Team Control Number : 61402

Problem Chosen : 2

2017 OSU Math Contest for Modeling Summary Sheet

For students who are still looking for the team completion, we have several advice for you.

- The writing skills will act as a multiplier and contribute more compared with mathematics and computer programming. Thus, you need to maximize the sum of scores of individual writing skills in general. However, due to the possible ego effect, we suggest you may only choose teammates with the highest writing skills under the 90% threshold (6.7824) if you already have someone with the extremely high writing skills in your incomplete team to avoid having two or three members with the extremely high writing skills in your team.
- For personality, you need to minimize the personality distances to achieve the best team performance. If you are looking two other students, we suggest you to find students whose personality scores are close to you within the range of 2. If you and the second student are looking for the third student, you may use both of your personality scores as references to find a third student whose personality score is located on the shorter arc in the modular arithmetic graph.
- Due to the robust and adjacent connection between mathematics and computer programming scores, we suggest you to consider them together when looking for your teammates. If right now there exists a heavy imbalance between mathematics and computer programming scores in your incomplete team, the complementary strategy may be applied which means you can look for someone who has the opposite skill set. By doing that, you may mitigate the possible negative imbalance effect. Also, the general team performance may be improved since you may have both high mathematics and computer programming scores on the team level.

Rise Up the Math Modeling Contest

Yiyang Li, Yunfei Luo, Cheng Zeng

November 5th, 2017

Contents

1 Introduction

In our report, we will provide three predictive models. The first model for Problem a is to select top 10 three-member teams that possibly perform the best out of 300 students, with or without permissions for the same students to participate in multiple teams. The second model for Problem b is to provide the algorithm to select the best-fit third person out of remaining 298 students given two chosen students without evaluating all of them. The third model for Problem c is to arrange all students into 100 three-member teams properly so that they will outperform the 100 randomly-selected teams.

2 Assumptions

Before the model building, we have several following assumptions.

Assumption 1 (Distribution Assumption) We assume that three attributes (mathematics, computer programming, and writing skills) of all students in this math modeling competition are in the normal distribution because the sample distributions of those attributes are approximately in the symmetrical bell shape. We also assume that the personality attribute is approximately in a discrete uniform distribution (around 10 students for each score) given its sample distribution we have. The sample distributions are shown in the Figure 1.

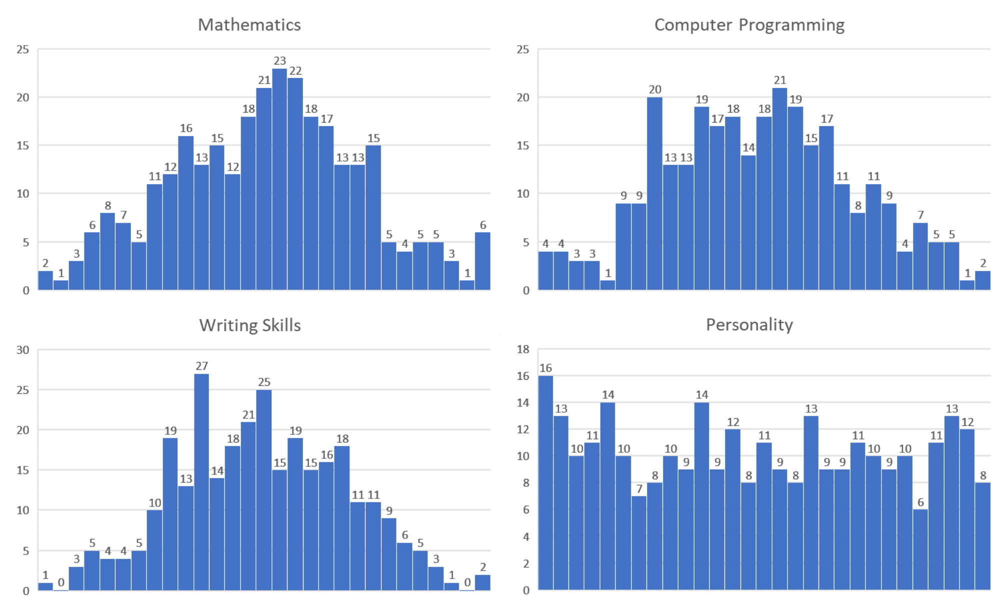


Figure 1: The Sample Distributions of Four Attributes

Assumption 2 (Writing Skills Assumption) Given that team members need to share the writing assignment and the weakest writer will lower the score, we assume that writing skills can be evaluated by the average score and the lowest score of writing skills together with certain adjustments, and the function of writing skills will work as a multiplier in our team performance function.

Assumption 3 (Mathematics and Computer Programming Assumption) We refer the carry effect to the phenomenon that one person who outperforms in mathematics and computer programming seems to carry his or her team. We assume that the abilities of mathematics and computer programming can be evaluated by their corresponding scores on the team level as well as the specific function designed to measure the carry effect.

Assumption 4 (Imbalance Assumption) In terms of the possible imbalance between mathematics and computer programming from both the individual level and the team level, we assume that this imbalance can be evaluated by the difference of the values of the functions related to mathematics and computer programming. We also assume that this factor will act more like an adder rather than a multiplier.

