
by

Kendell M. Timmers

B.S. Business Administration

University of North Carolina at Chapel Hill, 2000

Submitted to the Sloan School of Management in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Operations Research at the Massachusetts Institute of Technology

June 2004

© 2004 Massachusetts Institute of Technology All rights reserved

Signature of Author................................................................. Sloan School of Management Interdepartmental Program in Operations Research May 14, 2004

Certified by.................................................................................... Richard C. Larson Professor of Civil Engineering & Engineering Systems Thesis Supervisor

Accepted by.................................................................................. James B. Orlin Edward Pennel Brooks Professor of Operations Research Co-Director, Operations Research Center

by

Kendell M. Timmers


ABSTRACT

There is a great need among educators for a way to quickly assign teams in large or distance learning classrooms in a manner superior to random assignment or student self-selection. Forming teams based on knowledge of students’ characteristics is too time-consuming for large classrooms, yet research has shown that the characteristics of individuals greatly affect the quality of the teamwork experience. This thesis provides an automated method to quickly assign students to teams based on individual characteristics.

We begin with a thorough review of the literature on how individuals’ characteristics affect team behavior, focusing on the level of diversity of four main classes of traits – knowledge/skills/abilities, demographics, personality, and motivation. By forming teams that have diversity on some of these traits and homogeneity on others, we will be able to improve performance over randomly assigned teams.

We frame this problem from a group dynamics perspective, measuring the compatibility of every dyad of students within a team. We propose, for several group environments, which traits should be homogeneous and which heterogeneous, and how important each trait is, and use these values to create an equation for a student compatibility score, a number representing how well a pair of students will work together. We then simulate team assignment to determine which of several heuristics is most efficient. A combination of random generation and pairwise exchange is found to be the best, forming teams with average compatibilities 307% higher than the average randomly generated team. The code for this program is included in the appendices.

Additionally, we perform a classroom experiment in which sections of a class are divided into teams by three different methods – random assignment, intuition, and the method devised above. Although the experimental design was flawed, the results were encouraging, demonstrating that average student compatibility on a team was significantly positively associated with both the resulting team grade and the students’ perception of how much they learned about teamwork.

For a more detailed executive summary of this work, please see the Structure of the Thesis section on page 16.

Thesis Supervisor: Richard C. Larson
Title: Professor of Civil Engineering & Engineering Systems
Acknowledgements

This thesis would not have been possible without the guidance of Dr. Richard C. Larson, whose advice and encouragement played a major role in the achievement of this work. I am also indebted to Dr. Robert Freund for funding the first year of this research.

The classroom experiment section of this paper would not have been possible without the help of Dr. Lori Breslow, who not only served as an excellent source of teamwork information, but was also willing to help me get approval to use her students in my experiment, along with Dr. Jane Dunphy and Dr. Terence Heagney. Their cooperation and advice during and after the classroom experiment contributed greatly to my understanding of the difficulties of team assignment in a real classroom.

Finally, I'd like to thank my parents and my husband, Brad, for their support during my years at MIT.
# Table of Contents

I. Introduction ............................................................................. 11  
   A. The Benefits of Team Learning ........................................ 13  
   B. Forming Groups ................................................................ 15  
   C. Structure of the Thesis .................................................... 16  

II. What Makes a Good Team? ..................................................... 19  
   A. Knowledge/Skill/Ability .................................................. 19  
   B. Personality/Cognitive Behavior .......................................... 21  
      1. Personality tests ....................................................... 21  
      2. Learning style ......................................................... 24  
      3. Diversity of personality ............................................. 25  
   C. Demographics .................................................................. 27  
      1. Gender ..................................................................... 27  
      2. Age ......................................................................... 28  
      3. Race/Ethnicity .......................................................... 28  
      4. Overall effects of demographic diversity .................... 29  
   D. Values/Beliefs/Attitudes ................................................... 30  
   E. Other Factors .................................................................. 30  
      1. Task ......................................................................... 30  
      2. Moderating heterogeneity ......................................... 31  

III. Developing a Formulation ..................................................... 32  
   A. Framing the Problem ....................................................... 32  
   B. Group Learning Environments .......................................... 36  
      1. Local, high-skill, long-term, vague-task groups ............ 38  
      2. Local, high-skill, long-term, clear-task groups .......... 38  
      3. Local, high-skill, short-term, brainstorming groups ...... 39  
      4. Local, low-skill, short-term, clear-task groups .......... 39  
   C. Distance Groups ............................................................ 40  
      1. Distance, high-skill, long-term, vague-task groups ....... 40  
      2. Distance, high-skill, long-term, clear-task groups ...... 40  
      3. Distance, high-skill, short-term, brainstorming groups .. 41  
      4. Distance, low-skill, short-term, clear-task groups ...... 41  
   D. Student Compatibility Score ............................................. 42  
      1. Additivity .................................................................. 42  
      2. Value Functions ....................................................... 43  
      3. Example .................................................................... 45  
   E. Objective Function ........................................................... 46  
      1. Maximize the average ............................................... 47  
      2. Maximize the minimum ............................................. 47  

IV. Solving the Problem ............................................................ 48  
   A. Exhaustive Enumeration ................................................ 48
List of Figures

Figure 5.1  Maximize-the-minimum results for classes of 36 students – teams of 3 ....... 60
Figure 5.2  Maximize-the-minimum results for classes of 100 students- teams of 5 ....... 60
Figure 5.3  Maximize-the-average results for classes of 36 students – teams of 3 ........ 64
Figure 5.4  Maximize-the-average results for classes of 100 students – teams of 5 ........ 64
Figure 5.5  Overall ranking of heuristics by minimum compatibility from both objective functions ................................................................. 66
Figure 5.6  Minimum compatibility by class size and heuristic .................................. 70
Figure 5.7  Relationship of percentage female to change in performance .................. 74
Figure 5.8  Distribution of randomly generated min. compatibility scores – 12 student example .................................................................................. 75
Figure 5.9  Distribution of randomly generated minimum compatibility scores – 36 students ........................................................................ 76
Figure 5.10 Distribution of randomly generated minimum compatibility scores – 100 students ........................................................................ 77
Figure 6.1  Minimum and average compatibility scores by classroom ..................... 80
Figure 6.2  Change in average value of Total over time ............................................. 82
Figure 6.3  Change in average value of Learning over time ....................................... 83
Figure 6.4  Change in team characteristics satisfaction over time ............................. 84
List of Tables

Table 3.1  Measurements of diversity on a four-member team................................. 35
Table 3.2  Data coding for student data.................................................................. 45
Table 5.1  Probability estimates for generating sample data sets.......................... 58
Table 5.2  Percentage below best solution and average rank for maximize-the-
minimum objective function .............................................................................. 61
Table 5.3  Percentage below best solution and average rank for maximize-the-
average objective function ............................................................................... 65
Table 5.4  Top-ranked heuristics by ranking system................................................ 67
Table 5.5  Top-ranked heuristics by ranking system and class size.......................... 69
Table 6.1  Results on regression of Total3 ................................................................ 85
Table 6.2  Results of regression on Learning3 .......................................................... 86
Table 6.3  Significance of average compatibility to influence responses to survey
questions .............................................................................................................. 87
Table 6.4  Results of regression on Team Grade ....................................................... 88
Table 6.5  Results of regression on Total Grade ....................................................... 88
Table 6.6  Responses to skills and motivation survey questions.............................. 92

I. Introduction

The American institutions of higher education are going through a time of upheaval more drastic than any in the past forty years. In addition to financial pressures, colleges and universities also have to cope with new demands to prove their effectiveness, the internationalization of education, the question of affirmative action policies, and a flood of new, mostly non-traditional, students from other cultures and age groups (Berg, 2002). In the 1999-2000 academic year, nearly a third of undergraduate students were minorities (up from 25% 10 years before), and 21.4% of students were non-U.S. citizens, while the percentage of undergraduates working full-time during the school year increased 7 points in the past ten years to 39% (U.S. Department of Education, 2003). In the past ten years the number of associate's degrees has risen by 20%, while the number of bachelor's degrees has grown by 14% (U.S. Department of Education, 2003). In response to this influx of students, colleges and universities have turned to distance learning programs to accommodate the higher demand and the lifestyles of the older, working, students throughout the world, and have also continued to offer large introductory lecture courses to accommodate the demand within the physical space of the university.

Large lecture classes and distance learning classes both help educators reach more students, but come with drawbacks. Many students are introduced to their potential major through large, impersonal, lecture-based classes. The economics of running an institution of higher learning, be it a community college or a large public university, often require that there be such classes. However, Astin's longitudinal study of undergraduates
found that the environmental factors that had the most impact of academic achievement, personal development, and satisfaction with college were student-to-student interaction and student-faculty interaction (1993). These are the very types of interaction largely missing in distance and large classrooms, while noise and distraction from the large number of students is higher (Cooper & Robinson, 2000). Students often have a difficult time paying attention to large lecture classes, absenteeism is rampant, and the attrition rate from large classrooms is generally higher than that from smaller classes.

Distance classes share some of the same drawbacks. Birnbaum asserts that student-to-student interaction and networking in distance learning classes is equally limited in large classrooms and distance learning environments (2001), although the connections may be less personal in distance learning environments than they are in large classes, which at least have potential for face-to-face communications. Although the overall enrollment in distance learning classes is rapidly growing, the drop-out rate is still quite high (Rivera & Kostopoulos, 2001).

One solution to these problems that has been successfully applied to both local and distance classrooms is cooperative learning. Koppenhaver and Shrader define cooperative learning as “the instructional use of small class groups or teams where peer interactions plays the key role in learning.” (2003). Implementations of this concept can range from short in-class group discussion interspersed with lecturing, to groups that work together outside of class (virtually or in face-to-face meetings) on homework assignments and projects over a period of a semester or more; such popular techniques as the jigsaw method (in which each group member has a resource needed to complete the task) or think-pair-share (in which two attack a problem separately, then compare approaches and create a final solution jointly) are all types of cooperative learning. Cooperative learning is distinguished from simply putting students in teams by the inclusion of five additional conditions: positive interdependence, face-to-face or direct interaction, individual accountability, social skills, and group processing (self-evaluation by the group) (Johnson, Johnson, & Smith, 1991). Building these five conditions into a
team learning experience requires some thought, but can be easily done for almost any learning environment.

The Benefits of Team Learning

A large body of research exists demonstrating the usefulness of learning in groups. Johnson, Johnson, and Smith claim that, between 1900 and 1990, more than 600 studies have been conducted comparing cooperative learning to other methods (1991). The results of these years of experimentation have shown cooperative learning to have the potential for a wide range of benefits to students.

Clearly, an important goal in cooperative learning is to improve the academic performance and learning of the students. There is ample evidence of these educational benefits to teamwork. Students who experience learning in teams “tend to exhibit higher academic achievement, greater persistence through graduation, better high-level reasoning and critical thinking skills, deeper understanding of learned material, more on-task and less disruptive behavior in class, lower levels of anxiety and stress, greater intrinsic motivation to learn and achieve, greater ability to view situations from others’ perspectives, more positive and supportive relationships with peers, more positive attitudes toward subject areas, and higher self-esteem” (Felder & Brent, 1994).

A meta-analysis of 323 studies by Johnson, Johnson, and Smith showed that the average student in a cooperative learning environment performed at about 2/3 of a standard deviation above the average student in a competitive (students compete against one another for the best grade) or individualistic (students learn, work, and are graded on their own) environments, and were significantly more likely to use higher-level reasoning (1990). McConnell found that academic performance was higher in cooperative classes than in lecture-based classes, even though the average GPA prior to the class was actually slightly lower for the cooperative classes analyzed (1996). Lazarowitz and Karsenty also found that students taught in a team learning environment had significantly higher academic performance and self-esteem than those who attended a traditional lecture class.
Another meta-analysis, of experiments on students in science, math, engineering, or technology (SMET) classes, found not only higher achievement and self-esteem, but also a decrease in attrition from SMET classes and programs by 22% (Springer, Stanne, & Donnovan, 1997). Cooperative learning is also said to increase students' motivation to learn (Sharan & Shaulov, 1990) and the speed at which they learn (Shaw, 1981). Student satisfaction with the courses is also shown to increase with team learning. In Felder and Brent's experiment, 92% found the class taught with teamwork more useful than previous classes in chemical engineering, while 98% found group assignments to be more helpful than individual assignments, and 78% said that in-class group work was helpful (1994).

Part of the reason behind the success of team learning is the difference in student behavior in group learning environments and lecture-based classes. In a lecture format, professors often have a difficult time getting more than a small percentage of students to participate and answer questions. When groups are formed to discuss lecture material, nearly everyone in the class discusses the question, at least with other students, and, after discussing the question with a group, more students are inclined to volunteer to give an answer (Felder & Brent, 1994). It may also help in retention, a serious problem in lecture classes, where students have been shown to retain much of the information presented in the first few minutes of class, but little of the information given during the rest of the time period (Felder & Brent, 1994; Koppenahver and Shrader, 2003).

Aside from any educational benefits, the use of cooperative learning techniques prepares students for real-world interactions. The trend in business and industry is toward teamwork, and the use of teams in the college classroom will serve to prepare students for their future careers. Felder and Brent remind students who object to instructor-formed teams that such teams are typical of the type of teamwork situations frequently encountered in the business world (1994). The increasing level of diversity in society also makes understanding other viewpoints more important, and exposure to this diversity through diverse teams helps develop tolerance and understanding.
Forming Groups

One important question in implementing cooperative learning is how to put students into teams. There are three main ways of assigning students: allowing students to form their own teams, randomly assigning teams, or the professors assigning teams based on their understanding of how students work together and what abilities each student possesses. In an informal study in Scotland on team assignment, more than half of the instructors allowed students to choose their own teams, while almost all of the rest assigned students randomly (Huxham & Land, 2000). However, there are drawbacks to these methods. Generally, teams formed by students tend to be extremely homogeneous, with students choosing others with the same social and educational backgrounds and same skill level (Millis & Cottell, 1998). While this has the advantage of creating very cohesive teams, it also isolates people who have few friends in the class, and creates groups with low peer pressure, who will not push each other to work (McConnell, 1996). There is also evidence that such teams are more likely to cheat and to have members who engage in “social loafing” – not doing one’s share of the group work – because friends will not wish to report such behavior to teachers (Oakley, Felder, Brent & Elhaij, in press). We shall see in the next section that heterogeneity is conducive to good teamwork, so self-selected, homogeneous teams tend to do worse. Feichtner and Davis surveyed 155 students, who claimed in a 2 to 1 ratio that their worst group experience was with self-formed teams and their best was with instructor-formed teams (1985). A direct comparison of self-selected groups to those assigned to be heterogeneous on gender, race, and skills has also shown that the diverse groups performed significantly better (Imel, 1997).

Random teams also have drawbacks. While the randomness ensures that each student has an equal chance of being placed into a good team, such a method ignores the copious research that suggests that team characteristics can positively influence team performance. Shaw cites several studies in which the same people were put into different combinations of teams, with different cumulative class-wide success, demonstrating that
it was not the individual characteristics of the team members that determined the class’s performance, but the way they interacted as groups (1981). Siciliano also cites evidence that random and self-selected teams are less effective than instructor-assigned teams (2001). Further, there is still the possibility that random teams could be unbalanced, resulting in some good teams and some bad teams (Millis & Cottell, 1998).

Since randomly assigned teams and self-selected teams tend to perform more poorly than instructor-assigned teams, naturally professors in small classrooms should make use of the research on the best methods for assigning teams to do so in their own classes. However, for the large classes we are focusing on here, this can quickly become a time-consuming process as the number of students grows, often requiring the professor to use his or her own evaluation of how students would interact based on their observations; even assigning teams in a class of 30 students in this manner can take hours of thought (Dunphy, personal communication, February 2, 2004). So, how can professors in large, lecture-based or distance learning classes, where cooperative learning methods are most needed, take advantage of the benefits of assigning teams based on student characteristics, without burdening themselves with hours of guesswork about the compatibility of individual students?

Structure of the Thesis

This thesis is an initial attempt at designing an algorithm to quickly assign teams in large classrooms in a manner superior to random assignment or student self-selection. There is great need for this among today’s educators, especially those who teach large distance-learning and traditional classes. Carefully forming teams based on one’s knowledge of students’ characteristics is too time-consuming for use in large classrooms, and yet a large body of group dynamics research, especially over the last 30 years, has shown that the characteristics of individuals greatly affect the quality of the teamwork experience. In order to make use of this research in a large classroom, an efficient method for forming teams based on characteristics is required. This thesis provides such a method.
Section II of this thesis reviews the literature on how individual characteristics affect team behavior. The focus is on diversity or homogeneity of four main classes of traits – knowledge/skills/abilities, demographics, personality, and motivation. By forming teams that have diversity on some of these traits and homogeneity on others, we will be able to improve performance over random. Since this research will include an application of the method derived from this information, the best indicators of these traits are also discussed, such as the use of Myers-Briggs vs. the NEO-PI for personality traits.

Section III frames the problem. First, we discuss several different approaches, including the coverage problem, the assignment problem, and our approach, which involves comparing every dyad of students within a team to maximize diversity on some characteristics and minimize it on others. We also address the different methods of measuring diversity, and explain why diversity on the level of the dyad is the appropriate measure. Since the literature reviewed in Section II demonstrated that the characteristics that make a good team vary depending on time, type of task, and difficulty of task, we then address different group learning environments and propose, for several of them, which traits should be homogeneous and which heterogeneous, and how important each trait is. Finally, Section III uses these answers to create an equation for a student compatibility score, a number between 0 and 1 that represents how desirable it is for any two students to be on the same team, and defines two possible objective functions to improve sets of teams in a classroom.

Section IV briefly discusses five heuristics used to improve the objective functions discussed in Section III, which are exhaustive enumeration, single pairwise exchange, random generation, exhaustive enumeration followed by pairwise exchange, and random generation with repeated pairwise exchange. The code for these algorithms, written in C, is included in Appendix A.

In Section V, the results of simulation are analyzed to determine that the best of the heuristics was the random-pairwise method, which finds a good solution in a very short
period of time. Other interesting results from simulation include the differences in results between simulating large and small classrooms and large and small teams, and a comparison of the teams resulting from maximizing the minimum compatibility score between any two students on a team and maximizing the average compatibility score over all teams. Finally, we demonstrate that randomly-generated teams are highly unlikely to have compatibility scores as high as those found by the random-pairwise heuristic. In data sets of 100 simulated students, 10 million randomly generated sets of teams were compared to the best solution from the random-pairwise heuristic. No randomly generated teams for any of the data sets were as high as 66% of the best answer, and fewer than .01% of randomly generated teams were as high as 53% of the best answer. The best answer for each of the five data sets was, on average, 307% better than the average value of the randomly chosen teams.

Having demonstrated that, for the student compatibility score that we have defined, the algorithm decidedly beats random, Section VI describes a classroom experiment performed on three sections of a business communications class at MIT to determine the relationship between good teams and the student compatibility scores. In each section teams were formed with a different method – one random, one by the teachers’ intuition of how well students would work together, and one by the method simulated in Section V. Although there were problems with the experimental design, the results of the semester-long experiment were encouraging, demonstrating that average student compatibility on a team was significantly positively associated with both the resulting team grade and the student’s perception of how much they learned about teamwork.

Finally, section VII summarizes the work done and suggests avenues for future research, including comparing different frameworks or combining frameworks, altering the questions asked to students, and further exploration of some of the other details discussed in earlier sections.
II. What Makes a Good Team?

Although educators and business managers alike are convinced that teamwork is beneficial, the research on how best to compose a team is scarce and contradictory. McGrath broke down the types of diversity examined by researchers into four categories: knowledge/skill/ability, personality/cognitive behavior, demographics, and values/beliefs/attitudes (1998). We will briefly give an overview of these four categories and the current state of knowledge on how the level of diversity on these four scales affects team performance. The consensus from these four categories will then be used to develop a formulation to measure the overall compatibility of team members.

