

SPRING 2010

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

RICHARD A. LAYTON
Rose-Hulman Institute of Technology
Terre Haute, Indiana

MISTY L. LOUGHRY
Georgia Southern University
Statesboro, Georgia

MATTHEW W. OHLAND and GEORGE D. RICCO
Purdue University
West Lafayette, Indiana

ABSTRACT

A significant body of research identifies a large number of team composition characteristics that affect the success of individuals and teams in cooperative learning and project-based team environments. Controlling these factors when assigning students to teams should result in improved learning experiences. However, it is very difficult for instructors to consider more than a few criteria when assigning teams, particularly in large classes. As a result, most instructors allow students to self-select teams, randomly assign teams, or, at best, balance teams on a very limited number of criteria.

This paper describes the design of *Team-Maker*, a web-based software tool that surveys students about criteria that instructors want to use when creating teams and uses a max-min heuristic to determine team assignments based on distribution criteria specified by the instructor. The *Team-Maker* system was validated by comparing the team assignments generated by the *Team-Maker* software to assignments made by experienced faculty members using the same criteria. This validation experiment showed that *Team-Maker* consistently met the specified criteria more closely than the faculty members. We suggest that *Team-Maker* can be used in combination with the *Comprehensive Assessment of Team-Member Effectiveness (CATME)* peer evaluation instrument to form a powerful faculty support system for team-based and cooperative learning and for a variety of research purposes. Internet access to both the *Team-Maker* and *CATME* systems is freely available to college faculty in all disciplines by selecting the “request faculty account” button at <https://www.catme.org>.

Keywords: Team assignment, Teamwork, Cooperative Learning, Algorithm, Validation

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

INTRODUCTION

Undergraduate engineering programs were dominated by the paradigm of individual, rather than collective, excellence until the mid-1990's (Hilborn 1994). Outcomes-based accreditation requirements that engineering programs develop students' ability to function on multidisciplinary teams, implemented by the Accreditation Board of Engineering and Technology (ABET 2009-10), encouraged the engineering education community to shift this paradigm. As a result, instructional methods in which students learn from one another by working in groups have quickly gained ground in engineering classrooms. Note that although some authors use the terms "groups" and "teams" differently (Katzenbach and Smith 1993), in this paper, we use them interchangeably.

Various pedagogical approaches that use student teams have been shown to be effective for engineering education (Felder 2000). These include cooperative learning (Johnson 1998), collaborative learning (Bruffee 1993), and team-based learning (Michaelsen 2002). Team-based learning is treated as synonymous with cooperative learning in the engineering education literature (Felder 1993). Additional approaches, such as problem-based learning (Woods 1996) and active learning (Wankat 1993) are often used in conjunction with student teams, even though these approaches could be used while having students work individually. Because this emphasis on teaming has occurred relatively recently, there is limited research about what makes teamwork experiences successful in engineering education. There is a substantial body of research about the factors that influence the success of teams in general and student teams in particular. Much of this research is reported in the management and psychology literatures, however, where team processes have traditionally been studied and where teamwork has been used in classes for many years (Bacon, Stewart, and Silver 1999). Therefore, the generalizability of that work to engineering student teams is often unproven. One thing that is clear from existing research is how members are assigned to teams has important implications for team-member outcomes and team effectiveness.

This study adds to the engineering education literature and practice by demonstrating the validity of a computer-based system to assign students to teams according to instructor-specified and by making its heuristics available for public scrutiny.

METHODS OF ASSIGNING STUDENTS TO TEAMS: WHO SHOULD FORM TEAMS?

Although a number of external factors, team processes, and team-member characteristics have been shown to influence team success and team-member outcomes (Stewart 2006), one factor that is especially important in academic contexts is how team members are assigned to teams. This is a particularly important issue in student learning teams because instructors can directly control team

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

assignments, whereas many of the other factors that influence important outcomes are not within instructors' control. The three methods of assigning teams that instructors commonly use are self-selected teams, randomly assigned teams, and instructor-assigned teams. Each of these methods is described next. This section concludes with a discussion of computer-aided team formation, which has a number of advantages relative to the three alternatives.

Self-selected teams give students more responsibility and control over their learning experience than when instructors assign teams, which has a number of advantages and disadvantages (Dipinto and Turner 1997). Bacon and colleagues found that students often cite self-selected teams as their best team experiences, most likely due to increased group cohesiveness (Neal 1997; Strong and Anderson 1990; Wolfe, Bowen, and Roberts 1989; Wolfe and Box 1988), accountability (Mello 1993), and cooperativeness, which increases team members' feelings of indispensability and improves their satisfaction with deadlines (Bacon, Stewart, and Silver 1999). These benefits of self-selection were greater after the first academic term and are consistent with Gosenpud and Miesing's finding that knowledge of teammates prior to team formation is associated with improved team performance (1984).

In contrast to these findings is considerable evidence of negative effects of self-selection. Feichtner and Davis (1984) reported that self-selected teams resulted in 40% of students' worst group experiences and only 22% of their best group experiences. In a study of engineering students at the United States Military Academy, Brickell and colleagues found that self-selection had negative effects on students' opinions about the course, instructors, projects, classmates, and other criteria ([Brickell et al. 1994](#)). Self-selection can also lead to excessive homogeneity (Jalajas and Sutton 1984), such that the teams lack diversity (Bacon, Stewart, and Stewart-Belle 1998; Kirchmeyer 1993) and might not have all the skills required for their team's task (Mello 1993). Self-selection can also lead to clique behavior that erodes team cohesion and performance (Daly and Worrell 1993). Self-selecting teams is likely to be difficult and uncomfortable for students who do not have acquaintances in the class, particularly when other students already know one another, and for students who are introverts. Students can also feel uncomfortable about declining to be on a team with classmates who they believe would not be a good match, due to the social repercussions of rejecting peers. Another disadvantage of self-selected teams is that, due to their increased cohesion, self-selected teams may be more likely to experience "groupthink" (Janis 1982). Groupthink can be a particularly dangerous phenomenon in engineering teams, as exemplified in case studies of disasters (Moorhead, Ference, and Neck 1991), so it is vital that students in engineering learn the skills that it takes to avoid groupthink.

Self-selected teams with instructor-imposed constraints have been proposed to balance the benefits and drawbacks of self-selected teams (Bacon, Stewart, and Silver 1999). In this system, the instructor can insist, for example, that each team has at least one international student. This technique is still subject to many of the negative effects of self-selection.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Random assignment is another option for assigning teams, but this method has a number of disadvantages and no clear strengths relative to the alternatives. Random assignment does not necessarily result in a team with any more diversity, balanced skills, or blend of personalities than does self-selection (Cook 1981; Quirk 1989; Vora and Akula 1978), yet it raises concerns about fairness (Bacon, Stewart, and Silver 1999). Bacon and colleagues found that randomly assigned teams were negatively associated with students' best team experiences, and were not significantly associated with students' worst team experiences (Bacon, Stewart, and Silver 1999). Although random assignment lacks the advantages of the other team-assignment methods, it does avoid some of the negative effects of self-selection (Johnson, Johnson, and Smith 1991), so it is typically used for expediency. Instructors often use random assignment for short-term team assignments, when they do not see a clear benefit of using a more complex team-assignment strategy, and when they do not want to spend much time assigning teams. There are also situations when an experimental method requires that teams be randomly assigned.

Instructor-assigned teams enable instructors to control various criteria in an effort to create positive team experiences, and the preponderance of the available evidence suggests that controlling those criteria improves student outcomes (Oakley et al. 2004). Although there are clear advantages to assigning teams according to certain criteria, instructors assign teams relatively infrequently because the logistics can be challenging (Bacon, Stewart, and Silver 1999; Decker 1995). The complexity of team-assignment increases dramatically as the class size and number of variables to be considered increases. Therefore, implementing more than a few criteria for team formation can be inordinately time-consuming for instructors, especially when accounting for students' availability for team meetings outside of class and when working with the large classes that are typical in undergraduate engineering.

Computer-aided team formation makes instructor control of the team-assignment process feasible in more circumstances. To facilitate the assignment of students to teams using instructor-specified criteria, Bacon and colleagues developed a software program called "Team Maker" (Bacon, Stewart, and Anderson 2001), which administers a survey to students in order to collect demographic data and roles that students prefer to hold in teams. Instructors manually transcribe students' survey responses to a spreadsheet programmed to form teams that optimize the instructor's criteria. Although Bacon and colleagues recognized the potential of computer-aided team formation, they noted the drawbacks of their software system due to its complexity and the time required to create the survey and enter the data into the spreadsheet prior to program execution. Another software program for team formation is described by Redmond (2001). We do not provide details about this system because we believe that it is not as adaptable and user-friendly as the web-based program described next.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

THE *TEAM-MAKER* SYSTEM: AUTOMATING TEAM-FORMATION.