Assumption 5 (Personality Assumption) Judge and Ilies [1] state that personality can be viewed as the set of habitual behaviors, cognitions and emotional patterns and it varies among people. From a research about personality and its performance motivation, Sefcik, Prerost and Arbet [2] conduct a quantitative study including correlations to psychological variables. During this study, personality is measured by a five-factor model and the subsequent review indicates a medium multiple correlation with the motivation performance on average. From the academic researches mentioned above, we assume that personality distances have approximately a medium effect on the team performance in the short run. Given that the math modeling contest is from three to five days, personality should not have too much effect on the team performance and thus we set the range of the personality coefficient between 0.8 and 1.2.

Assumption 6 (Ego Effect Assumption) We define "extremely high" as his or her performance rated 90% or higher compared with other students among the sample distribution in the data set we choose. In order to simplify various conditions that the ego effect may be more or less, we treat the ego effects of mathematics and computer programming skills as the same, which are slightly more severe than that of the writing one. Given that the difference between the standard deviations of mathematics and computer science (2.0748 and 2.0851) is negligible while the standard deviation of writing (1.4125) is noticeably different, we separate ego effects into six conditions.

- Zero ego effect: zero or one team member is extremely high in any subject.
- One-attribute slight ego effect: any two or three team members have extremely high scores only for writing skills.

- One-attribute severe ego effect: any two or three team members have extremely high scores either for mathematics or for computer programming.
- Two-attribute slight ego effect: any two or three team members have extremely high scores for either mathematics or computer programming and for writing skills.
- Two-attribute severe ego effect: any two or three team members have extremely high scores for both mathematics and computer programming.
- Three-attribute ego effect: any two or three team members have extremely high scores for mathematics, computer programming and writing skills.

Standage, Duda and Pensgaard [3] have researched how different characteristics under the competitive settings affect the coordination task. They conclude that the ego effect influences the cooperation negatively and results in a poorer grade. No matter how egoistic a student is, however, he or she focuses on completing the tasks and desires to obtain the rewards. Thus, we estimate that the ego effect can not cause a decline to a large extent and thus assume the range of its coefficient between 0.7 and 1.

Assumption 7 (Interaction Assumption) It is indicated in this problem that mathematics and computer programming scores are not correlated and the personality is independent of other skills. However, since a student with lower combined computer programming skills and mathematics than median is likely to have a higher writing skills, we assume that there may exist a correlations between writing skills and the combination of mathematics and computer programming scores.

Assumption 8 (Error Assumption) The score is stochastic rather than deterministic. Therefore, we can assume that there exists the margin of errors.

3 Model Building

We will build three models to solve the following problems and choose Data Set 2 for our model building.

- Find out the top 10 teams that can possibly perform the best considering that one member can be in multiple teams or can only be in one team.
- Develop an algorithm so that we can search for the third student given any two students to form a possibly best-performed team.
- Develop an algorithm so that we select 100 teams where anyone can only be in one team, and the selected teams will outperform 100 randomly-selected teams.

3.1 The Top 10 Team Selection (Problem a)

We will demonstrate two steps to solve Problem a. The multi-factor function approaches this problem by measuring variables and their relationships. It aims at first selecting qualified and potentially well-performed individuals out of 300 students, and then we start forming teams from those individuals. We also developed a program in C++ that allows us to measure and compare different team combinations simply by data inputs of four attributes. The algorithm is based on the multi-factor function we construct.

3.1.1 The Multi-Factor Function

We denote m_i, c_i, w_i and p_i as the i^{th} scores of mathematics, computer programming, writing skills and personality respectively given the order of data entry where $i \in \{1, 2, \cdots, 300\}$. Since the random selections of 3 out of 300 students result in $\binom{300}{3} = 4455100$ possible combinations, we decide to approach this problem first by measuring individual performances.

On the individual level, we are allowed to skip the personality factor since only the differences between personalities affect the performances, and the ego effect can also be skipped since it only occurs on the team level. Thus, we define y_i as the individual performance score and write that

$$y_i = (m_i + c_i - f(m_i, c_i)) \cdot w_i + \epsilon, \tag{3.1}$$

$$f(m_i, c_i) = \begin{cases} 0, & \text{if } |m_i - c_i| < t \\ |m_i - c_i|, & \text{otherwise} \end{cases} \tag{3.2}$$

where t is the critical value to determine whether the difference between scores of mathematics and computer programming is sufficiently large to be considered as an imbalance and ϵ is the error with the normal distribution centered at 0. Since there is no way to estimate errors, we simply plug in the data and rank the performance scores. The result of top 30 is shown in the Table 1.