Knowledge/Skill/Ability

Of the four categories, skill level enjoys the most straightforward results. Teamwork researchers in K-12 and university education as well as researchers of teamwork in the workplace are nearly unanimous in their approval of diversity of skill level (Johnson et al., 1991; Bowers, Pharmer & Salas, 2000; Hooper & Hannafin, 1991; Morgan & Lassiter, 1992; Siciliano, 2001; Safford, 1997; Imel, 1997; Shaw, 1981; Michaelsen, 1994). Reasons for diversifying skill levels within a team include an increase in participation, more frequent giving and receiving of explanations, and higher self-esteem of both low- and high-ability students. Several studies of classrooms composed of children of mixed abilities have shown that, while groups composed entirely of high-ability students outperformed other groups, heterogeneous groups outperformed homogeneous groups of medium- and low-ability students (Hooper & Hannafin, 1991; Shaw, 1981). Although many claim that high-ability students gain from being in heterogeneous groups, learning the material more thoroughly by explaining it to others in the group, other studies have that shown low-ability students gain at the expense of high-ability students from heterogeneous groupings (Johnson et al., 1991; Hooper & Hannafin, 1991). One effort to calculate the amount of gain or loss caused by heterogeneous
grouping found that low-ability students in heterogeneous groups performed 50% better than low-skilled students in homogeneous groups, while high-ability students in homogeneous groups scored 12% higher than those in heterogeneous groups (Hooper & Hannafin, 1991). Another study reported that, while high-ability students participated at the same rates in heterogeneous and homogeneous teams, low-ability groups participated 30% more in heterogeneous teams (Hooper & Hannafin, 1991). Thus, low-ability students could have much more to gain by heterogeneous grouping than high-ability students have to lose.

Another justification for heterogeneous groups is the theory of process loss. Process loss occurs when the group performs worse than the sum of the individual group members' performances due to the difficulties of coordinating a task among multiple people (Penner & Craiger, 1992). Process gain is the opposite, when team members combine to perform better than the members could individually. Two possible causes of process gain include an increased capacity to learn when in groups, and the stimulation of discussion with other members (Bowers et al., 2000). Homogeneous teams of high-ability students have been shown to have process gain, while homogeneous teams of low-ability students have process loss, and heterogeneous teams have been shown to have additive performance, performing at the same levels as they would individually (Bowers et al., 2000; Tziner & Eden, 1985). In a classroom with diverse abilities, then, the only way to avoid process loss is to group students heterogeneously; it may or may not lead to more efficient learning, but the theory of process loss says nothing about the gains in social or teamwork skills that even a group performing at additive levels can make by virtue of accomplishing the same tasks with a group instead of individually.

Definitions of "high-skill" vary; specific skills related to the teamwork at hand tend to be more predictive than an overall measure of intelligence or performance like IQ or GPA (Shaw, 1981). However, on the whole, the body of empirical evidence gives us good reason to assume that, in almost every teamwork situation, diversity in skill is better than homogeneity in skill, and can have a significant effect on team performance and
participation, and our formulation of student compatibility should include measures of specific skills relevant to the team task at hand.

**Personality/Cognitive Behavior**

The second category of characteristics that influence team performance is much less specific than skill, because the methods of measuring personality are so numerous and widely varied. Currently more than 500 measures of personality have been used in group studies (Neuman, Wagner, & Christiansen, 1999). This fact is complicated by a similar profusion in categorizations of learning styles. This section must, therefore, answer three questions: which measures of personality type and learning style is best for this application? Can one test be used to measure both personality and learning style? Once we determine personality, which is preferable with respect to personality - a more diverse or a more homogeneous group?

**Personality tests**

The most common measures of personality type are the NEO-PI (also known as the Five Factor Model (FFM) or “the big five”), and the Myers-Briggs Type Indicator (MBTI). They are derived from different sources and are viewed differently by psychologists and businesspeople, so a closer examination is necessary to determine which one is most useful in this application.

The MBTI was created by Isabelle Myers and her mother, Katharine Briggs, in 1942, as an application of Jung’s theory of psychological types. The Jungian types were designed specifically to measure normal personalities, as opposed to those with psychiatric problems, and made no value judgments as to the desirability of one trait over another. Today, the MBTI is the most widely used personality test for non-psychiatric populations, the most often used in studies of the effect of personality on educational outcomes (Borg & Shapiro, 1996), and the most “enduring and empirically sound” method of measuring cognitive style (Volkema & Gorman, 1998). The MBTI is a set of
four non-judgmental scales that measure an individual’s preferences for relating to the world (Introvert/Extravert), processing information (Sensing/Intuitive), making decisions (Thinking/Feeling), and lifestyle and time orientation (Judging/Perceiving) (Ziegert, 2000). A questionnaire of about 80 questions will result in a four-letter type, listing the individual’s preferences on each of four scales.

Many descriptions of the eight categories can be found, but a brief description will be helpful, especially since the common conception of "introvert" and "extrovert" is somewhat different from the MBTI definition. Introverts are internally driven, while Extroverts are more intellectually stimulated by other people. Sensors tend to prefer data gathered by their five senses, while Intuitives focus more on implications and inferences. Thinkers use logic to make decisions, while Feelers more often rely on their sense of what is right. Judgers wish to live in an orderly world, while Perceivers prefer spontaneity and flexibility.

The other major instrument for measuring personality type is the NEO-PI, which measures five traits: conscientiousness, agreeableness, openness to experience, extraversion, and emotional stability (Neuman et al., 1999). The NEO-PI is not derived from theory, as the MBTI is, but purely from factor analysis of personality data. It began as three factors (the “NEO” stands for neuroticism, extraversion, and openness to experience) and, after further factor analysis indicated that personality consisted of five traits, was later expanded by Costa and McCrae to also include conscientiousness and agreeableness. Driskell, Hogan and Salas proposed an alternate version with six traits: intellectance, adjustment, ambition, prudence, sociability, and likeability (1987).

A brief description of the five factors follows:

Neuroticism: High scorers are anxious, hostile, self-conscious, and vulnerable, while low scorers are calm, even-tempered and confident.
Extroversion: High scorers tend to like large groups and gatherings, be outgoing and
social, and be assertive and talkative. Low scorers are introverts, and are more likely to be reserved, independent, and even-paced.

**Openness to Experience:** High scorers are more curious, imaginative, and tolerant of differences between people, while low scorers are conventional and conservative, preferring familiar things and having difficulty adapting to change.

**Agreeableness:** High scorers are easy-going, altruistic, and sympathetic, while low scorers are arrogant, egocentric, and skeptical.

**Conscientiousness:** High scorers have self-control and live ordered lives, following through on plans, while low scorers are less driven and more difficult to motivate.

One clear difference between the two tests is that one is based on theory, and the other on statistical analysis. Another is that the MBTI is non-judgmental about type preferences, while pejorative terms crop up frequently in FFM literature. Newman notes, for example, “Those low on the Agreeableness scale are not described by the FFM as excellent critical thinkers (as on the MBTI) but as ‘ruthless, suspicious, and uncooperative’ ” (1996). Despite these differences, the tests are actually quite strongly correlated. The NEO-PI scale of Extraversion has a correlation of .70 to the E/I dimension of the MBTI, and the Openness scale has a .70 correlation to the S/N dimension. Agreeableness has a .45 correlation to T/F, and Conscientiousness has a .47 correlation to J/P. The correlations for E/I and S/N are so high as to indicate equivalence between the two tests on these traits, but all four of these correlations are significant at p<.001 (Newman, 1996). This level of correlation on two personality measures derived in completely different manners is striking evidence that both tests are, indeed, valid measures of personality.

Personality researchers have also derived a fifth scale from the MBTI called C/D (Comfort-Discomfort), which has a .65 correlation to Neuroticism (Johnson, 1996). According to Newman, Myers found the same scale (which she called “Sufficiency-Insufficiency”) but did not include it as a separate scale because “the MBTI was specifically designed as a measure of normal personality, and clearly this fifth cluster tapped aspects of psychopathology” (1996).
In view of these correlations, it may seem that either test would be equally good for use in the classroom. However, in a real classroom environment, tests of personality will most likely be taken by students online, and the results of these tests will generally be shared with the students, for their growth as well as for convenience. This decision highlights the main reason why the FFM does not belong in a learning environment – it is hardly appropriate to rate students on agreeableness or neuroticism, or other traits that imply clearly negative personality aspects, when the results might be shared with the students, and possibly their teammates. MBTI, on the other hand, allows students to share this information without fear, possibly encouraging more and better interaction with teammates as they begin more quickly to understand one another’s personalities. The MBTI is also more accessible. Many self-scoring versions (some more accurate than others) can be found online; while it is more difficult to find an online NEO-PI. Myers-Briggs has the additional advantage of also serving as a measure of learning style.

Learning style

Personality is an important aspect of group behavior; however, in an educational context, cognitive behavior in the form of a learning style is also crucial. A learning style is defined as one’s preferred method of information processing and idea formation, the attitudes that direct one’s course of study, and a preference for some learning tools and environments over others (Brownfield, 1993). Just as personality type has been variously defined by a broad range of overlapping traits, the literature on learning style is incredibly diverse. Some typologies of learning style even include such details as the students’ preferences for time of day and temperature (Wilson, 1998). To complicate the issue, students often possess more than one learning style under certain categorizations - for example, a visual learner may also be good at learning by writing. This problem can be solved by using the same typology to measure personality and learning styles, namely, the MBTI.
There is strong evidence that MBTI measures one’s learning style. It is clearly correlated to several other measures of learning style, including the Grasha-Reichmann Learning Styles Questionnaire, Kolb’s Learning Style Inventory, Dunn, Dunn and Price’s Learning Style Inventory, and the learning style categories of both Anthony Gregorc and Howard Gardner, while being more accessible or clearly defined than these (DiTiberio, 1996; Herbst, Price & Johnston, 1996; Wilson, 1998; Borg & Shapiro, 1996; Millis & Cottell, 1998). In a learning style context, the eight categories of types are well-defined as follows:

- **Extroverts** like group work, use trial-and-error to learn, and dislike solitary reading and writing.
- **Introverts** prefer quiet study and lectures to group work.
- **Sensors** prefer to work with well-established theories and practical applications; they are generally good at memorization, and are slower and more methodical workers.
- **Intuitives** prefer open-ended assignments and dislike memorization. They are generally better at timed tests than Sensors because they can answer with their instincts.
- **Thinkers** need clear goals, motivations, and performance criteria. They talk and think systematically and tend to be blunt, but are often group leaders because they’re orderly and efficient.
- **Feelers** profit from group communication, are excellent mediators, and are more likely to need personal encouragement and attention.
- **Judgers** want a structured environment with clear goals and tend to be overachievers.
- **Perceivers** tend to procrastinate and enjoy discussion-oriented lectures and self-paced learning. (Brownfield, 1993).

Some of these types may seem contradictory when put together, but a person’s four categories generally combine to give a clear overall picture of a student’s learning style.

*Diversity of personality*

Even once the category of personality/cognitive style has been assigned an appropriate measure, the question remains as to whether groups perform better when the personalities of the group members are similar or different. There are two contradictory theories, similarity theory and equity theory, about the effect of personality diversity on teamwork.
Similarity theory states that homogeneous teams are more effective because differences in personality and demographics cause tension and difficulty in communication, while equity theory contends that performance is enhanced by the tension caused by such differences (Bowers et al., 2000). Educators have further contended that a diversity of learning style is important because it gives students a chance to observe the learning styles of other students, which may help them adopt techniques for learning when not in their preferred environment (Hooper & Hannafin, 1991).

Using the MBTI, Volkema and Gorman found that teams balanced on all four dimensions were more likely to establish rules for team government, which led to better performance, than groups consisting entirely of Thinkers/Judgers, who, because of their logical and orderly orientation, were originally expected to be better at problem solving (1998). Another study used the MBTI to calculate team diversity and found a positive correlation between diversity and the quantity of output and the effectiveness of a team (Hammer & Huszczzo, 1995). These two studies provide some evidence that heterogeneity of personality, as measured by the MBTI, has a positive effect on team effectiveness.

The wide variety of personality measures makes generalizations about the empirical evidence for similarity or equity theory difficult. Some researchers are in favor of diversity of personality in general (Shaw, 1981; Neuman et al., 1999; Hooper & Hannafin, 1991) while others have found negative or insignificant results (Huxham & Land, 2000; Bowers et al., 2000). The answer to the question of diversity of personality, as we will discuss in more detail later, is that the effects of personality diversity are modified by the type of task and by several other factors. Thus we cannot make an overarching claim that personalities on a team should always be diverse or similar, as we could with ability.
Demographics

The influence of demographic diversity on team performance is even more contentious than that of personality, although the methods of measurement are much more straightforward. The three main demographic factors are gender, age, and race; while some studies have made general comments about demographic diversity, each of these characteristics have also been studied individually and thus need to be addressed separately to form a context for a discussion about the overall effect of demographic diversity.

Gender

The role of gender in teams is difficult to determine. There is much disagreement over whether differences in team performance related to gender are caused by actual differences between males and females or in the perceived differences between the sexes. While Lichtenstein, Alexander, Jinnett and Ullman claim that “women are more likely to focus on group processes and interpersonal relationships and men are more likely to focus on achieving specific outcomes” (1997), Younglove-Webb, Gray, Abdalla, and Thurow contend that it is perceptions of “traditional behavior” of the genders which cause communication difficulty in the form of men “not crediting women fully for contributions… and devaluing female leadership efforts” (1999) rather than actual behavioral differences. In 1981, Shaw wrote that, while males and females did appear to behave differently now, “these differences are probably due to different socialization patterns for males and females, [and] they may disappear as differences between expected sex roles narrow” thus decreasing the negative aspects of gender diversity, such as poor communication.

Overall, several researchers were in favor of gender diversity (Johnson et al., 1991; Hooper & Hannafin, 1991; Siciliano, 2001; Imel, 1997; Shaw, 1981), while Bowers et al. found insignificant results (2000). Morgan and Lassiter found the effect of gender diversity to be dependent on type of task, with single-gender groups performing better at
structured, production-oriented tasks, while mixed gender groups performed better at creative tasks (1992).

Age

Less empirical research has been done investigating age, especially in the educational context, where most students are very close in age. Lichtenstein et al. found different age groups tended to use different cognitive processes to solve problems, which could indicate that age diversity, like diversity in learning style, can be beneficial in teaching students new methods of learning (1997). However, they also found a strong negative correlation between diversity in age and perception of “team integration”, a measure of personal satisfaction between team members. Similarly, some studies have shown that groups diverse in age have more negative evaluations of peers’ performance, and, in work groups, more turnover (Harrison, Price, & Bell, 1998).

Race/ethnicity

Race and culture are difficult items to measure, because there are many different levels on which to determine similarity or difference. For example, should an Asian-American be classified as Asian or American, or in a separate category from either? The number of possible gradations of difference (first-generation, non-citizen, etc.) are staggering. In the interest of simplifying what is already a complex analysis, we simply use white vs. non-white, as self-reported by students, in our formulations. There is ample justification for this seeming overgeneralization. For one thing, determining the level of difference between each race or ethnicity was beyond the scope of this investigation. Further, more than 70% of U.S. college students are white, so fragmenting the minorities into subsections may not be useful. Finally, there are cultural justifications for differences between whites and non-whites. The cultural value associated with race/ethnicity that has the strongest effect on team performance is that of collectivism vs. competition. White Americans of European descent tend to be competitive in nature, while Asian-Americans, African-Americans, Hispanic-Americans,
Latin-Americans, Africans, and Asians all tend to be more collaborative in nature (Cox, Lobel, & McLeod 1991). The Cox et al. experiment found that ethnically diverse groups worked more collaboratively than did all-white groups. In that sense, ethnic/racial diversity could be helpful in team performance. Another study by Watson, Kumar, and Michaelsen found that, while all-white groups initially outperformed ethnically diverse groups, after 13 weeks, overall performance was the same for both groups, and after 17 weeks, racially diverse groups generated better alternative solutions than did the all-white groups (1993). Another study showed race as task-dependent, with lower-skilled tasks performed better by homogeneous groups, and high-skilled tasks performed better by heterogeneous groups. Finally, there is the argument that, regardless of the impact on team performance, contact with other races and ethnicities decreases prejudice for students and workers of all age groups and is thus beneficial regardless of the educational outcomes (Cooper & Robinson, 2000).

**Overall effects of demographic diversity**

Some of the contradictory results mentioned above can be explained by the results of a study conducted by Harrison et al (1998). They theorized that the effect of what they called “surface differences,” particularly race and gender, would decrease over time, while the underlying attitudinal diversity in the form of motivation and personality, would become more important. Thus, “initial categorizations are accompanied by perceptions of similarity or dissimilarity that are based on surface-level demographic data; these perceptions change when deep-level information is obtained” (Harrison et al., 1998). They found significant results that the initial negative influence of demographic diversity on team cohesiveness was lessened over time, which provides more evidence for the idea that homogeneous teams are initially more efficient than heterogeneous teams, but over time improve more quickly and are eventually more effective at problem-solving. Again, since many of the specific demographic factors listed above depended on task, demographic diversity as a whole will as well. Thus, the benefits of diversity in demographics, like personality, can only be discussed within the context of the task at hand and the time that task takes to complete.
Values/Beliefs/Attitudes

The final category in McGrath’s classification was that of values, beliefs, and attitudes. The most relevant of these to the classroom is motivation, which could be measured by answering such questions as “How motivated are you to get an A in the class?” or “How interested are you in the class work?” (McGrath, 1998). There is empirical evidence that motivation contributes to performance in an additive way – that is to say, the more motivated people on a team, the better the performance – but does not cause process gain the way ability does (Tziner & Eden, 1985). Further, it is clear that a team composed entirely of low-motivated students will not do very well; thus Johnson et al. recommend, and common sense dictates, that teams should always be heterogeneous on motivation (1991).

Other Factors

Task

Many studies have shown that the type of task at hand mediates the impact of diversity on team performance. However, the definition of types of tasks is varied, and an exploration of the relationship between diversity and task type cannot take place until the concept of task type is more thoroughly investigated. In their meta-analysis on diversity in teams, Bowers et al. categorized tasks as low or high difficulty, with result that in low difficulty tasks, homogeneous teams performed moderately better, while in high-difficulty tasks, heterogeneous teams performed much better (2000).

There are several other ways in which to differentiate tasks, including the degree of collaboration required, the balance of creativity vs. skills, and the type of personality best suited to the task (Driskell et al., 1987). Amount of time needed for the task also appears to be important, given the theory that demographically diverse teams improve over time – thus, a short-lived team should probably be more demographically homogeneous than a longer-term team. There is little empirical evidence on how the amount of collaboration needed influences the impact of diversity on performance, but it seems clear that the
more interaction required between the students to complete the task, the more important communication and coordination skills become. Since diversity of personality and demographics tend to have, at least initially, a negative effect on communication, teams working on a highly collaborative task operating for a short amount of time should be more homogeneous on those scales than a team operating with a less collaborative task over a long period of time (Millis & Cottell, 1998).

*Moderating heterogeneity*

Finally, the last factor to take into account when developing teams is the idea of moderating heterogeneity. It’s well established that the more heterogeneous a team is, the more difficult communication becomes. This effect, as we’ve seen, is moderated by time in the case of demographic diversity, but underlying personal characteristics such as personality and motivation can continue to pose these problems. Thus, several researchers have suggested that teams should be heterogeneous on some characteristics to get the benefits of diversity, and homogeneous on other characteristics to maintain some level of ease in communication (Lichtenstein et al, 1997; Shaw, 1981; Brufee, 1993; Miller & Harrington, 1992).

We must now combine the research on diversity of skill, personality, values, and demographics, with considerations of the task type and balancing heterogeneity and homogeneity, and apply these ideas to different learning contexts to determine the appropriate composition of teams in each situation.
III. Developing a Formulation

In this section, we will synthesize the conclusions taken from the literature review in section II into a simple formula for diversity, after first examining the various methods for measuring team-wide diversity on an individual trait.

Framing the Problem

Before we can derive any formulation, we need to decide how best to frame the problem. One way to approach the problem of creating teams is a coverage problem, i.e. to formulate an optimization problem to maximize the number of teams that contain at least one member with each desirable characteristic: for instance, to ensure gender diversity, one characteristic is female, another characteristic is male. Skill level could have such categories as “person who ranks in the top 20% of the class in skill A”. An objective function could be devised to assign different penalties for not having any members in each category, allowing one to prioritize diversity. For instance, diversity in skill level and motivation seem essential, and thus could have high penalties assigned to solutions that do not provide teams with such diversity. One problem with coverage is that it does not take into account degrees of difference between people. If the goal was, say, to include at least one person who ranks in the top 20% of the class in skill A, the objective function would not distinguish between a team where (at one extreme) everybody was far above average on this skill (with at least one of these people in the top 20%), and (at the other extreme) one in which one student is in the top 20% and the rest are in the bottom 20%. Similarly, requiring at least one male and at least one female doesn’t provide for a balance of genders, just the inclusion of both genders. Ideally, we would like a mix of people from all levels of skills in this case, but this formulation doesn’t examine the differences between individuals, but only the overall team characteristics.
An implementation similar to the coverage problem that solves this difficulty was described by Fourer (1997). In attempting to assign 1000 people into discussion groups for a dinner event, he wanted to assign groups whose populations roughly matched those of the entire assembly – for example, if 24% of the participants were from Chicago, 24% of the group should also be from Chicago; the objective was to minimize the deviation of the actual proportion from the target. In this format, the problem became a variant of the assignment problem. While this ensured groups who were diverse on each characteristic separately, it did not prevent groups in which two people could have the same characteristics, which runs counter to the idea of “cross-cutting” in which each dyad of people is different in some way and similar in some way (Miller & Harrington, 1992; Webb, 1992). Additionally, the formulation was such that he could only solve on two characteristics at a time, and then refine the answer to include additional characteristics.

