

# 系统工程导论第二次作业

何舜成

2015 年 4 月 6 日

## 1 问题描述

小明已经是一名本科三年级的学生了，面对大四即将到来的毕业选择，他常常在思考：出国，读硕，直博，还是直接工作？

在今天的系统工程课后，小明突然想到可以用AHP方法帮自己做一个决策。

对于上述4个毕业选择，小明有3个考虑的原则：

- 1) 以自己的成绩和能力，作这个选择的难度如何；
- 2) 从自己的性格和以往的经验来说，自己是否适合或者喜欢这个选择；
- 3) 这几个选择对自己的职业发展影响如何。

对于第三个原则，小明认为过于宽泛，经深思熟虑，觉得这条原则可分为3小点考虑：

- 3.1) (毕业后) 找工作的难度；
- 3.2) 工作得到的待遇；
- 3.3) 学位和履历对自己长期发展影响。

请同学们合理构想一个小明，简要描述他的基本情况。并利用AHP方法，替他为这四个选择排序，给出权重。

## 2 参数确定

四个选择：出国、读硕、直博、工作

三个原则：难度、喜爱程度、职业发展

原则三的子规则：找工作的难度、工作待遇、学历的长期影响

	原则一	原则二	原则三
原则一	1	1/5	1/4
原则二	5	1	2
原则三	4	1/2	1

表 1: 判断矩阵A: 三个原则的排序

	规则一	规则二	规则三
规则一	1	1/6	1/2
规则二	6	1	4
规则三	2	1/4	1

表 2: 判断矩阵A1: 原则三子规则排序

	选择一	选择二	选择三	选择四
选择一	1	1/3	5	1
选择二	3	1	7	3
选择三	1/5	1/7	1	1/8
选择四	1	1/3	8	1

表 3: 判断矩阵B1: 四个选择相对于原则三子规则一（找工作难度）的排序

	选择一	选择二	选择三	选择四
选择一	1	4	1/2	7
选择二	1/4	1	1/4	4
选择三	2	4	1	9
选择四	1/7	1/4	1/9	1

表 4: 判断矩阵B2: 四个选择相对于原则三子规则二（工作待遇）的排序

	选择一	选择二	选择三	选择四
选择一	1	4	1/2	7
选择二	1/4	1	1/6	4
选择三	2	6	1	9
选择四	1/7	1/3	1/9	1

表 5: 判断矩阵B3: 四个选择相对于原则三子规则三（学历的长期影响）的排序

	选择一	选择二	选择三	选择四
选择一	1	1/5	3	1/3
选择二	5	1	7	3
选择三	1/3	1/7	1	1/8
选择四	3	1/3	8	1

表 6: 判断矩阵C1: 四个选择相对于原则一（难度）的排序

	选择一	选择二	选择三	选择四
选择一	1	6	3	4
选择二	1/6	1	1/3	1/3
选择三	1/3	3	1	1
选择四	1/4	3	1	1

表 7: 判断矩阵C2: 四个选择相对于原则二（喜爱程度）的排序

### 3 运行结果

7个矩阵的C.R.分别为:

	矩阵A	矩阵A1	矩阵B1	矩阵B2	矩阵B3	矩阵C1	矩阵C2
C.R.	0.0212	0.0079	0.0550	0.0395	0.0071	0.0582	0.0170

表 8: 7个矩阵的C.R.

均符合一致性要求。

最后得到的权重-得分矩阵为:

	难度	喜爱程度	职业发展影响			总得分
			0.3331			
			找工作难度	工作待遇	学历长期影响	
	0.0974	0.5695	0.1061	0.7010	0.1929	
出国	0.1111	0.5609	0.2019	0.3354	0.3203	0.4363
读硕	0.5546	0.7010	0.5176	0.1227	0.0991	0.1477
直博	0.0482	0.1898	0.0456	0.4988	0.5376	0.2654
工作	0.2861	0.1783	0.2349	0.0431	0.0431	0.1505

表 9: 最终结果

依照此结果，可以有较大的把握认为选择出国是四个选择中最好的。

### 4 具体实现

Matlab代码如下所示:

```

1 %%Codes:
2 clc;
3 clear all;
4 A = [1 1/5 1/4; 5 1 2; 4 1/2 1];
5 A1 = [1 1/6 1/2; 6 1 4; 2 1/4 1];
6 B1 = [1 1/3 5 1; 3 1 7 3; 1/5 1/7 1 1/8; 1 1/3 8 1];

