Team Assignment Model Based on Analytic Hierarchy Process (AHP)

Yifan Sun, Zeyuan Xu, Zhiyuan Li

November 5, 2017

Summary

In order to predict the scores of 100 teams of three, a predictor function of team assessment was developed.

we first categorize metrics into three aspects: basic skills, writing skills and modifier item. Basic skills are subdivided by two elements, mathematical skills and programming, while modifier includes ego modifier and imbalance modifier. One of the three datasets provided can be used to verify the predictor function.

Analytic Hierarchy Process (AHP) is established to evaluate the weights for each criteria and subcriteria to the predictor function. The technique of normalization of the data could allow the aggregation of attributes with comparable data. With the statement of relationship among different attributes in the problem, we improve our predictor function to further assess the scores of each team.

We decide to use a combination of additive model and multiplicative model to develop our team assessment equation. The additive model part consists of mathematical skills, programming and writing skills, with multiplicative part consists of personality and modifier item. Noted that the personality factor was constructed with a linear function that can have either the positive or negative impact on the function based on the closeness of personality among each team member. Using a random dataset of 300 students with specific values of attributes, the 300 students are grouped into 100 teams of three and the resultant team scores based on the predictor function matches the expectation.

After we fit the model, we try to answer two questions: How to select the top ten team? And how to assign 300 students to 100 teams so that overall they can have better scores. For the first question, we use a pure exhaustive method. For the second question, we apply a grouping method to simplify the algorithm. Namely, we assign students to a certain number of groups based on their Personality index. Then we use combination to figure out the best possible assignment.

We find that to the key element of a good team is a large personality multiplier (i.e. all the members in the team have similar personality). Secondly, a good team should have a person with exceptionally high score in math and a person with exceptionally high score in coding. This means that a good team should have members who complement each other in math score and coding score. Lastly, a good team’s members should all have high or moderately high writing scores.

Keywords: AHP; team assignment
Contents

1 Introduction ............................................. 3

2 Assumptions ........................................... 3

3 Symbols Definition .................................... 4

4 Model Design ........................................... 4
   4.1 Overview ......................................... 4
      4.1.1 Special Note ................................ 5
   4.2 Predictor Evaluation ............................... 5
      4.2.1 Team Math Score $X_{Mi}$ ................... 5
      4.2.2 Team Coding Score $X_{Ci}$ ................. 6
      4.2.3 Team Writing Score $X_{Wi}$ ............... 6
      4.2.4 Ego Effect Modifier $X_{Egoi}$ ............. 6
      4.2.5 Imbalance Team Modifier $X_{Imbi}$ ........ 7
      4.2.6 Personality Multiplier ....................... 7
   4.3 Analytic Hierarchy Process (AHP) Model ........ 7
      4.3.1 Model Construction .......................... 7
      4.3.2 Calculation ................................... 8
      4.3.3 AHP Evaluation ............................... 9
   4.4 Normalization ..................................... 9
   4.5 Final Equation ................................... 10

5 Results using data2.csv ............................... 10
   5.1 Model Testing ..................................... 10
   5.2 Answer to Question a ............................. 11
   5.3 Answer to Question c ............................. 13

6 Conclusion ............................................ 13
   6.1 Strengths and Weakness .......................... 14
   6.2 Plan for Improvement ............................. 14

7 Summary for Contest Organizer: Team Section Guide 15
Appendices

Appendix A  MATLAB Code

A.1  Code to calculate team score ........................................ 16
A.2  Code used in question a ............................................... 17
A.3  Code used in question c ............................................... 17
1 Introduction

With 300 students at a large online university signed up for the math modeling competition, all the participants will be formed into 100 teams of three. The competition will reply on four different attributes in each team member, which includes:

1. Mathematics ($M$): scale of 0 to 10 with 10 at the highest
2. Programming Skills ($C$): scale of 0 to 10 with 10 at the highest
3. Writing ($W$): scale of 0 to 10 with 10 at the highest
4. Personality ($P$): scale of 0 to 29, indicating one of 30 types of personalities

In order to help these participants form a team, as well as to assess the performance of each team, we are required to develop at least one mathematical model to predict the score of a team with the dataset of the attributes for 300 participant being given. After the model of predictor being developed, it will be able to help fulfill three different ways of forming teams:

