Algorithm Design Document

Overview

To prepare university graduates for future employment, many university courses require students to work on team projects. This helps the students develop important lifelong skills, such as teamwork, cooperation and delegation of work. A common issue that arises in this situation is teams of students who are not compatible. Some contributing factors are skill set, personalities and work habits. To help address the problem and make more compatible teams, a system called the Carleton University Project Identifier (cuPID) was proposed to be developed. For a given project, the cuPID system would organize students into compatible teams.

To be a useful system, cuPID has to be able to add and store projects and students. To ensure compatible students are grouped together, the cuPID system keeps track of student profiles and has a scientific algorithm, to match students in their groups, called the Project Partner IDentifier (PPID). The algorithm allows the administrator to set the size of each group within a project. The PPID can be run on different projects and will group the students who are registered in the project based on their compatibility and group size. The PPID uses the latest research in psychology and group dynamics and calculates how compatible students are to each other based on a set of qualification questions that the students have to answer when registering in the system.

Qualification Questions

The qualification questions in the student profile are used when calculating how compatible students, who are registered in a project, are to each other. The questions are

subdivided into four categories; personality questions, group dynamics questions, grades questions and miscellaneous. They are listed below in their category. Students answer these questions about themselves and what type of group members they are looking for. Students answer the questions by picking a value between 1 and 5 to indicate the degree to which they associate with each trait.

Personality Questions

The first set of questions is the personality questions. These questions are designed to evaluate the student's personality. The set identifies if the student is an introvert or extrovert, a logical decision maker or an intuitive one and if the student is open to criticism. The personality questions identify the student's self-perception and where the student feels his or her strengths and weaknesses are. This is an important set due to the fact that students have to learn how to identify their own traits. This will help them recognize situations and qualities where they lack and what type of group members they need to help make up for abilities they require.

The first question is designed to see how comfortable the student is presenting in front of a group. It will help find out if the student is comfortable expressing her/his ideas and can answer questions about the idea on the fly. It also shows if the student is an introvert or extrovert. This is an important distinction to be made because of how the two behave. Introverts prefer more to work separately, whereas extroverts prefer more to work in a group. Introverts would rank themselves lower on this question and extroverts will rank themselves higher. This separation will help create groups with members who are more similar to each other and have more cohesive work environments.

The second question is designed to rate if a student is comfortable taking criticism of themselves and their ideas. Students who rate higher on this will be more comfortable taking

criticism of their ideas and will probably be willing to criticize other group member's ideas. Students who rate high on this should be grouped together so the group as whole is leaning more towards students who criticize and take it or do not like to take it, but will also not be criticizing others. Students who rate themselves low on this should be grouped together so the group is more cohesive and people do not get in each other's ways.

The third question is designed to find out if the student is an emotional thinker. The higher students rate themselves on this question, the more emotional decision makers they are. The reason to group students who are more emotional thinkers together is due to how decisions are made. Emotional thinkers use certain heuristics when making decisions. Grouping people who use similar decision making heuristics will help streamline decisions compared to grouping students with different decision making heuristics. This is because there will be less time spent on justifying the reasons for a decision to other group member.

The fourth question in this set is designed to figure out how much of a logical thinker a student is. Logic is very important for computer science. Computer science and logic have a very pervasive deep connection. It is important that in a group a person can study how information is captured in sentences and how it is possible for one statement to be a consequence of another. It is important that bunch of logical students are put together in a group as an emotional person and a logical person might cause friction.

Group Dynamic Questions

The second set of questions is the group dynamic questions. These questions are designed to assess how a student works in a group. The set detects if the student is punctual, able and willing to resolve a conflict, help a member or is confident when expressing her or his ideas to the team. This set is used to help ensure group members are somewhat similar is specific traits to ensure some harmony in the group. Students can use this set to see how they work in a group environment and what type of group members they will need to ensure the work on the project is done in a timely manner.