The biggest problem with such formulations as the coverage problem and the assignment problem variant is that, as we have seen, diversity is not viewed as uniformly beneficial. In fact, there is a strong case for moderating heterogeneity, so that team members share one characteristic in common, while being different on others. It would be difficult, however, to formulate a coverage problem that wanted “non-coverage” in some categories, and coverage in others (i.e. teams homogeneous in gender would require an objective function for which teams had no coverage in the female category or no coverage in the male category.) Similarly, in the assignment problem explained above, there’s no allowance for some variables having to be as similar as possible to each other rather than as similar as possible to the overall population. (In other words, if we wanted gender-homogeneous groups, and we had a 50/50 mix in the population, what we need is teams closer to 100/0 or 0/100, not 50/50.)

The difficulties with the two formulations discussed above highlight the fact that we are measuring the diversity of individual characteristics aggregated on a team-wide level. However, no theoretical approach has yet been established using individual’s traits to measure a group’s diversity (Barrick, Stewart, Neubert, & Mount, 1998). Many team
studies have been performed in labs, where analysis on the effects of diversity of a trait could be performed on carefully orchestrated teams. Often, the level of analysis was the dyad – homogeneous teams were composed of two people who shared a trait, while heterogeneous teams had one person with the trait and one without. For traits that were not 0/1, such as skill level, participants are often categorized as low, medium, and high, and put into teams of three, some of which consist of one member from each skill level, and the others consisting of three members from the same level. Aside from the somewhat questionable nature of results from teams formed and studied in the lab, these experiments failed to look at the degree of diversity on a trait within teams, creating teams only as “diverse” or “homogeneous”. In field experiments on existing teams, the common approaches are to measure the mean, the standard deviation, the highest value, or the lowest value of the trait and use this number as a team index of diversity (Barrick et al., 1998). The highest and lowest value of a graduated trait like skill level or GPA reflects an assumption that the best or worst member of the team on that trait will dramatically affect the performance of the entire group. The mean is only appropriate when one believes that more of a trait is always better, or worse, regardless of how it is distributed among the team members. Only the standard deviation examines the overall diversity on a trait.

The standard deviation is a good measurement for the level of diversity on a team, but it is not the only valid measurement. One can think of diversity on an individual level – how each student compares to every other student on a given trait, to allow our formulation to promote cross-cutting. If one views students as vectors of N characteristics, then, envisioning them in this N-dimensional space, we would like to do something akin to clustering in which we maximize the difference between students in the same group on some dimensions, and minimize the difference between them on other dimensions for each pair of students. We could do this by developing a compatibility score that measures to what degree any two students satisfy our desires for diversity and similarity on each of the N characteristics. This would allow us to make direct comparisons between each student, and to create objective functions that would easily
take into account our desire to create diversity on some characteristics and homogeneity on others. We could also weight each dimension so that the degree of difference or similarity on some dimensions contributes more to the compatibility than do others.

The difference between this sort of formula, which examines each student in comparison to each other student on a team, and using the standard deviation, is simply that more emphasis is placed on the dynamics of the group on a dyadic level. For instance, if we had one 0/1 variable and a team of four students, let’s examine the possible teams and the values of three types of measurement – the average of the differences on a dyad level, the standard deviation, and the mean.

<table>
<thead>
<tr>
<th>Table 3.1 Measurements of diversity on a four-member team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>Member 1</td>
</tr>
<tr>
<td>Member 2</td>
</tr>
<tr>
<td>Member 3</td>
</tr>
<tr>
<td>Member 4</td>
</tr>
<tr>
<td>Average of absolute value of differences between each student</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Mean (i.e. proportion of 1’s)</td>
</tr>
</tbody>
</table>

If we were looking for a diverse team on this characteristic, clearly the preferred team would be team 3 above. The mean is obviously an inappropriate measure because it views more people who possess the trait as better, and thus rates the team with all 1’s the highest. The standard deviation and the average of the differences both correctly identify team 3 as the best team; the difference is that the average of differences approach puts even greater emphasis on the value of the evenly balanced team over the slightly diverse teams (one 1 and three 0’s or vise versa). Since we are building a model to be highly sensitive to changes in diversity, this seems appropriate. The average of differences approach has been used in several other team compatibility experiments, such as Lott and Lott, who measured cohesiveness within a team (1961). Because cohesiveness was clearly related to how each member felt about every other member, an aggregate of the
dyads was clearly necessary. More recently, this method has been frequently used in studies of how diversity of personality affects teamwork (Hammer & Huszczo, 1996). Among researchers of the MBTI, a measure of the dyadic differences of each of the four parts of the MBTI using the average of differences method is called the TCAI, the Team Communication Index, and it has been shown that the greater the team diversity as measured by the TCAI, the higher the overall quantity of work produced, effectiveness, and the group's supervisor's ratings of quality and external viability (Hammer & Huszczo, 1996). Since the MBTI is one of the aspects of diversity we are measuring, the TCAI provides further support for the average of differences method of measuring diversity. Once we have such a formula for a compatibility score, we can create an objective function that uses this score as a measurement of team compatibility.

The backbone of a compatibility score based on this method will consist of two types of information for each characteristic: whether the distance between two students on this scale should be maximized or minimized (i.e., whether the students on the teams should be diverse or homogeneous on this dimension), and some weighting of how important this characteristic should be relative to the other characteristics. For the sake of simplicity, let us not define these numbers for individual characteristics, such as gender, but of the four major categories of characteristics outlined above: skill, demographics, personality, and motivation. With the information from section III in mind, we shall now proceed to define these numbers that will make up the compatibility score.

**Group Learning Environments**

One important conclusion from the last section is that one formulation will not work equally well for all team learning situations, but varies depending on the type of task, the amount of time a team will have to do the task, and other factors. Before we can really start on a formulation, we must decide which types of teams we will consider. First, we will consider the four categories of traits listed above in different group learning
environments and determine whether each should be homogeneous or heterogeneous, and assign weights to each category indicating the relative level of importance of that category over the others. Then we will use this information to derive equations for determining a compatibility score for each dyad of students in a given learning environment, using information about their characteristics described above.

The four aspects of group learning environments are location (local or distance learning), skill level (low-skill or high-skill), duration (long or short), and task purpose (clear, vague, or brainstorming). Task purpose is a category that requires further explanation. This category is used to distinguish between groups that are focused on solving a clear-cut problem, groups focused on solving an open-ended problem, and groups whose purpose is to brainstorm and generate many alternate solutions without necessarily pursuing any of them to completion. This is clearly related to the level of creativity required, but to simply call these groups “highly-creative” or “un-creative” seems a misnomer, as a clear-cut but difficult problem may need much creativity to solve, and no professor would wish to assign a task that is “un-creative.” Skill level refers to how difficult the task is in terms of the type of skills the class is imparting. Although there are 24 possible combinations of these characteristics, we will only discuss eight of the most common combinations, which should provide intuition for creating other combinations should they be needed.

For each of these common combinations, we must determine whether we want students to be diverse or homogeneous on the four categories discussed in the previous section. We then need to assign some value, or weight, to each of these categories, indicating the relative importance of the diversity or homogeneity of each category. As we have seen above, some categories have stronger evidence in favor of diversity, whereas others are more controversial and therefore should perhaps be given less of a say in determining compatibility. Also, some categories are more important in some environments than they are in others. For instance, demographics are largely invisible in distance-learning environments, and may therefore be assigned lower weights.
Local, high-skill, long-term, vague-task groups

Probably the most common college-level group learning experience is a long-term team formed for most or all of a semester to complete a difficult assignment that requires high levels of skill and creativity to define and solve. High-skill, open-ended tasks require a diversity of skills. Diversity in demographics, although initially counter-productive, will eventually produce more creative solutions. To moderate this diversity, personality should be more homogeneous. Since the evidence in favor of diversity of skill is very strong, it should receive more weight than those assertions that are not backed up with as much empirical evidence. In his experiments, McConnell also places a higher weight on his measurement of skills than on demographic variables (1996), while Koppenhaver and Shrader assigned by skills and gender, followed by personality (2003). As always, motivation should be heterogeneous. Thus we assign the following direction and weight to each category of classification:

<table>
<thead>
<tr>
<th>Category</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics</td>
<td>Heterogeneous with weight 15%</td>
</tr>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 50%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 15%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>

Local, high-skill, long-term clear-task groups

Another common group experience is the long-term group formed to solve a difficult but clear-cut task. While very similar to the above group, it requires slightly less creativity, and thus diversity of demographics becomes less important compared to diversity of skills.

<table>
<thead>
<tr>
<th>Category</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics</td>
<td>Heterogeneous with weight 10%</td>
</tr>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 60%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 10%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>
Local, high-skill, short-term, brainstorming groups

Groups in this category are more likely to be working on projects with a smaller scope, the object of which is to explore alternative solutions to a difficult problem without necessarily conclusively solving the problem. Such teams engage frequently in brainstorming-type activities where diversity of personality is even more important, but due to the short-term nature of the group, differences in demographics create more difficulties in communication. Skill, while still important, is less important since the focus is on generating alternate paths of exploration. Thus, demographics should be homogeneous, while the other categories should be heterogeneous, with the following weights:

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Homogeneous with weight 15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 35%</td>
</tr>
<tr>
<td>Personality</td>
<td>Heterogeneous with weight 30%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>

Local, low-skill short-term clear-task groups

Although most group work is difficult in nature, sometimes a task is assigned more to the purpose of getting students to interact than to combine their abilities for a task that would be too difficult for any one student to solve. While high-skill tasks have been clearly shown to be better performed by heterogeneous groups, there is also some evidence that homogeneous teams are better at low-skilled tasks. A short-term task is generally easier where demographics are also homogeneous. Motivation, as always, should be heterogeneous.

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Homogeneous with weight 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Homogeneous with weight 40%</td>
</tr>
<tr>
<td>Personality</td>
<td>Heterogeneous with weight 20%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>
Distance Groups

Distance learning provides some interesting wrinkles to the question of group composition. Communicating with team members by electronic means creates more opportunities for miscommunication as the contextual clues of facial expression and tone of voice are lost (Nemiro, 2000). However, it also creates an opportunity to avoid some of the initial preconceptions caused by race, gender, and physical appearance, as students generally don’t know what their team members look like, and names are not always gender-specific (Salmon, 2000). There is also evidence that over time, virtual groups overcome the potential difficulties of their situation and begin to behave like local groups (Ahuja et al., 2003). With these points in mind, we generate weights for the virtual equivalents to the teams listed above.

**Distance, high-skill, long-term vague-task groups**

Long-term distance-learning groups will be similar to their local counterparts; however, demographics will play less of a role, while personality (and its influence on communication) will play a wider role. Thus the weights shift to:

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Heterogeneous with weight 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 45%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 25%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>

**Distance, high-skill, long-term clear-task groups**

Once again, demographics plays a smaller role, and personality a wider one in a virtual team. Like its local counterpart, these teams have more need for a diversity of skill.

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Heterogeneous with weight 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 55%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 15%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>
*Distance, high-skill, short-term, brainstorming groups*

In the short term, communication within a distance-learning group can be very difficult, which would indicate a need for homogeneity of personality and demographics. However, the nature of brainstorming requires diversity in personality. To further complicate the issue, demographic differences may be hidden in virtual teams. Since one assumes that without good communication, diverse ideas and viewpoints cannot be shared, this formulation sacrifices diversity of personality for the sake of good communication, in the hope that diversity of skills and demographics may compensate by increasing the diversity of ideas.

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Heterogeneous with weight 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Heterogeneous with weight 50%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 10%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>

*Distance, low-skill, short-term, clear-task groups*

As above, in the short term, good communication is necessary immediately, which suggests a homogeneity of personality, rather than diversity of personality possible in a local group, where visual clues help smooth communication difficulties.

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Homogeneous with weight 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills</td>
<td>Homogeneous with weight 50%</td>
</tr>
<tr>
<td>Personality</td>
<td>Homogeneous with weight 20%</td>
</tr>
<tr>
<td>Motivation</td>
<td>Heterogeneous with weight 20%</td>
</tr>
</tbody>
</table>

These weights and directions are initial estimates based on the existing literature on teamwork. They represent a starting point and an opportunity for further research and experimentation, rather than a definitive final answer to the question of diversity in different teamwork environments.
Student Compatibility Score

The next step is to use the weights and directions of our characteristics as described above to determine a compatibility score for each pair of students in a given population. This score will rate, on a scale of 0 (worst) to 1 (best), how well two students would work together as part of a team in a given group learning environment. With these scores we will fill a matrix comparing each student to every other student, and be able to use the scores from each dyad in a team to determine the desirability of the team overall.

Additivity

The first step in designing the value function to fill this compatibility matrix is to determine some basic aspects of the relationships between the variables. Since the compatibility score is essentially a measure of utility, we can use the rules of utility functions to guide us in our formulation. Rather than consider lower-level variables, such as gender or extrovert/introvert, let us continue to think in terms of the four major categories of variables: skills, demographics, personality, and motivation. If we were, for example, to hold personality and motivation constant, do the tradeoffs between skill levels and demographic differences depend upon those fixed values of personality and motivation or are they independent? (In other words, if we were comparing two students with the same amount of difference in personality and motivation, would the amount of difference in skill levels that we would exchange for a certain amount of difference in demographics remain the same, regardless of the current values of the differences in personality and motivation?) Since there’s no apparent relationship in the literature relating skills and demographics to personality and motivation, we can assume skills and demographics are independent of the other two; in fact, we assume here that all pairs of variables are independent of their complements. That is to say, we assume that each of our variables is preferentially independent. Mutual preferential independence implies an additive utility function (Keeney & Raiffa, 1976). The values of the four variables must then be scaled and weighted to produce a function of the form:
\[ v(x_1, x_2, x_3, x_4) = \sum_{j=1}^{4} \lambda_j v_j(x_j), \text{ where} \]

\[ v_j (\text{worst } x_j) = 0 \]
\[ v_j (\text{best } x_j) = 1 \]

\[ 0 < \lambda_j < 1 \]

\[ \sum_{j=1}^{4} \lambda_j = 1 \] (Keeney & Raiffa, 1976).

In the previous section we have already established the \( \lambda_j \)'s, in the form of weights for each of the four characteristics in each learning environment. The next section will develop utility functions based on these criteria.

**Value functions**

If we were to consider a simple goal of maximizing diversity on each type of characteristic equally, a compatibility score between any two students would be easy. The distance between two students can be calculated as the Euclidean distance between the two points in n-dimensional space, i.e. \[ \left( \sum_{i=1}^{N} (x_{1i} - x_{2i})^2 \right)^{1/2} \] where \( x_{1i} \) represents the value of student 1 on characteristic i and \( x_{2i} \) represents the value of student 2 on characteristic i. If we were simply interested in maximizing the distance between students in every dimension, this could be scaled to a number between 0 and 1 and used as the compatibility score. However, as discussed above, the educational literature suggests not only that heterogeneity should be moderated by elements of homogeneity, but also that some factors are more important than others in given situations. Therefore, we need utility functions that modify the above equation for a given group learning environment, conforming to Keeney and Raiffa's definition of an additive utility function rather than to the Euclidean distance formula. This will require a change from Euclidean distances to city-block distances, so that each dimension can be considered separately, which would change the above equation like so: \[ \sum_{i=1}^{N} (| x_{1i} - x_{2i} |) \]. Now we can isolate different parts of the equation to make it reflect the direction (homogeneous or
heterogeneous) and weight of each characteristic. For instance, assume that we would like to maximize the difference between the two students on characteristic i, the value of which is either 0 or 1. The section of the above equation relating to characteristic i would remain $|x_{1i} - x_{2i}|$, and the value of this part of the equation would be either 0 or 1, with 1 being the desired result. If instead we are interested in minimizing the difference between the two students on this characteristic, the proper equation for characteristic i would be $- (|x_{1i} - x_{2i}| - 1)$. If the students are different, then the value will be 0. If they are the same, the value will be 1. Thus, a value of 1 will always indicate the desired result, and the compatibility score will increase when the desired condition (be it heterogeneity or homogeneity) is achieved. In front of either of these equations, we would place a scaling variable, the $\lambda_i$ from the Keeny and Raiffa definition. Thus, the modified version, if all variables were 0/1, would be

$$\sum_{i=1}^{N} \lambda_i a_i (|x_{1i} - x_{2i}| - b_i),$$

where

$$a_i = \begin{cases} 
1 & \text{if diversity is desired for characteristic } i, \\
-1 & \text{otherwise}.
\end{cases}$$

$$b_i = \begin{cases} 
0 & \text{if diversity is desired for characteristic } i, \\
-1 & \text{otherwise}.
\end{cases}$$

and $\sum_{i=1}^{N} \lambda_i =1$.

If the variables are not 0/1, we can make a small adjustment to this equation by dividing the absolute value term by $y_i$, where $y_i$ is the difference between the highest possible value of $x_{1i}$ or $x_{2i}$ and the lowest possible value of $x_{1i}$ or $x_{2i}$. Thus,

$$\frac{\sum_{i=1}^{N} \lambda_i a_i (|x_{1i} - x_{2i}| - b_i)}{y_i}.$$
Example

Let's apply this formulation to a realistic set of characteristics. As we determine a formulation, more detailed information on the type of data we expect to have available to us is necessary. We are given each student's data for age, race, gender, motivation, Myers-Briggs type, and skill level on 3 skills determined relevant for the team project. The data can be coded in the following manner:

<table>
<thead>
<tr>
<th>Data</th>
<th>Range</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>0-1</td>
<td>0=Male, 1=Female</td>
</tr>
<tr>
<td>Race</td>
<td>0-1</td>
<td>0=White, 1=Non-White</td>
</tr>
<tr>
<td>Age</td>
<td>2-6</td>
<td>Age in decades, to the nearest decade</td>
</tr>
<tr>
<td>Skill 1</td>
<td>1-3</td>
<td>1=lowest level, 3=highest level</td>
</tr>
<tr>
<td>Skill 2</td>
<td>1-3</td>
<td>1=lowest level, 3=highest level</td>
</tr>
<tr>
<td>Skill 3</td>
<td>1-3</td>
<td>1=lowest level, 3=highest level</td>
</tr>
<tr>
<td>Motivation</td>
<td>1-3</td>
<td>1=lowest level, 3=highest level</td>
</tr>
<tr>
<td>E/I</td>
<td>0-1</td>
<td>0=I, 1=E</td>
</tr>
<tr>
<td>S/N</td>
<td>0-1</td>
<td>0=S, 1=N</td>
</tr>
<tr>
<td>T/F</td>
<td>0-1</td>
<td>0=F, 1=T</td>
</tr>
<tr>
<td>J/P</td>
<td>0-1</td>
<td>0=J, 1=P</td>
</tr>
</tbody>
</table>

Note that assigning a value of 0 to Introvert and 1 to Extrovert is not meant to make any statement about the relative value of this personality trait – for the purpose of the compatibility score, it is only the absolute value of the difference, rather than the actual value of the student, that is considered. This holds true to all variables thus assigned.

