



# Team Assignment Algorithm- overview

## Project #35 (Skills based assignment of capstone project teams)

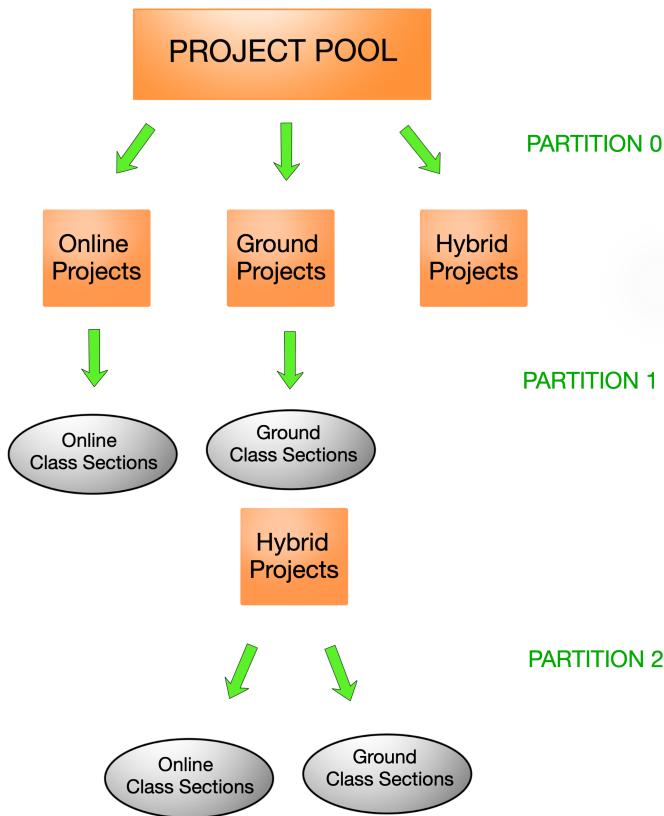
The GUI application for the team assignments system is launched. The user must enter certain information, and class section, and student data is collected directly from canvas using the ASU instructor's login access.

Project proposal data is read in directly from a csv file, that the user provides.



ASU SER Instructor access GUI application

## Projects to Class Sections



First, we sort the projects based on type, Online Ground or Hybrid, as well as by project priority.

Then we will partition the online projects to the online Class Sections, (based on percentage score) until there are no online projects left.

The same is done for the Ground projects to ground Class Sections, until all the ground projects are assigned.

Last we partition the hybrid projects, starting with the highest priority projects, assigning them the best class section (based on percentage score). Also taking into account the minimum required projects for each class section. Once each class section has the required amount of projects, the rest of the projects will remain unassigned. (The projects to class section assignment only assigns the minimum number Of projects that each class section requires. This is based directly off of the number Of students in that class section.)

We assign the projects to class sections one at a time until all the class sections have been assigned a project. So if there are 4 class sections, We assign the best project to class section match, and pull that one out. We then do the same for the 3 remaining, then 2, then 1, and start back over until All the projects have been assigned.

	Class Section 1	Class Section 2	Class Section 3	Class Section 4
Project 1	88	92	36	100
Project 2	70	73	60	90
Project 3	45	99	45	78
Project 4	56	89	39	98
Project 5	91	55	70	85
...	...	...	...	...

### PERCENTAGE MATRIX

We calculate a percentage score of how well the students in a Class section compare to each project. This is based on the aggregate of the student's skills in a class section, Compared to the project's skills. This is used to determine which class section a project Would be most suited for.

# Assignment of Students to Project Teams

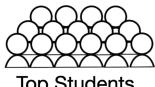
For each Class section,  
3 calls to the Project Team assignment function  
will be made on separate Threads.  
This will keep the runtime down  
Roughly equal to the time it takes one  
Class section to be done.

We use the 3 calls as way to lower  
the number of students and projects assessed.  
If the number of projects is 30 or less, we make sure  
That each call has no more than 10 projects, and no  
Less than 2. If there are 30 or more projects assigned to  
A class section, the projects will be evenly distributed.

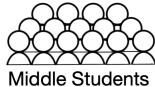
Students will be partitioned based  
on their skill averages,  
and the number of students in that partition  
will relate to the number of projects selected  
for that priority.

(If 10 high priority projects,  
the top 50 students are selected).

Highest Priority Projects



Middle Priority Projects



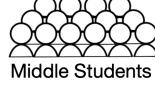
Lowest Priority Projects



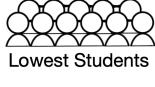
Highest Priority Projects



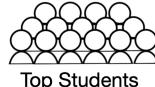
Middle Priority Projects



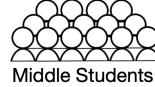
Lowest Priority Projects



Highest Priority Projects



Middle Priority Projects



Lowest Priority Projects



Thread 1  
Class Section 1

Thread 2  
Class Section 2

Thread N  
Class Section N

## Project Team Assignment Algorithm

All team combinations  
For Project 1



Team: 1



Team: 2



....



