

Optimizing Capstone Team Selection

Dr. B. Matthew Michaelis, Eastern Washington University

Matthew Michaelis is an Assistant Professor of Mechanical Engineering and Mechanical Engineering Technology at Eastern Washington University in Cheney, WA. His research interests include additive manufacturing, advanced CAD modeling, and engineering pedagogy. Before transitioning to academia, he worked for years as a design engineer, engineering director, and research scientist and holds MS and PhD degrees from University of CA, Irvine and a B.S. degree from Walla Walla University.

Dr. Heechang (Alex) Bae, Eastern Washington University

Assistant Professor Mechanical Engineering/Mechanical Engineering Technology Program Department of Engineering & Design

Optimizing Capstone Team Formation

Abstract

For senior capstone teams, team composition is one of the primary factors in student satisfaction and project success. Previous team formation were done manually after students submitted their top five choices from the available projects and were time consuming and ineffective. To improve team composition and reduce formation time, mixed-integer linear programming is utilized to optimize the team formation process. The presented approach allows control of team size (individually and globally), the number of total teams, assignment of specific students to a particular project, and whether a specific project is a “Go” or “No-go”. An online survey is used to collect each student’s top choices and the results are used to select the projects that will move forward and optimize the team compositions based on student preference. To accomplish this, a team formation score is calculated for possible sets of teams by quantifying the average student “happiness” as measured by how close students get to their first choice. Many possible team compositions are then tested until the optimum teams and projects are found. In application, with class sizes of 60 or 30 students and an average team size of 6, the algorithm was utilized in eight capstone sections to form 40 projects teams using 230 students. Of these students, 74% got their first choice and 94% got one of their top 3 choices. Another key result is a reduction in team formation time from 2 days or so down to less than 1 hour.

Introduction

It was my first day of Senior Capstone during my second quarter of full time teaching and we were faced with a room full of approximately 80 senior Mechanical Engineering students buzzing with the excitement of a new quarter and the realization that they were getting close to the end of a long, and often challenging, journey. Our primary goals for our Senior Capstone course are to give students an engineering project/job like experience that builds on the knowledge/tools/experiences they’ve gained and further develop their teamwork skills. The task for the course is to select projects and form teams; but how do you divide students into teams that will have a decent chance at success?

For that first course, students were allowed to self-select teams and projects. Several proposed projects already had industry sponsors and some students had put some previous thought into capstone and came prepared with projects and preferred teammates. Others, however, formed teams first and then came up with a project during the first week or so of class. While this semi-ad hoc approach worked out (in spite of the initial chaos) for many of the projects, upon reflection it was clear that there was room for improvement.

The following year, potential projects were all found and screened before class began and a project interest survey was introduced through which students could communicate their project preferences by selecting their top five projects. This gave the instructors an idea of what students thought they’d like to work on but then there remained the arduous task of taking that data and constructing project teams. A few teams were simple/obvious since the exact number of students needed selected those projects as their top choice. However, several projects were very popular

and had twice the number of 1st choices as needed so it was clear that a decent number of students were definitely not going to get their first choice. The last, and most challenging projects had mostly 2nd, 3rd, or even 4th, 5th, or no student selections. Forming teams for these “left over” projects was challenging and time consuming and then were typically the most challenging for students, instructors, and sponsors throughout the capstone course. Thus, while the student interest survey was effective, the manual team formation method proved cumbersome, time intensive, and didn’t generate the optimum teams. Therefore, there was an opportunity to further improve the team formation process.

In past work, Brickell [1] linked common interests with better grades and showed self-selected groups to be less effective and Bosco [2] showed that student-selected teams performed better than random, instructor-selected ones. Team formation algorithms have been developed but were limited to meeting times or with the goal of forming the best single “superstar” team possible as opposed to forming the best set of teams having a similar chance of success from a large pool of students (which is our goal) [3]. Academic teams are also somewhat unique since they are not formed using people with established relationships (e.g. coworkers); but they aren’t complete strangers either since many have been taking classes together for several years. Further, many of the developed algorithms have not been used in practice [4].