1. Find the best teams can be put together, specifically considering to find the top 10 teams with replacement and without replacement
2. With two members assigned in a team, search for the 3rd member in a team without evaluating the predictor function with all 298 combinations
3. Based on the predictor function of each team, develop algorithm(s) to assign students in 100 teams such that the average (or median) score of all teams is higher than that of randomly picking 100 teams from the 300 people

To develop a model with good reasoning and justification, we started with applying Analytic Hierarchy Process (AHP) model and get the weights to each group factor and sub-group factor. We decided to use a combination of additive model and multiplicative model to develop our equation. The datasets from 300 students will be used for model testing and demonstration purposes.

2 Assumptions

1. Assume that the individual Mathematics, Computer programming, Writing skills and Personality can perfectly reflect the impact of a student on the team.
2. Assume that the individual Mathematics of the three students in the team can perfectly reflect the team overall ability of Mathematics. The same for Computer programming, Writing skills, personality.
3. Assume that the Team Mathematics, Team Computer programming, Team Writing skills, modifier factor and Team personality multiplier can perfectly reflect the final team score.
4. Assume that students with relatively higher Mathematics or Computer programming can have a higher impact on the team Mathematics or Computer programming.

5. Assume that having any imbalance between the Team Mathematics and Team Computer programming will result in a net decrease in the team score, which depends on the scale of imbalance.

6. Assume that the closest personality results in best teamwork and vice versa.

7. Assume that the best teamwork can result in 30% net increase of the team score while the worst teamwork can result in 50% net decrease.

8. If a student are top 20% among all students in M, C or W, then assume that such student has a high skill in M, C or W, respectively.

9. Assume that two persons on the team with extremely high and similar skills sets in M, C or W will have slight decrease in the team score.

3 Symbols Definition

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{ij}$</td>
<td>Mathematic scores of the $j^{th}$ individual in the $i^{th}$ team</td>
</tr>
<tr>
<td>$C_{ij}$</td>
<td>Computer Programming scores of the $j^{th}$ individual in the $i^{th}$ team</td>
</tr>
<tr>
<td>$W_{ij}$</td>
<td>Writing scores of the $j^{th}$ individual in the $i^{th}$ team</td>
</tr>
<tr>
<td>$P_{ij}$</td>
<td>Personality index of the $j^{th}$ individual in the $i^{th}$ team</td>
</tr>
<tr>
<td>$X_{Mi}$</td>
<td>Team Mathematics scores for the $i^{th}$ team</td>
</tr>
<tr>
<td>$X_{Ci}$</td>
<td>Team Computer Programming scores for the $i^{th}$ team</td>
</tr>
<tr>
<td>$X_{Wi}$</td>
<td>Team Writing scores for the $i^{th}$ team</td>
</tr>
<tr>
<td>$X_{Ego_i}$</td>
<td>Ego Modifier for the $i^{th}$ team</td>
</tr>
<tr>
<td>$X_{Imb_i}$</td>
<td>Imbalance Modifier for the $i^{th}$ team</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Personality Multiplier for the $i^{th}$ team</td>
</tr>
<tr>
<td>$w_i$</td>
<td>Weight vector for the $i^{th}$ team</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>Final performance for the $i^{th}$ team</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The largest eigenvalue in the pairwise comparison matrix</td>
</tr>
<tr>
<td>$X^*$</td>
<td>a vector of predictor variables after transformation</td>
</tr>
</tbody>
</table>

4 Model Design

4.1 Overview

To predict the overall performance of the team in the competition, we have to find possible predictors for all the teams. Based on the description of the question and what we have experienced so far, we postulate that the performance of a team is determined by a
combination of Team Math Scores, Team Coding Scores, Team Writing Scores, two modifiers (Ego Effect Modifier and Imbalanced Team Modifier) and a Personality Multiplier.

We apply Analytic Hierarchy Process (AHP) to determine the weight of five criteria, excluding the Personality Multiplier and the stochastic term (the rationale behind this will be discussed later). After we acquired the weights, we have to normalize all predictors to the same scale. Later, a Personality Multiplier will be multiplied to the equation. This will adjust for team corporation (See Figure 1).