For this type of data, it is easy to apply the above equation. Recall that the weights and directions for the local, high-skill, long-term, vague-task groups was:
Demographics  Heterogeneous with weight 15%
Skills       Heterogeneous with weight 50%
Personality  Homogeneous with weight 15%
Motivation   Heterogeneous with weight 20%

Since the data set has multiple variables for each of these categories except motivation, these weights should be distributed equally over each of the variables that make up the category. For instance, since the category demographics has three variables (age, race, and gender), and the weight is 15%, each of these variables gets a weight of 5%. Thus, our equation is:

\[0.05 \cdot \left| (\text{gender}_1 - \text{gender}_2) \right| + 0.05 \cdot \left| (\text{race}_1 - \text{race}_2) \right| + \frac{0.05}{4} \cdot \left| (\text{age}_1 - \text{age}_2) \right| + \frac{0.20}{2} \cdot \left| (\text{motivation}_1 - \text{motivation}_2) \right| + \frac{0.20}{3} \cdot \left| (\text{skill1}_1 - \text{skill1}_2) \right| + \frac{0.20}{3} \cdot \left| (\text{skill2}_1 - \text{skill2}_2) \right| + \frac{0.20}{3} \cdot \left| (\text{skill3}_1 - \text{skill3}_2) \right| - \frac{0.15}{4} \cdot \left| (\text{E/I}_1 - \text{E/I}_2) \right| - 0.15 \cdot \left| (\text{S/N}_1 - \text{S/N}_2) \right| - 0.15 \cdot \left| (\text{T/F}_1 - \text{T/F}_2) \right| - 0.15 \cdot \left| (\text{J/P}_1 - \text{J/P}_2) \right| - 1\]

The derivation of such an equation for any of the other group learning environments should be equally straightforward. This equation also demonstrates how flexible this function is – although we assume in our formulation that all variables in a category receive the same weight, it would be quite simple to adjust them – this is probably most important for the skills category, where one skill could easily be more important than the other two.

**Objective Function**

Like all difficult problems, a great deal of ambiguity exists in the formulation of an objective function. We have compatibility scores for every dyad of students – but what are we trying to maximize? We do not wish to improve one team at the expense of another team, unless we know that the first team benefits more than the second team loses.
Maximize the average
The most intuitive thought is to maximize the average pairwise compatibility between all students on the same team, over all teams. The average is a robust measure, including all students. However, this does not describe how evenly the teams are distributed: one team may have very low compatibility scores, while another has correspondingly higher ones. Although the average over all teams is good, it is still not the complete picture, and an algorithm that maximizes the average may thus help some teams at the expense of others.

Maximize the minimum
Another possible objective function is to minimize the worst of the compatibility scores between any two students on any team, so that no team has a pair of students with a very low compatibility compared to all others. This is important because if two people in a group do not get along, it can affect the entire group. In this case, the value of the objective function will essentially say “No two students on any team are less than x% compatible” with each other. The benefit of this objective function is that it allows us to say something definite about the state of every team. Unfortunately, it also is not especially robust – switching two students, as can sometimes happen in a real classroom, can drastically change the value.

Both of these objective functions have some validity; later experiments will test which objective function works better in simulation, and in the real world. With these objective functions in mind, we next turn to heuristics for solving this NP-hard problem.
IV. Solving the Problem

Now that we have formulated a matrix of student compatibility scores, and determined that the best way to frame the problem is to examine dyads on teams, maximizing either the minimum or the average pairwise compatibility across all teams, we need to find heuristics to improve the objective function efficiently.

Given the difficulty of solving an integer program for a large number of students, five heuristics were instead designed to quickly improve the objective function to find a good solution. In general, these heuristics used three tools, alone or in combination: enumeration, random generation, and pairwise exchange. Each addresses the same problem in a different way – enumeration lists out the different possibilities in order, always keeping the best one, while random generation provides solutions without any order, which is much quicker but not guaranteed to ever find the optimal solution. Pairwise exchange, rather than generating solutions, improves an existing solution.

In addition to the tractability issue, there is another benefit to using heuristics rather than an integer program – the objective function becomes quite flexible. For all of the programs described below, the algorithm will compare existing solutions to new solutions, and adopt the new solutions when, depending on which objective function is being used, the minimum compatibility score (average compatibility score) is higher than the current solution’s minimum (average), or when the two minimums (averages) are equal and the average (minimum) of the new solution is greater than that of the current solution.

Exhaustive Enumeration

The simplest concept is exhaustive enumeration – working through every possible permutation of students, and storing the best permutation. Obviously, this method, if left
to run forever, would find the optimal solution; however, in a combinatorial optimization problem such as this, the number of permutations quickly becomes too large to solve to completion this way in any reasonable time. Surprisingly, this most intuitive of concepts is also the most difficult to program.

The version of exhaustive enumeration used here enumerates by lexicographical order; that is, it begins with the students in ascending order, 1 through n (where n=the total number of students), and calls this set of teams the current optimal, and the enumeration continues until the last sequence, descending order n through 1, is found. The program contains a recursive function, which we will now briefly describe.

Imagine the set of students as an array, x. When we are evaluating which student to place in a given position, p, in the array x, all students in positions > p are put into another array, called “available”, consisting of students who have not been assigned. When the student in position p is replaced by a different student from the available array, the old student is placed back in the available array. The available array is always kept in ascending order. We let

\[ n = \text{total number of students} \]
\[ p = \text{current position of algorithm.} \]
\[ x_p = \text{the sequential ID number of the student in the current position.} \]
\[ k= \text{total number of teams.} \]

Let’s call the function Increment. It works as follows:

While \( x_p < \) the ID number of the highest available student:
  - Place the next higher available student in \( x_p \).
  - Place remaining available students in ascending order in all positions \( >p \).
  - Calculate minimum and average
- If \( \text{min} > \text{optimal min} \), \( \text{optimal min} := \text{min} \), \( \text{optimal ave} := \text{ave} \), 
  \[ \text{optimal}_x \_p := x_p \text{ for all } p. \]
- Else if \( \text{min} = \text{optimal min} \) AND \( \text{ave} > \text{optimal ave} \), \( \text{optimal ave} := \text{ave} \), 
  \[ \text{optimal}_x \_p := x_p \text{ for all } p. \]
- If \( x_p \neq \text{highest available student} \), then
  - \( p := n-n/lk \).
  - Do Increment.
- If \( x_p = \text{highest available student} \) AND \( p > 1 \), \( p := p-1 \).

We would start this function at \( p = n-n/lk \), the last student in the second-to-last team. The reason that \( p \) is never greater than \( n-n/lk \) (we never increment on the last team) is that once the other teams have been assigned, there is only one possible set of members for the remaining team.

In words, the above pseudo-code is a series of nested loops, with the innermost loop incrementing the last student on the second-to-last team, and each loop outside of that incrementing the previous student. We begin in the innermost loop, incrementing the last student on the second-to-last team, and assigning the remaining students to the last team.

In other words, if the first set of students is \((1,2,3) (4,5,6) (7,8,9), (10,11,12)\), the next set is \((1,2,3) (4,5,6) (7,8,10), (9,11,12)\). That position continues to be incremented until the number \( n \) is reached, at which point this innermost loop ends and the next loop increments the second-to-last student, triggering the innermost loop to cycle through the remaining possibilities. This continues, with each outer loop triggering the innermost loop to start and progress back out to the position that was just incremented. For instance, after the set \((1 2 3) (4 5 6) (12 11 10) (9 8 7)\), the position \( p \) will be 6, and the next set will be \((1 2 3) (4 5 7) (6 8 9) (10 11 12)\). The value of \( p \) will then return to 9, and all possible permutations of the last two teams will cycle through before \( p \) will return to 6 and increment again to \((1 2 3) (4 5 8) (6 7 9) (10 11 12)\).
The first five sets of teams generated by this method when \( n=12 \) and \( k=4 \) are:

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 8 \ 10) \ (9 \ 11 \ 12)
\]

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 8 \ 11) \ (9 \ 10 \ 12)
\]

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 8 \ 12) \ (9 \ 10 \ 11)
\]

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 9 \ 8) \ (10 \ 11 \ 12)
\]

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 9 \ 10) \ (8 \ 11 \ 12)
\]

For the maximize-the-average function, this slow progression is the only way to explore all possibilities (with the above code altered so that first the average is compared to the current optimal’s average, and then if they are equal, the minimum is compared to the current optimal’s minimum). However, with the maximize-the-minimum objective function, a better method can be applied.

Let’s say that, as we increment numbers to get a solution, every time we have a complete team (i.e. after incrementing the last student on a team) we can calculate the minimum compatibility on that team. If we find that this team has a minimum less than the minimum of the current optimal solution, no combination of students in subsequent teams will result in a complete set of teams that beats the current optimal solution. Therefore, we should skip those combinations altogether and proceed with the next incrementing.

As an example, say that the current optimal set of teams was

\[
(1 \ 2 \ 3) \ (4 \ 5 \ 6) \ (7 \ 8 \ 12) \ (9 \ 10 \ 11),
\]

with minimums of .25, .13, .28, and .33. Currently, the enumeration is on: \((1 \ 2 \ 3) \ (4 \ 6 \ 7) \ (5 \ 8 \ 9) \ (10 \ 11 \ 12)\), which was the first combination found after incrementing the last number of the second team. The minimums of these teams are .25, .07, .32, and .45. We can see that no combination of the last six students is going to replace the current optimal. Therefore, instead of cycling through the possible values of the last two teams, starting with \((5 \ 8 \ 10) \ (9 \ 11 \ 12)\), as you would have done in the above algorithm, you can jump back to the last student on the second team, which is the team with the lowest minimum, and increment there instead. In fact, as this example demonstrates, we should always work on the team with the lowest minimum. If we increment again and get \((1 \ 2 \ 3) \ (4 \ 6 \ 8) \ (5 \ 7 \ 9) \ (10 \ 11 \ 12)\) with minimums of .25, .33, .32,
and .45, we notice that we have a new optimal set of teams. We also see that now the lowest team is the first team, and that all subsequent combinations in the last three teams will not improve the optimal, so our next move is to increment the last student on the first team, to (1 2 4) (3 5 6) (7 8 9) (10 11 12) and start again at the innermost loop. This time-saving method is not available when we are trying to improve the average, because we can no longer say that further combinations of students will not improve the average. Thus, in order to preserve the exhaustive aspect of this heuristic, the algorithm above has been modified to include the following for the maximize-the-minimum function only:

- After each team is formed, calculate the min and compare to the optimal min.
  - If the min of the team is less than the optimal min, do not call Increment or change the value of p.
  - After a new solution is found (optimal or not optimal), change p to the last student in the team with the lowest min. If the last team has the lowest min, then change p to the last student in the second-to-last-team.

Exhaustive enumeration may be very slow, but it has the advantage that it is deterministic – it is guaranteed to run the same way each time and return the same answer, and that answer is non-decreasing over time. For very small problems, this program can quickly solve to completion, allowing us to examine, for small situations, the true optimum.

**Pairwise Exchange**

A well-known algorithm for improving solutions in vehicle routing and other such problems that deal with grouping things is pairwise exchange, in which two items in different groups switch places to improve the solution. The general algorithm is as follows:

1. Find an initial solution.
2. Examine every possible two-student exchange.
3. Choose the one that most improves the objective function.
4. If no solution improves the objective function, stop.
We need to add some specifics for the first three steps in order to completely explain our application of pairwise exchange. In step one, for this first application of pairwise exchange, we set the initial solution at the same initial solution of the exhaustive enumeration method – all students in ascending order 1 through n. In step two, we must define “every possible” exchange. In the case of the maximize-the-minimum objective function, we will replace the current optimal solution with the new set of teams if the minimum is greater than the optimal minimum or if the minimums are equal and the average is greater than the optimal average. In such a case, we can take a very limited view of which exchanges are possible.

If we have 100 students in teams of five, for example, and we know that the minimum pairwise compatibility is between students x and y, who are together in team j, we know that any pairwise exchange which doesn’t involve either student x or student y will not improve the minimum. Therefore, we only need to look at student x exchanged with every student on every team except team j, and student y exchanged with every student on every team except team j – a total of 95*2=190 exchanges, instead of the 95*100 = 9500 exchanges one would need in order to exchange every student with every other student on a different team.

Although we cannot say for the maximize-the-average objective function that students x and y above are the only choices for improving the average objective function, we do know that they are good choices to start improving from. In the case of exhaustive enumeration, we could not take “shortcuts” with the maximize-the-average objective function program, because that would prevent the program from being an exhaustive enumeration of every valid possibility. With pairwise exchange, however, we already know the algorithm isn’t certain to find us the best solution if run indefinitely – it is only going to find us a good solution. Using the 190 exchanges rather than the 9500 exchanges will still get us a good solution, so the purpose of the heuristic is not destroyed by this shortcut. Therefore, since the difference in the number of operations required is
so great, we will apply the same method to the average objective function that we do to the minimum. Although this difference in computation time means little in the case of pairwise exchange on a single solution, we shall see in later applications of pairwise exchange that speed is desirable.

This pairwise exchange algorithm, improving from one initial solution until it can no longer improve, is not guaranteed to find the best solution, because the paths from the initial solution to the best solution may involve switches that make only small improvements, or even do not improve the solution, along the way. However, this method is extremely quick, and therefore should be attempted to see how it compares to the slow-but-steady exhaustive enumeration method. Some sacrifice of objective function value may be worth the shorter calculation time, depending on results.

**Enumeration and Pairwise Exchange**

Having developed the two methods above, it also seems reasonable to attempt some combination of them. Rather than start pairwise exchange with the initial solution of students arranged in ascending order, we can run enumeration for some period of time and use the resulting solution as the starting point for pairwise exchange.

This method has the advantage of starting pairwise exchange at a solution that is already better than the initial solution. This method is guaranteed to work better than exhaustive enumeration alone for any given time period, because pairwise exchange can only improve on the final answer from the enumeration, and will do this for little additional time since the pairwise algorithm runs extremely fast. However, it is important to note that this method is *not* guaranteed to work better than pairwise exchange alone, nor is it guaranteed to work better as enumeration runs for a longer period of time. Just because it starts at a better solution doesn’t guarantee that it will end with a better solution. The reason for this is that pairwise exchange is limited in the paths it can choose to improve a given solution. A better initial solution may have fewer possible exchanges that will
improve the solution than a worse initial solution has. Thus, although it is likely that a better initial solution will lead to a better answer, it is not guaranteed.

Random Generation

Another intuitive method of finding the best set of teams is to randomly generate many sets of teams and pick the one with the highest objective function value. The advantages of this method are that it runs quickly and is quite easy to code in C: the programmer can add a random seed generator so that the program runs differently each time it runs. One disadvantage is that randomness introduces difficulties in assessing the method -- since it will not run the same way twice, the answers are likely to be different each time, and it is difficult to tell if a given answer is typical of the answers provided by the method, without multiple runs of the same program. Another disadvantage is that this algorithm does nothing active to improve the solutions. Thus, if there are very few good solutions, it is unlikely that this method will find them.

Random Generation and Pairwise Exchange

A final combination of the above methods is a random-pairwise method, in which pairwise exchange is performed on many randomly-generated initial solutions. While this preserves the disadvantages of randomness and of pairwise exchange, it eliminates the problem with random generation alone being too passive and not directly seeking to improve solutions.

Specifically, one iteration of the algorithm runs as follows:

1. Generate a random set of teams
2. Generate every possible two-student exchange.
3. Choose the best of these exchanges
   - Compare to the current optimal.
   - If it beats the current optimal, replace current optimal and return to step 2.
This algorithm is then run for a given number of iterations. Since pairwise exchange and random both run quite quickly, it is expected that many randomly-generated sets of teams could be produced and improved in a short period of time. The advantages of this method make it the most promising of the five.

The code for the maximize-the-minimum objective function versions of these five programs, written in C, is available in Appendix I. With these five methods, we can now begin to assess which type of heuristic works best to improve the objective function value.
V. Simulation

Now that we have developed an equation for student compatibility, and several methods to improve the objective function value, for two definitions of the objective function, we face several unanswered questions: What method is the best? Does the answer depend on the size of the classroom? The number of students in a team? The objective function? How well do these methods do at beating a randomly chosen set of teams? This chapter explains the simulations designed to provide the answers, then briefly summarizes and analyzes the results to form conclusions about the nature of these heuristics.

Experimental Design

In order to begin simulation, we need several data sets containing information on student characteristics. Since a large number of actual student responses was not readily available, generated data sets, based on reasonable assumptions about the distributions of student characteristics, are required.

For these experiments, ten data sets were generated. Five of these sets contained generated data for 36 students, to be used in creating 12 teams of 3, and five data sets consisted of generated data for 100 students, to be used in creating 20 teams of 5. In this way, the effectiveness of these methods for different numbers of students per classroom and different numbers of students per team could both be evaluated. These data sets were each run on ten different programs, which used the five optimization heuristics described above (exhaustive enumeration, pairwise exchange, exhaustive enumeration with pairwise exchange on the best answer, random generation, and random generation with pairwise exchange on each answer), to improve two objective functions – maximize the minimum student compatibility, or maximize the average student compatibility.

In order to determine how the length of time a program was run affected the quality of the result, we simulated all of these methods except single pairwise exchange for five
different lengths of time: .5, 15, 30, 45, and 60 minutes. (Single pairwise exchange is only run once for each data set because it always completes within one minute.) As discussed previously, only exhaustive enumeration is guaranteed to return answers that are non-decreasing functions of the amount of time the program is run due to the randomness of the random methods and the fact that better starting points are not guaranteed to lead to better answers in pairwise exchange. However, running these methods over five different time periods will show whether extra time will, on average, produce an upward trend in objective function value.

Table 5.1 Probability estimates for generating sample data sets.

<table>
<thead>
<tr>
<th>Race: White</th>
<th>Non-white</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.708</td>
<td>0.292</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;30</td>
</tr>
<tr>
<td>0.750</td>
</tr>
<tr>
<td>30&lt;x&lt;40</td>
</tr>
<tr>
<td>0.150</td>
</tr>
<tr>
<td>40&lt;x&lt;50</td>
</tr>
<tr>
<td>0.050</td>
</tr>
<tr>
<td>&gt;50</td>
</tr>
<tr>
<td>0.050</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skill 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (low)</td>
</tr>
<tr>
<td>0.330</td>
</tr>
<tr>
<td>2 (medium)</td>
</tr>
<tr>
<td>0.330</td>
</tr>
<tr>
<td>3 (high)</td>
</tr>
<tr>
<td>0.340</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skill 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (low)</td>
</tr>
<tr>
<td>0.250</td>
</tr>
<tr>
<td>2 (medium)</td>
</tr>
<tr>
<td>0.330</td>
</tr>
<tr>
<td>3 (high)</td>
</tr>
<tr>
<td>0.420</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skill 3:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (low)</td>
</tr>
<tr>
<td>0.150</td>
</tr>
<tr>
<td>2 (medium)</td>
</tr>
<tr>
<td>0.350</td>
</tr>
<tr>
<td>3 (high)</td>
</tr>
<tr>
<td>0.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How Motivated are you to do well in this class?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (low)</td>
</tr>
<tr>
<td>0.170</td>
</tr>
<tr>
<td>2 (medium)</td>
</tr>
<tr>
<td>0.330</td>
</tr>
<tr>
<td>3 (high)</td>
</tr>
<tr>
<td>0.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gender: Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.440</td>
<td>0.560</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First letter of MBTI:</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
</tr>
<tr>
<td>0.459</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>0.541</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>0.525</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>0.475</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second letter of MBTI:</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
</tr>
<tr>
<td>0.717</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>0.283</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>0.749</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>0.251</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Third letter of MBTI:</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
<tr>
<td>0.565</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>0.435</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>0.245</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>0.755</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fourth letter of MBTI:</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
</tr>
<tr>
<td>0.050</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>0.292</td>
</tr>
<tr>
<td>J</td>
</tr>
<tr>
<td>0.292</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>0.560</td>
</tr>
</tbody>
</table>

**Probability Estimates**

We use the set of characteristics established in section III as our representatives of skill, demographics, personality, and motivation. Estimates for the probabilities of these characteristics came from several different sources. Census data provided the age and gender estimates for students in American universities (US Census Bureau, 1998). Skill
estimates were created by imagining progressively easier skills, with half of all students at a high level of the easiest skill. Motivation was similarly estimated. All of these characteristics were presumed to be independent of one another. However, the results of the Myers-Briggs Personality Type Indicator have been shown to be dependent on gender (Type Statistics and Surveys, 2002), so the last four characteristics were generated dependent on the gender of the student. A SAS program was used to generate distributions of these variables, with different distributions of the personality variables for each gender.

**Results for the Maximize-the-minimum Objective Function**

To determine the “best” solution under this objective function, the solutions were ranked by minimum pairwise compatibility, sum of pairwise compatibilities, and time to solution. In all cases, the best solution was generated by the pairwise exchange on randomly generated solutions, although in the 36 student case it was the pairwise exchange on random after 30 or 45 minutes, while for 100 students it was the same method either after 45 or 60 minutes. The worst solutions were consistently from the random generation by itself.