It’s definitely been our experience that students who are excited about a project have better outcomes and Bricknell’s [1] findings support the interest survey approach since students are grouped based on having a common interest in one of the projects. In related work, Aller [5] explored the mingling method as “a way to enhance the likelihood of a common interest” when selecting teams but it is not utilized for this investigation.

The primary goals of capstone is to achieve high quality student projects (and the resulting high grades) and have the course be a fulfilling, growth achieving, meaningful, and potentially enjoyable experience. Since team composition is a significant factor in achieving these goals, we decided to improve the capstone project team formation process and work toward the following goals:

1. Optimization of capstone teams based on student project preference
2. Quick and efficient team formation (less than one day)

To achieve these goals, an optimization algorithm was implemented for several capstone sections that maximizes average student happiness when measured by how close students get to their first choice. A five choice student interest survey was still utilized since it’s been our experience that unmotivated students have poor capstone experiences and the student interest survey is an effective method for determining how motivated a student is for each available project. Further, a reduction of team formation time enables students to quickly focus on their projects. A basic assumption of our approach is that students are good at predicting their optimum project and it doesn’t account for whether other students interested in similar projects would make good teammates.

Application/course background

The mechanical engineering capstone course at Eastern Washington University is a required, senior level course that represents the culmination of student's engineering education experience for mechanical engineering and mechanical engineering technology students. Students are required to complete an engineering project with a team of their fellow students and the course is spread over two consecutive quarters. The first quarter focuses on team structure, project definition, project scope, concept generation, final design, project management and documentation. The second quarter is dedicated to design implementation, performing experiments, data analysis, etc. While there are a few individual course deliverables, the majority are team deliverables covering report writing, presentations, engineering documentation, fabrication/testing/experimenting etc. The team deliverables account for 85% and the individual deliverables 15% of student's final grade.

Since the capstone course sequence gives students a chance to apply the skills, knowledge, tools, and techniques they've developed during the past 3+ years to a real-world engineering problem, it is designed to mimic real engineering projects. Therefore, students often face typical projects challenges such as confusing or unresponsive clients, budgeting issues, tight deadlines, communication breakdowns, and/or team conflicts. A significant aspect of many of these challenges is working with other engineers and colleagues as part of a team. Thus, this experience reinforces the truism that being an effective teammate is critical to being an effective engineer.

Potential capstone projects typically come from one of three sources: local industry, faculty, or students. To be considered, projects must be submitted using a standardized project proposal form. From the pool of potential projects, teams of 4 to 7 students are formed depending on project complexity and student interest. Typically, classes have around 30 or 60 students and 5 to 10 projects are selected from a varying sized pool of projects. Projects range from high power rockets, to electric vehicles, to accident reconstruction science, to factory automation cells, etc. Matching students with projects they are excited about and teammates they are compatible with is a daunting challenge since with up to 60 students and 10 teams, the number of possible team combinations is often on the order of 10^6 . Therefore, team formation methods are sought that efficiently yield high quality teams.

Team formation is somewhat manageable manually if the team size is predetermined and the exact number of needed projects is presented. However, it is undesirable to restrict the number of potential projects since there would be a real risk of forcing students onto a project they didn't want and it would put the instructors in the challenging position of having to reject some projects before actual interest is determined. Additionally, if a student proposes a project that other students are really interested in, it is important that that student be placed on that project if it is selected. Further, as mentioned earlier, it is desirable to have varying team sizes based on project complexity. Based on these complexities, an optimization algorithm is warranted and the constraints can be refined as detailed below.

Global Constraints

1. Team size minimum and maximum (our capstone course typically has teams between 4 and 7 students so this can be set explicitly).
2. Number of projects (for example, if 13 projects are proposed, the instructor can choose to have exactly 10 teams formed).