Figure 1: Model Overview

4.1.1 Special Note

We did not include $P_i$ in the AHP model because we believe that, $P_i$ behave more like a multiplier. If we think about a situation where three team members really don’t get along with each other, then, they don’t cooperate, maybe even argue all the time. So their overall performance will be decrease, vice versa. Hence, instead of treating $P_i$ as an addition to the model, we think it is the best if we multiply it to the equation.

4.2 Predictor Evaluation

Given the dataset, we derived these following predictors in our model.

4.2.1 Team Math Score $X_{M_i}$

In reality, we expect that if a team has a member whose math skill is exceptional, the overall math performance of the team will increase drastically (i.e. the member with
the highest math score will "carry" the team). In light of this, we developed a weighted average method to calculate the team’s math score. Further, we assume the member with the highest math score will contribute 70% of the team’s math score, the member with middle math score will contribute 20% of the team’s math score, and the member with the lowest math score will contribute 10% of the team’s math score.

Let \( M_{i(1)} \leq M_{i(2)} \leq M_{i(3)} \),

\[
X_M = 0.7 \times M_{i(3)} + 0.2 \times M_{i(2)} + 0.1 \times M_{i(1)}
\]

### 4.2.2 Team Coding Score \( X_{C_i} \)

Similarly for coding score, we also expect to see an "Anti-Wooden-Bucket Effect". Namely, the overall coding score of a team is largely determined by the member with the highest coding score. Just like Team Math Score, we use a weighted average method with weight 0.7, 0.2 and 0.1 for the team member with the highest, middle and the lowest score, respectively.

Let \( C_{i(1)} \leq C_{i(2)} \leq C_{i(3)} \),

\[
X_{C_i} = 0.7 \times C_{i(3)} + 0.2 \times C_{i(2)} + 0.1 \times C_{i(1)}
\]

### 4.2.3 Team Writing Score \( X_{W_i} \)

Based on the description of the question, we know that all members have to write part of the paper and the overall quality of the paper is largely determined by the member with the lowest writing score (i.e. the overall team writing score will be drag down by the member with the lowest writing score). Thus, assign the weight 0.2, 0.3 and 0.5 to the team member with the highest, middle and the lowest score, respectively.

Let \( W_{i(1)} \leq W_{i(2)} \leq W_{i(3)} \),

\[
X_{W_i} = 0.2 \times W_{i(3)} + 0.3 \times W_{i(2)} + 0.5 \times W_{i(1)}
\]

### 4.2.4 Ego Effect Modifier \( X_{Ego_i} \)

We define that if two team member both has high scores (defined as top 20%) in the same area (Math, Coding or Writing), there is a ego effect, which means that they will try to steal each other’s thunder and decrease their overall performance. Ego Effect Modifier is a categorical variable written as follows.

\[
X_{ego_i} = \begin{cases} 
-1, & \text{if two person in the same team have high scores in W or C or M} \\
0, & \text{otherwise}
\end{cases}
\]

Note: Negative value because ego effect cause net decrease on overall performance.
4.2.5 Imbalance Team Modifier $X_{Imb_i}$

We assume that if there is an imbalance between Team Coding Score ($X_{Ci}$) and team math score ($X_{Mi}$), there will be a penalty for overall performance. In reality, if a team has very good math skills, but lacks in coding skills, their ideas would not have been tested and performed and vice versa.

$$X_{Imb_i} = -|X_{Ci} - X_{Mi}|$$

4.2.6 Personality Multiplier

We assume that if the 3 people in a team share common or similar personalities, there will be an improvement in the final team score as a result of good teamwork. Contrarily, if the personalities differ a lot, it will be difficult for the 3 people to work together. Assume that best teamwork results in +30% team score and worst teamwork leads to -50% team score.

$$\Delta_{1,2} = |P_1 - P_2|$$

$$Dist(P_1, P_2) = \begin{cases}  
\Delta_{1,2}, & \text{if } \Delta_{1,2} \leq 15 \\
30 - \Delta_{1,2}, & \Delta_{1,2} > 15.
\end{cases}$$

$$P_i = -0.0377\sqrt{Dist(P_{i1}, P_{i2}) + Dist(P_{i1}, P_{i3}) + Dist(P_{i2}, P_{i3})} + 1.3$$

