过 C 语言代码可求出所有方案。

运行结果部分如下图所示：



其中结果的第一列为所有方案的个数，第二列为每个方案的分数，第三列为每个方案的序号，例如有 8x8 组得分数据则从 1-8 分别编序号为 0-7，将运行出的所有得分复制到 Excel 表格中筛选出最高分，根据最高分对应的序号可找出具体的每个学院该如何配对。最终组数为 12 组时选择出的最佳方案如下图所示：

得分		E 0	K 1	M 2	Q 3	B 4	T 5	A 6	R 7
P	z	7.35	7.21	8.04	7.97	8.28	8.22	7.63	7.09
F	x	6.47	7.15	7.14	7.25	7.46	7.34	7.89	7.99
I	c	7.09	7.82	7.08	7.58	7.41	7.42	7.89	7.71
L	v	7.25	7.24	7.96	8.02	8.30	8.14	7.70	7.15
S	b	7.47	7.55	7.87	8.18	8.19	8.23	7.68	7.14
O	n	7.23	7.66	7.71	7.96	8.04	8.02	7.87	7.33
C	l	7.66	7.50	7.97	8.13	8.16	8.28	7.54	6.99
H	m	7.26	6.74	8.12	7.62	8.04	7.92	7.51	6.82

其中做出标记的表格为最佳选择方案，即选择分组组数为 12 组：将 P 学院与 M 学院组合，F 学院与 R 学院组合，I 学院与 K 学院组合，L 学院与 B 学院组合，S 学院与 Q 学院组合，O 学院与 A 学院组合，C 学院与 T 学院组合，H 学院与 E 学院组合，最佳方案的最终得分为 95.96593786 分。

5.2 问题 2 的模型建立与求解

此小问与第一小问非常类似，只是从整个学校变成针对每一个学院，而每一个学院的班数不同，每个班的男女生人数也不同，这里需要先算出每个学院的男女生总人数，又因为题目要求对各个学院进行甲组和乙组分类，所以只需要分为两组即可，再通过 Excel 算出每个学院的男女生平均人数，以此作为分组的评判标准。

假设学院拥有的班级数为 z	
对于拥有双数班级数的学院，其分组的排列组合最多需要计算	Z/2
对于拥有单数班级数的学院，其分组的排列组合最多需要计算	(Z-1)/2

根据此排列组合方案，对每个学院的男女生人数分别进行降序排序，并计算出各个学院在分配组合时男女生人数与平均人数的最大差值 c，以此来设定扣分标准，即每与平均人数相差一个人需要扣多少分，设置男女生扣分权重各占 50%，最终扣分分数 f=50/（c 向上取一位整数），也就是如果最大差值 c 为 103，即取 110。

结果如下表所示：

学院	男生平均人数	女生平均人数	男生每相差 1 人扣分	女生每相差 1 人扣分
A	714	101.5	0.172	0.625
B	570	147.5	-	-
C	180	43	0.417	2.500
D	864.5	371	0.167	0.294
E	757.5	304.5	0.085	0.208
F	259	240.5	0.385	0.455
G	331	858.5	0.417	0.227
H	474	17.5	0.167	5.000
I	279.5	130	0.217	0.500
J	899	299	0.109	0.294
K	376	491	0.313	0.294
L	159.5	302.5	-	-
M	481.5	212.5	0.200	0.385
N	962.5	233.5	0.143	0.714
O	67.5	190	-	-
P	408.5	135	0.714	2.500
Q	540	165.5	0.147	0.385
R	437	144	0.200	0.625
S	372.5	99	0.714	2.500
T	343.5	262	0.455	0.500

(其中 B 学院、L 学院、O 学院只有两个班，所以不用进行分组，可不用计算数据)

通过 C 语言写出排列组合的代码，不需要去重，最终相同方案虽然顺序不同，但计算出的分数相同可通过 Excel 表格筛选出来，这里以 A 学院为例，具体部分代码如下所示：

```
#include<stdio.h>
int main(){
int z,x,c,v,b;
int man=714,woman=101,l=0;
double a[9]={135,145,41,170,112,192,143,250,240},
s[9]={5,10,7,21,33,34,4,42,47},
h,j,k,p=0.127,q=0.625;
for(z=8;z>=0;z--)
for(x=8;x>=0;x--)
for(c=8;c>=0;c--)
for(v=8;v>=0;v--)
{
if(z!=x)
if(z!=c)
if(z!=v)
if(x!=c)
if(x!=v)
if(c!=v)
{h=p*(man-(a[z]+a[x]+a[c]+a[v]));
j=q*(woman-(s[z]+s[x]+s[c]+s[v]));
if(h<0)
......(运行结果见附件)
```