Projects Constraints

1. Each project can be set to Go, No-go, or Maybe. If “Go” the project will be selected regardless of how popular it is, if “No-go” the project will not be selected regardless of its popularity, and if “Maybe” the project will only be selected if it is popular enough to increase the team formation score. The default is for all projects to be “Maybe” but often specifics about the project changes (e.g. sponsor pulls out, budget falls through, dean/provost/etc. strongly encourages, etc.) so having these easy switches minimizes reprogramming when the inevitable last minute changes/decisions come up.
2. Team size minimum and maximum for individual projects can be specified that overrides the default global team size minimum and maximum.

Student Constraints

1. Students can be assigned to specific projects (e.g. the project they proposed) if and only if a team is selected for that project. However, if their project isn’t selected, that student will then be assigned to another project using their list of choices. This is necessary since it’s not uncommon for the most popular project to have more than twice as many 1st choices as the maximum team size and thus there is an increased chance that the student(s) who proposed the project will not be assigned to that team.

Data is gathered with a student interest survey given using Canvas, a learning management system. A minimum of five questions are asked requesting the student to choose their first, second, third, etc. choice. Additional questions can be asked if certain projects require specific courses, certifications, or skills. Students are required to select 5 unique projects or the program will assume they don’t have a first selection (i.e. there is slim chance they will get their first choice)

Optimization algorithm

Based on the goals and constraints, this team formation challenge is formulated as a Binary Integer Linear Programming optimization problem. The binary integer constraint is necessary since each student is either assigned to a project or not (1 or 0) and each project are either a Go (1) or No-go (0). Formally, the optimization is formulated to:

$$\text{maximize } \sum_{i=1}^n f_i x_i \text{ such that } \begin{cases} x_i \text{ are 0 or 1} \\ A \cdot x \leq b \\ A_{eq} \cdot x = b_{eq} \end{cases} \quad (1)$$

For the explanation of the optimization variables, let k be the number of possible projects and j the number of students needing to be assigned to a project.

Objective Function

f is a 1D array that contains the students choices as determined through the interest survey and assigns a weight to each. For this application, the weights are arbitrarily chosen to be $60/n$ with n being the 1st, 2nd, etc. choice. Thus a higher weight is given for student's preferred projects. Other weighting strategies are certainly possible but haven't yet been explored. Also included in f are project indicators (all set to zero so the specific projects chosen will not influence the optimum solution) that are utilized when specifying the constraints. This yields an f with $n = (k \cdot j + k)$ elements.

x is also a 1D array, the same size as f , that represents a solution to the optimization problem. It specifies which project each student is assigned to and the Go/No-go state for each project. The primary goal is to find an optimum x , subject to the constraints, to specify teams and projects selected.

For example, if 5 students are to be assigned to 1 of 3 projects, one possible team assignment is listed in Table 1.

Table 1. Example team assignment and project Go/No-go tables

Student	Assigned Project
John	3
Mary	1
Phil	1
Gus	3
Kate	1

Project	
1	Go
2	No-go
3	Go

The example in Table 1 shows one possible team assignment and is represented by the x array in Figure 1. In the array shown, the first three elements indicate whether John is on project 1, 2, or 3, the second set of three pertain to Mary and so on. The final three element are the project indicators and specify the Go/No-go state for each of the three projects and indicate that projects 1 and 3 are a Go.

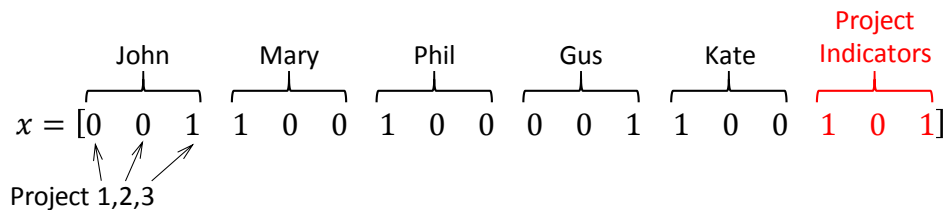


Figure 1. Example x array for teams and projects detailed in Table 1.