Cavanaugh, Ellis, Layton and Ardis (2004), who were unaware of Bacon and colleagues' work, developed a different system that forms teams using instructor-defined criteria. Layton, an engineering professor with several years of experience manually assigning students to cooperative learning teams, worked with Ardis, a software-engineer, and undergraduate students Cavanaugh and Ellis to develop a web-based system to automate the team-assignment process. By coincidence, they named the program "*Team-Maker*" (as they were unaware of the similarly named program by Bacon and colleagues). The remainder of this paper describes the development, use, and testing of the *Team-Maker* system, followed by a brief discussion of the factors that the literature suggests instructors should consider when assigning students to teams.

Cavanaugh et al's. (2004) main objective was to create an algorithm to codify the team-assignment process and implement it in an easy-to-use Internet-based interface. Their specific goals for the system included:

- automating the team-assignment process consistent with well-established methods for manually assigning students to cooperative learning teams
- increasing the likelihood that instructors' team-formation criteria are met compared to manually-assigned teams
- providing a team "compliance score" to assess the extent to which all of the team-formation criteria have been met
- allowing instructors to explore multiple solutions to the team-assignment problem; and
- availability of the program to faculty everywhere.

The system, initially developed and tested in 2002-03, provides separate interfaces for instructors and students. The instructor's interface is used to create a student survey and to specify the criteria by which the survey results should be used to make team assignments. The student's interface allows a student to complete the survey confidentially and on-line, so that the data do not have to be re-keyed before they are used.

To use *Team-Maker*, an instructor is given a login by the system administrator. This prevents unauthorized users from creating surveys or viewing confidential data. The instructor creates a survey by providing the names and e-mail addresses of the students to be surveyed, choosing questions, and specifying a deadline. The system generates an email to each student that includes a customized link to the survey. After the students complete the survey, the instructor specifies how their responses are used to form teams. The software generates a set of teams according to those criteria, along with a statistical summary of how all of the surveyed variables are distributed on each team. *Team-Maker* also provides for manual reassignment of team members for situations when an

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

instructor knows something relevant that is not provided to the software (for example, that two students may not work well together).

The *Team-Maker* interface allows instructors to specify any size team and accepts various question formats so that instructors have substantial flexibility in creating teams. Question formats include multiple choice, choose-any-or-all-of, and schedule availability, each with its own scoring algorithm. *Team-Maker* asks instructors to choose from several different criteria when specifying how variables should be distributed when the system forms the teams. These include maximizing the diversity of a variable on a team (such as to distribute sub-disciplines/majors evenly across teams), minimizing the diversity of a variable on a team (for example, to have students with similar interests on the same team), and a special distribution criterion that allows instructors to prevent women or minorities from being outnumbered on a team. *Team-Maker* also asks instructors to assign weights to the variables when they specify the distribution criteria, so that the algorithm can give higher priority to the criteria that the instructor feels are the most important for assigning students to teams.

Instructors can also use *Team-Maker* to collect information that is not used for assigning students to teams. This feature of the *Team-Maker* program allows the instructor to ask questions of students for classroom use or research purposes, beyond what is needed to form teams. For example, instructors can use the survey to get to know their students, replacing the index cards and paper questionnaires that many instructors collect from students early in the semester. In addition, instructors can conduct research by collecting information on variables that might affect important individual or team-level outcomes, allow those variables to distribute randomly (by instructing *Team-Maker* to ignore them when assigning teams), then study the effects of those variables on the outcomes of research interest. This feature of *Team-Maker* facilitates the creation of new knowledge about what variables should be considered when forming teams and how to effectively distribute those variables across teams.

THE *TEAM-MAKER* ALGORITHM FOR ASSIGNING STUDENTS TO TEAMS

The *Team-Maker* algorithm assigns students to teams based on their responses to an online survey. Instructors create the student survey by choosing the variables that they want to be included in their survey from the list of variables in *Team-Maker's* "inventory". The variables have associated with them the questions that the students will be asked and the responses from which they will be permitted to choose. When *Team-Maker* was first developed (*Team-Maker* Version 1), it allowed instructors to write and edit their own questions. This feature was eliminated when *Team-Maker* was moved to a web-based interface (*Team-Maker* Version 2), but instructors can still add, remove, or re-order questions for each survey. Having a list of available questions makes it easier

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

to compare responses across surveys, facilitates research, deters instructors from using the system to ask inappropriate questions, and simplifies the interface. *Team-Maker's* inventory of variables can be expanded as users request that new questions be added to the current choices.

After the students complete the survey, the instructor assigns a decision/distribution rule/weight to each survey variable that indicates 1) whether the instructor wants students with similar or dissimilar responses to be grouped, and 2) how heavily that variable should be weighted when creating teams. The team-assignment algorithm generates a “question score” for each variable characterizing how well the team’s distribution of that variable complies with the instructor’s wishes—higher positive values are better. *Team-Maker's* algorithm then generates a “compliance score” for each team characterizing how well the team’s distribution of all variables complies with the instructor’s wishes—again, higher positive values are better. The team’s compliance score is the average of the team’s question scores on all variables. *Team-Maker* works by randomly assigning students to teams of the size specified by the instructor, calculating question scores and compliance scores, then iteratively changing the team assignments to attempt to maximize the minimum compliance score of the set of teams.

Team-Maker supports four types of questions: multiple-choice, choose-any-or-all-of, schedule-compatibility, and underrepresented-member. A different heuristic is used to generate the question scores for each type of question. The heuristics generally return a value on the interval [0, 1] with 0 representing homogeneity and 1 representing heterogeneity for that particular question. In the subsections that follow, we describe for *Team-Maker* Version 1 the computation of the question scores for the four types of questions, the computation of the compliance scores for the team, the max-min procedure that uses the compliance scores to find a good set of teams, and the summary statistics that the system provides to the instructor.

Multiple-choice question. In a multiple-choice question, the student picks exactly one item from a list of choices. A survey may have an unlimited number of multiple-choice questions. The score S_k^{mul} for the k th multiple-choice question (the superscript *mul* indicates multiple-choice) is given by

$$S_k^{mul} = \frac{1}{n} \sum_{i=1}^{N_k^{mul}} \bigvee_{j=1}^n r_{ij} \quad (1)$$

where student responses r_{ij} are interpreted using

$$r_{ij} = \begin{cases} 1 & \text{if student } j \text{ selects choice } i \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of students in the team and N_k^{mul} is the number of choices in the question. The OR operator (\vee) returns a value of 1 if *any* student in the i th team selects the i th choice and a value of zero if

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

no one in the team selects the i^{th} choice. The greater the number of options that the team members select in common, the closer the score approaches zero, indicating team homogeneity on this question.

To illustrate computing this question score, suppose the multiple choice question asks: "My overall GPA is in the range of (select one): a) 4.0-3.5; b) 3.4-2.8; c) 2.7-2.0; d) 1.9 or below." If a 5-member team had the GPA set {3.8, 3.6, 3.3, 2.7, 2.5} the heuristic returns a 3 (three of the possible four choices were checked by at least one student) divided by 5 (the number of students in the team), or $3/5 = 0.6$, a number closer to one than zero, indicating a team with some heterogeneity. If, in contrast, the team GPA set were all in one bin, for example, {2.8, 2.9, 3.0, 3.3, 3.4}, then the result of the heuristic is $1/5 = 0.2$, a number close to zero, indicating a team with homogeneity on this question. In this case, the GPA is entered as free response, but the aggregate data is still grouped into bins in this way.

Choose-any-or-all-of question. In a choose-any-or-all-of question, the student picks all appropriate items from a list of choices. A survey may have an unlimited number of choose-any-or-all-of questions. The score s_k^{aoa} for the k th choose-any-or-all-of question (the superscript *aoa* indicates any-or-all) is given by

$$s_k^{aoa} = \max \left\{ 0, 1 - \frac{1}{nR_k^{aoa}} \sum_{i=1}^{N_k^{aoa}} b_i^2 \right\} \quad (2)$$

where student responses r_{ij} are interpreted using

$$r_{ij} = \begin{cases} 1 & \text{if student } j \text{ selects option } i \\ 0 & \text{otherwise} \end{cases}$$

$$a_i = \sum_{j=1}^n r_{ij}$$

$$b_i = \begin{cases} 0 & a_i = 0 \text{ or } 1 \\ a_i & a_i \geq 2 \end{cases}$$

where n is the number of students in the team, N_k^{aoa} is the number of options in the question, and R_k^{aoa} is the number of responses by all team members to all options. The condition $b = 0$ for $\alpha = 0$ or 1 expresses our decision that having a choice selected by no students is the same level of heterogeneity as having a choice selected by only one student. The greater the number of options that the team members select in common, the closer the score approaches zero, indicating team homogeneity on this question. The heuristic includes the square of b so that the numerator (b^2) and the denominator ($n \times R$) are of the same order.

