AC 2007-778: AN ALGORITHM FOR PROJECT ASSIGNMENT IN CAPSTONE DESIGN

Theodor Freiheit, University of Calgary

Julian Wood, University of Calgary

An Algorithm for Project Assignment in Capstone Design

Abstract

This paper presents an algorithm to automate the assignment of students to project teams. Students bid on a limited set of choices of the projects being offered. The algorithm then attempts to place students into projects such that the overall project assignment solution provides the highest 'satisfaction'. Satisfaction is defined by a scoring methodology for assigning students to their preferred project. The algorithm allows pre-assignment of students to a particular project, closes projects after they have been sufficiently subscribed, provides a bumping routine to move students around in finding a best solution, and eliminates 'unpopular' projects. It has successfully been tested in assigning students to project teams, reducing the time required to one quarter that taken using a similar manual system.

Introduction

Project based learning in design education is an effective method for students "to learn design by experiencing design as active participants" in the context of the "complex processes of inquiry and learning that designers perform in a systems context,…, often working collaboratively on teams…."[1] Team formation for project-based learning can present administrative difficulties for design educators, however, especially in large classes. Projects used in capstone design courses are usually established by one of two methods: projects suggested by the enrolled students and then vetted by the instructor, and projects sponsored by members of the university, industry, or non-profit communities. When projects are sponsored by outside organizations, the department and course instructors have a stake in facilitating a positive outcome for all parties.[2] Therefore, it is desirable to provide a design environment for each project that maximizes the potential educational and curricular goals for the students and the university, and the problem solution, recruitment needs, and/or community support goals of the outside sponsors.

Anecdotally, it has been found that when projects match a student's interests, whether it is in a particular discipline or the type of project, there is a better project outcome. Students that identify with the project objectives and believe in their team's collective potential will achieve high quality results.[3] Moreover, students want relevance and experiences that enable them to be successful practicing engineers. Todd & Magleby comment that "students enjoy learning more if their learning is focused on things that they choose" and that if they "have some choice in the project they work on they may have greater motivation than if they are assigned arbitrarily".[2] This is frequently one factor contributing to the organization of capstone design courses around student-proposed projects.

Many departments want to broaden their student experience to include real engineering problems sponsored by outside entities[4]. This requires student teams to be assigned to these projects either as individuals or through self-formed teams. However, when self-formed student teams do not receive their first choice project, such as with very popular projects, individual group members may not be interested in the project that they are assigned.

The capstone design course at the University of Calgary is similar to other capstone design courses, e.g. Moore & Berry[4] and Doepker[5]. Twenty-five to thirty projects are conducted for a full year, with an approximately equal number sponsored by industry and not-for-profits groups, faculty, and students. Students are given a description of the projects during the first week of classes, and they apply to work on specific projects. They are allowed to bid on three projects, and provide a brief description of their interest, capabilities and background. Teams of between four and six members are then assigned based on student interest. Unlike Todd *et al.*[6], no attempt is made to form teams based on cohesiveness, required talents, grade point average, personality type, or gender. The problem with this approach is that fair team formation is a time-consuming task for the course instructor. However, the administrative overhead required to assign students to a project team based on individual choices lends itself well to automation.

Team Formation Tool

A project to develop an automated student assignment process was initiated with the Teaching and Learning Centre at the University of Calgary. The goal of this project was to provide a reusable, automated, fair system in which student project groups could be created.