运行结果中的第一列数据表示所有方案的个数，第二列是具体方案的分数，第三列是序号，其中序号 0-8 对应的是 A 学院中班级的 1-9 班，第四列是男生相差人数所扣的分数，第五列是女生相差人数所扣的分数。

将运行出的所有数据复制到 Excel 表格中，并筛选出最高分，因为代码没有去重，所以筛选出的最高分会出现多条方案，但每条方案的其实是一样的，只是顺序不一样，

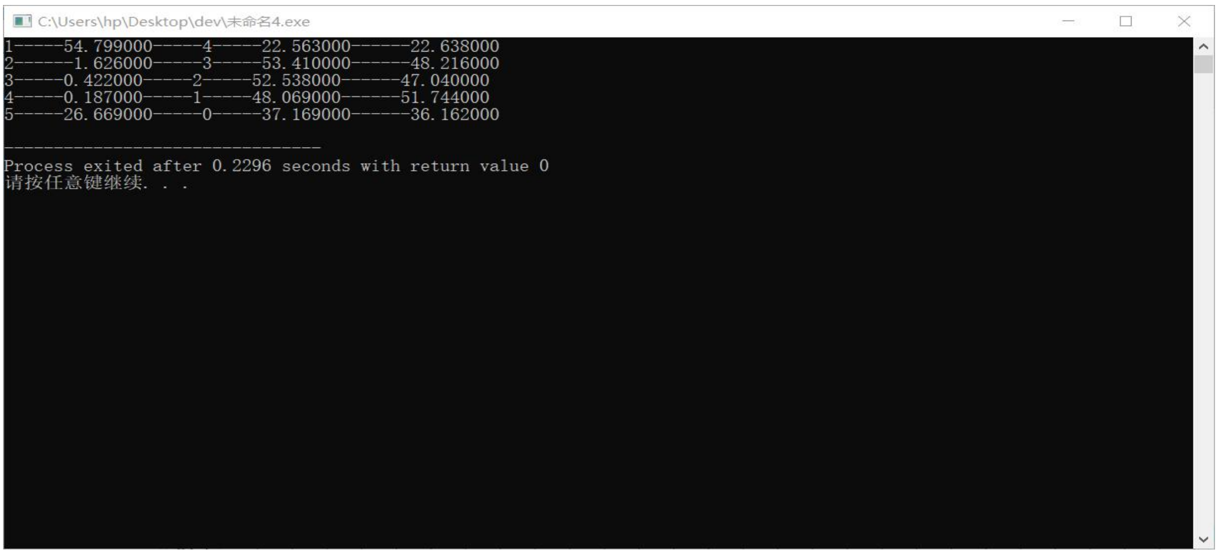
所以代码不去重并不会影响最终的结果，具体的 A 学院所有方案详情见附件，A 学院通过筛选后的最高分数据及方案如下表所示：

序号	分数	序号	男生扣分	女生扣分
2706	96.621	8510	0.254	-3.125
2727	96.621	8150	0.254	-3.125
2820	96.621	5810	0.254	-3.125
2851	96.621	5180	0.254	-3.125
2985	96.621	1850	0.254	-3.125
3001	96.621	1580	0.254	-3.125

对 A 学院	
甲组	A1 班+A2 班+A6 班+A9 班
乙组	A3 班+A4 班+A5 班+A7 班+A8 班

通过计算其他剩下的所有学院（除了 B 学院、L 学院、O 学院），最终所有学院的分组方案，其中筛选出 J 学院有特殊班，即某个班人数与其他班人数相差非常大，最终打分非常低，所以需要打赌考虑此学院，由于 J 学院共有 6 个班，其中 J5 班男生人数 827 人，女生人数 288 人，都远超另外五个班的人数，这里分别采用 1x5 和 2x4 和 3x3 的排列组合方式带入 C 语言代码中运行，最终得出 J 学院的运行结果如下图所示：