To illustrate computing this question score, suppose the choose-any-or-all-of question asks: "In which sports are you active this quarter (choose all that apply): football, soccer, baseball, basketball, swimming, lacrosse." For a team of four, with one member playing baseball and one on the

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

swim team, the b^2 term is zero and the resulting question score is 1 (perfectly heterogeneous). In contrast, for a team of four with three members on the soccer team, $b = 3$ (three members on the same team), $n = 4$ (four team members), and $R = 4$ (four responses), resulting in a question score of $(1-3^2/4^2)=(1-9/16) = 0.44$, a number close enough to zero to indicate a degree of homogeneity.

The weighting of the choose-any-or-all-of questions is treated in the same manner as described above for multiple-choice questions.

Schedule compatibility question. In the schedule-compatibility question, students enter times when they are unavailable to meet with their team outside of class. A survey may have only one schedule-compatibility question. The score s^{sch} for the schedule-compatibility question (the superscript sch indicates schedule) is given by

$$s^{sch} = \min \left\{ \frac{1}{h} \sum_{i=1}^H \left(1 - \prod_{j=1}^n r_{ij} \right), 1 \right\} \quad (3)$$

where student responses r_{ij} are interpreted using

$$r_{ij} = \begin{cases} 1 & \text{if student } j \text{ selects time block } i \\ 0 & \text{otherwise} \end{cases}$$

and where n is the number of students in a team, h is the number of compatible hours beyond which the developers deemed further compatibility unnecessary ($h = 40$ in *Team-Maker* Version 1), and H is the number of blocks of time in a week ($H = 119$ in *Team-Maker* Version 1). The OR operator returns a value of 1 if one or more students in a team are busy in the i th time block and a value of zero if everyone is available for team work in the i th time block. A high percentage of zeros in the schedule matrix indicates a high level of schedule compatibility, meaning that the team has an adequate number of hours to meet for team work outside of class. In (3), the only time blocks that improve a team's score are those in which *all* members of a team are free.

This heuristic returns a value on the interval $[0, 1]$ but unlike the previous heuristics, here the value of zero indicates complete heterogeneity (undesirable: the entire team never is free at the same time) and a value of one indicates adequate homogeneity (desirable: the entire team has at least 40 free hours in common). While it is possible to adjust (3) to reverse the heuristic, the inconsistency is rooted in the fact that we collect data from students on when they are **unavailable**, but we are ultimately interested in when they are **available**. Thus, the reverse-coded nature of this heuristic is expected and adjusting the equation would actually confuse matters further.

To illustrate computing this question score, suppose the aggregate responses of a team of four to the schedule compatibility question are summarized as shown in Table 1. Here, a result of "0%" in a time block indicates that no one on the team is busy (thus everyone is available for team work)

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Hour	Mon	Tue	Wed	Thu	Fri	Sat	Sun
7 am	0%	0%	0%	0%	0%	0%	25%
1 st hr	75%	75%	75%	75%	50%	25%	25%
2 nd hr	50%	50%	75%	50%	50%	25%	25%
3 rd hr	50%	50%	75%	50%	25%	25%	25%
4 th hr	0%	25%	0%	25%	0%	0%	0%
5 th hr	25%	50%	0%	50%	25%	0%	0%
6 th hr	75%	25%	75%	50%	50%	0%	0%
7 th hr	25%	50%	50%	50%	25%	0%	0%
8 th hr	75%	100%	100%	100%	100%	0%	25%
9 th hr	75%	100%	100%	100%	100%	0%	50%
10 th hr	25%	75%	25%	100%	25%	0%	50%
5 pm	25%	25%	25%	25%	25%	0%	25%
6 pm	25%	50%	50%	50%	25%	25%	25%
7 pm	25%	0%	25%	25%	0%	25%	0%
8 pm	50%	0%	25%	0%	0%	25%	25%
9 pm	50%	0%	25%	0%	0%	25%	25%
10 pm	25%	0%	25%	0%	0%	25%	25%

Table 1: Sample schedule-compatibility results: % of team busy.

and a result of “100%” in a time block indicates that everyone on the team is busy (thus not available for team work).

This sample team has 34 time blocks with a result of 0% (everyone available). In this case the summation in (3) returns a value of 34, and the score s^{sch} is given by $34/40 = 0.85$, a number close to 1, indicating schedule compatibility. Three additional cases illustrate the heuristic:

- Complete incompatibility (no time blocks are 0%): this occurs if every time block has been marked busy by at least one member of the team. The summation in (3) returns a zero, and the score $s^{sch} = 0$.
- Further compatibility unnecessary (forty time blocks are 0%): the summation in (3) returns a $40/40 = 1$, and the score s^{sch} is 1.
- Complete compatibility (all time blocks are 0%): the summation in (3) returns a $119/40 = 2.975$, and the score s^{sch} is 1.

Because schedule compatibility is always desirable, the weights for the scheduling question are in the reverse order from the weights for the previous two questions, i.e., the 11 positions on the scale correspond to the set of weights $\{+5, \dots, +1, 0, -1, \dots, -5\}$ where a positive weight indicates an instructor’s desire for homogeneity (gather similar). The negative values are not expected to be used. In *Team-Maker* Version 2, negative weights are disallowed—instructors may not intentionally group for schedule incompatibility.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

To illustrate the use of weights with the schedule-compatibility question, suppose a heterogeneous team's question score is 0.85 as in the example given above. If the instructor selects a weight of +4 (compatibility desired, but not the most important question), then the score-weight product is $0.85 \times 4 = 3.4$, which adds to the team's compliance score. A higher weight would increase the influence of schedule compatibility on the compliance score. If the team has a question score of zero (complete incompatibility), then to increase the team's compliance score, the algorithm attempts to improve schedule compatibility. In contrast, if the team already has a score of 1 (the maximum compatibility score attainable), then the algorithm seeks no greater levels of schedule compatibility.

Pairing of underrepresented team members. A survey usually has no more than two questions regarding underrepresentation (gender and/or race/ethnicity). Females and all racial/ethnic groups that are not "white, non-Hispanic" are treated as underrepresented groups. The score s_k^{urm} for the k^{th} underrepresented member (the superscript *urm* indicates underrepresented member) is given by

$$s_k^{urm} = \begin{cases} -1 & a_i = 1 \\ 0 & a_i = 0 \\ 1 & a_i \geq 2 \end{cases} \quad (4)$$

where student responses r_{ij} are interpreted using

$$r_{ij} = \begin{cases} 1 & \text{if student } j \text{ selects underrepresented category } i \\ 0 & \text{otherwise} \end{cases}$$

$$a_i = \sum_{j=1}^n r_{ij}$$

and where n is the number of students in a team. For a team with only one underrepresented member, $a = 1$ and the question score is -1 , decreasing the team's compliance score to account for the fact that the underrepresented member is outnumbered on the team. For the case in which a team has no underrepresented members, the question is irrelevant, so $\alpha = 0$ and the question score is 0 and has no effect on the team's compliance score. For cases where specific underrepresented members are at least paired on a team, $\alpha \geq 2$ and the question score is $+1$. Note that the algorithm implements "not being outnumbered" on a team by "at least pairing" underrepresented members on a team. Consequently, if a class includes only three women, for example, the algorithm would tend to place them all on the same team (subject to the weight given to other criteria).

To illustrate computing this question score, suppose that the instructor wishes to prevent the outnumbering of women and Black students on teams of four. Two survey questions would be categorized as underrepresentation questions: one for gender and one for race. Consider the following examples for teams of four (undesignated members are white males):

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

- For a team with one White female and one Black male, the gender question score is -1 and the race question score is -1 .
- For a team with two Black males, the gender question score is 0 and the race question score is $+1$.
- For a team with one Black male, one Black female, and one White female the gender question score is $+1$ and the race question score is $+1$.

This last case illustrates an overly simple characteristic of this heuristic: with a Black male and a Black female on the same team, the heuristic computes that Black team members are not outnumbered (meeting the instructor’s goals) but we know that gender can confound this pairing. Likewise, with two women on the team, the heuristic computes that women are not outnumbered (meeting the instructor’s goals), but we know that race can complicate this pairing because Black women and White women are less likely to have shared experience. In *Team-Maker* Version 2, a more sophisticated heuristic accounts simultaneously for race and gender if “outnumbering” is to be prevented.