Optimization Constraints

To evaluate the quality of each possible group of teams, a team formation score (i.e. $\sum_{i=1}^n f_i x_i$) is used to quantify how well students were matched to their projects of choice. To accomplish this, the algorithm assigns a weight of $60/n$ to each student where n is the choice number of each

student. So if the student gets their first choice, their contribution to the team formation score is $60/1=60$ but if they only get their 4th choice their contribution is only $60/4=15$. Thus, if all students got their first choice, the team formation score would be $60/1=60$ times the number of students and be the maximum possible. Obviously, not everyone can get their top choice, so the algorithm maximizes the overall teams to give students as close to their top choice as possible by finding the teams that give the highest team formation score possible. Essentially, the goal is to maximize the average student “happiness”.

As specified above, x_i are constrained to be binary. Thus, each student is either assigned to a specific project (1) or not (0), and each project is either a Go (1) or No-go (0).

Inequality constraints, $A \cdot x \leq b$, where A is a matrix with n columns and b is a column vector, are used to:

1. set the minimum and maximum team size for each project.
2. assign a student to the project they proposed if, and only if, the project is a “Go.”

Equality constraints, $Aeq \cdot x = beq$, where Aeq is a matrix with n columns and beq is a column vector, are used to:

1. ensure each student is assigned to one, and only one, project.
2. ensure the requested number of projects is selected.
3. force select projects to be either a Go or a No-go regardless of what the students selected.

To solve the optimization problem posed, MATLAB’s mixed integer linear programming (MILP) solver `intlinprog` is utilized and it solves the optimization problem using linear program preprocessing, noninteger linear programming, mixed-integer program preprocessing, cut generation, heuristics, and/or branch and bound algorithms [6].

The Team Formation Process

To give some perspective on how the optimization is utilized, the team formation steps will be detailed. To start, a curated list of available projects is developed and accessible to students during the 4 or so weeks between when they register and the course starts. The following steps generally ALL occur before the second day of class.

1. During the week before class starts
 - a. Instructors post the final list of available projects with full descriptions.
 - b. Instructors send an announcement to the class to encourage students to fully consider all potential projects before making their choices.
2. 1st day of class
 - a. Instructor(s) and/or project sponsors give a brief (~1 min.) overview of each project and answer questions.
 - b. Students then have the opportunity to mingle together to gauge their classmates interests and ask further questions.
 - c. After class ends, students have several hours to select their top five project choices using the online student interest survey.

3. Running the optimization
 - a. Export survey data
 - b. Set number of desired projects
 - c. Set the default minimum and maximum team sizes
 - d. Run algorithm once to determine the numbers that the algorithm assigns to specific projects and students.
 - e. Set minimum and maximum team size for specific projects (optional)
 - f. Assign specific students to projects if the projects are a Go
 - g. Set specific projects as Go/No-go/Maybe if needed (optional)
 - h. Run optimization
4. Before the 2nd class period announce the projects and teammates students will be spending the next 5 or so months or their lives working on/with.

Unlike the 2-3 weeks Aller [5] uses, team formation is accomplished in 1 day with the actual optimization (step 2) taking less than an hour. The disadvantage is probably less optimum teams but the time gained to use to work on the project is maximized and the process is very efficient.

Results

So far, the optimization algorithm has been utilized to create project teams for eight capstone sections with each optimization run forming teams from a pool of approximately 30 or 60 students. This resulted in 230 students being placed on 40 projects teams. While tweaks were made to reduce or eliminate some of the housekeeping and data presentation tasks, the optimization algorithm was consistent throughout. The output of the latest version automatically generates a spreadsheet with the project teams and a plot of the overall results like the one in Figure 2 that can immediately be distributed to the students.