Rank	Index	Scores	Rank	Index	Scores	Rank	Index	Scores
1	110	111.1249	11	268	92.8320	21	278	78.5999
2	228	109.3904	12	274	89.0638	22	99	77.9379
3	84	105.2842	13	71	88.6637	23	67	77.4562
4	201	104.2076	14	81	88.5624	24	120	77.2364
5	27	102.4487	15	102	83.1701	25	283	76.7336
6	12	100.6996	16	206	80.3286	26	5	76.6234
7	164	98.6225	17	203	80.2640	27	140	76.3357
8	259	97.5926	18	231	79.8306	28	191	75.7823
9	292	96.7371	19	186	79.6930	29	104	75.6438
10	69	92.8320	20	74	78.5999	30	25	75.1967

Table 1: The Top 30 Individuals

Remark 1 *Given the limited information, it is difficult to measure and verify the exact value of t since the definition of "sufficiently large" can be arbitrary. After observing the distribution of $|m_i - c_i|$ shown in the Figure 2, we discover that Q3 (75%) is an ideal critical value because the occurrence of data beyond that seems fluctuating and irregular. To be discreet, we use Q1 (25%), the median, and Q3 (75%) where t equals 0.9534, 2.3364 and 3.2968 respectively to compute top 50 under each condition and it turns out that 36 of 50 students are overlapped among those results under three critical values.*

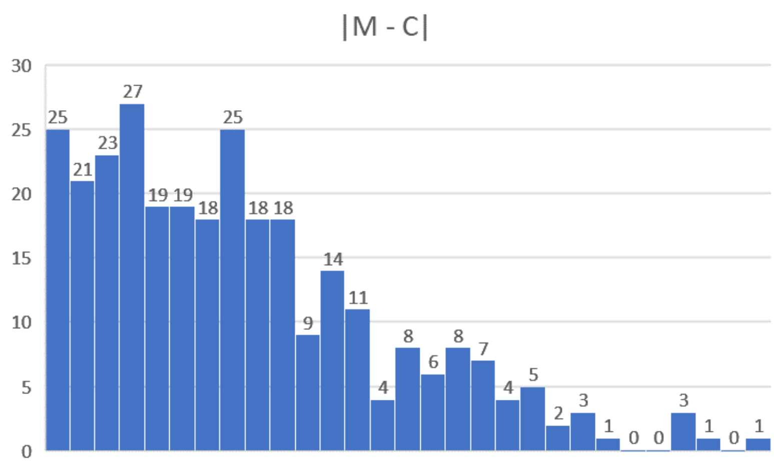


Figure 2: The Sample Distribution of $|m_i - c_i|$

Now that we have selected students, we need to form three-member teams for them. We denote their attributes in the Table 2 where $i, j, k \in \{1, 2, \cdots, 300\}$.

Person	Mathematics	Computer Programming	Writing Skills	Personality
1	m_i	c_i	w_i	p_i
2	m_j	c_j	w_j	p_j
3	m_k	c_k	w_k	p_k

Table 2: The Denotation of Attributes in a Team

We start considering the personality and the possible ego effect from here and combine them with our performance score function. We denote Y as the team performance score and write the general function that

$$Y = (M^* + C^* - F_{MC}) \cdot W^* \cdot P^* \cdot E + \epsilon. \tag{3.3}$$

M^* : The function of scores of mathematics on the team level
C^* : The function of scores of computer programming on the team level
F_{MC} : The function of the imbalance between mathematics and computer programming
W^* : The function of scores of writing skills on the team level
P^* : The function of scores of personality
E : The function of the ego effect
ϵ : The error distributed as $\mathcal{N}(0, \sigma^2)$

According to the Assumption 2, the function W^* should reflect both the average score and the lowest score of the team. Thus, we write that

$$W^* = \frac{w_i + w_j + w_k}{3} + \min\{w_i, w_j, w_k\}. \tag{3.4}$$

According to the Assumption 3, the functions M^* and C^* should reflect the average scores and the "carry" effects which can be simulated by the segmented function.

$$M^* = \frac{m_i + m_j + m_k}{3} + g(m_i, m_j, m_k), \tag{3.5}$$

$$C^* = \frac{c_i + c_j + c_k}{3} + g(c_i, c_j, c_k), \tag{3.6}$$

$$g(x_i, x_j, x_k) = \begin{cases} 0, & \text{if } \max\{x_1, x_2, x_3\} - \bar{x} \leq 0 \\ \left|\max\{x_1, x_2, x_3\} - \bar{x}\right|, & \text{otherwise} \end{cases} \tag{3.7}$$

where $\bar{x}$ is the mean of x. The function F_{MC} is to measure the imbalance of mathematics and computer programming both on the individual level and on the team level. That means we need to measure the comprehensive difference to have a better insight about the imbalance, and thus we write that

$$F_{MC} = p \cdot (|M^* - C^*| + |m_i - c_i| + |m_j - c_j| + |m_k - c_k|) \tag{3.8}$$

where p is the parameter to adjust the influence of the imbalance.

Remark 2 *Since we do not know how this imbalance will influence the team performance exactly, we set $p = \frac{1}{2}$ as our best guess. It can be modified if we have more information.*

In order to build the function P^*, we need to first define the personality distance rigorously.

Definition 1 *A personality distance is the shortest difference between two values on the modular arithmetic scale with the modulus 30. We denote $dist(a, b)$ as the personality distance between a and b.*

We write

$$dist(a,b) = \begin{cases} |a-b|, & \text{if } |a-b| \leq 15 \\ 30 - |a-b|, & \text{otherwise} \end{cases} \tag{3.9}$$

$$P^* = \frac{\alpha}{dist(p_i,p_j) + dist(p_i,p_k) + dist(p_j,p_k) + \beta} \tag{3.10}$$

where α and β are the parameters to adjust the range of the function P^*.