- Reusable – the solution should be reusable from year to year
- Automated – minimize the data entry and manual manipulation required by the instructors
- Fair – students should be assigned to groups based on preference; conflicts should be resolved randomly

This tool would automate the process of assigning students to groups based on their preferences. It is especially useful for classes with large numbers of students, because the tool can sort hundreds of students into groups. The basic solution would consist of an *Administration Page* where the required project information would be entered, a *Selection Page* where the students would be able to enter in their group preferences, and a *Report Page* that would list the final Student Project Groups and any errors encountered during the group creation process. The student assignment algorithm was implemented in Blackboard as a 'Team Formation Tool' (TFT). Blackboard is a web-based, content-management system designed for educators. It is intended to facilitate the work involved in placing a course or associated course resources online.[7]

Blackboard was selected to implement this algorithm because of a number of factors. It already handles student lists for course enrollment and its security features already addresses user name and password issues. Course information can be kept confidential and can be limited to registration in the course, including students, instructors, TA's, and guests. Blackboard is also the main communication tool used for the design course and many other courses on campus. Finally, Blackboard had an existing group management feature, allowing for the potential of automated group creation, with the facilitation of the communication of detailed project information, the ability to restrict access to administrative processes to instructors and TAs, and modularity permitting a seamless integration to existing course websites.

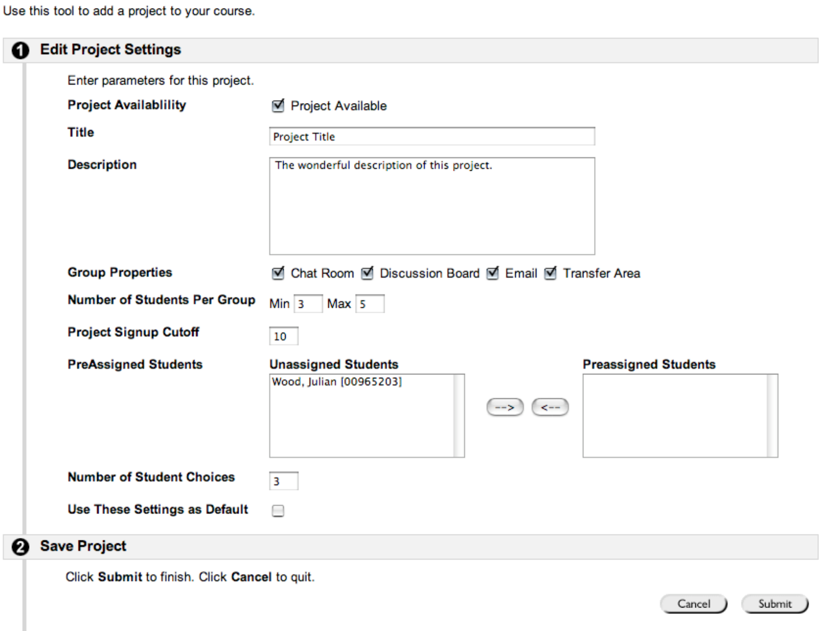
Features implemented into the TFT include the ability for:

- Students to decide in which groups they want to partake.
- Students to rank preferred groups.
- The automatic assessment of the most popular groups, given student rankings.
- The automatic creation of discussion groups *and* forums.
- The ability to download group reports for import into Excel.
- Instructors to pre-assign students to groups.
- Instructors to easily monitor student rankings as they are made.
- Creation of groups without student ranking.

Project Group Administration and Creation – as Implemented in Blackboard

The instructor creates projects by entering project data in the *Administration Page*, illustrated in Figure 1. Projects are created one at a time by entering the project title, description, a minimum and maximum group size, and a Project Signup Cutoff to limit to the number of students a particular project can have selected as a preference. This last feature enables better distribution of students between the projects. Blackboard features that may be enabled for student teams include chat, discussion board, and email lists. Students are not able to see projects unless explicitly made available. This permits projects to be reserved and not given as general access to all students, for example when a project is brought to the instructors by a student team.

Once projects have been created, students can rank their project preferences using the *Selection Page*, Figure 2. Students must be notified that projects are available and may be given a deadline for submitting their choices. Students can be pre-assigned to a project, if, for example, it is a student-sponsored project. Students see a list of all the projects that have been made available and have not already reached their Project Signup Cutoff limit. Students use radio buttons to rank


Use this tool to add a project to your course.

1 Edit Project Settings

Enter parameters for this project.