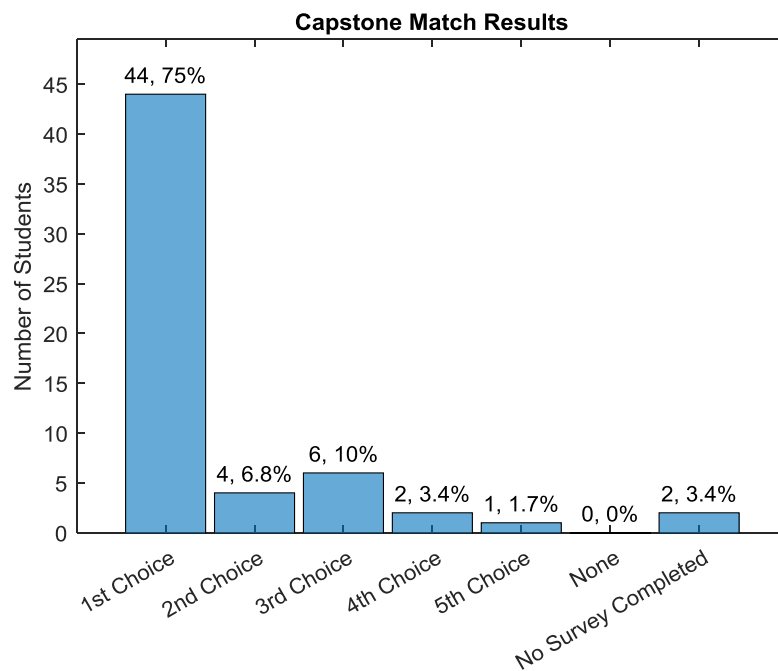


Figure 2 Example team formation results for a course.

When the results of all sections where the algorithm was used are considered, the algorithm matched 74% of the students to their 1st choice, 11% to their 2nd, and 9% to their 3rd choices as illustrated in Figure 3. On one end, there was a section where the lowest match was a 3rd choice while there was one section with two students who didn't get any of their choices. However, the only time students didn't get any of their choices there were more than 2.5 times the number of projects proposed than needed and those students had the misfortune of not choosing a single project that was a Go.