Remark 3 *The sum of the performance distances of a and b, a and c, and b and c is ranged from 0 to 30. Without adjustments, the value of P^* can be extremely large when there is little or no personality distances among three members, and this could lead to the extremely large value of the performance function Y. That indicates that any team with little or no personality distances can outperform anyway regardless other factors and this does not fit the reality. Thus, we introduce α and β here to mitigate this possible disturbance. In order to reach the range between 0.8 and 1.2 under the Assumption 5, we choose $\alpha = 72$ and $\beta = 60$.*

According to the Assumption 1 and the Assumption 6, we assume the normal distributions of scores of mathematics, computer programming and writing skills and decide to define "extremely high" as outperforming at least 90% of all students in the sample. We summarized the numerical results of the means, standard deviations and 90% thresholds of four attributes in the Table 3.

	Mathematics	Computer Programming	Writing Skills	Personality
μ	5.0717	4.8952	4.9722	14.48
σ	2.0748	2.0851	1.4125	8.6649
$\mu + \sigma \cdot z_{0.9}$	7.7306	7.5674	6.7824	N/A

Table 3: The Means, Standard Deviations and 90% Thresholds of Attributes

For the function E, we also simplify the complicated real-life scenarios and separate ego effects into six cases under the Assumption 6. If at least two members have scores higher than the 90% threshold of the same attribute, there is an ego effect to that attribute. Meanwhile, given those computational results in Table 3, we find that the standard deviations of mathematics and computer programming are similar, while the one of writing skills is significantly different. We will use three indicator functions to demonstrate whether there is an ego effect in any of three attributes.

We write

$$i_m = \begin{cases} 1, & \text{if } m \geq 7.7306 \text{ is true for at least two members} \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

$$i_c = \begin{cases} 1, & \text{if } c \geq 7.5674 \text{ is true for at least two members} \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

$$i_w = \begin{cases} 1, & \text{if } w \geq 6.7824 \text{ is true for at least two members} \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

as indicator functions of scores of mathematics, computer programming and writing skills. We combine these functions into one re-scaled indicator function and write

$$i_{total} = i_m + i_c + 0.75 i_w = \begin{cases} 0, & \text{if there is zero ego effect} \\ 0.75, & \text{if there is one-attribute slight ego effect} \\ 1, & \text{if there is one-attribute severe ego effect} \\ 1.75, & \text{if there is two-attribute slight ego effect} \\ 2, & \text{if there is two-attribute severe ego effect} \\ 2.75, & \text{if there is three-attribute ego effect} \end{cases} \tag{3.14}$$

Notice that the definitions of these ego effects are already defined in the Assumption 6.

Remark 4 *Although being aware that the ego effect to all members is definitely more severe than the one to any two members which will lead to a larger i_{total} and therefore a smaller E, we still decide to skip this consideration to avoid over-complication of our indicator functions.*

We write

$$E = \frac{\gamma}{i_{total} + \gamma} \tag{3.15}$$

as the function of the ego effect where γ is the parameter to adjust the ego effect.

Remark 5 *Under the Assumption 6, we need to control E between 0.7 and 1, and thus choose $\gamma = 7$.*

We have already defined every component of the team performance function thoroughly and should finally continue our computations given 30 individuals we selected waiting for the team forming. In order to achieve the final goal, first we consider the situation where one student can participate in multiple teams. We rearrange them from the lowest personality score to the highest one and then construct 30 teams from three students who are close to each other on the modular arithmetic scale of the personality scores (See Appendix B). After calculating the values of the team performance function defined above, we rank those results and get the top 10 list under the first situation shown in the Table 4.

Rank	Index Set	Rank	Index Set
1	(27, 110, 206)	6	(164, 203, 259)
2	(69, 201, 228)	7	(25, 84, 268)
3	(71, 201, 231)	8	(5, 12, 191)
4	(5, 12, 74)	9	(27, 110, 191)
5	(71, 201, 228)	10	(69, 186, 228)

Table 4: The Top 10 Teams (Situation 1)

For the situation where one student can only join in one team, we have developed an algorithm to remove students who already have teammates from the waiting list. We rank those scores and get the top 10 list under the second situation shown in the Table 5.

Rank	Index Set	Rank	Index Set
1	(27, 110, 206)	6	(25, 84, 268)
2	(69, 201, 228)	7	(99, 120, 274)
3	(71, 81, 231)	8	(104, 140, 292)
4	(5, 12, 74)	9	(67, 102, 186)
5	(164, 203, 259)	10	(191, 278, 283)

Table 5: The Top 10 Teams (Situation 2)