Project Availablility ☑ Project Available

Title Project Title

Description The wonderful description of this project.

Group Properties ☑ Chat Room ☑ Discussion Board ☑ Email ☑ Transfer Area

Number of Students Per Group Min 3 Max 5

Project Signup Cutoff 10

PreAssigned Students **Unassigned Students** Wood, Julian [00965203] **Preassigned Students**

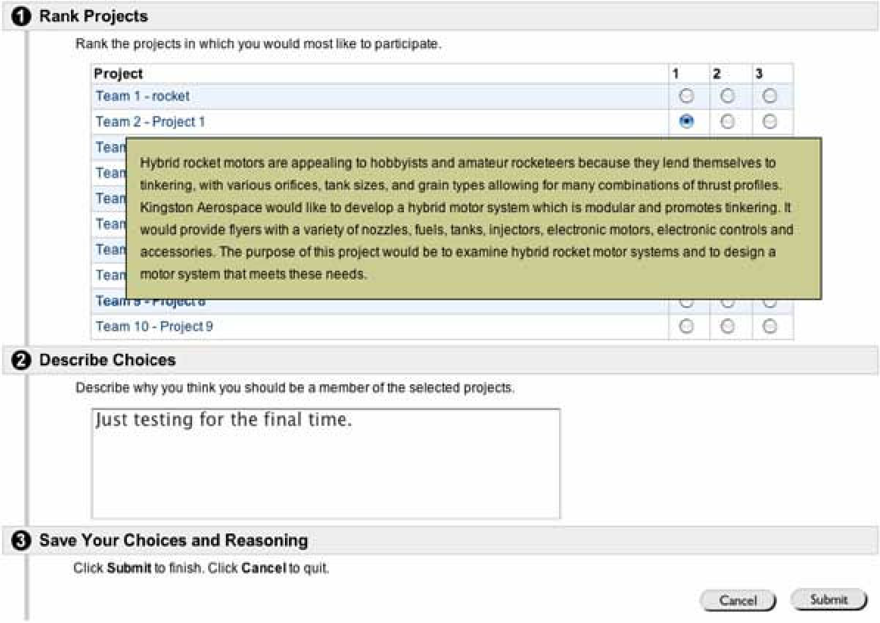
Number of Student Choices 3

Use These Settings as Default □

2 Save Project

Click **Submit** to finish. Click **Cancel** to quit.

Figure 1 - Project Administration Page



1 Rank Projects

Rank the projects in which you would most like to participate.

Project	1	2	3
Team 1 - rocket	○	○	○
Team 2 - Project 1	●	○	○
Tean			
Tean			
Tean			
Tean			
Tean			
Tean			
Team 9 - Project 8	○	○	○
Team 10 - Project 9	○	○	○

Hybrid rocket motors are appealing to hobbyists and amateur rocketeers because they lend themselves to tinkering, with various orifices, tank sizes, and grain types allowing for many combinations of thrust profiles. Kingston Aerospace would like to develop a hybrid motor system which is modular and promotes tinkering. It would provide flyers with a variety of nozzles, fuels, tanks, injectors, electronic motors, electronic controls and accessories. The purpose of this project would be to examine hybrid rocket motor systems and to design a motor system that meets these needs.

2 Describe Choices

Describe why you think you should be a member of the selected projects.

Just testing for the final time.

3 Save Your Choices and Reasoning

Click **Submit** to finish. Click **Cancel** to quit.