1x5 运行结果：



(其余运行结果见附件)

通过 Excel 筛选出 J 学院的最高分的所有方案及其数据如下表所示：

1x5 筛选数据：

序号	分数	序号	男生扣分	女生扣分
1	54.799	4	22.563	22.638

2x4 筛选数据：

序号	分数	序号	男生扣分	女生扣分
1	83.094	43	6.322	-10.584
5	83.094	34	6.322	-10.584

3x3 筛选数据：

序号	分数	序号	男生扣分	女生扣分
16	60.888	510	21.364	-17.748
20	60.888	501	21.364	-17.748
26	60.888	432	21.364	-17.748
30	60.888	423	21.364	-17.748
46	60.888	342	21.364	-17.748
50	60.888	324	21.364	-17.748
66	60.888	243	21.364	-17.748
70	60.888	234	21.364	-17.748
84	60.888	150	21.364	-17.748
97	60.888	105	21.364	-17.748
104	60.888	51	21.364	-17.748
117	60.888	15	21.364	-17.748

J 学院的最佳分组方案	
甲组	J5 班+J5 班
乙组	J1 班 J2 班+J3 班+J4 班

最终所有学院的分组结果如下表所示：

学院	甲组	乙组	得分
A	2+6+9+1	3+4+5+7+8	96.621
B	1	2	-
C	1+4	2+3+5	86.247
D	3+4+7+8	1+2+6+5	93.6
E	1+5+7	2+3+4+6	92.621
F	1+5	2+3+4	91.3725
G	2+7+4+1+3	5+6+8+9+10+11	98.485

H	1+5	2+3+4	84.98
I	2	1+3	92.6365
J	1+2+3+6	4+5	83.094
K	1+5	2+3+4	65.298
L	1	2	-
M	1+2	3+4+5	88.0075
N	1+4+6	2+3+5+7	94.855
O	1	2	-
P	1+2	3+4	48.59
Q	1+2	3+4+5	95.7545
R	1+2	3	60.825
S	3+4	1+2	48.59
T	3+4+6	1+2+5	95.725

5.3 问题 3 的模型建立与求解

由题目已知条件：运动会的积分规则为：第一名得 9 分，第二名至第八名获得 7 至 1 分；各学院男生和女生累积得分最终构成团体得分可得出，体育特长生的分配会直接影响到各学院的得分以及排名情况，为了使各学院实力更为均衡，比赛更为公平，所以允许体育特长生跨学院参加比赛。

根据实际情况，假设每个体育特长生最终都获得第一名，也就是得 9 分，根据每个学院的人数基数来分配所有的体育特长生，由题目给出的所有学院的数据可通过 Excel 算出体育特长生共有 73 位，每个学院的学生总数减去体育特长生的人数的数据表如下表所示：

A	B	C	D	E	F	G	H	I	J
1626	1435	446	2464	2117	999	2363	983	819	2386

K	L	M	N	O	P	Q	R	S	T
1728	919	1388	2391	515	1087	1404	1157	943	1207

通过 Excel 计算出每个学院占整个学校所有人数减去体育特长生人数的比，比值数据如下表所示：

A	B	C	D	E	F	G	H	I	J
0.05 7152	0.0504 3936	0.015 67662	0.086 60808	0.0744 1124	0.0351 1423	0.083 05799	0.034 55184	0.02878 7346	0.0838664 32

K	L	M	N	O	P	Q	R	S	T
0.06 07	0.0323 0228	0.0487 8734	0.084 04217	0.0181 0193	0.038 20738	0.049 34973	0.040 66783	0.03314 587	0.0424253 0

按照此比例给各学院分配体育特长生，计算出的数据如下表所示：

A	B	C	D	E	F	G	H	I	J
4.172161	3.68207381	1.14439367	6.32239015	5.43202109	2.56333919	6.06323374	2.52228471	2.10147627	6.122249561

K	L	M	N	O	P	Q	R	S	T
4.43388	2.35806678	3.56147627	6.13507908	1.32144112	2.78913884	3.60253075	2.96875219	2.419648506	3.097047452

