A Greedy Algorithm Assignment of Capstone Course Students to Teams and Projects Using Skill Heuristics

Robert D. Plumley and Charles C. Tappert

Seidenberg School of CSIS, Pace University, Pleasantville, New York
rplumley@pace.edu, ctappert@pace.edu

Abstract
Collaborative project work is an increasingly prevalent pedagogy across nearly all academic and vocational disciplines. However, there are many variables involved in the formation of teams when choosing from a pool of students with widely varied skills and levels of experience, leaving many instructors no choice but to use time consuming superficial grouping methods or simply allow students to self-group for team projects. This paper describes a comparative analysis of the skill distributions of teams formed conventionally and teams formed programmatically, using target skill distribution heuristics and a greedy assignment algorithm. Test populations for this comparison came from two courses that utilize group project work as the primary course deliverable – an information systems graduate Capstone Project course of 40 students fall 2015 and 60 students spring 2016, and an Introduction to Computing for Non-Computing Majors undergraduate course of 105 students fall 2015 and two sections spring of 2016 totaling 160 students.

1. Introduction
In an ever evolving professional work environment which is skewing increasingly toward freelance contactors and remote workforces, the ability for newly minted graduates to effectively communicate with unfamiliar team members and contribute positively to group projects is a must [4]. Project Based Learning (PBL) courses are a means by which many higher education institutions are instilling these skills in their students while also increasing the overall effectiveness of the course itself with respect to increasing student competence in the respective skills targeted [5]. However, in most academic settings the number of students and/or a lack of information about the students in a class prevent any means other than random assignment or self-segmentation from being utilized because other methods such as manual assignment by the instructor assignment are time prohibitive [2].

Although, random team assignments may provide benefits with respect to simplicity and avoiding the formation of cliques there is the risk that teams may be unwittingly being setup for failure if a randomly selected team does not possess team member skills required to properly execute on the project. To address these skill-related concerns, we developed a program that collects the customer’s expectation of the skills needed to complete a project and the skills of the potential team members, and then uses a scoring heuristic and a greedy assignment algorithm to produce teams that should be a better fit for the project’s needs than a randomly selected team.

This paper provides an analysis of five unique team assignment approaches where the resulting team skill heuristics and their fit to their respective projects’ needed skills are compared. These approaches are evaluated with data from two courses.

1.1. Capstone project course
At Pace University the Computer Science and Information Technology Capstone Project course is a semester long group project based course. Those enrolled in the capstone course are tasked with working with teammates and project customers to conduct technical research and/or develop customized technical solutions to address the customer’s specified problem. Group members are expected to meet with customers to evaluate product specifications, provide weekly updates, make mid-semester and end-of-semester presentations, and ultimately provide a deployable solution and user manual [3].
1.2. Computing for Non-Computing Majors

Introduction to Computing for Non-Computing Majors (CIS 101) at Pace University is a three credit course designed for freshman and sophomore non-computing major undergraduate students. With both personal assignments and a semester-long group project, the course maintains two parallel tracks throughout the semester where the students’ individual assignments are designed to prepare them to contribute to the group project deliverables. The primary focus of the first half of the course to teach students the concepts of variables, functions, and conditional statements using Microsoft Excel for the eventual development of personal and project budgets. The second half of the course focuses on web design using HTML, CSS, and JavaScript, with the development of a project webpage as the final goal [1].

2. Methodology

The primary goal of the software program is to optimize the grouping of students into teams. Currently, the Capstone Project course instructor reviews students’ skill assessment surveys and assigns students to projects, while the Introduction to Computing for Non-Computing Majors course instructor allows the students to form the teams.

2.1. Data collection

Collecting data with a consistent format and data range are essential components to the program, allowing it to score student skill levels and assign students to groups of evenly distributed skill levels or matching project needs.

Capstone project skill needs assessment

Utilizing an online survey, capstone project customers were prompted to rank twelve areas of expertise and nine computer programming languages with respect to their importance to the successful completion of respective project, using a 0-10 scale with zero meaning “Not Applicable” and ten “Mission Critical”. Project customers were also asked to indicate the minimum and maximum number of team members they anticipate needing for their project.