Figures 5.1 and 5.2 below document the results after 60 minutes for all methods except pairwise alone, which always completes within one minute. Clearly, although random-pairwise is always the best and random alone is the worst, the other three methods (pairwise alone, exhaustive enumeration, and enumeration with pairwise) behave differently depending on the data set. For a graphical view of each data set’s results over the five time periods (5, 15, 30, 45, and 60 minutes), see Appendix II.

After the methods were ranked, the highest objective function value (where the value is the minimum pairwise compatibility between any two students on the same team) was recorded for each data set, and the percentage difference between each method’s
objective function value and the highest value was calculated. Table 5.2 shows the average results over all five data sets for these calculations.

**Figure 5.1 Maximize-the-minimum results for classes of 36 students - teams of 3**

**Figure 5.2 Maximize-the-minimum results for classes of 100 students - teams of 5**
As the classes get larger, the percentage below the best solution increases for all methods, but the average ranking for each method remains similar between the two problems. Overall, for the maximin objective function, the best method is the combination of random and pairwise exchange, followed by enumeration plus pairwise exchange and pairswitch alone, with enumeration alone and random alone the worst two methods. Random alone is fairly close behind enumeration alone for 36 students, but trails behind significantly for 100 students.

Table 5.2 Percentage below best solution and average rank for maximize-the-minimum objective function

<table>
<thead>
<tr>
<th>For 36 Students</th>
<th>average % below best solution</th>
<th>average rank</th>
<th>For 100 Students</th>
<th>average % below best solution</th>
<th>average rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairswitch only</td>
<td>-9%</td>
<td>9.8</td>
<td>Pairswitch only</td>
<td>-13%</td>
<td>9.6</td>
</tr>
<tr>
<td>Enumerate only (1)</td>
<td>-21%</td>
<td>17.6</td>
<td>Enumerate only (1)</td>
<td>-64%</td>
<td>21</td>
</tr>
<tr>
<td>Enumerate only (15)</td>
<td>-13%</td>
<td>12.8</td>
<td>Enumerate only (15)</td>
<td>-21%</td>
<td>14.2</td>
</tr>
<tr>
<td>Enumerate only (30)</td>
<td>-13%</td>
<td>13.8</td>
<td>Enumerate only (30)</td>
<td>-18%</td>
<td>12</td>
</tr>
<tr>
<td>Enumerate only (45)</td>
<td>-13%</td>
<td>14.8</td>
<td>Enumerate only (45)</td>
<td>18%</td>
<td>12.4</td>
</tr>
<tr>
<td>Enumerate only (60)</td>
<td>-12%</td>
<td>14</td>
<td>Enumerate only (60)</td>
<td>-17%</td>
<td>13</td>
</tr>
<tr>
<td>Enumerate and pairswitch (1)</td>
<td>-12%</td>
<td>10.6</td>
<td>Enumerate and pairswitch (1)</td>
<td>-15%</td>
<td>9.6</td>
</tr>
<tr>
<td>Enumerate and pairswitch (15)</td>
<td>-8%</td>
<td>6.8</td>
<td>Enumerate and pairswitch (15)</td>
<td>-14%</td>
<td>8.6</td>
</tr>
<tr>
<td>Enumerate and pairswitch (30)</td>
<td>-8%</td>
<td>7.8</td>
<td>Enumerate and pairswitch (30)</td>
<td>-13%</td>
<td>8.6</td>
</tr>
<tr>
<td>Enumerate and pairswitch (45)</td>
<td>-8%</td>
<td>8.8</td>
<td>Enumerate and pairswitch (45)</td>
<td>-12%</td>
<td>8.8</td>
</tr>
<tr>
<td>Enumerate and pairswitch (60)</td>
<td>-7%</td>
<td>9.2</td>
<td>Enumerate and pairswitch (60)</td>
<td>-11%</td>
<td>8.2</td>
</tr>
<tr>
<td>Random and pairwise (1)</td>
<td>0%</td>
<td>5.6</td>
<td>Random and pairwise (1)</td>
<td>5%</td>
<td>5</td>
</tr>
<tr>
<td>Random and pairwise (15)</td>
<td>0%</td>
<td>3.6</td>
<td>Random and pairwise (15)</td>
<td>2%</td>
<td>3.6</td>
</tr>
<tr>
<td>Random and pairwise (30)</td>
<td>0%</td>
<td>2.4</td>
<td>Random and pairwise (30)</td>
<td>2%</td>
<td>3.4</td>
</tr>
<tr>
<td>Random and pairwise (45)</td>
<td>0%</td>
<td>1.4</td>
<td>Random and pairwise (45)</td>
<td>1%</td>
<td>1.6</td>
</tr>
<tr>
<td>Random and pairwise (60)</td>
<td>0%</td>
<td>2.6</td>
<td>Random and pairwise (60)</td>
<td>0%</td>
<td>1.4</td>
</tr>
<tr>
<td>Random alone (1)</td>
<td>-33%</td>
<td>21</td>
<td>Random alone (1)</td>
<td>-50%</td>
<td>20</td>
</tr>
<tr>
<td>Random alone (15)</td>
<td>-19%</td>
<td>17.2</td>
<td>Random alone (15)</td>
<td>-39%</td>
<td>17.4</td>
</tr>
<tr>
<td>Random alone (30)</td>
<td>-18%</td>
<td>17.4</td>
<td>Random alone (30)</td>
<td>-40%</td>
<td>18.6</td>
</tr>
<tr>
<td>Random alone (45)</td>
<td>-19%</td>
<td>16.8</td>
<td>Random alone (45)</td>
<td>-38%</td>
<td>17.4</td>
</tr>
<tr>
<td>Random alone (60)</td>
<td>-18%</td>
<td>17</td>
<td>Random alone (60)</td>
<td>-37%</td>
<td>16.6</td>
</tr>
</tbody>
</table>

There are several reasons that 36-student solutions were better than those of 100-student classes. An obvious reason is that the 36-student situation can loop through more iterations of any method than the 100-student problem can within the same time period. However, this result is clearly also a result of having teams of 5 instead of teams of 3, which increases the difficulty of the problem, since more constraints will be added by the
increase in the number of dyads per team than will be subtracted by the decrease in the number of teams.

*Performance of the maximize-the-minimum objective function at improving average pairwise compatibility*

In order to get a feel for how different the two measures (minimum compatibility and average compatibility) are, the average compatibilities resulting from the maximize-the-minimum objective function simulations, were also analyzed. Recall that the primary objective of this objective function is to maximize the minimum, with a secondary objective to maximize the average. The results of the maximize-the-minimum objective function runs were re-ranked with the priorities being highest average compatibility followed by minimum compatibility, followed by time to solution. The clearest result is that the difference between the best average compatibility score and any of the other scores is much smaller, because the average is a much more robust measure than the minimum. For example, the minimum compatibility for 100 students after .5 minutes of random generation is 64% worse than the best minimum, while the average compatibility for this method is only 16% worse than the best average.

If we were to use the results of the maximize-the-minimum simulations to assign teams based on average compatibility, the best solution would be the same for the datasets containing 36 students, but for the 100-student data sets, the best average compatibility solution was sometimes from the random-pairwise method, but other times from the enumeration-pairwise method. However, neither of these comparisons tells us anything about how the average compatibility resulting from this objective function differs from the average compatibilities attainable from an objective function that maximizes the average compatibility, with a secondary goal of maximizing the minimum compatibility. The simulations were re-run with this new objective function.
Results for the Maximize-the-average Objective Function

The same five methods – pairwise exchange on one solution, exhaustive enumeration, exhaustive enumeration with pairwise exchange on the best result, random number generation, and pairwise exchange on randomly generated solutions – were re-run with the objective to maximize the average compatibility, with a secondary goal of maximizing the minimum compatibility. Rather than recalibrate the number of iterations each program ran in order to get the time increments (.5, 15, 30, 45, and 60 minutes) established for the first study, each average compatibility version of these methods was run for the same numbers of iterations that the minimum compatibility version had been run in the previous experiment.

The solutions were then ranked in a similar manner to the earlier results, by average pairwise compatibility, minimum pairwise compatibility, and time to solution. For the data sets of 36 students, the best method was similar to the best method under the other objective function, pairwise exchange on randomly generated solutions for 15, 30, or 45 minutes. For the 100-student data sets, however, two of the five data sets found the best solution through exhaustive enumeration with pairwise exchange on the best result, while the other three found the best solution through pairwise exchange on randomly generated solutions for 15 or 45 minutes. However, it is important to note that in the cases in which exhaustive enumeration and pairwise exchange were the best methods, the results found from 45 minutes of pairwise exchange on randomly generated solutions were less than 1% worse than the exhaustive-pairwise solutions; thus overall, the best method for the majority of classes is still random-pairwise, for both 36 and 100 students. Figures 5.3 and 5.4 below show the results after 60 minutes for all methods except pairwise alone, which completes within a minute.
These data were also ranked, and the percentage difference between the best average pairwise compatibility for each class and every other method's average pairwise.
compatibility was calculated, as in the minimum compatibility data set above. These results are shown below in Table 5.3.

Table 5.3 Percentage below best solution and average rank for maximize-the-average objective function

<table>
<thead>
<tr>
<th></th>
<th>36 students</th>
<th>100 students</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave. % worse (ave compat.)</td>
<td>Ave rank - Ave. compatibility</td>
</tr>
<tr>
<td>Pairswitch only</td>
<td>-3.4%</td>
<td>9</td>
</tr>
<tr>
<td>enumerate only (1)</td>
<td>-19.0%</td>
<td>20.8</td>
</tr>
<tr>
<td>enumerate only (15)</td>
<td>-14.9%</td>
<td>17.8</td>
</tr>
<tr>
<td>enumerate only (30)</td>
<td>-14.9%</td>
<td>18</td>
</tr>
<tr>
<td>enumerate only (45)</td>
<td>-14.2%</td>
<td>18.4</td>
</tr>
<tr>
<td>enumerate only (60)</td>
<td>-13.8%</td>
<td>17.4</td>
</tr>
<tr>
<td>Enumerate and pairswitch (1)</td>
<td>-3.2%</td>
<td>8.8</td>
</tr>
<tr>
<td>Enumerate and pairswitch (15)</td>
<td>-2.9%</td>
<td>7.2</td>
</tr>
<tr>
<td>Enumerate and pairswitch (30)</td>
<td>-3.2%</td>
<td>8.4</td>
</tr>
<tr>
<td>Enumerate and pairswitch (45)</td>
<td>-3.9%</td>
<td>9.6</td>
</tr>
<tr>
<td>Enumerate and pairswitch (60)</td>
<td>-3.1%</td>
<td>8</td>
</tr>
<tr>
<td>Random and pairwise (1)</td>
<td>-0.5%</td>
<td>5</td>
</tr>
<tr>
<td>Random and pairwise (15)</td>
<td>-0.1%</td>
<td>3</td>
</tr>
<tr>
<td>Random and pairwise (30)</td>
<td>-0.1%</td>
<td>2.4</td>
</tr>
<tr>
<td>Random and pairwise (45)</td>
<td>0.0%</td>
<td>1.8</td>
</tr>
<tr>
<td>Random and pairwise (60)</td>
<td>0.0%</td>
<td>2.8</td>
</tr>
<tr>
<td>Random alone (1)</td>
<td>-14.1%</td>
<td>18.6</td>
</tr>
<tr>
<td>Random alone (15)</td>
<td>-7.4%</td>
<td>13.8</td>
</tr>
<tr>
<td>Random alone (30)</td>
<td>-7.5%</td>
<td>13.8</td>
</tr>
<tr>
<td>Random alone (45)</td>
<td>-7.4%</td>
<td>13.6</td>
</tr>
<tr>
<td>Random alone (60)</td>
<td>-6.8%</td>
<td>12.8</td>
</tr>
</tbody>
</table>

Comparative Results

Now that we have the results from both objective functions, we can use the data as a whole to answer the questions we posed at the beginning of this section.
Which heuristic is best?

Looking at the data from each of the objective functions separately, we can compare the ranking of each method under each objective function. For instance, in the case of 36 students, Figure 5.5 below shows each method’s average rank under the maximin objective function and under the average compatibility objective function. Stacking the two, we can see that the best-ranked methods, under either objective function, for getting the best minimum compatibility, are the random-pairwise methods, followed by pairwise exchange on one solution, enumeration and pairwise exchange, with enumeration alone and random alone performing the worst. The results are quite similar for the best methods of finding the best average compatibility, and the same pattern holds for 100-student data sets, although, in the 100-student groups, enumeration and pairwise at larger amounts of time beats pairwise alone.

![Figure 5.5 Overall ranking of heuristics by minimum compatibility from both objective functions](image)
These results use rankings from each objective function alone, rather than finding an overall ranking. However, for a closer look at the rankings, we can analyze the data as a whole by combining the results from all 42 types of runs (four methods at five time periods and one method at one time period, for two objective functions) and ten data sets. We can now calculate an overall ranking for each method under each objective function.

For each of the 42 types of runs, we have now three rankings – average compatibility, minimum compatibility, and overall (the average of the two other rankings). Table 5.4 below shows the top five for each type of ranking.

<table>
<thead>
<tr>
<th>Rank of average compatibility</th>
<th>Rank of minimum compatibility</th>
<th>Overall Rank</th>
</tr>
</thead>
</table>
| randompair (45) - ave compat. | randompair (45) - minimax obj. fc | randompair (45) - minimax obj. fc |}

Clearly, both the separate rankings and these combined rankings indicate that there is one heuristic that is better than the others, regardless of which objective function is used: randomly generated solutions with pairwise exchange performed on each solution. Even running this heuristic for less than one minute ranks higher than running any of the other heuristics for sixty minutes!

Notice that this heuristic, when run for 45 minutes, ranks higher than the same heuristic at 60 minutes. This is due to the fact that, in most situations, the values of the minimum and average compatibilities were the same for both time periods, and so the method that took less time was ranked higher. The differences between the results for the shortest time period (.5 minutes) and the longest (60 minutes) were never more than 2%.
(maximize-the-average) or 5% (maximize-the-minimum) for this method; thus, a fairly good solution can be found even after a very short amount of time.

*Differences between minimum and average objective functions*

One purpose of these experiments is to test the differences between the objective functions. Obviously, the maximize-the-minimum objective function does a better job at finding the highest minimum compatibility, while the average compatibility objective function does a better job at finding the highest average compatibility. However, what if a professor considered both measures important – that is, he or she desired that no group be saddled with a poor match, and also that the overall compatibility be high? In this case, the maximize-the-minimum objective function is the better method, according to Table 5.4 above. The reason for this is that the maximize-the-minimum programs do a better job of raising the average compatibility than the maximize-the-average programs do of raising the minimum compatibility. This was true across all five methods, and probably a result of the fact that the average is a more robust measure and thus far less variable than the minimum.

*Differences between large and small classes*

In terms of which programs worked the best, large and small classes were quite similar. However, there were a few subtle differences. The rankings for 36 students and 100 students on average and minimum compatibility are listed in Table 5.5 below. For both average and minimum, the major difference between the two is that the longer lengths of time are generally ranked higher for 100 students than they are for 36 students. This indicates that improvement is more likely to occur with additional time (within the one hour period that we tested) for 100 students than it is for 36 students. For average compatibility, we have the additional result that for 100 students, an hour of enumeration with pairwise exchange on the best answer is ranked in the top five rather than less than one minute of random-pairwise.
Table 5.5 Top-ranked heuristics by ranking system and class size

<table>
<thead>
<tr>
<th>Rank of average compatibility - 36 students</th>
<th>Rank of average compatibility - 100 students</th>
<th>Rank of minimum compatibility - 36 students</th>
<th>Rank of minimum compatibility - 100 students</th>
</tr>
</thead>
<tbody>
<tr>
<td>randompair (45) - ave compatibility obj. fon.</td>
<td>randompair (45) - ave compatibility obj. fon.</td>
<td>randompair (45) - minimax obj. fon.</td>
<td>randompair (60)- minimax obj. fon.</td>
</tr>
<tr>
<td>randompair (30) - ave compatibility obj. fon.</td>
<td>randompair (30) - ave compatibility obj. fon.</td>
<td>randompair (30) - minimax obj. fon.</td>
<td>randompair (45) - minimax obj. fon.</td>
</tr>
<tr>
<td>randompair (60) - ave compatibility obj. fon.</td>
<td>randompair (15) - ave compatibility obj. fon.</td>
<td>randompair (60)- minimax obj. fon.</td>
<td>randompair (30) - minimax obj. fon.</td>
</tr>
<tr>
<td>randompair (15) - ave compatibility obj. fon.</td>
<td>randompair (30) - ave compatibility obj. fon.</td>
<td>randompair (15) - minimax obj. fon.</td>
<td>randompair (15) - minimax obj. fon.</td>
</tr>
<tr>
<td>randompair (1) - ave compatibility obj. fon.</td>
<td>Enumeration/ pairwise - ave compatibility obj. fon. (60)</td>
<td>randompair (1) - minimax obj. fon.</td>
<td>randompair (1) - minimax obj. fon.</td>
</tr>
</tbody>
</table>

Differences between large and small groups

The two types of data sets that we have discussed so far were 36 students in groups of three and 100 students in groups of five. As expected, the objective function values for the 100-student data sets were lower than those for the 36-student data set, as a result both of increased size (and thus fewer iterations made within each time period) and the larger number of students in each group. We now modify the experiment somewhat in order to separate these two factors and see what effect each one has on objective function value. One student was randomly removed from each of the 100-student files in order to form five new data sets of 99 students each, to be made into 33 teams of 3. These new data sets were used on all five methods and all five time periods, with the maximize-the-minimum objective function.

The results were extremely interesting. Since the datasets used for 100 students and 99 students were practically the same, one can directly compare the results. The best results in all five of the 99-student data sets were, like for the 100-student sets, from the random-pairwise method, although for the 99-student data sets the best answer came after shorter time periods than for the 100-student data set, consistent with the behavior of the 36-student data sets. This indicates that the additional time to reach the best solution that we see with the 100-student sets is primarily a function of the larger teams, not the larger classroom. Even when the random-pairwise method was run for less than one minute, the
objective function value for the 99-student data set was 28% higher than the same program for the same time period for 100 students.

Figure 5.6 Minimum compatibility by class size and heuristic

Figure 5.6 shows the relationship between the three sizes of data set, comparing 100 students in teams of 5 to 99 students in teams of 3 and 36 students in teams of 3. For most of the methods, the difference between 99 students and 36 students is very small, with the exception of enumeration only for less than one minute and random only for all time periods – in these cases, 99 students performed much worse (between 24 and 29% worse) than 36 students. From the above, we can conclude that, for most methods, the difference between small classes with small teams and large classes with large teams is mostly attributable to the number of students in the team, not the size of the class.

However, professors should not take this conclusion too far. The difference between 36 students and 100 students may be minimal, but that does not mean that the difference between 100 and 500 is also small. Since one would assume a non-linear relationship between the increase in time necessary to achieve the best answer and the number of
students, the best method for a professor with several hundred students is probably to break down the class into groups of around 100, and optimize each of these separately. This would allow the professor to use a small period of time (1 minute or 15 minutes) with a high likelihood of getting very close to the best answer available by these methods; since the professor is seeking a good set of teams rather than the absolute optimal, splitting the classroom into sets of 100 should not materially affect the quality of the teams in the entire classroom.

The fact that larger teams will get worse minimum and average compatibilities, and take longer to get good compatibilities, is a necessary result of assessing team worth using dyads of students within the team, since there are more dyads in larger teams than there are in smaller ones. Framing this problem as a coverage problem would not have gotten this result, because the more students on a team, the more likely that one of them will possess a given trait.

There has been a great deal of research on the appropriate size of learning teams. Several experiments have shown that, as group size grows, the inequality of participation also grows, such that one person dominates the group (Shaw, 1981; Jaques, 2000; Huber & Effer, 1990). Additionally, an increase in the number of group members leads to more disagreement, an increasing probability of subgroups forming, less cohesion, lower satisfaction, and possibly less consensus (Shaw, 1981). The concept of social loafing, in which individual group members decrease their efforts as the size of the group grows, has been shown to occur in groups doing cognitive tasks, although the effect is mitigated or eliminated when the task is especially difficult or individual team members are held accountable for their contributions (Fleishman & Zaccaro, 1992). As a consequence of this and other research, “working groups (students doing research projects together for several days, weeks, or months, for example) seem to be most successful with three members. Long-term working groups larger than three often become logistically cumbersome.” (Brufee, 1993). Huber and Effer also recommend groups of two or three (1990), while some others set the magic number at four — “because they are small enough
to promote interaction, large enough to tolerate an occasional absence, and balanced enough to permit focused activities in pairs” (Millis & Cottell, 1998). Others simply advise that the team should be as small as possible to complete the appointed task (Johnson et al, 1991; Shaw, 1981).