```

```

7 B2 = [1 4 1/2 7; 1/4 1 1/4 4; 2 4 1 9; 1/7 1/4 1/9 1];
8 B3 = [1 4 1/2 7; 1/4 1 1/6 3; 2 6 1 9; 1/7 1/4 1/9 1];
9 C1 = [1 1/5 3 1/3; 5 1 7 3; 1/3 1/7 1 1/8; 3 1/3 8 1];
10 C2 = [1 6 3 4; 1/6 1 1/3 1/3; 1/3 3 1 1; 1/4 3 1 1];
11
12 %%Step 1: Calculate the weights of the three
    principles
13 [x, lambda] = eig(A);
14 r = abs(sum(lambda));
15 n = find(r==max(r));
16 max_lambda = lambda(n,n);
17 max_x = x(:,n);
18 sum_x = sum(max_x);
19 weight_A = max_x/sum_x;
20
21 cr = (max_lambda-3)/2/0.58;
22 disp(cr);
23 pause();
24
25 %%Step 2: Calculate the weights of the three subrules
26 [x, lambda] = eig(A1);
27 r = abs(sum(lambda));
28 n = find(r==max(r));
29 max_lambda = lambda(n,n);
30 max_x = x(:,n);
31 sum_x = sum(max_x);
32 weight_A1 = max_x/sum_x;
33
34 cr = (max_lambda-3)/2/0.58;
35 disp(cr);
36 pause();
37
38 %%Step 3: Calculate the score of the first subrule

```

```
39 [x, lambda] = eig(B1);
40 r = abs(sum(lambda));
41 n = find(r==max(r));
42 max_lambda = lambda(n,n);
43 max_x = x(:,n);
44 sum_x = sum(max_x);
45 score_B1 = max_x/sum_x;
46
47 cr = (max_lambda-4)/3/0.90;
48 disp(cr);
49 pause();
50
51 %%Step 4: Calculate the score of the second subrule
52 [x, lambda] = eig(B2);
53 r = abs(sum(lambda));
54 n = find(r==max(r));
55 max_lambda = lambda(n,n);
56 max_x = x(:,n);
57 sum_x = sum(max_x);
58 score_B2 = max_x/sum_x;
59
60 cr = (max_lambda-4)/3/0.90;
61 disp(cr);
62 pause();
63
64 %%Step 5: Calculate the score of the third subrule
65 [x, lambda] = eig(B3);
66 r = abs(sum(lambda));
67 n = find(r==max(r));
68 max_lambda = lambda(n,n);
69 max_x = x(:,n);
70 sum_x = sum(max_x);
71 score_B3 = max_x/sum_x;
```

```
72
73 cr = (max_lambda-4)/3/0.90;
74 disp(cr);
75 pause();
76
77 %%Step 6: Calculate the score of the first principle
78 [x, lambda] = eig(C1);
79 r = abs(sum(lambda));
80 n = find(r==max(r));
81 max_lambda = lambda(n,n);
82 max_x = x(:,n);
83 sum_x = sum(max_x);
84 score_C1 = max_x/sum_x;
85
86 cr = (max_lambda-4)/3/0.90;
87 disp(cr);
88 pause();
89
90 %%Step 7: Calculate the score of the second principle
91 [x, lambda] = eig(C2);
92 r = abs(sum(lambda));
93 n = find(r==max(r));
94 max_lambda = lambda(n,n);
95 max_x = x(:,n);
96 sum_x = sum(max_x);
97 score_C2 = max_x/sum_x;
98
99 cr = (max_lambda-4)/3/0.90;
100 disp(cr);
101 pause();
102
103 %%Step 8: Calculate the final result
104 s1 = score_B1*weight_A(3)*weight_A1(1);
```

```
105 disp(s1);
106 s2 = score_B2*weight_A(3)*weight_A1(2);
107 disp(s2);
108 s3 = score_B3*weight_A(3)*weight_A1(3);
109 disp(s3);
110 s4 = score_C1*weight_A(1);
111 disp(s4);
112 s5 = score_C2*weight_A(2);
113 disp(s5);
114
115 score = s1+s2+s3+s4+s5;
116 disp(score);
```