Because avoiding outnumbering underrepresented members on a team is desirable if this type of question is used for forming teams in engineering classes (Rosser 1998), the weights for this question have positive values only, from 0 to $+5$. To illustrate, suppose the instructor selects a weight of $+5$ to indicate a strong desire to prevent outnumbering of underrepresented members. Using the cases listed above:

- For a team with one White female and one Black male, the gender question score is -1 and the race question score is -1 . Both score-weight products are $-1 \times 5 = -5$, and both results lower the team’s compliance score. To increase the team’s compliance score, the algorithm attempts to pair the White female with another female and to pair the Black male with another Black student, if possible.
- For a team with two Black males, the gender question score is 0 and the race question score is $+1$. The gender score has no effect on the compliance score and the race score increases the compliance score by $+5$.
- For a team with one Black male, one Black female, and one White female the gender question score is $+1$ and the race question score is $+1$. The two combined raise the compliance score by $+10$.
- A weight of zero, as with all survey questions, causes the algorithm to ignore the underrepresented member question.

Team compliance score. The compliance score for a team, C , is computed from the weighted sum of the question scores given by

$$C = \sum_{k=1}^{Q^{mul}} S_k^{mul} W_k^{mul} + \sum_{k=1}^{Q^{aoa}} S_k^{aoa} W_k^{aoa} + S^{sch} W^{sch} + \sum_{k=1}^{Q^{urm}} S_k^{urm} W_k^{urm} \tag{5}$$

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

where S_k^{mul}, W_k^{mul} score and weight for the k th multiple-choice question

S_k^{aoa}, W_k^{aoa} score and weight for the k th choose-any-or-all-of question

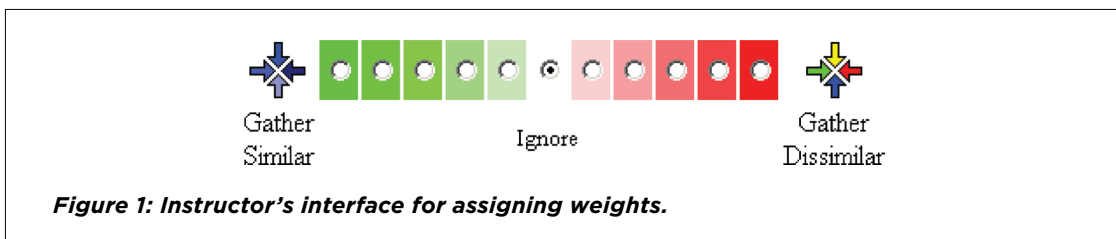
S_k^{sch}, W_k^{sch} score and weight for the schedule-compatibility question

S_k^{urm}, W_k^{urm} score and weight for the k th underrepresented-member question

and where Q^{mul} is the number of multiple-choice questions, Q^{aoa} is the number of choose-any-or-all-of questions, and Q^{urm} is the number of underrepresented-member questions in the survey. It is assumed that only a single schedule-compatibility question is included in the survey. Again, relatively high positive scores indicate a greater degree of compliance with the instructor's wishes than relatively lower scores. In *Team-Maker* Version 2, dividing by the maximum possible team score (the sum of the weights) normalizes each survey so that team compliance scores can be compared from one survey to another.

Assigning weights to each question score. The instructor assigns a decision rule/weight (hereafter called "weight" for simplicity) to a multiple-choice question on a scale as illustrated in Figure 1. The 11 positions on the scale correspond to the set of weights $\{-5, \dots, -1, 0, +1, \dots, +5\}$ where a negative weight indicates an instructor's desire for homogeneity (gather similar) and a positive weight indicates a desire for heterogeneity (gather dissimilar). The larger the magnitude of the number, the greater the importance placed by the instructor on that particular question. A weight of zero causes the question to be ignored when computing question scores. If all the weights are set to zero, the effect is that teams are randomly assigned.

To illustrate the use of weights, suppose a heterogeneous team's GPA question score is 0.6 as in the example given above. If the instructor selects a weight of +5 (largest weight for heterogeneity), then the product of the score and the weight is $0.6 \times 5 = 3.0$. This positive value increases the team's compliance score, where a higher relative compliance score indicates "meeting the instructor's criteria". Suppose instead that the instructor wishes students with similar GPAs to be grouped together and so selects a weight of -5 (largest weight for homogeneity). The score-weight product is negative, $0.6 \times 5 = -3.0$, decreasing the team's compliance score. Recall that meeting the instructors' wishes means increasing the compliance score, so the algorithm iteratively



Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

attempts to group students together with similar GPAs. From the example earlier (at the end of the *Multiple-choice question* section), if a team is assembled such that all the team members are in the same GPA bin (complete homogeneity), then the question score is 0.2, the score-weight product is $0.2 \times 5 = -1.0$, and the compliance score, by being reduced by as little as possible, is as large as it can be in this case. Again, in *Team-Maker Version 1*, specific numeric values of compliance scores have no particular significance; only the relative values of compliance scores among teams and from iteration to iteration have significance.

Optimization strategy. The search for a “best” set of teams is based on the hill-climbing algorithm (Russell and Norvig 1995), which at best finds local maxima. The cost function of the search is the weighted compliance score. Repeating this algorithm from various starting points makes it more likely that the maximum found is global, but that cannot be assured.

The search begins by randomly assigning the entire class to teams of a size selected by the instructor. The compliance scores for the first two teams are computed. Next, a team-member exchange (swap) is made and new compliance scores are computed. If the lowest of the two original compliance scores has been improved, the swap is kept; if not, the swap is undone. This process is repeated for every combination of paired members of these two teams and then repeated for every combination of paired teams in the class. Pseudo-code that explains the swapping process is shown in Figure 2. There is a built-in limit of 20 passes through the team-swapping loop because convergence is not guaranteed.

The result of these 20 passes is a set of teams with the highest minimum team compliance score (the max-min) for a particular starting condition. This approach makes the worst-fitting team have as good a fit as possible. Because the algorithm started from a particular random assignment of students to teams, the max-min score is at best a local maximum. Thus the random assignment of students to teams as a starting point is conducted multiple times, creating a team formation, or “outer”, loop

```

repeat
  for each teamA from 1 ... (num_teams-1)
    for each teamB from (teamA+1) ... num_teams
      for each studentA in teamA
        for each studentB in teamB
          old = min(score(teamA), score(teamB))
          swap studentA and studentB
          new = min(score(teamA), score(teamB))
          keep swap if (new > old), otherwise revert
        end
      end
    end
  end
end
until no swaps succeed (maximum 20 passes)

```

Figure 2: Pseudo-code for the team-member swapping procedure.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

that repeats the team-swapping process. If the new max-min compliance score is greater than the previous max-min score, then the set of teams from the second iteration is saved; if not, the set of teams from the first iteration is kept. This “outer” iterative loop is repeated 50 times.

Idiosyncrasies of the algorithm include:

- The algorithm runs through all students in all teams in order in every pass.
- The algorithm does not specifically try to improve the score of the lowest team—it just runs through all possible pairs of teams and tries member swaps; if the lower score between those two teams improves, then it keeps the swap.
- The built-in limit of 20 passes through the team-swap loop is necessary because of the possibility of infinite loop swapping, i.e., a swap improves the two teams under immediate consideration but gets undone by later swapping.
- The “outer” loop limit of 50 was found to be effective through experimental trials. Additional iterations did not seem to significantly improve the minimum compliance score and, for the size of the classes being tested (30 students in a class, teams of 4 or 3), 50 iterations were not computationally expensive.

Recently, the 50-trials outer-loop limit has been found acceptable even in large classes, as the program typically converges within 20 minutes with a class size as large as 1500. While additional iterations of the outer loop limit help ensure finding the highest possible compliance score, that improvement comes at the cost of additional computational time. The 50 iterations used by the algorithm provides acceptable results (as shown below) in a reasonable time-time that is offset by the time saved by automating team assignments. Further, automated team assignment can be run as a background task on the instructor’s computer while the instructor engages in other work, because the intensive computation is performed by a remote computer.

Summary statistics for the instructor. Once a “best” set of teams is found, a set of summary statistics is reported to the instructor. If the assigned teams are not to the instructor’s liking, he or she can assign new question weights, obtain a new set of teams, review the summaries, and repeat until the assignments are satisfactory.