Team: N



Top 5 Teams

All team combinations  
For Project 2



Team: 1



Team: 2



....



Team: N



Top 5 Teams

All team combinations  
For Project N



Team: 1



Team: 2



....



Team: N



Top 5 Teams

Then we will find the best  
project to team set for all the projects  
based on the top 5 teams for every project.

We will search for the highest scoring project  
set with all unique students. If one is found,  
then we will assign all those student teams  
to the respective projects.

If no set is found with unique students,  
we will find the highest scoring project set  
with the least amount of duplicate students.

We will then swap out the duplicate students  
with the best unassigned replacement student  
from the list, (starting with the lowest scoring  
teams first, until there are no duplicate students).

## Duplicate Student Swapping

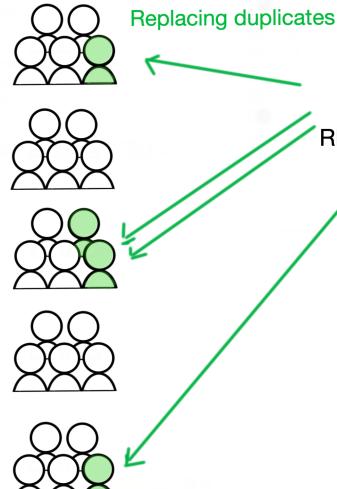
Once we get the set of teams to projects that has the least amount  
of duplicate students, we then begin to swap out those students  
With students in the remaining pool who have not been assigned yet.

We iterate over the list of teams, and once we come across a duplicate  
student, we swap out that student with the best replacement student.  
The best replacement will be the highest scoring student in the reserve  
for that particular project. The team is reassessed for negative affinity, and  
If there is, then we move onto the next student.  
If not, that student is now assigned.

DUPLICATE STUDENT

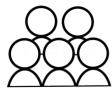
Replacing duplicates

RESERVE STUDENT POOL

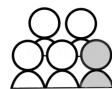


### TEAM SIZE OF 4

We have also taken into consideration having team sizes of 4.  
So depending on the number of teams of 4 that will be needed  
(depends on the original number of students)  
we will swap in a fake student into the required number of teams.  
So all the teams will have 5 students, but the teams of 4 will have 1  
Fake student.



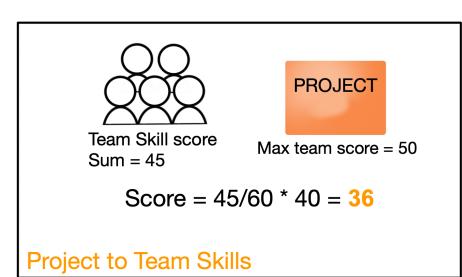
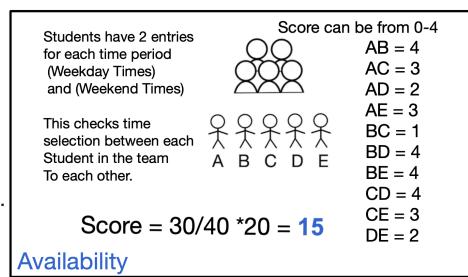
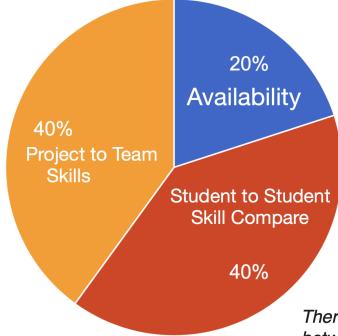
Team of 5



Team of 4

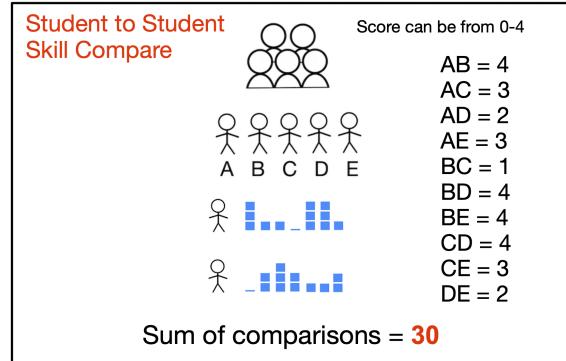
## Team Scoring

This all depends on giving a team a quantifiable score depicting the quality of that team assignment.  
 This will be a 3 part scoring mechanism.  
 So for the final team score, a team can score 0-100.  
 20% of that will be availability  
 40% of it will be student to student skill comparison for the team.  
 40% will be team skill to project comparison



**Team Score = 15 + 30 + 36 = 81**

*There is a negative affinity check that makes sure there is no negative affinity between any of the members on the team. A team with negative affinity will not be assessed at all, and the system will move onto the next team combination.*



Drawn By: Myles Colina