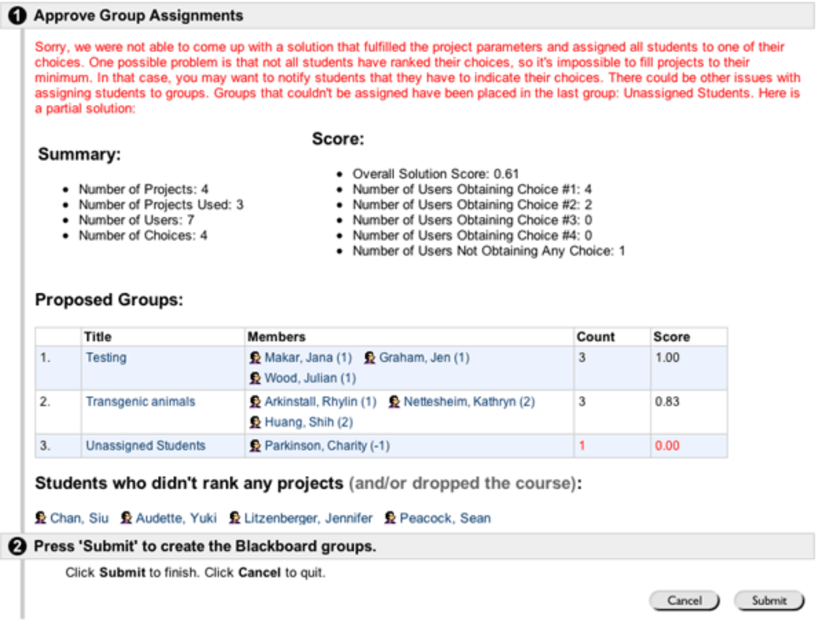
Figure 2 - Project Selection Page

their choices, and have a text box to give the instructor feedback on their project preferences. This textbox feature is useful for resolving assignment errors.

Students can change their project selections and ranking at any time, with only their last selections taken into account when assigning them to groups. Once the groups have been created, changes submitted by the students are ignored. To facilitate communication of project details to the student, as illustrated in Figure 2, the TFT will provide the project description in a pop-up box when the cursor is moved over the project title.

After students have indicated their project preferences, groups can be created for the projects. The *Administration Page* provides a list of students and their preference rank next to each project. In its Blackboard implementation, group creation from this list is as simple as clicking on the 'Create Groups From Project' button. As illustrated in Figure 3, the instructor is presented with a listing of the projects with a tentative assignment of students to each group on a temporary *Group Creation Report Page*. The program is designed to assign as many students as possible to their preferred group; however, some students will be assigned to their second or third choice. Information presented to the instructor includes summary statistics on the number of projects used and the number of students assigned, the overall solution satisfaction score for the proposed solution with a breakdown of assignment rankings, a count of the number of students assigned to each group, and a satisfaction score for each group. Like the project description on the *Selection Page*, a student's rationale for their choices can be seen by moving the cursor over that student's name.

A list of students who either never ranked any projects, or dropped the course at some point is also provided. By clicking on a student's name, a reminder email can be sent, if desired. In the situation where not every student could obtain one of their preferred projects, a new group named 'Unassigned Students' is created. Students in this group can be moved to other project



❶ Approve Group Assignments

Sorry, we were not able to come up with a solution that fulfilled the project parameters and assigned all students to one of their choices. One possible problem is that not all students have ranked their choices, so it's impossible to fill projects to their minimum. In that case, you may want to notify students that they have to indicate their choices. There could be other issues with assigning students to groups. Groups that couldn't be assigned have been placed in the last group: Unassigned Students. Here is a partial solution:

Summary:

- Number of Projects: 4
- Number of Projects Used: 3
- Number of Users: 7
- Number of Choices: 4

Score:

- Overall Solution Score: 0.61
- Number of Users Obtaining Choice #1: 4
- Number of Users Obtaining Choice #2: 2
- Number of Users Obtaining Choice #3: 0
- Number of Users Obtaining Choice #4: 0
- Number of Users Not Obtaining Any Choice: 1

Proposed Groups:

	Title	Members	Count	Score
1.	Testing	Makar, Jana (1) Graham, Jen (1) Wood, Julian (1)	3	1.00
2.	Transgenic animals	Arkinstall, Rhylin (1) Nettesheim, Kathryn (2) Huang, Shih (2)	3	0.83
3.	Unassigned Students	Parkinson, Charity (-1)	1	0.00

Students who didn't rank any projects (and/or dropped the course):

Chan, Siu   Audette, Yuki   Litzenberger, Jennifer   Peacock, Sean

❷ Press 'Submit' to create the Blackboard groups.