由于人数分配只能取整数，首先筛选出每个数据小数点后面大于 5 的数据，此类数据全部向大的方向去整数，计算得出已经分配 72 位体育特长生，还剩一位未被分配，所以对剩下的小数点后面小于 5 的数据进行排序，找出其中小数点后数据最大的一个值，对它进行向大的方向取整，最终找出的是 K 学院的 4.433884007 向上取整得 5，所有结果如下表所示（体育特长生跨院参加比赛后个学院分配的特长生人数表）：

A	B	C	D	E	F	G	H	I	J
4	4	1	6	5	3	6	3	2	6

K	L	M	N	O	P	Q	R	S	T
5	2	4	6	1	3	4	3	2	3

当体育特长生不参加比赛时，讨论一个对各学院相对公平的加权积分方案，依旧根据题目给出的得分条件：第一名得 9 分，第二名至第八名获得 7 至 1 分；各学院男生和女生累积得分最终构成团体得分可考虑给人数和男女比例设置加权积分，采用熵值法计算权重，首先根据各学院男女人数比例减去对应的体育特长生人数，结果如下表所示：

A		B		C		D		E	
男	女	男	女	男	女	男	女	男	女
1424	202	1140	295	360	86	1725	739	1511	606

F		G		H		I		J	
男	女	男	女	男	女	男	女	男	女
518	481	657	1706	948	35	559	260	1791	595

K		L		M		N		O	
男	女	男	女	男	女	男	女	男	女
750	978	317	602	963	425	1924	467	135	380

P		Q		R		S		T	
男	女	男	女	男	女	男	女	男	女
817	270	1076	328	870	287	745	198	685	522

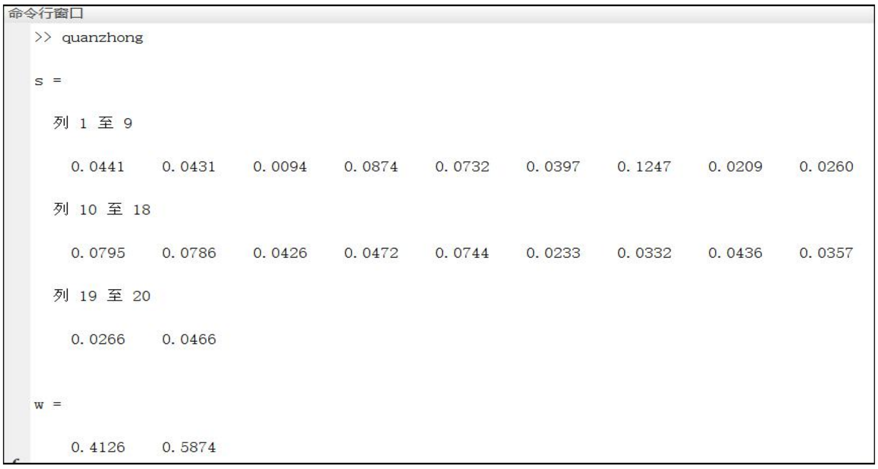
将每个学院的男女生人数作为一列矩阵，整个学院为一个大矩阵，通过熵值法输入矩阵带入 MATLAB 代码中进行计算加权指数，熵值法[3]是一种用来判断一个事件的随机性及无序程度或者某个指标的离散程度的方法，根据指标的特性，我们可以用熵值来判断某个指标的离散程度：指标熵值越小，离散程度越大，该指标对综合评价的影响（即权重）也越大。

最终的 MATLAB 代码如下所示：

```
clear
x=[1424,202;1140,295;360,86;1725,739;1511,606;
518,481;657,1706;948,35;559,260;1791,595;
750,978;317,602;963,425;1924,467;135,380;
817,270;1076,328;870,287;745,198;685,522];
[s,w]=shang(x)
function [s,w]=shang(x)
[n,m]=size(x);
[X,ps]=mapminmax(x');
ps.ymin=0.002;
ps.ymax=0.996;
ps.yrange=ps.ymax-ps.ymin;
X=mapminmax(x',ps);
X=X';
for i=1:n
    for j=1:m
        p(i,j)=X(i,j)/sum(X(:,j));
    end
end
k=1/log(n);
for j=1:m
    e(j)=-k*sum(p(:,j).*log(p(:,j)));
end
d=ones(1,m)-e;  % 计算信息熵冗余度
```

```
w=d./sum(d);      % 求权值 w
s=w*p';           % 求综合得分
end
```