The first question assesses a student's time management skills. Punctuality is an important part of group harmony. Punctuality is a trait that is valued differently among students. It is very important that students with good sense of time management are put together, whereas students who are more flexible be grouped together to ensure there is little friction among group members. Planning a strategy to go about designing the software and setting timely milestones is very important in a group, but how strictly those timelines are followed varies among students who view punctuality differently, therefore students who are similar in this trait should be grouped together.

The second question in this set deals with conflict resolution. When a group of students work on a project it is imminent that conflicts are going to arise. Students with good conflict management skills will likely to be more efficient when working together and not let a conflict or an argument become a setback for the group. A student with good conflict management skills might also be a good mediator when it comes to resolving a conflict between two different students. Students can use this question to judge themselves and also decide if they need a conflict resolver.

Third question in this set deals with how fluent a student is when it comes to explaining his or her ideas to other group members. Communication is a vital part of group dynamics and a student with good communication skill will fit well in a group with other students who share the same quality. Communication is also crucial for keep everyone in the group up-to-date with the progress and plans of a group. Students can use this question to judge themselves and decide if the student needs other students who are good communicators.

The last question in this group deals with student's willingness to lend a helping hand to another student who is struggling. Being helpful and co-operative to each other is necessary for a healthy group dynamics. Being supportive to one another helps foster an environment of mutual respect and strong cohesion in a group. A group consisting of cooperative members will always find a way out of any problem. Students who rank low on this may rank high on their preference to want students who are helpful to ensure when others in the group need help, someone is there.

Personality Questions
1) How comfortable are you presenting in front of an audience?
2) How well do you take criticism?
3) How much do you consider your feelings when making a decision?
4) How much of your decision making is done though logical thinking?

Group Dynamic Questions
1) How important is punctuality to you?
2) How likely are you to resolve a conflict in the group?
3) How comfortable are you stating and explaining your ideas to a group
4) How likely are you to help a struggling team member?

Grades Questions

1)	**What was your mark in 2401?**
2)	**What was your mark in 2404?**
3)	**What was your mark in 1805?**

Miscellaneous Question
What is your preferred programming language? C++, JAVA, Python

Grades Questions

The third set of questions is the grades questions. These questions show how a student performed in courses that are relevant to the project but teach different skills. The set is used as a measuring stick to track how students performed in a structured environment where pressure exists to meet deadlines and complete projects. Students can use these questions to find other group members to excel in these situations to ensure they have group members who are able to follow instructions. It will also help identify to the students that they may have shortcoming in these situations and to be mindful of this when working with her or his project group.

The first question in this category deals with a student's grades in 2401. Comp 2401 was one of the most important courses in computer science. If a student has done well in Comp 2401, it's highly likely that this student has a very good knowledge and understanding of proper memory management, pointers, help and stack and also different data types. Students that have good understanding of these concepts can work together in efficient planning and implementation of a software design.

The second question deals with student's grades in Comp 2404. Comp 2404 is an important course because students who have done well in this course have command over the skills needed to build large, efficient and reusable object-oriented systems. Group of students

working together who are good at these skills will be able to set goals, perceive problems in terms of programming and eliminate risk of setting unachievable milestones.

Third question in this set deals with the grade a student has achieved in Comp 1805. Discrete math is very important for computer scientists. It gives a student various tools to work on a given problem. This course pushes students to look outside the box when solving a problem, while being very eloquent and descriptive about your steps to do so. Students who have done well in 1805 will surely demonstrate concise and logical thinking and a group working together with these skills will be able to break-down the steps needed to solve a problem and at the same time make sure each step is optimized.

Programming Preference

The twelfth question is used to find what programming language the student feels comfortable in and what programming language the student wants her or his group members to be comfortable in. This will help bring consistency and cohesion to the group.

These four sets of questions clarify the skills and traits the student possesses and where the student lacks. This, in turn help the student find group members who are similar in some aspects to ensure there is compatibility in the group. It will also make sure some variety exists to safeguard against the entire group lacking a key component to finish the project on time and well. Once the questions are answered, the system calculates the totals of each set to be used by the algorithm.