3.1.2 The Computer Programming

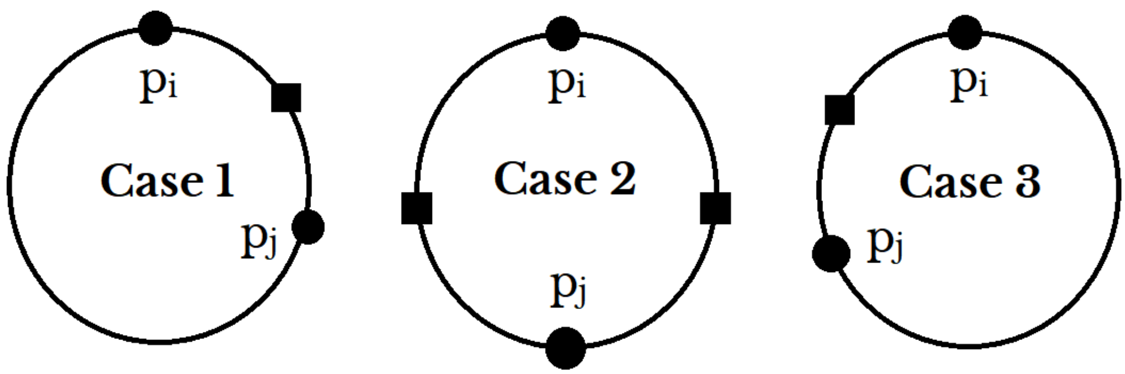
We develop an algorithm to compute the results of the team performance function given attributes of any three students (See Appendix A). This algorithm allows us to evaluate the decisions made on our own but cannot automatically generate the best-selected combinations of three-member teams.

3.2 The Third Student Selection (Problem b)

In this section, we need to build a model (more like an algorithm precisely) to find the best-fit third student out of remaining 298 given any two students without evaluating every option. We first consider this problem under the ideal situation where we do not have to compromise our rules and cope with challenges. Then, we move to the non-ideal situation and discuss about how to deal with reality.

3.2.1 The Ideal Situation

Under the ideal situation, we will not encounter the dilemma that either we have to lower our previously set-up standards or there will be no one left to choose. We denote the attributes of three students in the same way as shown in the Table 2 and define the functions in the same way shown in Section 3.1. We start our analysis from the personalities of two given students since there may be potential ego effects in mathematics, computer programming and writing skills. Our goal is to minimize $dist(p_i, p_j) + dist(p_i, p_k) + dist(p_j, p_k)$ and consequently maximize P^* with respect to the variable p_k. We consider three cases and summarize our results as following.



$$\text{Case 1: } \min\{p_i, p_j\} \le p_k \le \max\{p_i, p_j\} \qquad \text{if } |p_i - p_j| < 15 \tag{3.16}$$
$$\text{Case 2: } p_k \in \{0, 1, \cdots, 29\} \qquad \text{if } |p_i - p_j| = 15 \tag{3.17}$$
$$\text{Case 3: } p_k \le \min\{p_i, p_j\} \text{ or } p_k \ge \max\{p_i, p_j\} \qquad \text{if } |p_i - p_j| > 15 \tag{3.18}$$

We can also calculate the sum of performance distances for each case. We write that

$$dist(p_i, p_j) + dist(p_i, p_k) + dist(p_j, p_k) = \begin{cases} 2\,|p_i - p_j|\,, & \text{for Case 1} \\ 30, & \text{for Case 2} \\ 60 - 2\,|p_i - p_j|\,, & \text{for Case 3} \end{cases} \tag{3.19}$$

Given the ideal situation, we still have candidates and subsequently use writing skills to continue sifting. Generally, we need someone who has better writing skills to improve the team performance, but possible ego effect may occur if two or three members in the eventually formed team have excellent writing skills. Thus, our goal is to maximize $W^* \cdot E$ with respect to the variable w_k.

We denote $W_f(w_k)$ as the function to find out the best third team member only considering his or her writing skills and write

$$\begin{aligned}W_f(w_k) &= W^* \cdot E \\ &= \left(\frac{w_i + w_j + w_k}{3} + \min\{w_i, w_j, w_k\}\right) \cdot \frac{\gamma}{i_{total} + \gamma} \\ &= \left(s_1 + \frac{w_k}{3} + s_2\right) \cdot \frac{\gamma}{i_{total} + \gamma}\end{aligned} \tag{3.20}$$

where s_1 is a constant and s_2 is a negligible variable given that we are looking for high writing skills.

We conclude that the higher writing skills usually make better team candidates from Equation 3.20. The influence of the ego effect can be evaluated and discussed in three cases.

- **Case 1:** If neither of chosen students have extremely high writing skills, there is no opportunity for the ego effect regardless the condition of the third student.
- **Case 2:** If both chosen students have extremely high writing skills, there already exists the ego effect and it will not worsen the situation more under the Assumption 6.
- **Case 3:** If only one chosen student has extremely high writing skills, the ego effect will occur when the third student also has them. In this case, we suggest to choose the third student with the highest score under the 90% threshold (6.7824) given the ego effect.

Remark 6 *Although we try to remain consistent and objective in the previous analysis, the difficulty to measure the ego effect and the randomness to define "extremely high" leave our results easy to be attacked. There exists, however, a general pattern about how the ego effect influences the overall performance. In the case that the number of students who have extremely high writing skills increases from 1 to 2, we need to pay more attention to the threshold since there may be a sudden decline of the team performance score right due to the ego effect.*

Due to the robust and adjacent connection between mathematics and computer programming (both higher leading to a better result, the imbalance leading to a poorer result, etc.), we decide to evaluate the third student's scores of mathematics and computer programming together. The ego effect will possibly influence mathematics and computer programming just like writing skills. Thus, our goal is to maximize $(M^* + C^* - F_{MC}) \cdot E$ with respect to the variables m_k and c_k.