Click **Submit** to finish. Click **Cancel** to quit.

Figure 3 - Group Creation Report Page

groups using Blackboard's built-in group management tools. If the proposed groups are satisfactory, clicking on a second 'Submit' button will permanently create the groups within the Blackboard course website with the Blackboard features selected. After the groups have been created, group settings can be edited or students can be assigned to a different group using the group management tools, provided by Blackboard. For example, some students may not have indicated any project preferences and need to be manually added to a group.

Student Assignment Algorithm

The following section describes the algorithm that places students into project groups. Because the algorithm uses random selection processes, it generates different results every time a proposed solution is generated. A summary of the algorithm can be found in Figure 4.

Student Placement

For each project, all students enrolled in the class having ranked their first preference projects are temporarily pooled and students are randomly selected and assigned to their first rank project until the maximum number of students for the project group is reached. Next, all remaining students are pooled according to their second preference and again assigned randomly to their second rank project group until the maximum number of students for each group is reached. Finally, all remaining students are pooled according to their third preference and again assigned randomly to groups until the maximum number of students for each group is reached. Students that have not been assigned to a project, because their project preferences have been filled, are now eligible for the bumping routine. Students who did not select any projects or dropped the class are ignored by the algorithm, but are retained for reporting purposes.

1. sort projects from least popular to most popular
    - calculate satisfaction points for each student ranking
    - popularity is a measure of the number of students in a project and the satisfaction weighting assigned to each ranking
    - empty projects are least popular
    - pre-assigned members are assigned
2. shuffle members and loop for each ranking, up to the designated number of student choices:
    - assign each member to current ranking (start with first) up to the maximum permissible group size
    - pool remaining unassigned members and proceed to next loop iteration
3. remove under-assigned projects, placing any members into the **unassigned members** list, if and only if they can't be put in one of their other choices
    - proceed in direction of sort
    - keep track of the total number of spots available, and don't eliminate a project, even if under-assigned, if it would decrease the number of spots available below the number of students still needing assignment
4. assign **unassigned members** using a bumping algorithm
    - take member and see if we can assign to any of their other choices without problem
5. if not, take his or her first choice project and find someone else in that project (at random) that can be removed (the displaced user)
    - "can be removed" refers to anyone that is not pre-assigned
    - try to assign this displaced user to one of their choices by the same process, up to 3 levels deep (three "bumps")
    - any remaining students are placed in the catchall project, **Unassigned Students**
6. loop back to step 2.
    - save best solution based on overall satisfaction score
    - terminate after 30 iterations

Figure 4 – Algorithmic Implementation of TFT

Student Pre-assignment

Students that are pre-assigned to a project group by the instructor prior to group creation are assigned immediately. The algorithm will not move their assignment under any circumstance. If no other student ranks the project in which a student was pre-assigned, the project will simply be presented as part of the solution, even though it likely does not satisfy the project's minimum enrollment criterion. Projects with pre-assigned students are never eligible to be eliminated by the project elimination subroutine, outlined below. It is up to the instructor to decide what to do with a pre-assigned project that is undersubscribed.

Satisfaction Score

Each project group has a satisfaction score calculated as a basis for optimizing student placement in the project groups. Satisfaction is defined in this algorithm by a scoring methodology that attempts to place students in their first, second, or third, etc. choice, in that order. More specifically, a point score is calculated for each project ranking, and the overall group satisfaction score is generated by totaling the points for the rank each student has given to the project to which they have been assigned. Satisfaction points allotted to a student project ranking is given by the following formula:

$$\text{Points} = \frac{[(n-i)+1](n-i)}{2} + 1$$

where n is the total number of possible project choices the student can rank, and i is their rank of the project. For example, in a scenario where each student can rank three projects, 4 points are awarded for a successful first place assignment, 2 for a second, and 1 for a third place assignment.