The summaries include both numerical statistics and graphs of the response distributions for each question. These team-by-team summaries serve as measures of how well the teams meet the instructor’s team-formation criteria. For example, Figure 3 (from *Team-Maker* Version 1) shows a representative team’s summary of responses to a question asking for their grades in a prerequisite course. This is a team of four in which one student (25%) reported an “A”, one student (25%) reported a “B+”, and two students (50%) reported “B or C+”. If the instructor desired team heterogeneity of this attribute, then this team is satisfactory.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Question #6: Your grade in ES203 Electrical Systems (a prerequisite for ES205) was:

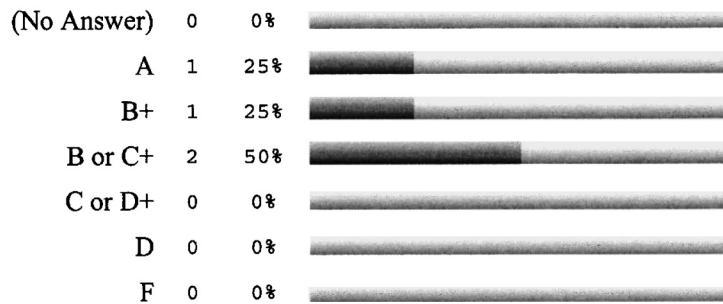
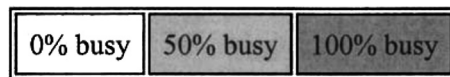


Figure 3: A representative team's prerequisite-grade summary.



Percent busy by hour

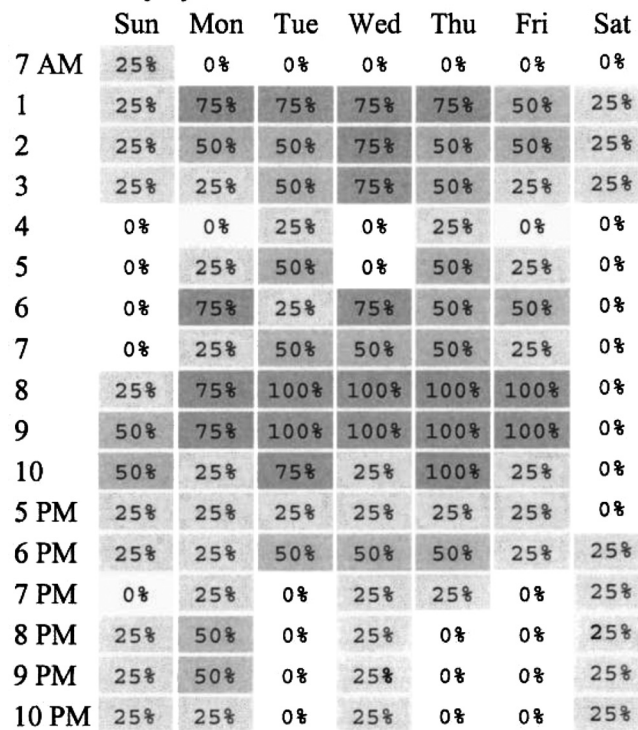


Figure 4: A representative team's schedule-compatibility summary.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Figure 4, also from *Team-Maker* Version 1, shows a representative team's summary of responses to the schedule-compatibility question. The hours marked "1" through "10" correspond to "first hour" through "tenth" hour—an idiosyncrasy of the class schedules at Rose-Hulman Institute of Technology. (For broader applicability, *Team-Maker* Version 2 simply uses the normal hours of the day.) All boxes labeled "0%" indicate times at which all members of the team are free to meet outside of class; boxes labeled "100%" indicate times at which all team members are busy and are unable to meet for team work. For the team shown, the members have 34 time blocks in common (this is the summary from which Table 1 was taken). More importantly, the summary shows that there are five blocks of time during the week during which all four team members have *three hours or more* available to meet. This attribute of schedule compatibility is not purposefully sought by the *Team-Maker* algorithm, hence the importance of the summary (when viewed by both faculty and students). If this team had not had at least one or two of these 3-hour blocks of time, the instructor would very likely deemed this an unsatisfactory team assignment and would increase the weight of the schedule-compatibility question and re-run the algorithm.

The summary statistics generated by *Team-Maker* will be useful for research to model how individual and team-level variables affect team performance or other dependent variables of the researcher's choice. Once *Team-Maker* Version 2 establishes compliance scores that have absolute rather than relative meanings, researchers will be able to use the team's question scores and compliance scores as independent variables predicting outcomes of interest. Additional details of the *Team-Maker* interface are available ([Cavanaugh et al., 2004](#)).

TESTING THE VALIDITY OF THE *TEAM-MAKER* ALGORITHM

The success of the *Team-Maker* (Version 1) software at meeting its objectives was tested in a study conducted at Rose-Hulman Institute of Technology during the Spring 2003 semester. Three instructors, teaching 86 students in four sections of a sophomore-level System Dynamics course, formed teams in their own sections using student responses to a paper-and-pencil survey asking questions about the students' engineering disciplines, GPAs, grades in a prerequisite course, schedule compatibility, and gender. The students were primarily mechanical engineering, electrical engineering, and computer engineering majors. Students were assigned to teams to do work associated with the weekly 3-hour lab period. A total of 24 teams were assigned.

Later that summer, the paper-and-pencil survey responses from the quarter were transcribed to the *Team-Maker* program. New teams were assigned using *Team-Maker* and the resulting set of 24 compliance scores—the "automated set"—was recorded. Then the program feature allowing

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

manual team assignment was used to reassign teams to match the set originally (and manually) assigned by the three instructors, producing a “manual set” of 24 compliance scores. In all cases, we sought to form teams of four with heterogeneous disciplinary interests in which women were not outnumbered and student schedules were compatible. The comparison of the two sets of compliance scores is our validity test.

Since we are comparing the performance of the software to the performance of only three instructors, our validity test has a sample size of three, suggesting a need for future work in which the performance of the software is compared to the team-assignment performance of a larger sample of experienced instructors.

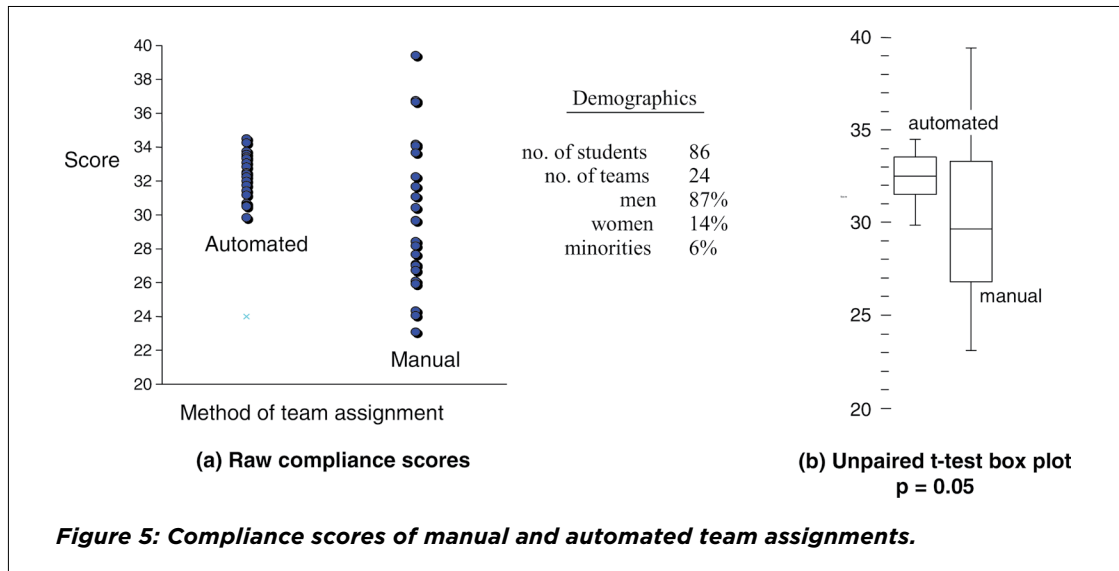
The comparison of the descriptive statistics of the two sets of compliance scores are shown in Table 2; higher scores are better. The mean compliance score for automated team creation is 8% higher than the mean score for teams created manually. What may be of even greater importance, however, is that the **lowest** compliance score for the teams created by the automated system is 29% higher than the lowest compliance score for the teams created manually. This means that the team that least meets the specified criteria comes considerably closer to meeting the instructor’s criteria with automated team selection.