Our method, based as it is on dyads, is able to reflect this preference for smaller groups, because it is the increase in the number of interactions that make group dynamics more difficult as the group grows. Thus, our method is reflecting a reality that larger teams present more difficulties.

**Gender constraints**

Another set of simulations was performed as the result of feedback from the classroom experiment described in the following chapter. One of the most contentious issues raised in the experiment was that of gender balance on teams. All three of the professors involved mentioned that they had had trouble in the past with teams that had only one female undergraduate, and several male undergraduates. The two other methods use in the experiment (Professor Dunphy’s intuition method and Professor Breslows’ pseudo-random method) specifically avoided assigning one and only one female to any team. There is some support for this idea in research. Felder’s experiences with cooperative learning groups in a series of chemical engineering courses at NC State led him to also advise that no group should have one and only one female or minority member (1994). Other researchers included females as a type of minority, and asserted that minorities should not be placed so that there is one on each team (Miller & Harrington, 1992; Webb, 1992). This would mean that in classes with groups of three, teams could consist only of all males, all females, or one male and two females. For teams of four and five, this limitation is not quite as binding – by this policy, two females and three males on a team of five are still acceptable, even though the females are outnumbered.
Preliminary exploration

A small simulation was designed to determine the effect such a constraint as "no teams with exactly one female" would have on the solution. To do this, we developed a new algorithm for the random-pairwise heuristic. In each iteration, the following steps are carried out:

1. Generate a random set of teams.
2. Enumerate all possible exchanges of two students.
   a. Determine if the exchange will cause any teams with exactly one female.
   b. If any team resulting from the exchange has exactly one female, then eliminate that exchange from the list.
3. Choose the best of the remaining exchanges.
   a. Compare this to the current optimal. If the exchange beats the current optimal, make the exchange and repeat step 2.

This algorithm is not guaranteed to always return only sets of teams that meet the constraint. For instance, in the case in which only one iteration is performed, if the randomly-generated set of teams has no possible improvements which both meet the constraint and improve the objective function, then the original randomly-generated set of teams is returned as the solution; these teams are not guaranteed to meet the constraint. However, experimentation has shown that as few as 1000 iterations (taking less than 20 seconds to run) will consistently yield an answer that does meet the constraint. Since this algorithm was developed as a simple test to determine the effect of the gender constraint on the objective function value rather than an ironclad method of guaranteeing after even one iteration the non-existence of teams with one female, this level of accuracy is acceptable.

Preliminary results are encouraging. For five sets of 36 students and five sets of 100 students, this modified random-pairwise program was run for five different time periods with the maximize-the-minimum objective function. The minimum compatibility score was an average of 4% below the same program run without the restriction. The program
also ran much faster, on average 90% faster, because the number of pairwise exchanges it compared was much smaller than before. There was no difference between the changes seen in the 36-student data sets and the 100-student data sets. Although the average decrease in the minimum compatibility score was 4%, results for the 10 data sets ranged from 1% to 9% worse than the unconstrained score. A regression clearly shows the relationship between the percentage of female students and the resulting decrease in minimum compatibility score, significant at p = .005, and a scatter plot (Figure 7.1) visually demonstrates the relationship. This relationship is logical – the more females there are, the easier it is to form groups that do not have exactly one female.

![Figure 5.7 Relationship of % female to change in performance](image)

Although this is a small trial of the addition of a gender constraint, the small decrease in the values of the results demonstrates that, should professors find such constraints a necessary addition to a team assignment algorithm, our formulation can be adapted to meet that need without a large decrease in effectiveness.

**Beating Random**

The goal of this thesis is to show that there is a relatively quick and easy way to form teams intelligently in large classrooms that is more successful than random assignment.
Thus, as part of these simulations, we need to examine the nature of randomly chosen teams, and compare them to the teams found with our other methods. One of the simplest methods used in these simulations was to simply generate many random sets of teams, and pick the best one. This program can also be used to approximate the distribution of attributes such as the minimum compatibility score for random teams.

*Simple example*

For a data set of 12 students, exhaustive enumeration quickly runs to completion. In this example, the true optimal minimum compatibility score is .425. When 10 million randomly generated sets of teams are examined, we get the same answer, along with a distribution of the minimums for the 10 million sets, and the average. As Table 5.7 below illustrates, a very small percentage of simulated results (.46%) came within the same bucket as the answer; only 6.3% of simulated scores came within 70% of the optimal score. The average minimum compatibility, .187, is less than 50% as good as the optimum. Thus, even for a very simple problem, the odds of randomly choosing teams with scores as high as these heuristics do are quite small.
Larger examples

For the data sets of 36 and 100, we do not know the optimal answer; however, comparing the distributions of the datasets to the results obtained by the random-pairwise heuristic demonstrate even more strongly how unlikely is the event that a randomly chosen set of teams will be as good as the heuristic's results. For the sets of 36 students, while the random-pairwise result was on average 23% better than the best of 10 million randomly generated sets of teams, it was a striking 178% better than the average of the 10 million sets. In other words, we would expect a randomly chosen set of teams to be far worse than the result of our heuristic.

![Figure 5.9 Distribution of randomly generated min. compatibility scores - 36 students](image)

For 100-student data sets, the results were even clearer. No randomly generated teams for any of the five data sets came within 66% of the best answer, and fewer than .01% of randomly generated teams came within 53% of the best answer. The best answer for each of the five data sets was, on average, 307% better than the average value of the randomly chosen teams. Thus, we can say that, in terms of the compatibility measure we have designed, the random-pairwise heuristic conclusively beats random team assignment by a wide margin.
Conclusions

In examining the results of simulations, we have discovered several important things about the nature of this problem. The random-pairwise method is superior to the other methods attempted, for both objective functions and all numbers of students and team sizes. The maximize-the-minimum function improves the average more than the maximize-the-average function improves the minimum. The decrease in objective function value between 36 students in teams of three and 100 students in teams of five is caused primarily by the increase in students per team, not the overall increase in number of students. And finally, there is less than a one in ten million chance that a randomly chosen set of teams will be as good as a set of teams generated by the random-pairwise heuristic, on our measure of student compatibility.

With this knowledge, we are ready to apply our heuristic to the real world in a classroom experiment. We now know that our heuristics work efficiently to improve the average
and minimum compatibility scores, but we still need some evidence that this compatibility score is positively correlated to some aspect of team satisfaction.
VI. Classroom Experiment

Having developed a mathematical method for team assignment, and tested various algorithms on artificially generated data sets, we are now ready to apply this method to a real classroom environment to determine the method's feasibility, as well as its ability to improve team satisfaction and team grades.

We were fortunate to have just such an opportunity in the Fall 2003 semester at the Sloan School of Management at MIT. Three professors who taught sections of 15.279, Management Communications for Undergraduates, were interested in team dynamics and proposed an experiment using their students.

Experimental Design

The three sections of 15.279 had 23-25 students per section. Each section was divided into teams of 3-5 students for a semester-long team project. The professor in section A used knowledge of the students' personalities (extroversion/introversion as seen in class), demographic characteristics (gender, race, and native-speaking ability), their grades from the first few weeks of class, and intuition about compatibility to form teams. Section B's teams were created with the method developed in sections III and IV. The objective function was to maximize the minimum pairwise compatibility score between any two students on the same team, and all heuristics mentioned above were attempted, with the best set of teams coming from the random-pairwise heuristic. Since the 25 students did not divide evenly into four, the student whose responses most closely matched the mean responses was removed, and the 24 remaining students were placed by the heuristic into six teams of four. Then the extra student was added to the team that best affected the objective function value. Section C teams were pseudo-random. The professor chose the first set of randomly-generated teams in which no team possessed exactly one female. For simplicity's sake, these three sections will now be referred to as Intuition, Optimized, and Random. All students in these classes received a team formation survey which
gathered individual characteristics, as well as a team satisfaction survey three times throughout the semester. These surveys can be found in Appendix III.

Since the objective function of the Optimized method was to maximize the minimum, this method yielded the highest minimum and average compatibility score. Random, possibly as a result of the inclusion of a gender constraint, had an average compatibility almost as high as that of the Optimized method, while Intuition had the lowest average and minimum compatibilities.

![Figure 6.1 Minimum and average compatibility scores by classroom](image)

Students were surveyed three times throughout the semester. The first time was after their first team evaluation, and the third time was after they had received their final team grades. Questions were answered on a scale of 1 (most positive) to 4 (most negative).

**Formation Methods and Satisfaction**

In examining the survey results for each class, we expected to see two trends - improvement over time (especially in the Optimized group) and Optimization starting out worse than Intuition and Random, but ending up beating Random. We expect
improvement over time in all teams, as they go through the “norming” process and learn to function better together. We expect especially marked improvement for more diverse teams. Since the compatibility score was based largely on diversity, we expect to see, like Harrison et al. (1998), initially worse ratings from more diverse teams, gradually improving to work together better than less diverse teams by the end of the semester.

The second trend we expected was that intuition be the best method because, in a small classroom, a professor is likely to know far more about her students than one survey of 10 questions can reveal to any algorithm. A short survey is a poor substitute for personal acquaintance. However, in large classrooms, where personal acquaintance of every student is infeasible, the short survey is a definite improvement over total lack of information. We also expect optimized methods to beat random, because random methods will not ensure the mix of diversity and homogeneity that allows teams to learn and work well together. Thus, in a small classroom environment, we expect intuition to beat optimization, which should beat random. For the most part, both of these expectations are borne out.

*Average survey response*

The first variable we examine is Total, the average of the responses to all 19 questions, which bears out our expectation that Intuition beats the others methods in terms of team satisfaction. While Optimized teams began with worse satisfaction than Random, they ended with the same average value of 1.97, compared to 1.8 for Intuition. Recalling that lower numbers represent greater satisfaction, Figure 6.2 below shows this result.
Learning

While Total demonstrated Intuition's higher overall satisfaction levels, other sets of survey questions better demonstrated the difference between Random and Optimized. The variable Learning was the average of seven questions:

As a result of your work with this team, to what extent did your ability to do the following improve? (1=A great deal, 2=a fair amount 3=somewhat, 4=not at all).

- Communicate effectively with teammates
- Resolve conflict and reach agreement within team
- Pay attention to the feelings of all team members
- Listen to the ideas of others with an open mind
- Exercise my leadership skills
- Motivate other team members
- Work collaboratively with students of different backgrounds, cultures and levels of expertise.
Remembering again that lower values mean higher satisfaction, we see in Figure 6.3 that the variable Learning meets our expectations. Intuition, especially at the beginning of the semester, outpaces the other methods, while Random begins better than and ends worse than Optimized. Although all three groups end with students perceiving a greater amount of learning than they began with, the average improved by .33 for Optimized, while improving by less than .1 for the other two methods. The decrease of perceived learning for Intuition in the last period was mainly driven by one team, which was driven by internal conflicts to a steep decrease in satisfaction near the end of the semester.

**Team characteristics**

Another variable created by averaging a subset of the survey was team characteristics. This was an average of 10 questions on the survey, which asked, “**Rate your team’s characteristics (1=Excellent, 2=Good, 3=Fair, 4=Poor) on the following:**”

- Clear understanding of team goals
- Commitment to team goals
- Preparedness for meetings
- Efficiency of meetings
• Quality of group participation in discussions
• Ability to reach consensus on decisions
• Equal sharing of workload
• Motivation
• Support for each other
• Overall team performance

While learning improved over time for all classes, team characteristics worsened over time for Intuition and Random, while remaining essentially unchanged for Optimized. The lesson here is that in terms of team characteristics, the pseudo-random method was not materially different from the Optimized method, and there was a greater difference between Intuition and the other methods than in learning. This is an indication that higher compatibility scores do not necessarily lead to higher ratings of satisfaction on team characteristics.

The mean response to all survey questions, all learning survey questions, and all team characteristics survey questions give us some indication that the method of intuition does
form better teams for small classrooms than does optimized or random, and that that
optimized teams can surpass random in satisfaction with at least some aspects of team
formation. However, none of the differences between the mean responses are significant
due to small sample size, and even significant differences could be due to different
classroom environments or teaching methods. Therefore, another way of examining this
data is necessary to reveal a stronger link between team formation and satisfaction.

Compatibility and Satisfaction

Although the above analysis gives us a rough idea of the relationship between team
formation method and satisfaction, it does not take into account environmental factors,
such as teaching methods. In order to determine more directly if there is a link between
compatibility scores and satisfaction as measured by the three surveys given to students,
several regressions were performed. The first regression was on the variable Total3, the
average of responses to all questions in the third survey. Regressions show a relationship
between Total3 and average compatibility, significant at p=0.0977. Since lower numbers
of total indicate higher satisfaction, the negative parameter estimate for average
compatibility indicates a positive relationship between compatibility and the total
satisfaction as recorded in the survey.

Table 6.1 Results on regression of Total3

| Variable    | DF | Parameter Estimate | Standard Error | t Value | Pr > |t| |
|-------------|----|--------------------|----------------|---------|------|---|
| Intercept   | 1  | 0.08764            | 0.11411        | 0.77    | 0.4451|
| Ave_compatibility | 1  | -0.43141          | 0.25694        | -1.68   | 0.0977|
| total2      | 1  | 0.85768            | 0.09285        | 9.24    | <.0001|
| numstud     | 1  | 0.02982            | 0.02625        | 1.14    | 0.2599|

Similar regressions on other subsets of survey questions reveal that this relationship is
stronger for some questions than for others. For example, regression of the variable
Learning3, the average of seven questions measuring to what degree students felt they
learned teamwork skills from working with their teams, shows a highly significant positive relationship of average compatibility to Learning3 at p=.0112.

Table 6.2 Results of regression on Learning3

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter Estimate</th>
<th>Standard Error</th>
<th>Type II SS</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.55543</td>
<td>0.45853</td>
<td>3.82584</td>
<td>11.51</td>
<td>0.0012</td>
</tr>
<tr>
<td>AveCompatibility</td>
<td>-3.82448</td>
<td>1.46651</td>
<td>2.26116</td>
<td>6.80</td>
<td>0.0112</td>
</tr>
<tr>
<td>learning1</td>
<td>0.46412</td>
<td>0.12815</td>
<td>4.36094</td>
<td>13.12</td>
<td>0.0006</td>
</tr>
<tr>
<td>learning2</td>
<td>0.32632</td>
<td>0.12008</td>
<td>2.45548</td>
<td>7.39</td>
<td>0.0083</td>
</tr>
</tbody>
</table>

Similar regressions were performed on each of the individual survey questions, the results of which are shown in Table 6.3 below. The questions pertaining to how much students developed various team-related skills (questions 13-19) were all significant, with the exception of learning to exercise leadership skills. These results fit well with the current educational literature, which suggests that more diverse teams will learn more about working as a team. Since this experiment defined compatibility as diversity in all but one category of characteristics (personality), teams with higher compatibility scores are more diverse teams. Although an overall assessment of team performance was moderately significant (p=.095), satisfaction with specific team characteristics was, for the most part, unrelated to average compatibility. The failure of motivation to be significantly related to average compatibility indicates again that the answer to a single question may not be a good indicator of motivation.
### Table 6.3 Significance of average compatibility to influence responses to survey questions

<table>
<thead>
<tr>
<th>Survey Questions</th>
<th>Significance of average compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate your team’s Characteristics:</strong></td>
<td></td>
</tr>
<tr>
<td>1 Clear understanding of team goals</td>
<td>not sig</td>
</tr>
<tr>
<td>2 Commitment to team goals</td>
<td>not sig</td>
</tr>
<tr>
<td>3 Preparedness for meetings</td>
<td>0.0301</td>
</tr>
<tr>
<td>4 Efficiency of meetings</td>
<td>not sig</td>
</tr>
<tr>
<td>5 Quality of group participation in discussions</td>
<td>not sig</td>
</tr>
<tr>
<td>6 Ability to reach consensus on decisions</td>
<td>not sig</td>
</tr>
<tr>
<td>7 Equal sharing of workload</td>
<td>not sig</td>
</tr>
<tr>
<td>8 Motivation</td>
<td>not sig</td>
</tr>
<tr>
<td>9 Support for each other</td>
<td>not sig</td>
</tr>
<tr>
<td>10 Overall team performance</td>
<td>0.0951</td>
</tr>
<tr>
<td><strong>Please indicate the degree to which you agree or disagree with the following statements</strong></td>
<td></td>
</tr>
<tr>
<td>11 Individual team members brought valuable qualities to the team</td>
<td>not sig</td>
</tr>
<tr>
<td>12 I am enjoying working with the members of my team</td>
<td>not sig</td>
</tr>
<tr>
<td><strong>As a result of your work with this team, to what extent did your ability to do the following improve?</strong></td>
<td></td>
</tr>
<tr>
<td>13 Communicate effectively with teammates</td>
<td>0.0159</td>
</tr>
<tr>
<td>14 Resolve conflict and reach agreement within team</td>
<td>0.0548</td>
</tr>
<tr>
<td>15 Pay attention to the feelings of all team members</td>
<td>0.0815</td>
</tr>
<tr>
<td>16 Listen to the ideas of others with an open mind</td>
<td>0.1018</td>
</tr>
<tr>
<td>17 Exercise my leadership skills</td>
<td>not sig</td>
</tr>
<tr>
<td>18 Motivate other team members</td>
<td>0.0855</td>
</tr>
<tr>
<td>19 Work collaboratively with students of different backgrounds, cultures, and levels of expertise</td>
<td>0.0138</td>
</tr>
<tr>
<td><strong>Learning (Questions 13-19)</strong></td>
<td>0.0112</td>
</tr>
<tr>
<td><strong>Total (All Questions)</strong></td>
<td>0.0977</td>
</tr>
</tbody>
</table>

### Compatibility and Grades

Team success can be defined as increased team satisfaction or increased team grades. We have already shown that the results of this small survey hint at a relationship between certain aspects of satisfaction, especially development of teamwork skills, and average compatibility. We now examine the link between compatibility score and grades.
Grading for the team project parts of the class consisted of a total of 40 points. 35 points were awarded to the team as a whole, and the other 5 points were awarded to each student individually, representing the professor's evaluation of how well the student contributed to the team effort. The grade out of 40 points is referred to here as the variable total_grade, while the grade out of 35 points, common to all people in the same team, is called team_grade. Another important variable in these regressions is class1, which is equal to 1 if the student was in the Intuition classroom, 2 if the student was in the Optimized classroom, and 3 if the student was in the Random classroom.

A regression of team_grade on class1, average compatibility, minimum compatibility, and number of absences, with backward elimination, shows a significant relationship (p=.0778) between compatibility and team grade. It also demonstrates the significant relationship between which class students were in and grade.

**Table 6.4 Results of regression on Team Grade**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter Estimate</th>
<th>Standard Error</th>
<th>Type II SS</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>0.89102</td>
<td>0.03228</td>
<td>1.30541</td>
<td>761.80</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Ave_compatibility</td>
<td>0.19358</td>
<td>0.10816</td>
<td>0.00549</td>
<td>3.20</td>
<td>0.0778</td>
</tr>
<tr>
<td>class1</td>
<td>-0.02542</td>
<td>0.00653</td>
<td>0.02596</td>
<td>15.15</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

A similar regression, for total_grade, shows a relationship, not to average compatibility but to minimum compatibility, with the additional variable numstud – number of students in the team.

**Table 6.5 Results of regression on Total Grade**

| Variable         | DF | Parameter Estimate | Standard Error | t Value | Pr > |t| |
|------------------|----|--------------------|----------------|---------|------|---|
| Intercept        | 1  | 0.78509            | 0.04073        | 19.28   | <.0001|
| Min_compatibility| 1  | 0.12823            | 0.07001        | 1.83    | 0.0713|
| numstud          | 1  | 0.03603            | 0.00920        | 3.92    | 0.0002|
| class1           | 1  | -0.02361           | 0.00561        | -4.21   | <.0001|
This regression points out another complicating factor – the larger the number of students on each team in a classroom, the smaller minimum compatibility will be, since each team has more inter-student relationships to take into account. However, this regression appears to demonstrate that the larger the number of students on a team, the higher the grade. Numstud was not shown to have a significant relationship to satisfaction as determined by the survey responses. A larger data set would be necessary to fully explore the relationships between the number of students and satisfaction or grades.