4.3 Analytic Hierarchy Process(AHP) Model

4.3.1 Model Construction

Given the nature of our predictors/criteria, we divide them into three groups. They are Base Skills, Writing Skills, and Modifiers. (See Figure 2)

1. Base Skills We assume that in a math modeling contest, the most basic skills are math skills and coding skills. So Team Math Score and Team Coding Score will give us a baseline of a team’s performance.
   - Team Math Score $X_{Mi}$
   - Team Coding Score $X_{Ci}$

2. Writing Skill As discussion in the question, writing skills very important in the contest. Even more important than math and coding skills. It only takes one person who is very good at coding or math to "carry" the whole team. However, all members has to be at least moderately good at writing to finish a good paper, as a paper cannot be completed by one good writer. We want to stress the importance of writing, so we make writing its own category.
   - Team Writing Score $X_{Wi}$

3. Modifiers To account for the possible effect of ego effect and imbalanced team effect, we added two modifiers to adjust our model.
• Ego Effect Modifier $X_{Ego_i}$
• Imbalanced Team Modifier $X_{Imb_i}$

![Hierarchy Figure](image)

Figure 2: Hierarchy Figure

### 4.3.2 Calculation

We use Saaty’s Comparison Model of 1-9 to construct pairwise comparison tables. We can first Pairwise Comparison Matrix of Hierarchy I-II Table 2.

<table>
<thead>
<tr>
<th>Team selection</th>
<th>Base skills</th>
<th>Writing skills</th>
<th>Modifiers</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base skills</td>
<td>1</td>
<td>2/3</td>
<td>4</td>
<td>0.3735</td>
</tr>
<tr>
<td>Writing skills</td>
<td>3/2</td>
<td>1</td>
<td>5</td>
<td>0.5272</td>
</tr>
<tr>
<td>Modifiers</td>
<td>1/4</td>
<td>1/5</td>
<td>1</td>
<td>0.0992</td>
</tr>
</tbody>
</table>

The Pairwise Comparison Matrices of Hierarchy II-III are calculated below.

<table>
<thead>
<tr>
<th>Base Skills</th>
<th>Team Math Score</th>
<th>Team Coding Score</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Math Score</td>
<td>1</td>
<td>3/2</td>
<td>0.6</td>
</tr>
<tr>
<td>Team Coding Score</td>
<td>2/3</td>
<td>1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Ego Effect Modifier</th>
<th>Imbalanced Team Modifier</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ego Effect Modifier</td>
<td>1</td>
<td>1/3</td>
<td>0.25</td>
</tr>
<tr>
<td>Imbalanced Team Modifier</td>
<td>3</td>
<td>1</td>
<td>0.75</td>
</tr>
</tbody>
</table>
We combine weights acquired above to get a final weight for each criterion. (See Table 5)

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Math Score</td>
<td>0.2241</td>
</tr>
<tr>
<td>Team Coding Score</td>
<td>0.1494</td>
</tr>
<tr>
<td>Team Writing Score</td>
<td>0.5272</td>
</tr>
<tr>
<td>Ego Effect Modifier</td>
<td>0.0248</td>
</tr>
<tr>
<td>Imbalanced Team Modifier</td>
<td>0.0744</td>
</tr>
</tbody>
</table>

The weight vector and the largest eigenvalue $\lambda$ is displayed below.

$$w = (0.2241, 0.1494, 0.5272, 0.0248, 0.0744)^T,$$

$$\lambda = 3.02.$$  

4.3.3 AHP Evaluation

To evaluate our model, we perform a constancy ratio test. The consistency index is calculated as follows:

$$CI = \frac{\lambda - n}{n - 1} = 0.01$$

From Table 6, we can find the random consistency index $RI = 0.58$

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI</td>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0.90</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
</tr>
</tbody>
</table>

$$CR = \frac{CI}{RI} = 0.01724 < 0.1$$

Thus, we can conclude that our hierarchy matrix is consistent and the weights we obtained from the model is valid.

4.4 Normalization

To eliminate the scale effect of each variable, we perform a normalization procedure to all variables. Namely, we transform our variables so that each variable will have a range of 1. For variables with positive impact, they will have a minimum of 0 and maximum of 1, and for variables with negative impact, they will have a minimum of -1 and maximum of 0.