Group satisfaction scores are normalized to a maximum possible score for a group in which all members ranked that project first. Thus if there are 5 members in a proposed group where students had three project choices, the maximum score would be 20 (four points for first rank times five members). Summing the actual ranks and dividing by the maximum gives the project group satisfaction score. For example, if there are four first ranks and one third rank, the student ranking points would sum to 17, then be normalized to a project group satisfaction score of 17/20 = 0.85.

The overall solution satisfaction score is calculated by averaging all project group satisfaction scores, to give a number between 0 and 1. For both individual groups and the overall satisfaction score, the closer the score is to 1, the more students were placed in their most preferred project.

Bumping Routine

When there are students remaining who have not been assigned after student placement, a bumping routine is employed to try and assign them a project. This condition can occur for example when a student's preferred projects have filled-up in the initial student placement. An unassigned student is chosen at random from the unassigned group, and each of his or her choices is examined to see if they can simply be assigned to one of those projects. Since at this point, all of the projects available have at least the minimum number of members, the only reason the assignment could fail is if the project was full.

If an assignment is not possible for any of the student's choices, the unassigned student is temporarily assigned to their first choice, and a student is chosen at random to be removed from this project. This displaced student goes through the same bumping procedure, where each of his or her choices is examined. This procedure continues for up to three bumping cycles, seeking to place each displaced student in turn in one of their project choices. If the bumped student can not be satisfactorily placed, all changes are reverted and the original unassigned student is no longer eligible for bumping.

The bumping routine attempts to place each student from the unassigned pool only once. Sometimes, even after going through the bumping routine for each unassigned student, it is still not possible to assign all the students to a project. The bumping routine terminates at his point and places the remaining students in the special unassigned pool in a new project, aptly titled "Unassigned Students", and returns this project as part of the solution. Thus, the algorithm always returns its best attempt at solving the assignment problem, even if it can't place all the students.

Project Elimination

Projects are eliminated from the assignment pool when they are unpopular. Popularity is measured by scoring the project on a scale of zero to one, as determined by the ranking of all its potential members, even if assigning all those members would violate the project's minimum or maximum requirement. Project popularity is scored from both the ranking score and the number of students that chose the project with the following formula:

$$\text{Popularity} = \frac{\left(2\frac{\sum p}{np_{\max}} + \frac{n}{c}\right)}{3}$$

where p is the satisfaction points of each student based on their ranking of the project, $p_{\max}$ is the maximum possible satisfaction score given the number of student choices, n is the number of students that signed up for the project, and c is the project cut-off limit.

This formula provides a means to rank the overall interest in a project, weighing both the number of students interested and their specific level of interest in the project, with a higher weight on their interest. Thus, a project that received no rankings by any students is the least popular (would score zero), and will be potentially eliminated, as long as eliminating it does not reduce the total number of spots available to below the number of students needing to be assigned. Projects are eliminated one by one, proceeding from least popular to most popular, ensuring only that sufficient spots remain before the project is eliminated.

If the course instructor does not want any projects to be eliminated, e.g. if all projects must be staffed, the course instructor should set the cumulative maximum number of spots for each project equal to the enrollment in the course. This will prevent projects from being eliminated as it would reduce the spots available below the number of students needed to be assigned. Alternatively, the instructor can pre-assign a student (or dummy student) to a particular project that they do not want eliminated (projects with pre-assigned students are never eligible for elimination), and use the group management tools to re-assign student to this project if it is under-subscribed.

Stopping Criteria

There are several ways in which the algorithm can conclude. If after any step in the algorithm, the following criteria are satisfied, the assignment problem is considered solved:

- all of the projects are found to be "satisfactory", meaning they have the required minimum enrollment but do not exceed the maximum enrollment, or are empty.
- all students have been assigned to a project.
- no student is assigned to more than one project.