Due to the extremely high correlation of minimum and average compatibility, they could not be included in the same regression. These regressions do not tell the whole story about the influence of team formation on compatibility; nevertheless, they do provide some indication that, in a larger study, average compatibility, and possibly minimum compatibility as well, have an influence on team grades.

**Experimental Difficulties**

This experiment was not a perfect way to determine the effect of team formation on satisfaction or grades. It is clear from the above results that there is some link between the Optimized method and higher satisfaction, and a stronger link between the compatibility score developed earlier and higher satisfaction (especially in learning teamwork skills) and, to a lesser extent, higher grades; however, several problems combined to hinder this analysis.

*Experimental design*

The experimental design was less than ideal for providing firm results. Three small classes were used rather than one large one because the professors who were interested enough in this research to propose an experiment happened to be working with small classes. The Optimized method was designed to help professors assign teams in large classrooms, not in the small classrooms where intuition is not too time-consuming, and generally more accurate than any mathematical method. The sample size of 73 students
and 19 teams is very small. But there is a more serious flaw – each team formation method was used by one professor, so it is difficult to tell if differences in grades and satisfaction are the result of the classroom environment or the team formation. Although the three professors graded the first few individual assignments together until they were consistent with each other in grading (to within .5 points), all of the team grades were done separately by each professor. Thus, although the average grade is higher for the Optimized section was higher, it is not clear whether this is the result of the algorithm or the grader. Grading was also very uniform within each class. For two of the classes, the standard deviation was less than 3 percentage points. This made determining the effects of compatibility on grading somewhat difficult.

Further, in-class teamwork may have been handled differently among the classes – the Intuition class met many times as teams for small-group discussion during the class, the Optimized class met slightly less frequently, while the Random teams rarely met in their groups for in-class team learning unrelated to the team project. Meeting more frequently as a team could accelerate the processes of team development, making diversity work for the teams who met more often faster than for those who did not.

Methodology

In addition to the flaws in the original experimental design, there were further difficulties in applying the design to an actual classroom situation. There were two problems with application of team formation methods. First of all, the random method was not truly random, because it ensured a gender mix; in a class where females were slightly in the minority it made sure that no team had exactly one female. This method is therefore slightly closer to the Optimized algorithm than a true random method, making it more difficult to tell how much the algorithm improves over random.

The Optimized algorithm itself was also not perfectly implemented. The Optimized teams used in section B were not the teams originally formed from the algorithm. One student threatened to drop out, and so a student from one of the original teams was
moved onto another team, reducing the average of both teams and the minimum of one team. The professor determined that moving a student from one particular team (which they believed to be the highest-qualified) to the distressed team would cause minimum disruption in the classroom – re-optimizing the entire class would probably re-form all teams, some of which had already started working together on their projects, and the only decision to make was which of the two students the professor deemed most qualified should be moved onto the distressed team. Although this is the correct decision for the classroom environment, it does mean that these results do not necessarily reflect the best results possible from the Optimized method.

In addition to the problems with the team formation algorithms in an actual classroom environment, there was a consistency problem with the team satisfaction surveys. The surveys were not taken at the same time for every student. The professors preferred to hand them out in class; those students who were absent the day the surveys were handed out were tracked down at a much later date, or in some cases, missed altogether because of a lack of communication between the professor and the researcher. In a small sample, missing or late data only exacerbates difficulties. Altogether, ten surveys were missing, all from either the first or the second time period.

Survey design
A final difficulty with the actual experience came from the design of some of the inputs to the Optimized model, which were used to form the Optimized team and calculate compatibility for all teams. Better methods of determining skill levels and motivation will improve the Optimized method.

Skill
In order to determine skill level in the categories of writing, oral presentation, and teamwork skills, we asked students to self-evaluate their level of proficiency in these three skills on a score from 1 to 3 (with 3 as the highest rating). This rating level unfortunately fails to separate students into different categories, since the majority of them gave the same answer for most questions, as shown in Table 6.6 below.
Table 6.6 Responses to skills and motivation survey questions

<table>
<thead>
<tr>
<th>Survey questions:</th>
<th>writing skills</th>
<th>oral presentation skills</th>
<th>interpersonal communication skills</th>
<th>How motivated are you to do well in this class?</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (highest)</td>
<td>38%</td>
<td>36%</td>
<td>62%</td>
<td>91%</td>
</tr>
<tr>
<td>2</td>
<td>54%</td>
<td>53%</td>
<td>38%</td>
<td>9%</td>
</tr>
<tr>
<td>1 (lowest)</td>
<td>8%</td>
<td>9%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The idea that no students go into this class with low interpersonal communication skills is unrealistic, and does not help the model separate students into categories. Although the writing and oral presentation skills reflect slightly more realistic evaluations of overall writing and oral skills, what we’re more interested in is the students’ skills relative to one another. Obviously, sophomores in the Sloan School of Management are likely to have relatively high levels of these three skills compared to the general population; however, we are trying to get maximum diversity within this somewhat homogeneous group, so we must find a way to better distinguish students.

Dr. Jane Dunphy, the professor who created the Intuition classes, had similar goals to the Optimized method when she created teams, aiming to maximize diversity of demographic, personality, and skill levels within teams. Without having the survey results that were used to create the Optimized teams, she used the results of two assignments – one oral presentation and one difficult writing assignment – to get a feel for skill level. Her evaluation of individual skills was more granular (a grading scale on percentage of possible points rather than a three-point scale) and also more accurate than self-evaluation for getting a comparative score among students.

Many researchers, instead of using self-reported skill levels on various skills, used current GPA at start of class (Koppenhaver & Shrader, 2003; Oakley, Felder, Brent & Elhajj, 2003; Siciliano, 2001), or grades in relevant classes (Felder & Brent, 1994). Unless there are special skills involved in the team project that would not be necessarily reflected in grades from previous classes, this may prove to be a far better way to separate students.
Motivation

As one can see from Table 6.6 above, the motivation question on the survey was even less able to separate students into groups with different levels of motivation than the skill questions were. There are several possible explanations for this. One is that everyone was indeed highly motivated to succeed in this class – Business Communication is a required course for nearly all of these students, and also one that is especially important to MIT students, since “communication is a high-profile course within the curriculum”, one that students especially don’t want a poor grade for on their resume (Dunphy, personal communication, February 2, 2004). For these reasons, a good motivation question may be impossible to pose to students for core required courses.

Another possible explanation is that this question doesn’t accurately measure motivation because students have different definitions of “motivation”. Some students may consider themselves motivated to get a good grade, but that motivation doesn’t cause them to actually want to attend class, while others are passionately interested in the subject matter and attend every possible lecture in order to learn more. Ideally, the questions asked should try to get an overall value for all types of motivation that would cause a student to do well in a team.

Other Feedback

Each of the three classes had at least one team with difficulties. Professor Terence Heagney, who taught the Optimized teams, mentioned that the team that did poorly in his classroom consisted of three male students and one female student, and the female student reported difficulties working with her teammates (Heagney, personal communication, December 15, 2003). All three professors have reported problems with teams in which one female works with several males, in the past few semesters of teaching. While some published research attributes such problems to males failing to fully value the contributions or abilities of their female counterparts (Younglove-Webb et
al., 1999), Dr. Dunphy explains, "Sophomore males just aren't mature enough yet" to deal with a gender imbalance in their favor (personal communication, February 2, 2004).

**Lessons Learned**

This experiment served us in two ways - it revealed some difficulties with the survey questions and experimental design, which can be avoided or modified in future research, and it gave a measure of support for the idea that the method developed here can create compatibility scores that will predict to some degree how well a team will be able to achieve a high grade and learn teamwork skills. As a result of this experiment, we propose several alterations for future experimentation and application to real classrooms.

**Experimental design**

The appropriate design for any future experiments should be a very large classroom (300 or more students), which would be randomly split into three subsections, each assigned to a different method. In this way, we would eliminate differences in grades and satisfaction stemming from the classroom environment or the professor's grading policy, as well as preventing the professor from knowing which teams were formed by which methods, which could have added bias to grading in this experiment. These teams would, to a larger extent than in this experiment, be able to consist of the same number of team members, eliminating another possible influence on team satisfaction and compatibility scores. The students would also ideally be assigned exactly as the methods dictate. Randomly assigned teams should be truly random, not dependant on other preferences for team composition, and no switching of team members after optimization should be allowed.

**Team formation survey changes**

Future versions of the team formation survey should be somewhat different from the current version, and determined by the type of class being divided into teams. Obviously, what skills are being evaluated will change dependant on which skills are most important.
to the team work. Here, oral, written, and interpersonal communication skills were most important. In other classes, various math or science skills may be more important, for instance. Additionally, if possible, a teacher’s evaluation of the skill level should be used instead of or in addition to self-evaluation. If the class is small enough, the teacher should be able to evaluate at least some skill levels him- or herself, while in large classes grades already accumulated or current GPA could stand in as a good proxy for an intuitive evaluation.

Motivation should also appear differently than it does in the original version of the team formation survey. In terms of questions, a variety of motivation questions aimed at different types of motivations, for example “How motivated are you to get an A in this class?”, “How likely are you to drop this class?”, “How passionate are you about the class subject matter?”, “How much of your time are you willing to devote to this class?” may give a better idea of motivation in the sense that we are trying to diversify. We do not want a team full of people that don’t care about the subject matter and have little time to devote to it, even if they all want A’s.

Depending on the type of diversity generally found in the class, the white/non-white question may be less appropriate than the question “Are you a native speaker of English?” While many classrooms view diversity on a black/white racial level, MIT’s student body is more diverse in terms of national origin, so this aspect of diversity becomes potentially more important than racial diversity.

Formulation changes

Based on the feedback from the three participating professors, an optional constraint could be added to the heuristics, which does not accept teams in which exactly one female is on a team with multiple males. This idea will be addressed in more detail in the next section.
With the changes noted above, the Optimized method will continue to improve, better sorting students for increased diversity and balance while increasing its performance advantage over random.
VII. Conclusion and Avenues for Future Research

This thesis is a first attempt at designing a team-assignment heuristic that outperforms random assignment in large classrooms. In order to assess what has been accomplished and what opportunities exist for future research, we must consider the desirable characteristics of a program to assign teams in large classrooms, and how much progress has been made with this research towards satisfying those needs. The main features required are:

- Quick, better than random results
- Appropriate and easy-to-obtain survey questions
- Flexibility to meet individual educators’ needs
- Ease of use
- Robustness in a real classroom environment

Overall, this thesis has produced an initial method that satisfies all of these requirements. We have compiled the educational and group dynamics view of what characteristics make up a good learning team, synthesized these into a compatibility score that rates every dyad of students on the degree to which they will make good members of the same team, and created several heuristics to find a good set of teams by these standards. Simulations of the heuristics revealed that pairwise exchange on randomly generated sets of teams produced the best answers, and did so even after running for very short periods of time, beating the average randomly assigned teams by 307%. A survey was designed to obtain the appropriate information from students. A classroom experiment, applying this heuristic, proved to be somewhat flawed, but provided support that our measures of team compatibility – the average compatibility score and the minimum compatibility score – are significantly positively related to the team’s ability to learn teamwork skills, and that the heuristic-generated teams were overall more satisfied with their experience than a pseudo-random set of teams in another classroom. The code in the appendices can be quickly and simply implemented to assign students into teams in large classrooms, and is easily adapted to the professor’s preferences for variable weights and types of
questions asked. As such, this thesis has contributed a first version to meet the needs of educators who wish to assign teams intelligently.

**Future Research Areas**

This thesis has achieved its goal of setting up a new methodology for assigning teams that met the features described above. Future research in this area should focus on the five major features and investigating new ways to meet those needs by expanding the framework created here.

*Quick Results that Beat Random*

The most important feature of a team assignment program is that it work well, generating good team assignments very quickly, and the program created here does that. This research has clearly demonstrated that very small percentages of randomly assigned sets of teams come within 50% of the value of sets of teams assigned by the random-pairwise method, even when the method is run for very short periods of time. The classroom experiment has also shown that the student compatibility score is significantly positively related to team grades and students’ perception of learning about teamwork. However, since this implementation was a first attempt at attaching quantitative values to levels of diversity within learning teams, and forming teams that maximized these values, many decisions had to be made purely from research, without the results of detailed experimentation in a real classroom. Further research on several parts of this model will strengthen the evidence in favor of the methodology, and fine-tune the model where it is needed.

**Framing the Problem**

This thesis uses a group dynamics approach to computing the value of a team, focusing on the interactions between dyads of students on a team in order to provide the right levels of diversity on each team to positively influence team behavior. While this method succeeds in analyzing the effects of diversity and team size on some aspects of team
behavior, it may not tell the whole story. The coverage problem, in which teams are assigned in such a way as to maximize the number of teams that have at least one member with each of a list of desirable characteristics on a team, ignores the behavioral interactions, but does a better job of calculating the total resources each team has at its disposal. Since both methods deal with different pieces of the same puzzle, perhaps the methodology could be improved by implementing some hybrid, such as a system that uses one method to get a good initial solution and then modifies it with the other method, or a system that maximizes compatibility as calculated here, within the constraints formed by a coverage problem. The gender constraint program discussed in Section V is, in effect, this type of hybrid, except that it only constrains one variable. Experimentation with adding more constraints will better demonstrate the feasibility of such an approach.

Variable Weights
The variables used in the development of these initial heuristics were chosen because other experiments on team behavior had shown them to be likely to have an effect on team performance. The weights attributed to them came from a synthesis of the research and our estimate of the certainty of the consensus about each particular type of variable. However, the framework for the calculation of the compatibility score really provides a starting point for further research on the variables involved.

The weights used in this version of the compatibility score were derived from consensus on the effects of various types of diversity on team behavior, not as the result of experimentation. Further analysis of the classroom experiment, in which scores were recalculated using different weights, and additional regressions performed to determine if the new weighted compatibility scores have a stronger or weaker relationship to satisfaction, learning or grades, would help to validate or correct the initial weights. However, since this experiment was small and flawed in some ways, experimenting with the weights in a single, large classroom would be much more likely to result in better calibration of the weights. Additionally, the weights for variables in alternative group
learning environments, such as distance learning classrooms, or brainstorming teams in local classrooms, have not even been validated with a single experiment.

Another issue with variable weights is whether variables in the same category, such as demographics, all deserve equal weights. Experimentation may find that different skill, demographic, motivation, and personality variables each need different weights; for this attempt, the paucity of actual data prevented us from attempting such an assignment.

**Appropriate and Easy to Obtain Survey Questions**

Some of the variables used in this experiment were more successful than others. Appropriate variables are those that accurately measure some relevant characteristic, separate students into distinct groups, are relatively easy to obtain quickly through student surveys, and do not offend students. The variables chosen for the initial experiment met all of these qualifications except that the skill level and motivation questions did not separate students into distinct groups in the classroom experiment. While this was partly a result of a homogeneous classroom, the Intuition method demonstrated that there are some variables in skill and motivation that can more accurately define students, such as grades in this class prior to team assignment. If such grades are not available at the time teams are assigned, the grades for previous classes that are prerequisites to the class, or the student’s average GPA, may be good proxies (Felder & Brent, 1994). Depending on the class size, a professor or TA may even be able to give the student an overall rating on performance, if grades are not available but some class participation has allowed them to form opinions. These are all proxies for the more complete knowledge that a teacher in a small classroom develops and uses when assigning teams; although they can never be expected to outperform intuition in smaller classrooms, future experimentation may show that some or all of these are better predictors of skill than a self-evaluation on the part of students.

Motivation and demographics questions could also use further examination. As discussed in Section VI, more experimentation on the type of motivation question(s)
asked will help develop a better method of determining which students are most motivated, and discovering which types of motivation are most important to team satisfaction and success as well. Finally, the ethnicity question may need to be modified – the alternate question “Are you a native English speaker” may be more important in some classrooms than “Are you white or non-white?” depending on the type of ethnic/racial diversity common to the classroom environment.

In experimenting with survey questions in all categories, it is important to be mindful of asking questions that will not offend students or make them hesitant to answer. Getting no response from many students on a question will nullify its effectiveness, as well as creating a hostile audience for the educator. The students in our classroom experiment did not leave any questions blank, nor did they complain about the appropriateness of any of the questions – this should be an example for future attempts to modify the survey.

**Flexibility**

Any successful product must have flexibility to adjust to users’ differing goals. The code produced in this thesis is easily adjustable in terms of the number, type, and weights of the variables. However, professors may have additional requirements not addressed in this thesis, and if the teams created by these heuristics regularly violate them, this product will be of no use to them. One example of such a potential requirement is the prohibition of teams containing one and only one female, which the three professors involved in the classroom experiment all found desirable. Although a preliminary simulation in Section V found the addition of such a constraint feasible, a larger simulation with different variants of the gender constraint algorithm is necessary to confirm this.

Aside from the gender constraint, there may be other types of constraints similar to those found in coverage problems, such as “At least one student in the top 20% of the class must be in each team” that could be added to the existing model in a manner similar to the method for incorporating the gender constraint in Section V. If many of these constraints are highly desirable, then research should focus more on the coverage
problem approach, as discussed above. However, if a professor likes the current model but has one additional constraint (on any variable), a good team-assignment program should be able to incorporate that constraint. More research needs to be done on what additional needs professors may have for teams and how to best incorporate them into this framework or alter the framework to better serve these needs.

Ease of Use

The current implementation of the method described in this thesis is as a computer program; while it is fairly straightforward and easily adapted, it still provides a barrier for professors who do not have the time to make any small programming changes and run the program. Later versions could be adapted to be easier to use for the non-programmer by the inclusion of a GUI that asks simple questions such as

- How many students?
- How many students per team?
- Which variables should be homogeneous?
- What weight should be given to each variable?

And uses this information to create the teams and output them in a user-friendly format. If future research into the addition of constraints shows promise, there could also be a row of additional constraints that the user could click on to add. The modified program should continue to work as quickly as before, so that a user would have time to try several such simulations to decide which type of constraints most suit the team goals.

Robustness

Finally, the issue of robustness should be addressed in future research. While the average compatibility score is a very robust measure, the minimum compatibility score is not robust, as we saw when a student dropped out and teams had to be rearranged in the classroom experiment. In Section V we found that, if both goals – keeping compatibilities high overall without creating one team with an especially bad compatibility – were equally important, using the minimum compatibility score hurt the average less than using the average hurt the minimum. More examination should be
dedicated to finding an objective function that achieves both goals while maintaining
greater robustness than the minimum compatibility score does.

Our current model of team assignment takes a big step toward fulfilling the needs of
educators for a program that quickly assigns teams based on individual student
characteristics. As we learn more about incorporating alternative frameworks, additional
constraints, or different survey questions into the model, a workable end-product could be
put into use at the numerous colleges and universities endeavoring to improve their
traditional, lecture-based large classes and their new distance education programs.
Bibliography


Appendix I: C Code

A. Random and Pairwise code – the following C code carries out the random-pairwise heuristic found to be the most efficient of the heuristics tried here.