Capstone project student survey

Also utilizing an online survey tool, the students enrolled in the capstone project course were asked to provide a self-assessment, ranking their level of competence for the same areas of expertise and programming languages that the project customers provided, using a 0-10 scale with zero indicating no experience and ten expert level knowledge. Additionally, the students were asked to select their top five project choices. Pages 1 and 2 of the survey collected personal contact information and communication preference data not used in team assignment, however pages 3 and 4, seen in Fig1 and Fig2, contain the critical data used for the team assignment algorithm such as students’ technical self-assessments and project preferences.

---

**Project Preference and Skills**

*Required*

**Skills and Programming Languages:**

In this section you are asked to give your honest self-assessment of your experience level with the skills and programming languages listed.

**Skills:**

Rank your skill/experience level from 0 to 10 for each of the following topics: Zero being little to no experience, 10 being expert level knowledge.

**Languages:**

Rank your skill/experience level from 0 to 10 for each of the following languages: Zero being little to no experience, 10 being expert level knowledge.

---

**Fig1. Capstone Student Skills Survey – pg3.**
Manual instructor assignment. The Capstone Project course teams used for this analysis were the result of manual assignment by the instructor, an expert with over 15 years of experience teaching the capstone course. To further inform the project assignments students of the course were asked to provide a one-page survey in the form of a Word document where students were asked questions regarding their location, years of experience and skills, as well as their preference for projects from a list provided by the instructor (Fig. 1). Those documents were then evaluated and team assignments made manually by the instructor based on his knowledge of the projects’ needs and the preference of the students.

Fig. 3. Student Information Preference Form.

Team member self-selection. The section of Introduction to Computing that was used for this analysis was prompted to self-segment into groups of five, with students who were absent being assigned to existing teams as a sixth member. Since the teams were formed early in the semester and the vast majority of the students were freshman, very few intrapersonal relations existed prior to team formation, leaving general proximity in the class room as the primary factor used for group formation.

2.2.2 Programmatic assignment process

The program for systematic team member assignment utilizes a greedy algorithm approach by which a student’s fit for each of the respective projects was scored and the highest ranking students were assigned to projects by looping though the list of projects sorted in ascending order.

Fig2. Capstone Student Skills Survey – pg4.

Introduction to computing student survey

Finally, a third online survey was deployed to the students of the Introduction to Computing class. These students were asked to rank their level of experience for a list of knowledge domains, using the same no experience to expert level knowledge, 0 to 10 scale. These students were not asked for their project preferences because there are not predetermined projects for the course, instead each team is tasked with developing their own unique project idea.

2.2. Team assignment methods

In total five methods of team creation were analyzed, but only four methods were applied to each respective course. Both courses had pre-existing teams created manually by different methods, in the case of CIS101 self-selection was used while the professor created the teams for the capstone course. Additionally, the two programmatic approaches and a truly random method were used on both courses.

2.2.1 Manual unsystematic assignment processes

Teams were created by manual processes for both fall 2015 course populations.
by minimum team members needed. Once each project has reach its minimum team member needs, the process is continued for the remaining students using the maximum number of team members for each project.

This adjusted percentage array is used to dynamically recalculate the student pools’ project fit scores before assigning new team members. Then project preference multipliers seen in Fig. 5. Project Preference Multipliers, are applied to the new project fit scores.

Team leader scoring and assignment. Although the Intro to Computing course does not need a declared team leader in the case of the Capstone project course the first member assigned for each project is the team leader. To qualify as a potential team leader the student must be able to attend three on campus meeting and have experience in the tech industry.

Qualifying students are then given a team leader score on a scale from 0 to 25 based on their self-assessments of their leadership and organizational skills, availability, and years of experience where up to 10 points is granted for every year of experience. Qualifying students then have their project fit scores increased by up to 50% based on the student’s leadership score relative to the minimum and maximum leadership scores of all qualifying students.

<table>
<thead>
<tr>
<th>Leadership Score Category</th>
<th>Max Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leadership Assessment</td>
<td>5</td>
</tr>
<tr>
<td>Organization Assessment</td>
<td>5</td>
</tr>
<tr>
<td>Availability</td>
<td>5</td>
</tr>
<tr>
<td>Years of Experience</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>25</td>
</tr>
</tbody>
</table>

Leadership Multiplier Formula

\[ \text{Leadership Multiplier} = 0.5 \times \left( \frac{\text{Student Leader Score} - \text{minLS}}{\text{maxLS} - \text{minLS}} \right)\]

Fig. 6. Leadership score.