Originally $X_{Ego}$ has a minimum of -1 and maximum of 0, so we do not need to transform it. For $X_{M_i}, X_{C_i}, X_{W_i}$, they all have a minimum of 0 and maximum of 10. To
make it simple, we divide them all by 10. For $X_{Imb,i}$, we can also divide it by 10, so that it has a minimum of -1 and maximum of 0.

Finally we get,

$$X^* = \left( \frac{X_{M_i}}{10}, \frac{X_{C_i}}{10}, \frac{X_{W_i}}{10}, X_{Ego_i}, \frac{X_{Imb_i}}{10} \right)^T$$

4.5 Final Equation

The final equation to determine the $i^{th}$ team’s performance is expressed as such:

$$Q_i = \langle w, X^* \rangle 100 P_i$$

Note: For the sake of interpretation, we multiply $Q_i$ by 100, so that it looks like a real score for modeling contest. $Q_i$ has a theoretical maximum of 113.88 and a theoretical minimum of 0.

5 Results using data2.csv

5.1 Model Testing

Using the software R, we randomly assign 300 people to 100 teams and calculate each team’s $Q$. We select the first two group in our analysis. The followings are their statistics.

<table>
<thead>
<tr>
<th>Table 7: Randomized Team 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>Member1</td>
</tr>
<tr>
<td>Member2</td>
</tr>
<tr>
<td>Member3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 8: Randomized Team 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 2</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>Member1</td>
</tr>
<tr>
<td>Member2</td>
</tr>
<tr>
<td>Member3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 9: Score Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team.M(N)</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>Group 2</td>
</tr>
</tbody>
</table>

Team 2’s predicted final score is much higher than team 1’s. Here is our analysis and the reason why this model accurately reflect the reality.
First, the unweighted sums of all team members math scores for team 1 and team 2 are nearly the same (17.72 vs. 17.11). However, since team 1 has a student whose math score is exceptionally high. Thus, he will "carry" the whole team and increase the overall team math performance. This "carrier" effect is reflected in our weighted average calculation. This goes the same for Coding and Writing scores.

Second, when a team has an unbalanced math score and coding score, its overall performance will be decreased. For example, Team 1 has many students who are good with math, but not so many who are good at coding. In this situation, under our assumption, there will be a larger penalty for team 1 (represented by negative Imbalanced Team Modifier.

Third, a very important component in our model is the personality multiplier. In this situation, team 1 and team 2 are neck-to-neck, or at least not far by much, when we only consider the first five predictors. However, since team 1’s members have very different personalities, we expect that they would not cooperate very well and result in lower efficiency and lower scores. This is also reflected in team 1 and team 2’s final scores.

5.2 Answer to Question a

(The dataset gives the information of one student per row. For convenience, we label every student with a number according to their row)

Question A requires us to find top 10 possible teams with the team score predictor that we come up with. In this case, since our predictor is entirely mathematical, the pure exhaustive method using MATLAB should be an ideal one. We first create a MATLAB function using the predictor. It will return the team score $Q_i$, if given the information of three students in one team. Then we let MATLAB apply the function to every possible combination of teams among 300 students. The top 10 team scores are stored in a 10x4 matrix called "Top10", which includes the number label of the students and their team score.