In order to obtain a more optimal solution, the algorithm calculates thirty assignment solutions, presenting only the solution with the best overall satisfaction score.

Example

The following example illustrates the project assignment algorithm. Four projects are created by the instructor, with a desire to have project groups of at least three, but no more than four students. There are 12 students enrolled in the class. Each student will rank three project preferences, representing a total of 36 rankings, so the instructor sets the project cut-off limit to 10 for each project. Individual students, labeled a, b, c, etc., have ranked their preferences, as show in Figure 5. The algorithm then calculates the popularity of the projects, shown in the figure.

In step 1, the algorithm rank orders the projects according their popularity, as shown. In step 2, the algorithm randomly assigns students to each project based on their preferences, starting with their first rank. In this example, all students can be assigned their first rank preference except one (student j), but not all projects have reached their minimum allowable capacity.

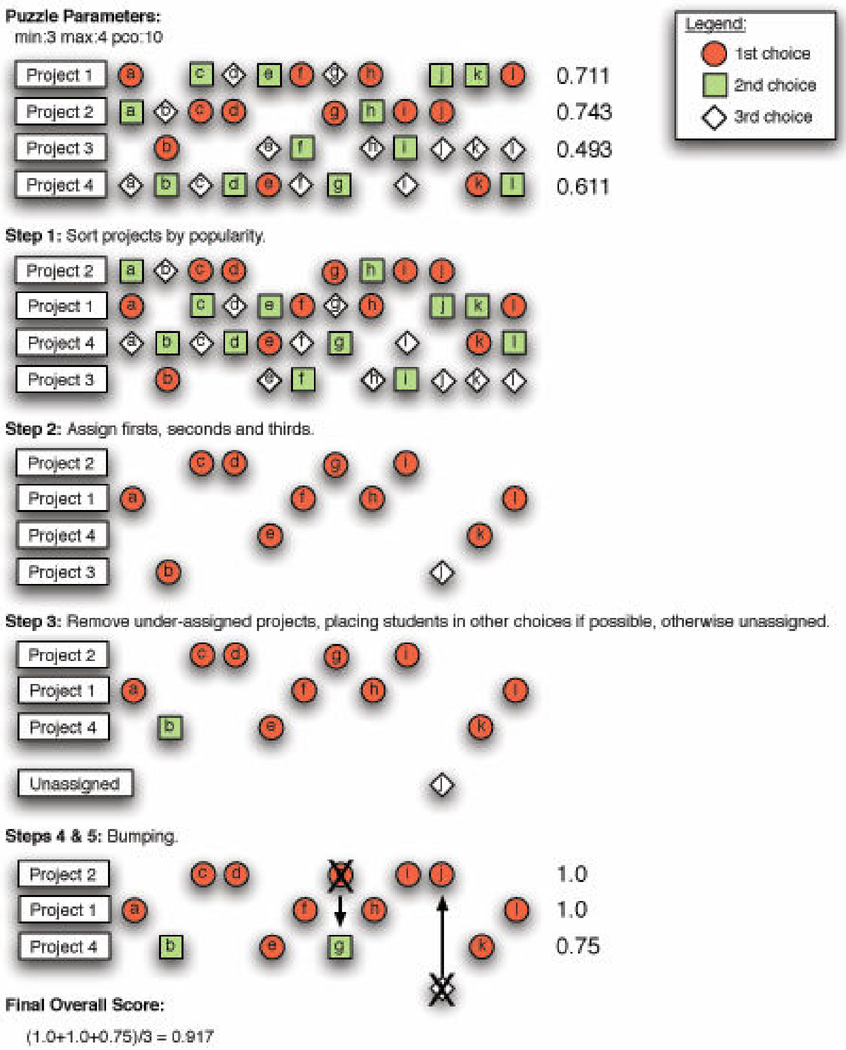


Figure 5 – Example Algorithm's Application to Student Assignment to Projects

In step 3, the algorithm begins to eliminate projects. Since there are 12 students, the algorithm must retain at least three projects. Beginning with the most unpopular project (project 3), the algorithm selects a student at random and tries to place that student in another project. In this example, student b is easily placed in his or her second choice. Student j, however, cannot be assigned to their first or second choice, as those projects are at their maximum capacity. Therefore, student j is placed in the 'Unassigned' pool.