The greedy assignment process is then used to fill the team leader role by selecting the leadership qualified student with the highest project heuristic score, for each project sorted in ascending order by the minimum number of project team members, ensuring that the smallest teams get the best fitted, most experienced leaders.

Additionally, because the team leader is the first team member assigned to a project, both the static and dynamic heuristic scoring methods result in the same team leader assignments, with the skills of the team leader thus having a significant impact on the dynamic scoring of the remaining team members to be assigned to a project.

Random assignment

To produce truly unbiased benchmarks for comparison both courses’ student populations were randomly assigned to groups. Each student was given a random score from 1 to 10 for each project which were then used by the greedy assignment process.

2.3. Assignment method comparison & scoring
Two metrics were used to measure and compare the relative success of each assignment method applied to the Capstone Project. The frequency with which a team creation method generates the highest cumulative team score for a respective project after the scores are adjusted to zero out skills indicated by project customers as unneeded. For the second success metric the Euclidean distance between the desired skill distribution and that of the resulting teams were measured and compared with the goal of generating the lowest distance possible.

For the Introduction to Computing course, the goal heuristic is a perfectly balanced team where skill rankings are distributed evenly to ensure a variety of skills are present on each team. For the capstone course the goal is to create teams that closely match the desired/needed project skills to help ensure a successful project.

3. Results

With respect to the capstone course, it became apparent that the availability of student expertise is the largest hurdle to overcome when assigning students to teams with a specific mix of skills and knowledge as the final goal. When multiple projects have ranked the same skills as high priority, the ability to allocate limited skill resources to both of them in the enough concentration to meet the goal heuristic is become challenging.

3.1. Capstone project course – Fall 2015

Student survey responses

Below in Fig. 7 and Fig. 8 are the frequency with which students of the capstone course described themselves as a particular skill level for a respective skill or programming language.
Team creation method comparison
In Fig. 11 the frequency with which a respective team creation methodology generated the highest scoring team for a particular capstone project is presented. Although with such a small number of groups no result can be considered definitive, the table shows that the current teams created manual by the professor outrank those teams of other creation methods more often. Also of note is the two-way tie for project 10 between the static and dynamic heuristic methods as both methods selected the same team leader for the one-person team.

<table>
<thead>
<tr>
<th>Team</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>Max Ct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Static</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Dynamic</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Random</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 11. Frequency of maximum team skill score.

In Fig. 12 and Fig. 13 the Euclidian distance between each project’s targeted skill and language heuristics and the skill distribution of each team creation method are compared. Here it is seen that the process of using a team’s current skill levels to dynamically adjust the scoring heuristic applied to the pool of potential team members generates teams that more closely match the goal heuristic a significant portion of the time.

<table>
<thead>
<tr>
<th>Team</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>0.22</td>
<td>0.25</td>
<td>0.26</td>
<td>0.25</td>
<td>0.27</td>
<td>0.29</td>
<td>0.25</td>
<td>0.36</td>
<td>0.34</td>
<td>0.37</td>
<td>2.86</td>
</tr>
<tr>
<td>Static</td>
<td>0.19</td>
<td>0.35</td>
<td>0.28</td>
<td>0.20</td>
<td>0.33</td>
<td>0.32</td>
<td>0.25</td>
<td>0.26</td>
<td>0.22</td>
<td>0.18</td>
<td>2.57</td>
</tr>
<tr>
<td>Dynamic</td>
<td>0.14</td>
<td>0.35</td>
<td>0.27</td>
<td>0.20</td>
<td>0.26</td>
<td>0.27</td>
<td>0.26</td>
<td>0.24</td>
<td>0.29</td>
<td>0.18</td>
<td>2.44</td>
</tr>
<tr>
<td>Random</td>
<td>0.31</td>
<td>0.31</td>
<td>0.22</td>
<td>0.25</td>
<td>0.34</td>
<td>0.34</td>
<td>0.31</td>
<td>0.24</td>
<td>0.19</td>
<td>0.88</td>
<td>3.38</td>
</tr>
</tbody>
</table>

Fig. 12. Euclidian distance of skill level distribution vs goal heuristic.