代码的运行结果如下图所示：



```
命令行窗口
>> quanzhong

s =

  列 1 至 9

    0.0441    0.0431    0.0094    0.0874    0.0732    0.0397    0.1247    0.0209    0.0260

  列 10 至 18

    0.0795    0.0786    0.0426    0.0472    0.0744    0.0233    0.0332    0.0436    0.0357

  列 19 至 20

    0.0266    0.0466


w =

    0.4126    0.5874
```

列 1 到列 20 分别表示学院 A 到学院 T,所以最终每个学院的加权积分方案如下表所示：

A	B	C	D	E	F	G	H	I	J
0.0441	0.0431	0.0094	0.0874	0.073	0.039	0.124	0.020	0.026	0.079

K	L	M	N	O	P	Q	R	S	T
0.0786	0.0426	0.0472	0.0744	0.02	0.033	0.043	0.035	0.026	0.046

即每个学院的团体得分算出之后需要乘以每个学院对应的加权积分。

5.4 问题 4 的模型建立与求解

在问题四中，需要通过计算机仿真模拟或理论推导来证明上面提出的四种组队模式哪一个组队模式最优，针对此问题，采用程序仿真模拟[4]的方式进行验证。假设参加运动会的有 n 个学院，学校编号为 1……n，比赛分成 m 个男子项目，和 w 个女子项目。项目编号为男子 1……m，女子 m+1……m+w。不同的项目取前八名积分，且前八名的积分分别为：9、7、6、5、4、3、2、1（m<=20，n<=20）。

功能要求：（1）可以输入各个项目的前八名的成绩；（2）能统计各学院的总分并排序；（3）可以按学院编号、学院总分、男女团体排序输出；（4）可以按学院编号查

询学院某个项目的情况；可以按项目编号查询取得前八名的学院。（5）可以查找汇总某名选手参加的项目和获取的名次和积分。（其他要求见附件）

对仿真模拟的部分程序见下：

```c
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include <malloc.h>
#include <stdlib.h>

#define N 10
typedef int ElemType;

typedef struct Node
{
    ElemType score;    //分数
    ElemType rank;     //排名
    ElemType sex;      //性别
    char name[20];     //姓名
    char college[20];  //学院
    char item[20];     //项目
    struct Node *next;
} LinkNode, *LinkList;

void Init(LinkList &Head);
int  GetScore(int rank);
void Input(LinkList &Head, int &CollegeNum, int &ItemNum, char Colleges[][20], char Items[][20]);
void Create(LinkList &Head, int score, int rank, int sex, char name[], char college[], char item[]);
void Show(LinkList Head);
void Save(LinkList Head, char Colleges[][20], int CollegeNum, char Items[][20], int ItemNum);
void Read(LinkList Head, char Colleges[][20], int &CollegeNum, char Items[][20], int &ItemNum);
void Collegescore(LinkList Head, char Colleges[][20], int CollegeNum);
void SexScore(LinkList Head, char Colleges[][20], int CollegeNum);
void QueryCollege(LinkList Head);
void QueryName(LinkList Head);
void QueryItem(LinkList Head);
void Insert(LinkList &Head, int score, int rank, int sex, char name[], char college[], char item[]);
void Print(Node *p);
void Free(LinkList &Head);
```

```c
    }

    Node *q;
    q = man_score->next;
    printf("学院男生团体的前八名为\n");
    while (q != NULL)
    {
        printf("%s男生团体得分为:   %d\n", q->college, q->score);
        q = q->next;
    }
    q = NULL;
    q = woman_score->next;
    printf("学院女生团体的前八名为\n");
    while (q != NULL)
    {
        printf("%s女生团体得分为:   %d\n", q->college, q->score);
        q = q->next;
    }

    Free(man_score);
    Free(woman_score);
}
//查询某学院某个项目获奖情况
void QueryCollege(LinkList Head)
{

    char item[20], college[20];

    printf("请输入您要查询的学院名:\n");
    scanf("%s", &college);

    printf("请输入您要查询的项目名:\n");
    scanf("%s", &item);

    Node *p = Head->next;

    while (p != NULL)
    {
        if (strcmp(college, p->college) == 0 && strcmp(item, p->item) == 0)
        {
            Print(p);
        }
        p = p->next;
```