The results are shown below

The answer is represented in the table below

<table>
<thead>
<tr>
<th>Rank</th>
<th>$Q$</th>
<th>Student 1</th>
<th>Student 2</th>
<th>Student 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>73.72083</td>
<td>69</td>
<td>85</td>
<td>228</td>
</tr>
<tr>
<td>2</td>
<td>72.79225</td>
<td>76</td>
<td>201</td>
<td>252</td>
</tr>
<tr>
<td>3</td>
<td>72.77944</td>
<td>69</td>
<td>228</td>
<td>290</td>
</tr>
<tr>
<td>4</td>
<td>72.4293</td>
<td>27</td>
<td>110</td>
<td>206</td>
</tr>
<tr>
<td>5</td>
<td>72.41608</td>
<td>69</td>
<td>201</td>
<td>228</td>
</tr>
<tr>
<td>6</td>
<td>72.34537</td>
<td>69</td>
<td>76</td>
<td>201</td>
</tr>
<tr>
<td>7</td>
<td>72.2737</td>
<td>110</td>
<td>206</td>
<td>243</td>
</tr>
<tr>
<td>8</td>
<td>72.2595</td>
<td>71</td>
<td>101</td>
<td>222</td>
</tr>
<tr>
<td>9</td>
<td>72.1357</td>
<td>27</td>
<td>70</td>
<td>110</td>
</tr>
<tr>
<td>10</td>
<td>72.12718</td>
<td>69</td>
<td>132</td>
<td>228</td>
</tr>
</tbody>
</table>
When it comes to the case that each student can only be in one team, the pure exhaustive method will not work well since it requires the MATLAB program to be more intelligent so that the computer can exclusively analysis out all students that may potentially be on the final Top10 team. Given limited time, we cannot reach a perfect automatic system with such exclusive feature. The combination of exhaustive method and manual analysis are applied to solve this problem. The mechanism is to run the previous exhaustive methods multiple times and remove the repentant students continuously. If one student appears in two Top10 teams, the teams with lower rank will be removed as well as the information of the Top10-team students from the given dataset. It guarantees that there will be no student repetency among the Top10 team and the next time we run the MATLAB program, the new Top10 team will by no means be the same with the current one. Once the "Top 10" matrix are filled, it should include 30 different students with highest team score possible.

The answer is represented in the table below

<table>
<thead>
<tr>
<th>Rank</th>
<th>Q</th>
<th>Student 1</th>
<th>Student 2</th>
<th>Student 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>73.72082</td>
<td>69</td>
<td>85</td>
<td>228</td>
</tr>
<tr>
<td>2</td>
<td>72.79225</td>
<td>76</td>
<td>201</td>
<td>252</td>
</tr>
<tr>
<td>3</td>
<td>72.4293</td>
<td>27</td>
<td>110</td>
<td>206</td>
</tr>
<tr>
<td>4</td>
<td>72.2595</td>
<td>71</td>
<td>101</td>
<td>222</td>
</tr>
<tr>
<td>5</td>
<td>72.08258</td>
<td>70</td>
<td>110</td>
<td>243</td>
</tr>
<tr>
<td>6</td>
<td>72.0699</td>
<td>106</td>
<td>114</td>
<td>242</td>
</tr>
<tr>
<td>7</td>
<td>72.06818</td>
<td>13</td>
<td>164</td>
<td>227</td>
</tr>
<tr>
<td>8</td>
<td>70.39777</td>
<td>7</td>
<td>132</td>
<td>290</td>
</tr>
<tr>
<td>9</td>
<td>68.88747</td>
<td>53</td>
<td>80</td>
<td>84</td>
</tr>
<tr>
<td>10</td>
<td>68.71163</td>
<td>62</td>
<td>113</td>
<td>268</td>
</tr>
</tbody>
</table>

Below are the top three team

<table>
<thead>
<tr>
<th>Student index</th>
<th>P</th>
<th>M</th>
<th>C</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>1</td>
<td>9.2513</td>
<td>0.6047</td>
<td>5.2283</td>
</tr>
<tr>
<td>85</td>
<td>1</td>
<td>2.2312</td>
<td>1.7755</td>
<td>7.4094</td>
</tr>
<tr>
<td>228</td>
<td>1</td>
<td>1.249</td>
<td>8.7909</td>
<td>6.2218</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student index</th>
<th>P</th>
<th>M</th>
<th>C</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>76</td>
<td>0</td>
<td>4.0142</td>
<td>4.28</td>
<td>6.6651</td>
</tr>
<tr>
<td>201</td>
<td>0</td>
<td>7.4333</td>
<td>10</td>
<td>5.9775</td>
</tr>
<tr>
<td>252</td>
<td>0</td>
<td>5.3921</td>
<td>4.8171</td>
<td>5.0191</td>
</tr>
</tbody>
</table>
Table 14: Team with the third top score

<table>
<thead>
<tr>
<th>Student index</th>
<th>P</th>
<th>M</th>
<th>C</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>11</td>
<td>7.123</td>
<td>1.596</td>
<td>7.1914</td>
</tr>
<tr>
<td>110</td>
<td>11</td>
<td>9.6409</td>
<td>4.8725</td>
<td>5.7632</td>
</tr>
<tr>
<td>206</td>
<td>11</td>
<td>3.3712</td>
<td>8.4593</td>
<td>4.9159</td>
</tr>
</tbody>
</table>