We denote $MC_f(m_k, c_k)$ as the function to find out the best third person only considering his or her mathematics and computer programming scores and write that

$$MC_f(m_k, c_k) = (M^* + C^* - F_{MC}) \cdot E. \tag{3.21}$$

Notice the symmetry of Equation 3.17 and let

$$u_x = m_x + c_x \text{ where } x \in \{i, j, k\} \tag{3.22}$$

and we have

$$
\begin{aligned}
MC_f(u_k) &\doteq \left(\frac{u_i+u_j+u_k}{3} + g(u_i,u_j,u_k) - F_{MC}\right)\cdot E \\
&\le \left(\frac{u_i+u_j+u_k}{3} + |\max\{u_i,u_j,u_k\} - (\bar{m}+\bar{c})| - F_{MC}\right)\cdot \frac{\gamma}{i_{total}+\gamma} \\
&= \left(s_1 + \frac{u_k}{3} + |\max\{u_i,u_j,u_k\} - s_2| - F_{MC}\right)\cdot \frac{\gamma}{i_{total}+\gamma}
\end{aligned}
\tag{3.23}
$$

where s_1 and s_2 are constants.

We can still roughly recognize that the higher mathematics and computer programming scores usually make better team candidates through the process. The rationale to replace $g(m_i, m_j, m_k)+g(c_i, c_j, u_k)$ with $g(u_i, u_j, u_k)$ is that the difference between these terms is negligible since we are looking for the highest scores or at least the highest scores below the thresholds. However, the term F_{MC} is relatively difficult for accurate measurement, so we turn to the descriptive way in order to simplify and estimate this term.

Remark 7 *Due to the time limit, we do not have a chance to evaluate the term F_{MC} analytically or even numerically. This can definitely be improved in the future.*

Recall that mathematics and computer programming attributes are in the normal distribution under the Assumption 1. We define strong performance, median performance and weak performance as following.

Definition 2 *A strong performance is the performance scored above $\mu+\sigma$. A median performance is the performance scored between $\mu-\sigma$ and $\mu+\sigma$. A weak performance is the performance scored below $\mu-\sigma$.*

By definition, we analyze totally nine possible combinations into three cases.

- **Case 1 - low or median in mathematics or computer programming on average**
  We suggest to find the third person with the highest sum of mathematics and computer programming scores within the range of our options since we do not need to worry too much about the ego effect and the imbalance issue.

- **Case 2 - high in either mathematics or computer programming on average**
  We suggest to find the third person with the opposite skill set. This strategy can mitigate the imbalance between two attributes on the team level. It may also create an opportunity that mathematics and computer programming scores are both high on the team level, which may trigger a better result for the team performance. For instance, if two chosen students have high mathematics scores and low computer programming scores on average, the third person may better have high computer programming scores and medium or high mathematics scores.

- **Case 3 - high in both mathematics and computer programming on average**
  We suggest to find the third person with the medium or high mathematics and computer science scores. The preferable choice is median since we want to mitigate the severe ego effect. If not, we will choose high instead of low since the overall performance is still much better in choosing high even though we may encounter the severe ego effect.

We already analyze four attributes independently. In the next step, we will rank their relative contributions to the performance of the team and summarize the result in the Table 6.

Rank	Attribute
1	Writing
2	Personality
3	Mathematics
3	Computer Programming

Table 6: The Rank of the Significance of Attributes

When it comes to analyzing the multi-factor function, multipliers are more significant than adders in general, and thus both mathematics and computer programming are ranked the lowest. We rank mathematics and computer programming scores at the same level because they share the symmetry in the function and there exists a robust connection between them. As for multipliers, we rank writing above personality since not only writing skills contribute a lot more than mathematics and computer programming but also the lowest score of writing skills influences negatively much more the one of other attributes. The rank of the ego effect is undetermined since it correlates with mathematics, computer programming and writing skills.

3.2.2 The Non-Ideal Situation

Since we do not live in a perfect world and cannot always get what we want, we may end up with no one by following the previous procedure too rigorously. One way to resolve this issue is to relax some rules we made gradually in order, starting with the least significant attribute up to the most significant one. Besides, we can also conduct the control variate method to verify or correct our theories.

3.3 The Team Arrangement Algorithm (Problem c)

In this section, we need to develop an algorithm to arrange 100 three-member teams that will outperform 100 randomly-selected teams. We begin our algorithm by randomly selecting one student out of 300. Given the personality score of that student, we set up a range around that score to reduce possible large personality distances. Then, we apply the complementary strategy for mathematics, computer programming and writing skills by choosing two students within the previous range who have the mostly-compatible opposite skill sets compared with the selected student. We eventually form a team and remove those three students from the waiting list, and we repeat this procedure from the beginning until we arrange all teams.

3.3.1 The Admissible Personality Range

We still first consider the personality since it is relatively independent compared with other attributes. If teams are randomly selected, since the number of possible combinations is sufficiently large, we know that the sample distribution of the sum of the personality distances is approximately normal distribution $\mathcal{N}(15, 25)$ according to the central limit theorem, which means the majority of teams will encounter medium total personality distances under random selection. We demonstrate that personality distances can be reduced to a large extent if we arrange teams properly.