Algorithm

The algorithm used to divide the students into group takes into account the traits, personality and skills of the student and what the student is looking for. Administrators can set group sizes and run the PPID on a project. The algorithm can also be changed to the administrator's preferences of which set of questions are given priority over the others.

At the PPID Edit page, an administrator can rank the sets of questions from 1 to 4. The algorithm starts with the range of ±3 that is used when evaluating how compatible students will be. At any point, if the algorithm cannot narrow down students, the range will be increased by 1 to help group students. The system will then start the sorting algorithm by grouping students, starting for the highest ranked set of questions to the lowest ranked set of questions.

The PPID starts by looking at the number of students registered in the project and the group size provided by the administrator. It checks if all the students in the project can be evenly divided into groups of the size provided. If not, the system notifies the administrator of the situation and asks for the confirmation to continue or to choose a new group size. If the administrator chooses to continue, the algorithm runs normally, but takes the students in the last group, which will be the uneven group, and puts each student in a different group to ensure groups are more even and one group with few members is not given an unattainable project.

Once the administrator has defined the parameters, the PPID starts with the first student that is registered in the project. It checks which set is given the highest preference by the administrator and retrieves from the first student the totals of the personal score and the partner score of the highest preferred set. For example, if the administrator chooses the Group Dynamics Questions set to be of the highest priority, the system will retrieve, from the first student, the total

of the Group Dynamic Questions answered about themselves and the total of the Group Dynamic Questions answered about the partners she or he is looking for. The algorithm then compares the student's partner score total to totals of other students registered in the project and the student's personal score total with the other students' partner score total for the same set. The algorithm uses the range, which by default ±3, when comparing totals to ensure students are grouped with other students who are similar to what the student is looking for, if not perfect.

The system makes a list of all the students who are closely compatible with the student registered, if there is more than one. Then it takes the list and reduces it further using the second highest preferred set of questions assigned by the administrator. This procedure is continued until the last preferred set of questions is used to compare the scores for the grouping of students. If, after the completion of the lowest priority set of questions, the students narrowed down and selected while comparing the scores with the first student, is equal to the group size assigned by the administrator, then the algorithm will make a group of the selected students for that particular group size. If the students selected or narrowed down with the algorithm are less than the group size assigned, then a group of the selected students will be made, an average of all their combined scores will be calculated for the highest preferred set of questions assigned and later compared with the remaining students registered in the project, for that particular set of questions. In any case where the students narrowed down or selected are greater than the group size assigned, the first student in the list will be picked, this allows the same groups to be made every time the algorithm runs. If at any point, while the algorithm is being run, the narrowed down list of students is 0, the algorithm increases the range and starts over again. A student is only added to the group if that student passes all 4 sets of questions to ensure compatibility. Once the following steps are finished, the grouped students are removed from the list of student for

that particular project. These steps are later repeated for the remaining students until perfect groups of the assigned group size are made. The remaining students in the end will be grouped together, once all other groups are made. If the last group made has a size that is less than or equal to 50% of the assigned group size, each student in the last group will be assigned to another group, with the first student being assigned to the first group, second student to the second group, etc.

PPID Example

This is an example for 8 students. The administrator has chosen to set group size to 4. 8 % 4 = 0, therefore even groups can be created. The administrator has chosen to given priority of the sets of questions in the order of 3, 2, 1, 4. The following tables provide the total for each student in each set of questions as well as their personal and partner score.

Student 1	Personal Score	**Partner Score**
Set 1	17	19
Set 2	19	15
Set 3	13	11
Set 4	2	2

Student 2	Personal Score	**Partner Score**
Set 1	14	20
Set 2	18	16
Set 3	13	11
Set 4	2	2

Student 3	Personal Score	**Partner Score**
Set 1	10	13
Set 2	9	18
Set 3	11	11
Set 4	1	2

Student 4	Personal Score	**Partner Score**
Set 1	16	15
Set 2	14	14
Set 3	8	5
Set 4	3	3

Student 5	Personal Score	Partner Score
Set 1	8	13
Set 2	10	16
Set 3	15	9
Set 4	1	2