5.3 Answer to Question c

The question C requires us to assign every student a proper team such that all 300 students can achieve the common good. In this case, the common good can be expressed as the highest sum of the scores for each team. It is doable to use the exhaustive method again but the calculation turns out that the team score function needs to be implemented for more than 10 to the power of 611 times, which is impossible for a personal laptop. One idea to efficiently decrease the complexity of computation while guarantee the accuracy is grouping. Considering the importance to have similar personality in one team, we first sort all students based on their personality number in the descending order. Then students are divided into 50 groups based on the personality order so that every group has 6 students with similar personality. Now the question is becoming how to achieve the common good for every individual group. It is much simpler because there are only 60 team combinations for each group. The optimization method does work. It turns out that the average team score for randomly assigned teams are only 36.5 based on the predicator. Using the algorithm above, the new average team scores go up to 59.7, which improves 64%.

6 Conclusion

We used AHP and variable normalization procedure to build to equation to evaluate each team. Based on that equations, we use different computer algorithms to answer two questions (A and C).

The question A is to find the best possible way to select the top 10 team. For the first problem of question A, the pure exhaustive method is used. The scores of all combinations of team among 300 students are calculated and compared using MATLAB. The Top 10 team score is then saved automatically as well as the team member information. For the second problem of question A, it requires exclusive team assignment. We mix the computer exhaustive method and manual analysis. Once the MATLAB generates results, we manually filter out the repentant student until all Top 10 teams are filled.

The question C is to find the best way to assign 300 students to 100 teams. For question C, the pure exhaustive method is not possible. Grouping method is to divide the 300 students into 50 groups (can also be 25, 20, 10 groups) with similar personalities. It transfers the question C from assigning 300 people to assign 6 people each time, which quantitatively lower the complexity from 10E+660 to 10E+2 so that the MATLAB can easily optimize the team assignment for each small group of people.

Besides these two questions, we find that to the key element of a good team is a large personality multiplier (i.e. all the members in the team have similar personality).
Secondly, a good team should have a person with exceptionally high score in math and a person with exceptionally high score in coding. This means that a good team should have members who complement each other in math score and coding score. Lastly, a good team’s members should all have high or moderately high writing scores.

6.1 Strengths and Weakness

Strength:

- We evaluated and weighted each available objective and subject elements for AHP model.
- The final model for predicting team assignment fits well with the datasets being given. and assumption being made, it also fits well to certain level of common sense.
- The final model is simple and easy to understand.

Weakness:

- Our time is quite limited, we are only able to take the available criterions in the problem listed into consideration, we could possibly include more subject/objective criterions into the AHP table.
- To achieve a better fitting model, we could consider using a combination model of AHP and Fuzzy Synthetic Evaluation (FSE) to cover more objective factors since AHP is more objective.
- An random error term could be added into our final model to make the model more realistic.

6.2 Plan for Improvement

To further improve the model, we could try to improve the model by using a better fitting nonlinear model and then normalize it, or we could come up with a random error term to make our final model more realistic. We could add more criterion in the modifier like leadership, which could possibly have positive or negative effect on the team etc. We could include more objective element including time weight and use Fuzzy Synthetic Evaluation (FSE) to evaluate each team’s hit scores. By comparing FSE model with AHP model, we might want to decide which one is better.
7 Summary for Contest Organizer: Team Section Guide

With the help of our predictor model, we would highly suggest students to start searching for their group mates by gathering along information of personality first with other students. Since all the students are not be able to meet in person, online tools like skype, facebook, even text message can be used for further knowing each other and have a closer feeling of whether each of us has the same, close or even distinct personality with respect to things like favorites and hobbies etc. The personality of each team member will determine whether the interpersonal conflict can be controlled and the team can be assembled together with a healthy team dynamics.

Next, the students can introduce their proficiency of Mathematical Modeling skills, computer programming skills and Writing skills, and talk about who is good at what skills based on some facts like previous or similar experience. We also suggested students to form a team with the other two students have the skills that you do not have. In this way, the diversity of skills in the team can be obtained. To be specific, if one student is good at coding, then ideally the other two students should be good at mathematical modeling and writing. This could result in a really high team performance, which is also verifiable based on the predictor modeling because of every student in the team are "carrying" their parts in the team as a whole. Noted that this precondition is their skills are quite proficient, then the efficiency would be boosted based on every team member that can contribute to the progress with a close personality.