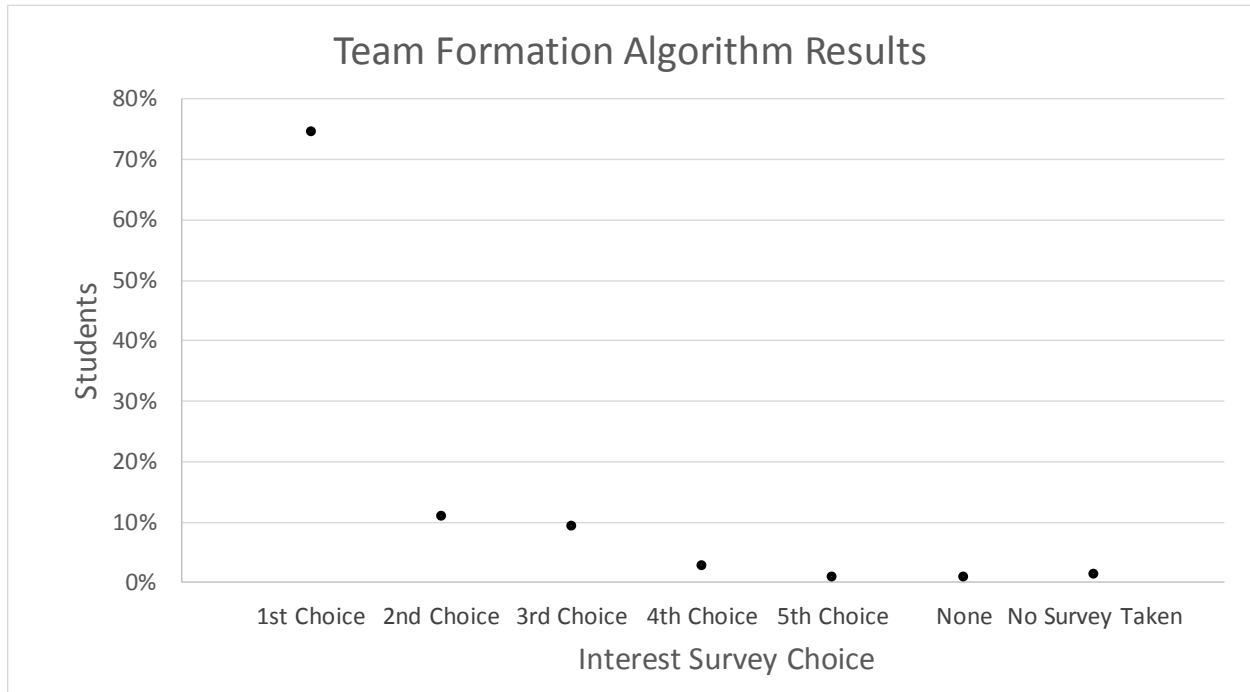


Figure 3. Cumulative results for team formation optimization algorithm for 40 projects composed of 240 students.

Overall, the algorithm did very good job of forming capstone teams very quickly with 94% getting one of their top 3 projects and enabled students to start on their project on the second day of class.

The algorithm was also retroactively used on 3 capstone sections (81 students) that used the same interest survey but where teams were manually formed. Instructors, after many hours of effort, were able to get 77% of the students their 1st choice. When ignoring all the extra effort required, this compares very favorably to the algorithm's results. However, feeding the interest survey data from those sections into the optimization algorithm resulted in an increase of the 1st choice matches to 79% with only 10 minutes of effort and thus validated the subsequently implemented optimization.

Conclusions

A very efficient team formation algorithm was developed and used to form capstone project teams in our Mechanical Engineering and Mechanical Engineering Technology programs. The

algorithm relies solely on a student interest survey and thus assumes that students are good at predicting what project they would enjoy working on. Being able to match 94% of the students to one of their top 3 projects in less than a day is very satisfactory and it will be used in our program moving forward. Of course, this method assumes students who get their top project choice will have a meaningful capstone experience but it's been observed many times that a solid, cohesive team can turn a mediocre project into a fulfilling experience while a poor team can turn a great project into a tedious, or even failed, exercise.

Moving forward, other factors that could be integrated into the team formation process include considering student's skills, classes (e.g. experience with machining, welding, or woodworking is very helpful for certain types of projects), available meeting times (some students have explicitly said this made their projects challenging), and/or GPA. Another factor to consider is the ratio of the number of available projects to those needed. If the ratio is too high, students who pick projects that don't go obviously don't get their top choice and this can lead to poor match results. Since the optimum available projects to needed projects ratio is not known additional investigation is warranted. Regardless, options being considered for dealing with this issue are limiting the number of available projects through instructor screening or integration of Aller's [5] mingling method into the 1st day of class whenever the number of proposed projects needs to be reduced.

References

- [1] L. C. J. L. Brickell *et al*, "Assigning Students to Groups for Engineering Design Projects: A Comparison of Five Methods," *J Eng Educ*, vol. 83, (3), pp. 259-262, 1994. . DOI: 10.1002/j.2168-9830.1994.tb01113.x.
- [2] S. M. Bosco, K. J. Jarvis and D. M. Harvey, "The Effect of Team Selection Method on the Occurrence and Nature of Conflict," *Journal of Applied Research for Business Instruction*, vol. 7, (1), 2009.
- [3] W. McHenry, "Selecting teams: The potential of "the Grid"," *Educ Inf Technol*, vol. 20, (1), pp. 183-199, 2015. . DOI: 10.1007/s10639-013-9274-8.
- [4] A. Baykasoglu, T. Dereli and S. Das, "PROJECT TEAM SELECTION USING FUZZY OPTIMIZATION APPROACH," *Cybern. Syst.*, vol. 38, (2), pp. 155-185, 2007. . DOI: 10.1080/01969720601139041.
- [5] B. M. Aller, D. M. Lyth and L. A. Mallak, "Capstone Project Team Formation: Mingling Increases Performance and Motivation," *Decision Sciences Journal of Innovative Education*, vol. 6, (2), pp. 503-507, 2008. . DOI: 10.1111/j.1540-4609.2008.00190.x.
- [6] (). *Mixed-Integer Linear Programming Algorithms*. Available: <https://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html>.