得出四种优化模式的特点。通过统计汇总，得出关于四种组队模式下的评分：

模式	优劣程度
模式 1	84.98
模式 2	92.6365
模式 3	83.094
模式 4	65.298

六、模型的评价及优化

6.2 模型的优点

1.本文对运动会比赛模式优化进行了探讨，对分组进行优化分配的问题得到了很好的解决，可以较好的推广到实际的运动会组队分配问题上。

2.对数据进行分类处理，从学院，专业，性别等特征上分别考虑，在保证模型准确度的情况下，减少了模型的复杂程度。

3.通过算法设计，减少对总体数据的处理难度，为得到最优的分组方案，在算法处理后，通过计算机模拟仿真正确证明了优化出的模型。

6.3 模型的缺点

1.对公平指数的评价体系的判定，有一定的局限性，适用的范围容易受到其他变量的影响，要考虑分配的综合性因素。

2.模型中的参数，决定了其推广的相对程度，没有进行简化处理，在对具体的使用过程中，需要经过专业的数据处理。

3.在制定的公平性中，其随机因素较多，得到的模型不能准确的表达出来，因模型的复杂因素较多，不能进行全面的考虑，会造成一定的与实际不符之处。

6.4 模型的推广

本文在处理运动会组队分配优化的问题上，在理想状态的环境下，通过建立的公平指数评价体系，对优化比赛模式的问题得到较好的解决，具有一定的合理性，模型是建立在对样本数据进行充分的挖掘的基础之上的，通过数据处理，建立合适的量化指标，建立准确的数学模型，根据问题的解决思路，分析其相关规律层次渐进，方便对问题处理，使用相关数学软件，同时进行优化，得出相应结果。

本文提出了关于运动会组队分配优化的数学模型，解决了比赛模式优化的问题，可以用于其他不确定性决策问题中，可以较好的推广到实际的关于人数组队分配的问题上，通用性较高，效果显著。

参考文献

[1]王晓,杨昔阳,董克强.一种新的席位公平分配方法[J].北京师范大学学报(自然科学版),2011,47(05):452-456.

[2]杨书影.贪心算法及其实例探究[J].安徽电子信息职业技术学院学报,2013,12(03):78-80.

[3]张红梅,贺颖,聂筑平.粗糙集与熵值法在综合评价中的运用述评[J].科技信息,2012(36):516-517.

[4]吴婷.基于计算机仿真模拟实验的应用分析[J].电子技术与软件工程,2013(19):208.

附录

问题一：

```
#include<stdio.h>
int main(){
//int a[]={},b[]={};
//int z=6,x=6,c=6,v=6,b=6,n=6,m=6,l=6;
int z,x,c,v,b;

int man=714,woman=101,l=0;
double a[9]={135,145,41,170,112,192,143,250,240},
s[9]={5,10,7,21,33,34,4,42,47},
h,j,k,p=0.127,q=0.625;
for(z=8;z>=0;z--)
for(x=8;x>=0;x--)
for(c=8;c>=0;c--)
for(v=8;v>=0;v--)
{
if(z!=x)
if(z!=c)
if(z!=v)
if(x!=c)
if(x!=v)

if(c!=v)
{h=p*(man-(a[z]+a[x]+a[c]+a[v]));
j=q*(woman-(s[z]+s[x]+s[c]+s[v]));
if(h<0)
{h=-h;}
if(j<0)
{j=-j;}
k=100-h-j;
l++;
printf("%d-----%lf-----%d%d%d%d%d-----%lf------%lf\n",l,k,z,x,c,v,b,h,j)
;
}}

return 0;
}
```