Finally, in steps 4 & 5, the bumping subroutine places student j in their first choice (project 2). A student is randomly selected to be bumped. In this case, student g, who is capable of being placed in their second choice (project 4), is chosen, concluding the placement of all students. A project satisfaction score is now calculated for each project, giving an overall solution score of 0.917. The algorithm now repeats (not shown) 30 times to see if a better solution (higher average group satisfaction score) can be obtained.

Conclusion

This paper presented an algorithm, implemented in Blackboard, to automate the assignment of students to project teams for capstone design courses. It allows students choices among the projects being offered, up to a limit set by the instructor (e.g. their top 3 choices). The algorithm then attempts to place students in projects such that the overall solution provides the highest 'satisfaction'. The algorithm allows pre-assignment of students to a particular project (e.g. to assure a student who proposes a project their assignment to that project), closes sufficiently subscribed to projects to maximize student distribution in all projects, provides a bumping routine to move students around in finding a best solution, and eliminates 'unpopular' projects as long as there is sufficient capacity within the remaining projects.

This algorithm has been successfully tested in assigning students to project teams, reducing the time to form teams to one quarter the time required when using a similar manual system. It has also been successfully rolled out to the entire University of Calgary, where to date 81 courses have used it for 471 projects involving 2588 student choices. Courses using the Team Formation Tool include those in Business, Science, Psychology, Philosophy, English, Linguistics and Engineering.

To increase the flexibility of this algorithm, future work proposed for its development is to develop staffing categories for project groups, e.g. allow projects to be assigned with multiple categories, each with a minimum and maximum group size, such as a major or a talent associated with students. This will better facilitate multi-disciplinary team formation. Further, it is unclear what optimal Project Signup Cut-off limit should be set for each project. If it is too small, student choices may be too restricted, as projects will fill too fast. If it is too large, popular projects will skew the distribution of students across all choices. A better understanding of this influence of the cut-off limit should be developed.

The features of this algorithm described in this paper are available in Version 1.2.4. This algorithm, in the form of a Blackboard module, can be downloaded from the following URL:

http://commons.ucalgary.ca/projects/tft/index.html

Bibliography

[1] Dym, C.L., Agogino, A.M., Eris, O., Frey, D.D., Leifer, L.J., “Engineering Design Thinking, Teaching, Learning”, *Journal of Engineering Education*, v 94, n 1, January 2005, pp 103-119.

[2] Todd, R.H., Magleby, S.P., “Elements of a successful capstone course considering the needs of stakeholders”, *European Journal of Engineering Education*, v 30, n 2, May 2005, pp. 203-214.

[3] Delson, N.J., “Increasing Team Motivation in Engineering Design Courses”, *International Journal of Engineering Education*, v 17, n 4-5, 2001, pp 359-366.

[4] Moore, D., Berry, F., “Industrially Sponsored Design Projects Addressed by Student Design Teams”, *Journal of Engineering Education*, v 90, n 1, January 2001, pp 69-73.

[5] Doepker, P.E., “Integrating the Product Realization Process into the Design Curriculum”, *International Journal of Engineering Education*, v. 17, n 4-5, 2001, pp 370-374.

[6] Todd, R.H., Sorensen, C.D., Magleby, S.P., “Designing a Senior Capstone Course to Satisfy Industrial Customers”, *Journal of Engineering Education*, v 82, n 2, 1993, pp 92-100.

[7] Blackboard Learning System. (n.d.). Retrieved January 10, 2007, from http://www.blackboard.com