However, if two of the three students in a team are both proficient to the same skill, then the situation of competitive conflict between those two students may occur. Because the team might have to choose between those two solutions, it would possibly lag the team process and excess labor effort would occur, resulting in negative effect on the whole team performance. But a good personality could offset this defect by one of the two actively giving up his own idea and follow the other. Then the whole performance of the team could still be very high. Another way to manage this conflict is by selecting a group leader from the beginning so that when conflicts happen, which means that the role of each team member must be established before the team starts moving forward. This method could facilitate the decision making and the development of the progress in a team.
References


Appendices

Appendix A  MATLAB Code

A.1  Code to calculate team score

Here is the program we used to calculate team score.

Input matlab source:

```matlab
function TeamScore= TeamScore_given(A ,x, y, z ) % The team score of student x, y, z
% read dataset from excel into MATLAB matrix
M=[A(x,:);A(y,:);A(z,:)];
M = sort(M);
ego= 0;
if max(M(2,2:4))>=6.5
    ego = 1;
end
TeamP=sqrt(Dist(M(1,1),M(2,1))^2+Dist(M(1,1),M(3,1))^2+Dist(M(2,1),M(3,1))^2);
TeamM=M(1,2)*0.1+M(2,2)*0.2+M(3,2)*0.7; %7:2:1
TeamC=M(1,3)*0.1+M(2,3)*0.2+M(3,3)*0.7; %7:2:1
TeamW=M(1,4)*0.5+M(2,4)*0.3+M(3,4)*0.2; %2:3:5
PersonalityMult = -0.03771*TeamP+1.3; %Personality Multiplier
imbalance = abs(TeamM-TeamC); %imbalance effect
% Team M: 22.4% , Team C: 15.0%, Team W: 52.7%, ego: 2.5%, imbalance: 7.4%
% (calculated by AHP)
TeamScore = ((TeamM/10)*0.224+(TeamC/10)*0.150+(TeamW/10)*0.527-ego*0.025-(imbalance/10)*0.074)*PersonalityMult*100;
end

function y= Dist( a,b ) % Calculate the modulo distance between a & b (mod 30)
y = abs(a-b);
    if y >15
        y = 30 - y;
    end
end
```

end
A.2 Code used in question a

Here is the program we used for question a.

Input matlab source:

```
Top10=zeros(10,4);
smallest=0;
filename='data2.csv';
M=xlsread(filename);
studentnumber=300; %The student number to find Best team score among
for i = 1:studentnumber-1
    for j = i+1:studentnumber-1
        for k=j+1:studentnumber
            Q = TeamScore_given(M,i,j,k);
            if Q>smallest
                [smallest,index] = min(Top10(:,1));
                Top10(index,:) = [Q,i,j,k];
            end
        end
    end
end
```

A.3 Code used in question c

Here is the program we used for question c.

Input matlab source:

```
filename='data2_sorted.csv';
filename2='data2.csv';
Group = [];
M=xlsread(filename);
M1=xlsread(filename2);
bestComb = 0;
newTeamscore = [];
for group = 1:50 %divide 300 students into 50 groups
    index = 6*group-5;
    for i=1:4
        for j=i+1:5
            Q1 = TeamScore_given(M,index,index+i,index+j);
            A = [1 2 3 4 5];
            A(A==i)=[ ];
            A(A==j)=[ ];
            Q2 = TeamScore_given(M,index+A(1),index+A(2),index+A(3));
            if (Q1+Q2)>bestComb
                bestComb = Q1 + Q2;
                add = [M(index,5) M(index+i,5) M(index+j,5) M(index+A(1),5) M(index+A(2),5) M(index+A(3),5)];
            end
        end
    end
end
```

Q1=TeamScore_given(M1,add(end-1,1),add(end-1,2),add(end-1,3));
Q2=TeamScore_given(M1,add(end,1),add(end,2),add(end,3));
newTeamscore= vertcat(newTeamscore,[Q1 Q2]);
Group = vertcat(Group, add);
bestComb = 0;
\textbf{end}