As shown in Table 2, the standard deviation for the automated process is less than 1/3 that of the manual selection, meaning that the automated process meets the specified criteria much more consistently than do the experienced instructors. Figure 5(a) illustrates that the raw compliance scores for the instructor-assigned teams have a greater range than the compliance scores for the *Team-Maker*-assigned teams. This result is consistent with the max-min logic of the *Team-Maker* algorithm, which attempts to maximize the score of the lowest-scoring team, even if it means lowering the score of the highest-score team. Therefore, as shown in Table 2, the manual assignment did have a higher maximum compliance score than the automated assignment did. However, because the objective of team assignment is to balance teams according to the specified criteria, having

Team formation	Team Compliance Scores			
	Mean	Min	Max	Std Dev
Automated, $N = 24$	32.4	29.8	34.5	1.3
Manual, $N = 24$	30.0	23.1	39.4	4.3
Result of using Automated System	+8%	+29%	-12%	1/3 the variability
P-value for unpaired t-test	0.05	0.001		0.001

Table 2: Comparing automated to manual team assignments, all sections.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria



one or a few teams that very closely meet those criteria while other teams are highly deficient is not beneficial for the learning environment.

We use an unpaired t-test to determine statistical significance. The “manual set” of 24 teams is independent of the “automated set” of 24 teams, even though they are assigned from the same pool of 86 students. Assuming the scores are normally distributed, the mean automated score is 32.43 ± 0.55 and the mean manual score is 30.01 ± 1.86 , both with 95% confidence. This gives a 95% confidence interval of [31.88, 32.98] for the automated teams and [28.15, 31.87] for the manual teams. Because the confidence intervals do not overlap, the difference of 2.42 in the means is statistically significant. Thus, the difference in means is significant for $\alpha < 0.05$ —evidence that *Team-Maker* (Version 1) forms teams that are a better fit to the criteria than teams created by these three experienced instructors. The lower variability of the compliance scores of the automated system (as measured by range and standard deviation) is evidence that the instructors’ team-formation criteria are met more consistently by the program than by the instructors. These results are presented graphically in the box plot in Figure 5(b).

The closeness of the average automated score to the average manual score is one indication that the heuristics are achieving the instructor’s objectives in forming teams, thus establishing concurrent validity of the *Team-Maker* algorithm. The application creates teams that meet the instructor-specified criteria both better and more uniformly than teams assigned manually by the instructor. Table 3 shows the results for each section separately. For every section, the automated heuristic score is greater and the standard deviations are smaller than the scores and deviations for the teams created manually, which is further evidence that the improved results are consistent.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Section	Automated score (std. dev.)	Manual score (std. dev.)
5	33.07 (0.93)	31.64 (5.11)
1	32.65 (1.46)	29.03 (3.77)
6	32.38 (0.56)	29.32 (4.23)
2	30.88 (0.90)	29.30 (4.17)

Table 3: Comparing automated to manual team assignments, by sections.

CRITERIA FOR TEAM FORMATION

Having demonstrated that the *Team-Maker* program can reliably assign members to teams based on instructor-specified criteria, we address our choice of criteria. We also briefly review literature related to criteria that instructors should consider when using the program to assign students to teams, particularly in undergraduate engineering classes.

Brickell and colleagues (1994) compared teams formed with all possible combinations of heterogeneity vs. homogeneity of ability and heterogeneity vs. homogeneity of disciplinary interest. The two combinations with one factor heterogeneous and the other homogeneous had significantly higher group grades than a comparison group of self-selected teams. Further, the teams formed this way developed better attitudes about the course and its administration, and made more efficient use of time spent on course work than the other types of teams in the study. The value of some providing some homogeneity in team formation may be in creating team cohesion (Wolfe and Box 1988; Hogg 1996; Gosenpud and Miesing 1984; [Jaffe 1990](#)). Distributing members to teams based on heterogeneity of ability (typically measured by GPA or a previous course grade) has been found to improve the average performance of teams ([Heller 1992](#)). The active and cooperative learning literature finds support for the learning benefit of forming teams of heterogeneous ability ([Hake 1998](#)). While teams with higher average cognitive ability among the team members consistently perform better (Horwitz 2007), creating teams of homogeneous ability would have a negative effect on teams comprised of lower-functioning students. Hilborn ([1994](#)) provides further (anecdotal) evidence to support the practice of forming engineering student teams on the basis of heterogeneous ability.

The value of disciplinary heterogeneity is highly dependent on the context—it should enhance team performance if the team's task requires a multidisciplinary team. In engineering education contexts, pressure from accreditation criteria ([ABET 2000](#)) makes heterogeneity of discipline preferable whenever feasible ([O'Doherty 2005](#); [Wesner et al. 2007](#)). Additional benefits of disciplinary diversity are increased team-member participation and more communication within the team

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

(Cohen 1995; Jacobson 2001) as well as improved knowledge transfer in multidisciplinary problem solving teams (Fenner 2001).

Some researchers assert that gender and race must be considered when forming teams in engineering education contexts (Rosser 1998; [Tonso 2006](#)). Heller and Hollabaugh (1992) observed that the voices of female students (even those having the highest ability on a team) tend to be silenced if they are outnumbered by dominant male voices in a group. Cady and Valentine's (1999) research suggests that members of underrepresented groups experience a similar loss of voice if they are outnumbered. However, if gender and race are used as criteria for team assignments, it is important for faculty to avoid drawing attention to ("spotlighting") differential treatment of women and minorities ([McLoughlin 2005](#)). For a more detailed discussion of using gender and race in engineering team formation, see ([Cordero 1997](#); Haag 2000). For a discussion of the complex mechanisms through which demographic composition might affect team performance, see Harrison et al. (2002) and Rentsch and Klimoski (2001). However, a 2007 meta-analysis found no significant affects of bio-demographic diversity (age, gender, and race/ethnicity) on the quantity of team performance in the 3 studies that examined it, and no significant effect on quality of team performance for the 14 studies that examined that relationship (Horwitz 2007).

Regardless of what challenges a team faces, team members must interact to address them. This requires at least some degree of schedule congruence. Up to 90% of student teams have difficulty finding a common time to meet ([Jaffe 1990](#)). Certainly, the challenge of finding a common meeting time increases with the team size. The Foundation Coalition for Engineering Education (Coalition 2002) recommended forming teams that have common time in their schedules to meet, and that teams establish meeting times early in the semester before extracurricular commitments complicate the students' schedules. Although schedule compatibility problems are a common complaint among students, there is little research on this criterion.

Additional engineering literature on team assignment describes research (Ogot and Okudan 2006; [Hunkeler 1997](#); McCaulley 1983, 1985; [Tonso 2006](#)) and practice ([Jack 2007](#); [Salama, Rizkalla, and Yokomoto 2004](#); [Brewer and Mendelson 2003](#); Drnevich 2007) on other criteria that could be considered when forming teams such as practical experience, personality type, and learning style. Literature from management and psychology shows that team composition criteria that may affect learning and team performance are varied and have complex direct and indirect effects (Stewart 2006; [Gibson 2003](#); Hamilton 2003).

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This paper described the development and validation of the *Team-Maker* program for the assignment of students to teams based on criteria that instructors specify. The results indicate that the

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

program is able to accomplish team-assignment tasks more effectively than experienced instructors. In addition, the program dramatically decreases the instructor time required to assign teams, making it possible for instructors to assign teams based on many criteria, even in large classes. Important benefits of the *Team-Maker* system are automating the team-assignment process, allowing faculty time to explore multiple solutions to the problem; increasing the likelihood that instructors' team-formation criteria are met consistently and to a greater extent than with manually-assigned teams, and providing a "compliance score" to assess the extent to which those criteria are met.

While the results presented here show that *Team-Maker* is already an effective and efficient means of assigning students to teams, there is still room for improving its algorithm. Some parts of the question scoring algorithm do not use the full range of the score scale. In the case of choose-any-or-all questions, for example, a completely homogeneous team has a score of nearly 0.5, rather than 0. Improved scoring algorithms are implemented for *Team-Maker* Version 2, and results from testing the new scoring algorithm will be published later. (Version 2 is the current, supported version of the software—Version 1 is no longer supported.)

Team-Maker Version 2 addresses some of the idiosyncrasies of the hill-climbing algorithm. After the initial random assignment, the teams are ordered by compliance score, lowest to highest. Then, starting with the lowest scoring team, a swap is tried. If the swap is successful (improves the lowest compliance score), then the teams are re-ordered, and the algorithm continues with the new lowest-scoring team. Once the lowest score can no longer be improved, the algorithm moves on to the next lowest scoring team, and so on until no improvements can be made. This approach improves the compliance score of all teams and is certain to converge.