Student 6	Personal Score	Partner Score
Set 1	18	14
Set 2	15	16
Set 3	9	6
Set 4	2	3

Student 7	Personal Score	Partner Score
Set 1	19	20
Set 2	18	17
Set 3	14	8
Set 4	3	1

Student 8	Personal Score	Partner Score
Set 1	14	20
Set 2	19	16
Set 3	13	11
Set 4	2	2

As per the rules, we start with the first student in the project, it is Student 1. Since the administrator has given set 3 the highest priority, this is where the algorithm will start. Student 1 is looking for someone who has scored an 11 for set 3; therefore the range used to asses other student's personal score is 8-14. According to the algorithm, Students 2, 3, 4, 6, 7 & 8 fit the criteria for personal score for student 1. Out of which student 1's personal score falls within

range for the partner scores of students 2, 3 and 8. Since there is more than one choice, the algorithm moves on to the next preferred set of questions assigned by the administrator, in this case it is set 2. For set 2, the range for what student 1 is looking for in a partner is 12 – 18, therefore student 2 is the only student that falls within this range from the previous students narrowed down. Student 1 also falls within student 2's partner range, thus creating a group with student 1 and 2.

Students 3, 4, 5, 6, 7 and 8 are left in the group. To find the other group members, the system will take the current group members, add up their priority 1 sets score and divide by the number of students in the group. In this case, Student 1 has scores of 13, 11 & student 2 has 13, 11. So the system will do (13+13)/2 = 13 and (11+11)/2 = 11.

Now the recently made group will look for someone whose personal score falls in the range of 8-14. Students 3, 5, 7 and 8 fit the criteria. So the algorithm will move to the second priority, which is set 2. Again, we take the average, (19+18)/2 = 18.9, round up to 19 and (15+16)/2 = 15.5, round up to 16. So for the second criteria we look for someone who has scored 13-19. Here when comparing scores the algorithm will find that students 5, 7 and 8 fall within student 1 and 2's average partner score ranger and student 1 and 2's average personal score also fits their respective partner range score of ±3.

Since it narrows down to more than one student again, we need to repeat the same iterations as done above. Later, when considering comparison for questions in set 1, we find that even though students 7 and 8 fall within partner score range for student 1 and 2's average (17 – 20), but in return student 1 and 2's average personal score (16) does not fall within range for students 7 and 8 partner score.

Student 8	Personal Score	Partner Score
Set 1	14	20
Set 2	19	16
Set 3	13	11
Set 4	2	2

Student 7	Personal Score	Partner Score
Set 1	19	20
Set 2	18	17
Set 3	14	8
Set 4	3	1

Here the algorithm with dynamically increase the range from ±3 to ±4 and repeat the last iteration again. This way it will allow the average partner score for student 1 and 2 to be 16 – 20 allowing students 7 and 8 to fall within range like last time, but after the following alteration the average personal score of 16 for students 1 and 2's will also fall within range for students 7 and 8 allowing algorithm to continue to the last iteration for set 4. After this last iteration students 7 and 8 will be grouped with 1 and 2 since looking at the score comparison for the last question set (Set 4) students 7 and 8 fall with the score ranges and the average score for 1 and 2 fall within range for students 7 and 8's partner score as well. Resulting in making a group of 4 students (1, 2, 7 and 8) leaving behind 4 students (3, 4, 5, 6) forming another group.

Conclusion

The algorithm we have proposed and will use in our system will take into account the preferences of the administrator and the students. It groups students, based on a range provided by the administrator, by looking at the student's personal and partner score in each question set and comparing it to other students to find compatible partners.

The algorithm also ensures groups of even sizes can be made or notifies the administrator if this is not possible. The administrator can choose to run it and have a few uneven groups or can resize the groups to ensure even groups.

Team work is an essential life skill, required in almost every work environment. Learning this skill in school will go a long way in making graduates more attractive to employees. This system will make cohesive groups so students are grouped with somewhat similar people to finish projects and help facilitate in learning this life skill.