x1=[]#x1 为对应组数的最低前 n 位学院的男生人数，需自行输入进列表

y1=[]#y1 为对应组数的最低前 n 位学院的女生人数，需自行输入进列表

x2=[]#x2 为对应组数的最低次 n 位学院的男生人数，需自行输入进列表

```
y2=[]#y2 为对应组数的最低次 n 位学院的女生人数，需自行输入进列表
renshu1=[]#此为对应组数的最低前 n 位学院的总人数
renshu2=[]#此为对应组数的最低次 n 位学院的总人数
nansheng=[]#此为对应组数的男生总人数
nvsheng=[]#此为对应组数的女生总人数
zongrenshu=[]
bili=[]
for i in range(len(x1)):
    a=x1[i]+y1[i]
    renshu1.append(a)
    b=x2[i]+y2[i]
    renshu2.append(b)
for j in renshu1:
    for m in renshu2:
        c=j+m
        zongrenshu.append(c)
n=1
for k in range(0,len(zongrenshu),n):
    zongrenshus=zongrenshu[k:k+n]
    print(zongrenshus)
for i in range(len(x1)):
    d=x1[i]+x2[i]
    nansheng.append(d)
    e=y1[i]+y2[i]
    nvsheng.append(e)
for j in nansheng:
    for m in nvsheng:
        f=j/m
        bili.append(f)
for k in range(0,len(bili),n):
    bilis=bili[k:k+n]
```

```
print(bilis)
52,104,54,73,66,43,46,22,68,83,51
192,201,151,148,121,177,163,76,155,188,145



#include<stdio.h>
int main(){
//int a[]={},b[]={};
//int z=6,x=6,c=6,v=6,b=6,n=6,m=6,l=6;
int z,x,c,v,b;
//double  q[8]={7.35 ,7.21 ,8.04 ,7.97 ,8.28 ,8.22 ,7.63 ,7.09    },
//w[8]={6.47 ,7.15 ,7.14 ,7.25 ,7.46 ,7.34 ,7.89 ,7.99  },
//e[8]={7.09 ,7.82 ,7.08 ,7.58 ,7.41 ,7.42 ,7.89 ,7.71 },
//r[8]={7.25 ,7.24 ,7.96 ,8.02 ,8.30 ,8.14 ,7.70 ,7.15 },
//t[8]={7.47 ,7.55 ,7.87 ,8.18 ,8.19 ,8.23 ,7.68 ,7.14  },
//y[8]={7.23 ,7.66 ,7.71 ,7.96 ,8.04 ,8.02 ,7.87 ,7.33  },
//u[8]={7.66 ,7.50 ,7.97 ,8.13 ,8.16 ,8.28 ,7.54 ,6.99  },
//i[8]={7.26 ,6.74 ,8.12 ,7.62 ,8.04 ,7.92 ,7.51 ,6.82 },
//h;
int man=331,woman=858,l=0;
double a[11]={52,104,54,73,66,43,46,22,68,83,51},
s[11]={192,201,151,148,121,177,163,76,155,188,145},
h,j,k;

//printf("%lf",y[0]);
//for(m=10;m>=0;m--)
//for(l=7;l>=0;l--)
for(z=10;z>=0;z--)
for(x=10;x>=0;x--)
for(c=10;c>=0;c--)
```

```
for(v=10;v>=0;v--)
for(b=10;b>=0;b--)
//for(n=7;n>=0;n--)
{

if(z!=x)
if(z!=c)
if(z!=v)
if(z!=b)
//if(z!=n)
if(x!=c)
if(x!=v)
if(x!=b)
//if(x!=n)
if(c!=v)
if(c!=b)
//if(c!=n)
if(v!=b)
//if(v!=n)
//if(b!=n)
//if(l!=z)
//if(l!=x)
//if(l!=c)
//if(l!=v)
//if(l!=b)
//if(l!=n)h
//if(m!=z)
//if(m!=x)
//if(m!=c)
//if(m!=v)
//if(m!=b)
```

```
//if(m!=n)
//if(m!=l)
//


{
h=0.417*(man-(a[z]+a[x]+a[c]+a[v]+a[b]));
j=0.227*(woman-(s[z]+s[x]+s[c]+s[v]+s[b]));
//printf("%lf--%lf\n",h,j);
if(h<0){
h=-h;
}
if(j<0){
j=-j;
}
k=100-h-j;


//if(z==10)  z=A;
//if(z==11)  z=B;
//if(x==10)  x=A;
//if(x==11)  x=B;
//if(c==10)  c=A;
//if(c==11)  c=B;
//if(v==10)  v=A;
//if(v==11)  v=B;
//if(b==10)  b=A;
//if(b==11)  b=B;
l++;
printf("%d-----%lf-----%d%d%d%d%d-----%lf------%lf\n",l,k,z,x,c,v,b,h,j);
}
}
```

（问题二三四涉及的程序较多，保存在附件支撑材料中）