Future research could examine the effects of cost functions other than maximizing the minimum compliance score, for example, minimizing the total deviation from the mean or minimizing the sum of the squares of the deviations. One could explore the sensitivity of the results to the type of cost function and compare the descriptive statistics of teams resulting from and the computational expense of different cost functions.

Future work might also include a new validity test to compare the performance of the software to the performance instructors experienced in manually assigning students to teams, where the size of the sample of instructors is larger than our sample of three. Such research would likely be based on *Team-Maker* Version 2.

Additional research is also needed to better understand how various team formation criteria affect student learning. There may be complex interactions among criteria that have not been considered in past research. The *Team-Maker* program will facilitate future research on how team formation strategies affect team success and student outcomes, both by making team assignments easier and because the survey function and summary statistics provided in the *Team-Maker* system will make

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

collecting data easier. It is important to note that although *Team-Maker* was developed for assigning students to learning teams, the program will work for any type of team assignments. Thus, *Team-Maker* could be used to facilitate team-related research in non-academic contexts.

We believe the risks of allowing student teams to self-select outweigh the benefits, especially in classes early in the curriculum, when students have both the least expertise and the least knowledge of other students. We suggest that automated team formation by *Team-Maker* is an excellent compromise, in that students still have input to the process. Today's students have a high level of comfort with technological solutions and are likely to welcome this solution and see it as fairer and less biased than team assignment by instructors. Enhancements to the *Team-Maker* system are ongoing. A new user interface coordinates *Team-Maker* with the *Comprehensive Assessment of Team-Member Effectiveness*, a web-based peer evaluation system. This provides instructors with a comprehensive solution for team formation and team-member monitoring. As development on this system continues, we will re-visit the definitions of each heuristic and consider whether additional heuristics should be developed to allow additional types of variables to be considered in team formation. In addition to the system's utility in the classroom for implementing what is already known about team formation, the validated software can be used for further research on teams.

FOR MORE INFORMATION

Online resources for readers interested in the *Team-Maker* and CATME systems include a [video walk-through](#) with commentary by an expert user, a step-by-step [written tutorial](#), and a [sample instrument](#). In addition, the combined *Team-Maker/CATME* system is one of two winners of the [2009 Premier Award for Excellence in Engineering Courseware](#). The [award application](#) includes a discussion of the combined system's student-focused characteristics such as learning objectives, interactivity, student cognitive change, use of media, and instructional use as well as software-design characteristics such as engagement, learner interface and navigation, and technical reliability. The functional interface is free for educational use. Interested users can [request a faculty account](#), which is approved after it is confirmed that the email address provided corresponds to someone with instructional responsibility.

ACKNOWLEDGEMENTS

We acknowledge the Educational Research and Methods division of the American Society for Engineering Education for providing financial support for the initial development of *Team-Maker*,

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

and the Rose-Hulman Institute of Technology for providing matching funds. Former RHIT students Ryan Cavanaugh and Matt Ellis developed the initial software package in 2003-2004 as part of CS414 and CS415 Software Engineering I and II and students Brant Gurganas and Phillip Doup continued the development in 2004-05. Other students on the project were David Aramant, Mark Newheiser, Brian Klimaszewski, Brian Kopecky, and Robert Drake, advised by Don Bagert. Mark Ardis helped bridge the cultural divide between engineering faculty and software-engineering students, Ray Bland maintained the server, and numerous RHIT faculty tested the package during development. Students in ES205 and ME311 at RHIT also participated in testing the system and provided valuable test data. Support for further development, including incorporating *Team-Maker* into the Comprehensive Assessment of Team Member Effectiveness, was provided by the National Science Foundation through Award ASA-0243254, *Designing a Peer Evaluation Instrument that is Simple, Reliable, and Valid*. Team-Maker is currently being maintained and integrated into a larger framework of teamwork tools with the support of National Science Foundation Award DUE-0817403, *SMARTER Teamwork: System for Management, Assessment, Research, Training, Education, and Remediation for Teamwork*.

REFERENCES

- [ABET Criteria for Accrediting Engineering Programs](#), 2009-2010 Accreditation Cycle, ABET Engineering Accreditation Commission, Baltimore, Maryland.
- Bacon, Donald R., Kim A. Stewart, and Elizabeth Scott Anderson. 2001. Methods of Assigning Players to Teams: A Review and Novel Approach.
- Bacon, Donald R., Kim A. Stewart, and William S. Silver. 1999. Lessons from the Best and Worst Student Team Experiences: How a Teacher Can Make the Difference.
- Bacon, Donald R., Kim A. Stewart, and Sue Stewart-Belle. 1998. Exploring Predictors of Student Team Project Performance.
- Brewer, W., and M. I. Mendelson. 2003. [Methodology and metrics for assessing team effectiveness](#). *International Journal of Engineering Education* 19 (6):777-787.
- Brickell, J. L., D. B. Porter, M. F. Reynolds, and R. D. Cosgrove. 1994. [Assigning students to groups for engineering design projects: A comparison of five methods](#). *Journal of Engineering Education* 83 (3):259-262.
- Bruffee, K. A. 1993. *Collaborative learning: Higher education, interdependence, and the authority of knowledge*. Baltimore: John Hopkins University Press.
- Cady, S. H., and Valentine, J. 1999. Team innovation and perceptions of consideration: What difference does diversity make? *Small Group Research* 30 (6):21.
- Cavanaugh, R., M. Ellis, R. Layton, and M. Ardis. 2004. [Automating the process of assigning students to cooperative-learning teams](#). Paper read at American Society for Engineering Education Annual Conference & Exposition, at Salt Lake City, Utah.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Coalition, Foundation. 2002. Student Teams in Engineering Sample Schedule Form. In *Website for Active and Cooperative Learning*.

Cohen, L.L., and Swin, J.K. 1995. The Differential Impact of Gender Ratios on Women and Men: Tokenism, Self-Confidence, and Expectations. *Personality Social Psych. Bull.* 21:9.

Cook, Robert W. 1981. An Investigation of Student Peer Evaluation on Group Project Performance.

Cordero, R., DiTomaso, N., and Farris, G.F. 1997. [Gender and race/ethnic composition of technical work groups: Relationship to creative productivity and morale](#). *Journal of Engineering and Technology Management* 13 (3-4):205-221.

Daly, Joseph P., and Dan L. Worrell. 1993. Structuring Group Projects as Miniature Organizations.

Decker, R. 1995. Management team formation for large scale simulations. In *Developments in Business Simulation & Experiential Exercises*, edited by J. D. Overby and A. L. Patz. Statesboro, GA: Association for Business Simulation and Experiential Learning.

Dipinto, V. M., and S. V. Turner. 1997. Students and teacher as co-conspirators in learning. *Current Issues in Middle Level Education* 6:29-39.

Drnevich, Vincent, and John Norris. 2007. Assigning Civil Engineering Students to Capstone Course Teams. Paper read at Proceedings of the ASEE Annual Conference at Honolulu, HI.

Feichtner, Susan Brown, and Elaine Actis Davis. 1984. Why Some Groups Fail: a Survey of Students' Experiences with Learning Groups.

Felder, R. M. 1993. [A longitudinal study of engineering student performance and retention. I. Success and failure in the introductory course](#). *Journal of Engineering Education* 82 (1):15-21.

Felder, R.M., D.R. Woods, J.E. Stice, and A. Rugarcia. 2000. The Future of Engineering Education II. Teaching Methods that Work. *Chem. Engr. Education* 34 (1):14.

Fenner, K., Jodicke, G., Alean-Kirkpatrick, P., and Hungerbuhler, K. 2001. Introducing Integrated product and process development into the education of science and engineering undergraduates: A lecture course with an accompanying case-study programme at the ETH chemistry department. *Chemosphere* 43:12.

Gibson, C. 2003. A healthy divide: Subgroups as a stimulus for team learning behavior. *Administrative Science Quarterly* 48 (2):202. ([abstract](#))

Gosenpud, J., and P. Miesing. 1984. Determinants of performance in computer simulations. *Developments in Business Simulation & Experimental Exercises* 10:53-56.

Haag, S.G. 2000. Teaming Backlash: Reframing Female Engineering Students. Paper read at Proceedings, American Society of Engineering Education Annual Conference, at St. Louis, MO.

Hake, R. R. 1998. [Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses](#). *American Journal of Physics* 66:64.

Hamilton, B. H. 2003. Team incentives and worker heterogeneity: An empirical analysis of the impact of teams on productivity and participation. *The Journal of Political Economy* 111 (3):465.

Harrison, D. A. 2002. Time, teams, and task performance: Changing effects of surface-and deep-level diversity on group functioning. *Academy of Management Journal*:1029.