<table>
<thead>
<tr>
<th>Team</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>Max Ct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Static</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dynamic</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Random</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 13. Euclidian distance of programming language distribution vs goal heuristic.

3.2. Introduction to Computing – Fall 2015
Student survey responses
As would be expected for an introductory course, the CIS section surveyed had a very limited number of student who considered them-selves proficient in the domains of knowledge the course is designed to address. As seen in Fig. 14 and Fig. 15, respondents reported very low levels of exposure to Web Design, HTML and CSS, and only moderate knowledge of Microsoft Excel.

Fig. 14. CIS 101 Student skill ranking frequency.

Fig. 15. Average CIS101 student skills assessment scores.

Team creation method comparison
As can be seen in Fig. 16 manual team creation, in the case of CIS the students self-selected to form project groups, once again has the highest frequency of maximum team skills scores in comparison to the other team formation models.

<table>
<thead>
<tr>
<th>Team</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
<th>P15</th>
<th>Max Ct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 16. Frequency of maximum team skill score.
### 3.3. Spring 2016 System Evolution

#### Prototype Testing

At the start of the Spring 2016 semester the system prototype was used to create the teams for one capstone section and two CIS101 sections. Utilizing the online surveys described in section 2.1 customer and student inputs were collected and processed to assign students to teams using the dynamic project needs heuristic assignment method described in section 2.2.2.

The use of the assignment system allowed each class’ team creation process to take less than 30 minutes to complete with much of that time being spent dealing with erroneous student inputs. As the semester is still underway, comparative analysis of team performance changes from Fall 2015 to Spring 2016 is pending.

#### System Improvements

Development efforts in the Spring 2016 semester focused primarily on the creation of a more robust user interface to improve usability and input integrity.

**Manual Override.** The first use case considered was the need for the instructor to manually override a student’s systematic team assignment. To address this use case an interface was created to allow the instructor to search for students by name, which would then display the student’s current team. The students team could then be changed via a drop-down list of the semester’s other projects.

**BlackBoard Interactions.** Blackboard is the learning management system (LMS) used by Pace University and is the primary means used by most instructors to access a class’ enrollment roster. As such, efforts to streamline the assignment system’s setup and updating processes required the development of methods to parse student information exports from BlackBoard. The efficiency of this process is particularly meaningful during the first few weeks of the semester when students are allowed to add and withdraw from courses, and such changes require team memberships to be updated accordingly.

Additionally, an export function was added to the assignment system to enable instructors to easily create groups and assign groups members in BlackBoard via its upload feature.

**Survey Updates.** In an effort to increase the data integrity of user inputs a new form built in MS Excel was developed to replace the online surveys used in the last two semesters. Although the use of online surveys proved convenient initially, it became apparent that the exceedingly dynamic data validation and user prompting options that could be built in Excel using formulas and conditional formatting could be used to reduce erroneous user input.

The majority of user input errors observed related to the input of the student’s email username which is used as a unique identifier by BlackBoard. Common email username errors include the forgetting or switching of characters and most common the inclusion of the email’s domain. The use of an Excel file as the means of data collection also allows the opportunity to use a script to provide each student with a customized survey file, pre-populated with the student’s unique identifiers.

### 4. Conclusions

Although manually formed groups have proven to be just as effective and diverse in their skills distribution as the informed selection algorithms tested, the process of forming groups manually can be very time consuming and exposes student to a number of potential issues driven by clique behavior among class goers. Both of these issues are addressed by the use of a well-informed algorithmic assignment system, which can cut instructors’ team creation efforts dramatically and puts a greater emphasis on student skills over student relationships.

Additionally, when the goal is to form project teams that are well matched to the unique mix of skill and expertise levels needed to complete the project, the utilization of system to analyze the student pool’s skill sets and programmatically assign those students to projects will yield teams that are more closely aligned with a project’s needed skill mix.
5. Future Work

With respect to the further development of the assignment system, future work will include the incorporation of team member availability window matching and customer office proximity into the primary, skill based project assignment algorithm.

References

[1] Reicher, Daniel, and Courtney, Mary, “Updating the CIS 101 Course in the Seidenberg School of CSIS”, Proceedings of Student-Faculty Research Day, Publisher, Pace University, May 1st 2015, pp. D2.1-D2.10.