Given any student, we denote $p_i \in \{0, 1, \cdots, 29\}$ as the personality score of that person where $i \in \{1, 2, \cdots, 300\}$. If we only need to consider the personality attribute, the smallest personality distances will definitely lead to better total team performances. However, since we still need other attributes, it is more reasonable to set up an admissible range around p_i. We denote the range as r_p and summarize the results as following.

$$\text{Case 1: } p_i - r_p \leq p_j, p_k \leq p_i + r_p \qquad \text{if } p_i - r_p \geq 0 \text{ and } p_i + r_p \leq 29 \tag{3.24}$$

$$\text{Case 2: } p_j, p_k \leq p_i + r_p \text{ or } p_j, p_k \geq 30 - p_i \qquad \text{if } p_i - r_p < 0 \tag{3.25}$$

$$\text{Case 3: } p_j, p_k \leq 30 - p_i \text{ or } p_j, p_k \geq p_i - r_p \qquad \text{if } p_i + r_p > 29 \tag{3.26}$$

Since this range is adjustable, we can shrink the range if we end up with too many students or extend the range if we end up with few students.

3.3.2 The Complementary Strategy

The complementary strategy, as we discussed briefly in Problem b, is a strategy to improve overall performance by choosing teammates who have the opposite skill sets when it comes to mathematics, computer programming and writing skills. By doing that, we can possibly reduce some negative influences such as imbalance and ego effect. Recall that scores of these attributes are in the normal distribution under the Assumption 1, and thus we can find their complementary scores by symmetry and set up ranges around them. We denote these ranges as r_m, r_c and r_w for mathematics, computer programming and writing skills respectively. We summarize the results as following.

$$\text{Mathematics: } |m_i - 2\mu_m| - r_m \leq m_j, m_k \leq |m_i - 2\mu_m| + r_m \tag{3.27}$$

$$\text{Computer Programming: } |c_i - 2\mu_c| - r_c \leq c_j, c_k \leq |c_i - 2\mu_c| + r_c \tag{3.28}$$

$$\text{Writing Skills: } |w_i - 2\mu_w| - r_w \leq w_j, w_k \leq |w_i - 2\mu_w| + r_w \tag{3.29}$$

Since these ranges are adjustable, we can shrink the ranges if we end up with too many students or extend the ranges if we end up with few students.

4 Model Analysis

In this section, we will analyze the advantages, the disadvantages and possible future improvements of our three math models.

4.1 Advantages

We have analyzed all problems on the individual level and on the group level. We also developed the optimizing algorithm for our multi-factor function. Given the limited information and resources, our models can also be potentially improved when more data with less errors become available. Besides, our models feature a large number of parameters with reasonable ranges which are derived from the studies we referred and can be adjusted to different scenarios. We also simplified several complicated assumptions such as the ego effect on the team performance. Students who participate the math modeling contest can use our models as well as algorithms to search their optimal teammates promptly.

4.2 Disadvantages

Although we have put our best effort to simulate the real scenarios, the inherent limitation of our models is that our assumptions may be inaccurate given the limited information and data from the description and the data set. Besides, we have no information about how exactly four attributes will influence the team performance, and thus must make several assumptions to include those variables in our functions properly. Also, since we lack information about how our proposed teams actually perform in the real contest, we find it difficult to check and modify our model. Furthermore, we assume that three attributes (mathematics, computer programming and writing skills) are approximately in the normal distribution and the personality is approximately in the uniform distribution. That may be biased since we only have a limited sample data set which does not necessarily reflect reality.

4.3 Possible Further Improvement

If we had had more time, we could have searched for more references and studies to propose the more accurate estimation for each attribute on the team performance in the math modeling contest and we would have projected the algorithm for the optimal result with a larger reliability. If we had acquired more data and available score of their corresponding final result, we could have estimated the error rather than ignoring them. Additionally, we could have included more factors that may affect the final performance of a team. For example, we could have considered "gender" as a factor to influence the team performance. It is reasonable to predict that a team with a balanced male-and-female ratio may have a higher average performance score than that of a team with an imbalanced ratio, but we do require more information to verify that prediction.

5 Conclusion

We have solved all problems by providing three models or algorithms. For Problem a, we have developed a multi-factor function which includes all variables we need to consider, and we also write a computer program in C++ for students who are still looking for the possible best combinations based on our model. For Problem b, we have developed an algorithm by first determining the optimization of the third person's attributes one by one. Then we rank the importance of four attributes and consider the resolution under the non-ideal condition. For Problem c, we have developed an algorithm by first determining an admissible personality range and then applying the complementary strategy when it comes to mathematics, computer programming and writing skills.