Heller, P. and Hollabaugh, M. 1992. [Teaching Problem Solving through Cooperative Grouping. Part 2: Designing Problems and Structuring Groups](#). *American Journal of Physics* 60 (7): 637-644.

Hilborn, R. B. 1994. Team learning for engineering students. *IEEE Transactions on Education*, 37 (2):207-211. ([abstract](#))

Hogg, M.A., and Hains, S.C. 1996. Intergroup relations and group solidarity: Effects of group identification and social beliefs on depersonalized attraction. *Journal of Personality and Social Psychology* 70 (2):15.

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

- Horwitz, S. K. 2007. The effects of team diversity on team outcomes: A meta-analytic review of team demography. *Journal of management* 33 (6):987.
- Hunkeler, David, and Julie E. Sharp. 1997. [Assigning Functional Groups: The Influence of Group Size, Academic Record, Practical Experience, and Learning Style](#). *Journal of Engineering Education* 86 (4):321-332.
- Jack, Hugh. 2007. [A Methodology for Assigning Project Teams](#). Paper read at American Society for Engineering Education Annual Meeting.
- Jacobson, D.L. 2001. A new agenda for education partnerships: Stakeholder learning collaboratives. *Change*:10.
- Jaffe, E.D., and Nebenzahl, I.D. 1990. Group interaction and business game performance. *Simulation & Gaming* 21 (2):14. ([abstract](#))
- Jalajas, David S., and Robert I. Sutton. 1984. Feuds in Student Groups: Coping with Whiners, Martyrs, Saboteurs, Bullies, and Deadbeats.
- Janis, Irving L. 1982. *Groupthink: psychological studies of policy decisions and fiascoes*. 2nd ed. Boston: Houghton Mifflin.
- Johnson, D.W., R.T. Johnson, and K.A. Smith. 1998. *Active learning: Cooperation in the college classroom*. Edina, MN: Interaction Book Co.
- Johnson, David W., Roger T. Johnson, and Karl A. Smith. 1991. *Cooperative learning: increasing college faculty instructional productivity*. Washington, DC: School of Education and Human Development, George Washington University.
- Katzenbach, Jon R., and Douglas K. Smith. 1993. *The wisdom of teams : creating the high-performance organization*. Boston, Mass.: Harvard Business School Press.
- Kirchmeyer, Catherine. 1993. Multicultural Task Groups: An Account of the Low Contribution Level of Minorities.
- McCaulley, M.H., Godleski, E.S., Yokomoto, C.F., Harrisberger, L., and Sloan, E.D. 1983. Applications of Psychological Type in Engineering Education. *Engineering Education* 73 (5):7.
- McCaulley, M.H., Macdaid, G.P., and Granade, J.G. 1985. ASEE-MBTI Engineering Consortium: Report of the First Five Years. Paper read at Proceedings, 1985 ASEE Annual Conference.
- McLoughlin, Lisa A. 2005. [Spotlighting: Emergent Gender Bias in Undergraduate Engineering Education](#). *Journal of Engineering Education* 94 (4):373-381.
- Mello, Jeffrey A. 1993. Improving Individual Member Accountability in Small Work Group Settings.
- Michaelsen, L. K. 2002. *Team-based Learning*.
- Moorhead, Gregory, Richard Ference, and Chris P. Neck. 1991. Group Decision Fiascoes Continue: Space Shuttle Challenger and a Revised Groupthink Framework.
- Neal, Derrick J. 1997. Group Competitiveness and Cohesion in a Business Simulation.
- O'Doherty, D. M. 2005. [Working as part of a balanced team](#). *International Journal of Engineering Education* 21 (1):113-120.
- Oakley, B., R. M. Felder, R. Brent, and I. Elhadj. 2004. Turning student groups into effective teams. *Journal of Student Centered Learning* 2 (1):9-34.
- Ogot, M., and G.E. Okudan. 2006. The Five-Factor Model Personality Assessment for Improved Student Design Team Performance. *European Journal of Engineering Education* 31 (5):517-529.
- Quirk, Thomas J. 1989. A Procedure Designed to Assist Students to Improve their Presentation Skills as Part of a Marketing Planning Team.
- Redmond, M. A. 2001. A computer program to aid assignment of student project groups. *SIGSCE Bulletin* 33 (1):134-138.
- Rentsch, J. R. and R. Klimoski, 2001. Why do 'great minds' think alike?: Antecedents of team member schema agreement. *Journal of Organizational Behavior* 22 (2):107. ([abstract](#))

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Rosser, S. 1998. Group work in science, engineering, and mathematics: Consequences of ignoring gender and race. *College Teaching* 46 (3).

Russell, Stuart J., and Peter Norvig. 1995. *Artificial intelligence: a modern approach*, Prentice Hall series in artificial intelligence. Englewood Cliffs, N.J.: Prentice Hall.

Salama, P., M. E. Rizkalla, and C. F. Yokomoto. 2004. [Screening and assessing a capstone senior design project: Video compression and error concealment over the Internet](#). *International Journal of Engineering Education* 20 (6):1055-1064.

Stewart, Greg L. 2006. A Meta-Analytic Review of Relationships Between Team Design Features and Team Performance.

Strong, James T., and Rolph E. Anderson. 1990. Free-Riding in Group Projects: Control Mechanisms and Preliminary Data.

Tonso, K. L. 2006. [Teams that work: Campus culture, engineer identity, and social interactions](#). *Journal of Engineering Education* 95 (1):25-37.

Vora, Jay A., and William G. Akula. 1978. Peer and Self Assessments in Small Groups.

Wankat, P. and F.S. Oreovicz. 1993. *Teaching Engineering*. New York: McGraw-Hill.

Wesner, J. W., C. H. Amon, M. W. Bigrigg, E. Subrahmanian, A. W. Westerberg, and K. Filipiski. 2007. [Student team formation and assignment in a multi-disciplinary engineering design projects course: a pair of suggested best practices](#). *International Journal of Engineering Education* 23 (3):517-526.

Wolfe, Joseph, Donald D. Bowen, and C. Richard Roberts. 1989. Team-Building Effects On Company Performance: A Business Game-Based Study.

Wolfe, Joseph, and Thomas M. Box. 1988. Team Cohesion Effects on Business Game Performance.

Woods, D.R. 1996. Problem-based learning for large classes in Chemical Engineering. In *Bringing Problem-based Learning to Higher Education: theory and practice*, edited by L. A. W. W. H. Gijsselaers. San Francisco, CA: Jossey-Bass.

AUTHOR BIOGRAPHIES

Richard A. Layton is the Associate Director of the Center for the Practice and Scholarship of Education (CPSE) and an Associate Professor of Mechanical Engineering at Rose-Hulman Institute of Technology. He earned a B.S. in Engineering from California State University, Northridge, and received his M.S. and Ph.D., both in Mechanical Engineering, from the University of Washington. His areas of scholarship include student teams, student learning in laboratories, visualization of quantitative data, and engineering system dynamics. He delivers workshops in student team-building and in laboratory assessment and he has been recognized with multiple conference best paper awards. Prior to his academic career, Dr. Layton worked for twelve years in consulting engineering, culminating as a group head and a project manager.

Misty L. Loughry is an Associate Professor in Georgia Southern University's Department of

Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria

Management, Marketing, & Logistics. She received her Ph.D. in Management from the University of Florida in 2001. She was a Co-PI on the NSF-supported research that developed the Comprehensive Assessment of Team Member Effectiveness (*CATME*). Her research focuses on control in organizations, particularly peer control and control in teamwork environments, including peer evaluations.

Matthew W. Ohland is an Associate Professor in Purdue University's Department of Engineering Education. He received his Ph.D. in Civil Engineering from the University of Florida in 1996. Dr. Ohland is the Past President of Tau Beta Pi, the national engineering honor society, and has delivered 95 volunteer seminars as a facilitator in the award-winning Engineering Futures program. He served as Assistant Director of the NSF-sponsored SUCCEED Engineering Education Coalition and a NSF postdoctoral fellow. His research on the longitudinal study of engineering student development, peer evaluation, and high-engagement teaching methods has been supported by over \$9.1 million in PI or Co-PI funding from the National Science Foundation.

George D. Ricco is a doctoral student in Purdue University's School of Engineering Education. He previously received an MS in Earth and Planetary Sciences studying geospatial imaging and an MS in Physics studying concentration in FT-IR studies in heavy water, both from the University of California at Santa Cruz. He has a BSE in Engineering Physics with a concentration in Electrical Engineering from Case Western Reserve University.