A Codes

```
#include <iostream>
#include <cmath>


using namespace std;

double MF(double a,double b, double c);
double CF(double a,double b, double c);
double FF(double a,double b);
double WF(double a,double b, double c);
double dist(double a,double b);
double PF(double a,double b,double c);
double EF(double a,double b,double c,double d, double e,double f,double g,double h,double i);
double gmax(double a,double b,double c);
double gmin(double a,double b,double c);



int main()
{
    double m1,m2,m3,c1,c2,c3,w1,w2,w3,p1,p2,p3;
    double M,C,F,W,P,Y,E;
    double data[400][4];

    for(int i=0;i<400;i++) {
        for (int j=0;j<4;j++) {
            cin>>data[i][j];
        }

    int i,j,k;
    cin>>i>>endl>>j>>endl>>k>>endl;
    p1=data[i][0]; p2=data[j][0]; p3=data[k][0];
    m1=data[i][1]; m2=data[j][1]; m3=data[k][1];
    c1=data[i][2]; c2=data[j][2]; c3=data[k][2];
    w1=data[i][3]; w2=data[j][3]; w3=data[k][3];
    M=MF(m1,m2,m3);
    C=CF(c1,c2,c3);
    F=FF(M,C);
    W=WF(w1,w2,w3);
    P=PF(p1,p2,p3);
    E=EF(m1,m2,m3,c1,c2,c3,w1,w2,w3);
    Y=(M+C-F)*W*P*E;
    cout<<Y;
    return 0;
}
```

```
double MF(double a,double b, double c) {
    double d;
    if(gmax(a,b,c)>7.73) {d=gmax(a,b,c)-7.73;}
    else {d=0;}
    double e=(a+b+c)/3+d;
    return e;
}

double CF(double a,double b, double c) {
    double d;
    if(gmax(a,b,c)>7.57) {d=gmax(a,b,c)-7.58;}
    else {d=0;}
    double e=(a+b+c)/3+d;
    return e;
}


double FF(double a,double b) {
    return 0.5*abs(a-b);
}


double WF(double a,double b, double c) {
    return ((a+b+c)/3/4.97+gmin(a,b,c))/4.97;
}

double dist(double a,double b) {
    if (abs(a-b)<=15)
        {return abs(a-b);}
    else
        {return 30-abs(a-b);}
}

double PF(double a,double b,double c) {
    double p=72/(dist(a,b)+dist(b,c)+dist(a,c)+60);
    return p;
}

double EF(double a,double b, double c,double d, double e,double f,double g,double h,double i) {
    int im=0;
    int ic=0;
    int iw=0;
    double itotal;
    if((a>=7.73 && b>=7.73) |(a>=7.73 && c>=7.73)|(b>=7.73 && c>=7.73)) {im++;}
    if((d>=7.57 && e>=7.57) |(d>=7.57 && f>=7.57)|(e>=7.57 && f>=7.57)) {ic++;}
    if((g>=6.78 && h>=6.78) |(g>=6.78 && i>=6.78)|(i>=6.78 && h>=6.78)) {iw++;}
    itotal=im+ic+0.75*iw;
    itotal=7/(7+itotal);
    return itotal;
}
```

```
double gmax(double a,double b,double c) {
    int d=a;
    if (b>d) {d=b;}
    if (c>d) {d=c;}
    return d;
}

double gmin(double a,double b,double c) {
    int d=a;
    if (b<d) {d=b;}
    if (c<d) {d=c;}
    return d;
}
```

B Computation

#	P	M	C	W	Y-individual
201	0	7.4333	10	5.9775	104.2076
228	1	1.249	8.7909	6.2218	109.3904
69	1	9.2513	0.6047	5.2283	96.7371
186	2	8.5189	4.2801	4.6855	79.8306
67	2	8.5748	2.5673	4.5165	77.4562
102	3	7.3166	0.7812	6.0194	88.0831
292	4	1.4146	7.927	6.1402	97.3467
104	4	9.9075	4.1936	3.8175	75.6438
140	5	3.5294	8.4682	4.5072	76.3357
74	6	6.5509	2.9033	6.0826	79.6930
12	8	4.0015	9.8991	5.0863	100.6996
5	8	8.1709	3.0241	4.6888	76.6234
191	10	1.6381	5.897	6.4255	75.7823
110	11	9.6409	4.8725	5.7632	111.1249
27	11	7.123	1.596	7.1914	102.4487
206	11	3.3712	8.4593	4.9159	83.1701
278	13	1.2583	5.7088	6.8841	78.5999
283	14	3.0267	6.5233	5.8815	76.7336
25	15	6.9093	2.056	5.4417	75.1967
84	16	2.8113	9.64	5.4608	105.2842
268	17	6.3461	2.5361	7.3141	92.8320
274	19	5.9792	1.7904	7.4478	89.0638
99	20	3.3719	8.3171	4.6854	77.9379
120	20	6.485	2.8095	5.955	77.2364
259	22	4.6528	8.808	5.54	97.5926
203	22	2.469	6.5507	6.1313	80.3286
164	23	7.8141	8.5809	6.0154	98.6225
81	24	0.9331	8.7495	5.061	88.5624
231	27	10	3.561	4.0132	80.2640
71	28	7.4906	7.7694	5.8102	88.6637

References

[1] T. A. Judge and R. Ilies, *Relationship of personality to performance motivation: A meta-analytic review*, Journal of Applied Psychology, **87** (2002), 797–807.

[2] D. J. Sefcik, F. J. Prerost and S. E. Arbet, *Personality Types and Performance on Aptitude and Achievement Tests: Implications for Osteopathic Medical Education*, The Journal of the American Osteopathic Association. **109** (2009), 296–301.

[3] M. Standage, J. L. Duda and A.M Pensgaard, *The effect of competitive outcome and task-involving, ego-involving, and cooperative structures on the psychological well-being of individuals engaged in a co-ordination task: A self-determination approach*, Motivation and Emotion. **29** (2005), 41–68.