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

typedef struct {
    char util1[1];
    char pad1[1];
    char util2[1];
    char pad2[1];
    char util3[1];
    char pad3[1];
    char util4[1];
    char pad4[1];
    char util5[1];
    char pad5[1];
    char util6[1];
    char pad6[1];
    char util7[1];
    char pad7[1];
    char util8[1];
    char pad8[1];
    char util9[1];
    char pad9[1];
    char util10[1];
    char pad10[1];
    char util11[1];
    char nl_null[2]; /* newline and '0' */
} util_matrix;

#define boolean int
#define TRUE 1
#define FALSE 0

#define N_VALUE 99 /* Number of students in class */
#define K_VALUE 3 /* Number of students per team */
#define totteam 33 /* Total number of teams */
#define ROWSIZE 25
float maxmin(int k, float a[], float b[][N_VALUE], int r);
float combos[N_VALUE+totteam];
float testcom[N_VALUE+totteam];
int availstud[N_VALUE]; /* 1 if student avail, 0 if not. */
int allavail[N_VALUE-K_VALUE];
float studmat[N_VALUE][11];
int i,d,j;
int b=0; /* where to begin writing a team sequence */
float k,dem,p,v; /*The weights for the variables in the compat. eqn. */
int teamgo=0; /* The team that contains the lowest-compatible pair */
int h=0;
int t=0;
float tempval[K_VALUE];
float min;
float setmin=5; /* minimum min. in set of teams. */
float ultmin=0; /* minimum compatibility of current best within a set of exchanges */
float optimal[N_VALUE]; /* optimal is where best teams are stored within pair comparison */
float utility [N_VALUE] [N_VALUE];
float sums[totteam];
float sum=0;
float ultsum=0; /* total compatibility of current best w/in pair exchanges. */
float ultsum2=0; /* we will maximize this as secondary goal. */
float ultmin2=0; /* minimum compatibility w/in best set of teams - maximize this */
int m=0;
int f=0;
int type=2; /*type of group environment. 2=local, long-term, high-skill, vague-task */
int alike=0; /*alike = 1 if the solution after listing pairwise exchanges is unchanged. */
int count=0;
int array1[N_VALUE];
int final[N_VALUE];
int maxroll;
int roller;
float optimal2[N_VALUE]; /*this is where overall optimals are stored!*/
int g;
int r;
int temp1;
int temp2;
int arraynew[2][N_VALUE];
int badpair[2*totteam];
int makepair;
int main() {

/*read in survey questions from csv file */
  util_matrix um;
  FILE* matrix_file;
char util_1[1+1];
char util_2[1+1];
char util_3[1+1];
char util_4[1+1];
char util_5[1+1];
char util_6[1+1];
char util_7[1+1];
char util_8[1+1];
char util_9[1+1];
char util_10[1+1];
char util_11[1+1];
float ut1, ut2, ut3, ut4, ut5, ut6, ut7, ut8, ut9, ut10, ut11;

memset( (char*)(&um), '0', ROWSIZE);
memset( util_1, '0', 2);
memset( util_2, '0', 2);
memset( util_3, '0', 2);
memset( util_4, '0', 2);
memset( util_5, '0', 2);
memset( util_6, '0', 2);
memset( util_7, '0', 2);
memset( util_8, '0', 2);
memset( util_9, '0', 2);
memset( util_10, '0', 2);
memset( util_11, '0', 2);

matrix_file=fopen("test995.csv","r");
if( matrix_file == NULL ) {
    printf("File does not exist. \n");
    exit(1);
}

while( fgets((char*)(&um), ROWSIZE, matrix_file) ) {
    h++;
    memcpy(util_1, um.util1, 1);
    memcpy(util_2, um.util2, 1);
    memcpy(util_3, um.util3, 1);
    memcpy(util_4, um.util4, 1);
    memcpy(util_5, um.util5, 1);
    memcpy(util_6, um.util6, 1);
    memcpy(util_7, um.util7, 1);
    memcpy(util_8, um.util8, 1);
    memcpy(util_9, um.util9, 1);
    memcpy(util_10, um.util10, 1);
    memcpy(util_11, um.util11, 1);
    ut1 = atof( util_1);
ut2 = atof( util_2);
ut3 = atof( util_3);
ut4 = atof( util_4);
ut5 = atof( util_5);
ut6 = atof( util_6);
ut7 = atof( util_7);
ut8 = atof( util_8);
ut9 = atof( util_9);
ut10 = atof( util_10);
ut11 = atof( util_11);

studmat[h-1][0]=ut1;
studmat[h-1][1]=ut2;
studmat[h-1][2]=ut3;
studmat[h-1][3]=ut4;
studmat[h-1][4]=ut5;
studmat[h-1][5]=ut6;
studmat[h-1][6]=ut7;
studmat[h-1][7]=ut8;
studmat[h-1][8]=ut9;
studmat[h-1][9]=ut10;
studmat[h-1][10]=ut11;
}
fclose(matrix_file);

if( ttype == 1 ) { /*city block distance */
for( i = 0; i < N_VALUE; i++ ) {
  for( j = 0; j < N_VALUE; j++ ) {
    utility[i][j]=0;
    for( h = 0; h<11; h++ ) {
      utility[i][j] += abs(studmat[i][h] - studmat[j][h]);
    }
  }
}

else if( ttype == 2 ) {
for( i = 0; i < N_VALUE; i++ ) {
  for( j = 0; j < N_VALUE; j++ ) {
    k = (1.0/6.0) *
      ( abs(studmat[i][4] - studmat[j][4]) +
      abs(studmat[i][5] - studmat[j][5]) +
      abs(studmat[i][6] - studmat[j][6]) );
    dem = (1.0/3.0) *
      ( abs(studmat[i][0] - studmat[j][0]) +

abs(studmat[i][1] - studmat[j][1]) +
.25*(abs(studmat[i][2] - studmat[j][2]));

p = (.25) *
( abs(studmat[i][7] - studmat[j][7]) +
abs(studmat[i][8] - studmat[j][8]) +
abs(studmat[i][9] - studmat[j][9]) +
abs(studmat[i][10] - studmat[j][10]) - 4 );

v = .5 * ( abs(studmat[i][3] - studmat[j][3]) );

utility[i][j] = (.5* k + .15* dem + .15* p + .2* v);
}
}

/*ensures that the compatibility between person and himself is 0. */
for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
        if( i == j ) utility[i][j] = 0;
    }
}

srand((unsigned int) time(0));  /*randomize seed*/

for(f=0; f<50; f++) {  /*loops through each random original combo*/

    /*set initial array*/
    for(i=0; i<N_VALUE; i++){
        arraynew[1][i]=i+1;
    }

    /*The following code generates N random numbers and uses these numbers to sort the N students into a random order. */

    maxroll=100; /*large number - will find a random number between 0 and maxroll*/
    for(i=0; i<N_VALUE; i++){
        arraynew[2][i]=genrand(maxroll);
    }

    for(i=0; i<N_VALUE; i++){
        for(h=i+1; h<N_VALUE; h++){  
            if(arraynew[2][h]>arraynew[2][i]){
                temp1=arraynew[2][h];
                temp2=arraynew[1][h];
                arraynew[2][h]=arraynew[2][i];
                arraynew[1][h]=arraynew[1][i];
            }
        }
    }

}
arraynew[2][i]=temp1;
arraynew[1][i]=temp2;
}
}

/* put new random start point into optimal */
for(i=0; i<N_VALUE; i++){
optimal[i]=arraynew[1][i];
}

for (i=0; i<totteam; i++){
combos[i]=0;
}
alike=0;
ultmin=0;
ultsum=0;
count=0;

/* this loop continues pairwise exchange until no feasible exchanges are possible for this randomly generated set of teams. */

while(alike==0){
    /*Put optimal into combos*/
    for(i=0; i<N_VALUE; i++){
       combos[i+totteam]=optimal[i];
    }

    /*find min for each team, put in first totteam slots of combos*/
    for(t=1; t<=totteam; t++){
        b=totteam+K_VALUE*(t-1);
        makepair=1;
        min = maxmin(K_VALUE, combos, utility, b);
        combos[t-1]=min;
    }

    setmin=5;
    for(i=0; i<totteam; i++){
        if(combos[i]<setmin) setmin=combos[i];
    }

    if(count==0) ultmin=setmin;
    sum=0;
    for(h=0; h<totteam; h++){
        sum += sums[h];
    }
if(count==0)ultsum=sum;

/*what team has setmin?*/
for(i=0; i<totteam; i++){  
    if(combos[i]==setmin){  
        teamgo=i+1;
    }
}

/*Place all students not in lowest team in allavail.*/
j=0;
for(i=0; i<N_VALUE; i++) availstud[i]=1; /*default: all avail.*/
for(i=0; i<K_VALUE; i++){
    availstud[K_VALUE*(teamgo-1)+i]=0;
}
for(i=0; i<N_VALUE; i++){
    if(availstud[i]==1){
        allavail[j]=combos[i+totteam];
        j++;
    }
}

/*Cycle through all switches involving the worst team*/
for(i=1; i<3; i++){
    for(j=0; j<N_VALUE-K_VALUE; j++){
        for(h=0; h<N_VALUE+totteam; h++){
            testcom[h]=combos[h];
        }
        b=totteam+(teamgo-1)*K_VALUE+badpair[2*teamgo-i];
        d=testcom[b];
        testcom[b]=allavail[j];
        for(h=totteam; h<N_VALUE+totteam; h++){
            if(h != b && testcom[h]==allavail[j]) testcom[h]=d;
        }
        for(t=1; t<=totteam; t++){
            d=totteam+K_VALUE*(t-1);
            makepair=0;
            min=maxmin(K_VALUE, testcom, utility, d);
            testcom[t-1]=min;
        }
    }
    setmin=5;
    for(h=0; h<totteam; h++){
        if(testcom[h]<setmin) setmin=testcom[h];
    }
}
sum=0;
for(h=0; h<totteam; h++){
    sum += sums[h];
}

if(setmin>ultmin){
    ultmin=setmin;
    ultsum=sum;
    for(h=0; h<N_VALUE; h++){  
        optimal[h]=testcom[totteam+h];
    }
}
else if(setmin==ultmin && sum>ultsum){
    ultsum=sum;
    for(h=0; h<N_VALUE; h++){  
        optimal[h]=testcom[totteam+h];
    }
}
}

/*@check to see if optimal is different from combos!*/

d=0;
for(i=0; i<N_VALUE; i++){  
    if(optimal[i] != combos[totteam+i]) d=d+1;
}
if(d==0) alike=1;
count++;

/*@end while loop*/

/*@If the result of the pw exchange beats the optimal found from  
 previous random sets, replace the old one. */
if(ultmin >ultmin2){
    ultmin2=ultmin;
    ultsum2=ultsum;
    for(h=0; h<N_VALUE; h++){  
        optimal2[h]=optimal[h];
    }
}
else if(ultmin==ultmin2 && ultsum >ultsum2){
    ultsum2=ultsum;
    for(h=0; h<N_VALUE; h++){  
        optimal2[h]=optimal[h];
    }
}


} 

} 

} /* end iteration of f loop */

printf("optimal2\n");
for(h=0; h<N_VALUE; h++){
    printf(" \%0f", optimal2[h]);
}
printf("\n");
printf("ultmin2= \%0f\n", ultmin2);
printf("ultsum2= \%0f\n", ultsum2);
return 0;
} /*end main */

float maxmin(int k, float a[], float b[][N_VALUE], int r) {
    int u, v, w, x, y, z, i;
    float fmin;
    float tempval[10];
    int worst;
    int index[3];

    if( k == 3 ) u=3;
    else if(k ==4) u=6;
    else if(k==5) u=10;

    for(i=0; i<k; i++){
        index[i]=a[r]-1;
        r++;
    }
    r=r-u;
    x=0;
    y=1;
    for(i=0; i<u; i++){
        tempval[i]=b[index[x]][index[y]];
        y++;
        if(y>(k-1)){
            x++;
            y=x+1;
        }
    }
}

fmin = 5;
for( i = 0; i < u; i++ ) {
    if( tempval[i] < fmin ){
fmin = tempval[i];
worst = i;
}
}

/* Find the worst pair of students in the team */

if(makepair==1){
x = 0;
y = 1;
for(i = 0; i < u; i++) {
  if(worst == i){
    badpair[2*t-2] = x;
    badpair[2*t-1] = y;
  }
  y++;
  if(y>(k-1)) {
    x++;
    y = x + 1;
  }
}
}
sums[t-1] = 0;
for( i = 0; i < u; i++) {
  sums[t-1] += tempval[i];
}
return fmin;
}

/* generates the random numbers */
int genrand(int maxroll){
  int roll;
  roll = rand() % maxroll + 1;
  return roll;
}

B. Exhaustive enumeration and pairwise exchange – this code contains all the code necessary to perform exhaustive enumeration alone, pairwise exchange alone, or exhaustive enumeration followed by pairwise exchange.

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
typedef struct {
    char util1[1];
    char pad1[1];
    char util2[1];
    char pad2[1];
    char util3[1];
    char pad3[1];
    char util4[1];
    char pad4[1];
    char util5[1];
    char pad5[1];
    char util6[1];
    char pad6[1];
    char util7[1];
    char pad7[1];
    char util8[1];
    char pad8[1];
    char util9[1];
    char pad9[1];
    char util10[1];
    char pad10[1];
    char util11[1];
    char nl_null[2]: /*newline and \0 */
} util_matrix;

#define boolean int
#define TRUE 1
#define FALSE 0

#define N_VALUE 99 /* Number of students in class */
#define K_VALUE 3 /* Number of students per team */
#define totteam 33 /* Total number of teams */
#define ROWSIZE 25

boolean enumeratecombos( int n, int k, int j[] );
float maxmin(int k, float a[], float b[][N_VALUE], int r);
void translate(int a[], int b, int t);
void maketeam();
int combosct=0;
float combos[N_VALUE+totteam];
float testcom[N_VALUE+totteam];
int availstud[N_VALUE]; /* 1 if student avail, 0 if not. */
int allavail[N_VALUE-K_VALUE];
int allcomb[N_VALUE];
float studmat[N_VALUE][11];
int i,d,j;
int b=0; /* where to begin writing a team sequence */
float k,dem,p,v;
int comb[N_VALUE] = { -1 };
int count = 0;
int teamgo=0; /* The team that needs to be worked on */
int lastct=0; /* counts the number of times the last team is called */
int loop=0; /* this var. breaks the first loop, starts pairs */
int h=0;
int t=0;
float tempval[K_VALUE];
int last[K_VALUE]; /* contains the last team */
float min;
float setmin=5; /* minimum min. in set of teams. */
float ultmin=0; /* minimum utility w/in best set of teams - we want to maximize this! */
float optimal[N_VALUE]; /* where best teams are stored */
float utility[N_VALUE][N_VALUE];
float sums[totteam];
float sum=0;
float ultsum=0; /* will maximize this as secondary goal */
int ttype=2;
int alike=0;

int main() {
/* Reads in survey results from a csv file */
    util_matrix um;
    FILE* matrix_file;
    char util_1[1+1];
    char util_2[1+1];
    char util_3[1+1];
    char util_4[1+1];
    char util_5[1+1];
    char util_6[1+1];
    char util_7[1+1];
    char util_8[1+1];
    char util_9[1+1];
    char util_10[1+1];
    char util_11[1+1];
    float ut1, ut2, ut3, ut4, ut5, ut6, ut7, ut8, ut9, ut10, ut11;

    memset((char*)(&um), '0', ROWSIZE);
    memset( util_1, '0', 2);
    memset( util_2, '0', 2);
    memset( util_3, '0', 2);
    memset( util_4, '0', 2);
    memset( util_5, '0', 2);
memset( util_6, '0', 2);
memset( util_7, '0', 2);
memset( util_8, '0', 2);
memset( util_9, '0', 2);
memset( util_10, '0', 2);
memset( util_11, '0', 2);

matrix_file=fopen("test995.csv","r");
if( matrix_file == NULL ) {
    printf("File does not exist. \n");
    exit(1);
}

while( fgets((char*)(&um), ROWSIZE, matrix_file) ) {
    h++;
    memcpy(util_1, um.util1, 1);
    memcpy(util_2, um.util2, 1);
    memcpy(util_3, um.util3, 1);
    memcpy(util_4, um.util4, 1);
    memcpy(util_5, um.util5, 1);
    memcpy(util_6, um.util6, 1);
    memcpy(util_7, um.util7, 1);
    memcpy(util_8, um.util8, 1);
    memcpy(util_9, um.util9, 1);
    memcpy(util_10, um.util10, 1);
    memcpy(util_11, um.util11, 1);
    ut1 = atof( util_1);
    ut2 = atof( util_2);
    ut3 = atof( util_3);
    ut4 = atof( util_4);
    ut5 = atof( util_5);
    ut6 = atof( util_6);
    ut7 = atof( util_7);
    ut8 = atof( util_8);
    ut9 = atof( util_9);
    ut10 = atof( util_10);
    ut11 = atof( util_11);

    studmat[h-1][0]=ut1;
    studmat[h-1][1]=ut2;
    studmat[h-1][2]=ut3;
    studmat[h-1][3]=ut4;
    studmat[h-1][4]=ut5;
    studmat[h-1][5]=ut6;
    studmat[h-1][6]=ut7;
    studmat[h-1][7]=ut8;
studmat[h-1][8]=ut9;
studmat[h-1][9]=ut10;
studmat[h-1][10]=ut11;
}
fclose(matrix_file);

/*Fills in compatibility matrix from survey questions */

if( ttype == 1 ) { /*city block distance */
  for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
      utility[i][j]=0;
      for( h = 0; h<11; h++ ) {
        utility[i][j] += abs(studmat[i][h] - studmat[j][h]);
      }
    }
  }
}

else if( ttype == 2 ) { /*15.279 */
  for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
      k = (1.0/6.0) *
      ( abs(studmat[i][4] - studmat[j][4]) +
      abs(studmat[i][5] - studmat[j][5]) +
      abs(studmat[i][6] - studmat[j][6]) );

      dem = (1.0/3.0) *
      ( abs(studmat[i][0] - studmat[j][0]) +
      abs(studmat[i][1] - studmat[j][1]) +
      .25*(abs(studmat[i][2] - studmat[j][2])) );

      p = (-.25) *
      ( abs(studmat[i][7] - studmat[j][7]) +
      abs(studmat[i][8] - studmat[j][8]) +
      abs(studmat[i][9] - studmat[j][9]) +
      abs(studmat[i][10] - studmat[j][10]) - 4 );

      v = .5 * ( abs(studmat[i][3] - studmat[j][3]) );

      utility[i][j] = (.5* k + .15* dem + .15* p + .2* v);
    }
  }
}

for( i = 0; i < N_VALUE; i++ ) {
for (j = 0; j < N_VALUE; j++) {
    if (i == j) utility[i][j] = 0;
}

} 

t = 1; /*Where t=the team that is currently being worked on. */

for (i = 0; i < N_VALUE; i++) {
    optimal[i] = 0;
}
maketeam();

printf("optimal teams are \n");
printf("ultmin = %.4f \n", ultmin);
printf("ultsum = %.4f \n", ultsum);
for (i = 0; i < N_VALUE; i++) {
    printf(" %.0f", optimal[i]);
}
printf("\n");
fflush(stdout);

count = 0;
/*alike =1 */ /*uncomment this to run ONLY enumerate combos, without pw exchange */
while (alike == 0) {

    /*Put optimal into combos*/
    for (i = 0; i < N_VALUE; i++) {
        combos[i + totteam] = optimal[i];
    }

    /*find min for each team, put in first totteam slots of combos*/

    for (t = 1; t <= totteam; t++) {
        b = totteam + K_VALUE * (t - 1);
        min = maxmin(K_VALUE, combos, utility, b);
        combos[t - 1] = min;
    }

    setmin = 5;
    for (i = 0; i < totteam; i++) {
        if (combos[i] < setmin) setmin = combos[i];
    }

    if (count == 0) ultmin = setmin;

    /*what team has setmin?*/
    for (i = 0; i < totteam; i++) {

    

}
if(combos[i]==setmin){
    teamgo=i+1;
}

/*Place all students not in lowest team in avail.*/
j=0;
for(i=0; i<N_VALUE; i++) availstud[i]=1; /*default: all avail.*/
for(i=0; i<K_VALUE; i++){
    availstud[K_VALUE*(teamgo-1)+i]=0;
}

for(i=0; i<N_VALUE; i++){
    if(availstud[i]==1){
        allavail[j]=combos[i+totteam];
        j++;
    }
}

/*Cycle through all switches involving the worst team*/
b=totteam+K_VALUE*(teamgo-1);

for(i=0; i<K_VALUE; i++){
    for(j=0; j<N_VALUE-K_VALUE; j++){
        for(h=0; h<N_VALUE+totteam; h++){
            testcom[h]=combos[h];
        }
        d=testcom[b+i];
        testcom[b+i]=allavail[j];
        for(h=totteam; h<N_VALUE+totteam; h++){
            if(h != b+i & testcom[h]==allavail[j]) testcom[h]=d;
        }
        for(t=1; t<=totteam; t++){
            d=totteam+K_VALUE*(t-1);
            min=maxmin(K_VALUE, testcom, utility, d);
            testcom[t-1]=min;
        }
        setmin=5;
        for(h=0; h<totteam; h++){
            if(testcom[h]<setmin) setmin=testcom[h];
        }
        sum=0;
        for(h=0; h<totteam; h++){
            sum += sums[h];
        }
        if(setmin>ultmin){

ultmin=setmin;
ultsum=sum;
for(h=0; h<N_VALUE; h++){
optimal[h]=testcom[totteam+h];
}
else if(setmin==ultmin & & sum>ultsum){
ultsum=sum;
for(h=0; h<N_VALUE; h++){
optimal[h]=testcom[totteam+h];
}
}
}

/*check to see if optimal is different from combos!*/
d=0;
for(i=0; i<N_VALUE; i++){  
if(optimal[i] != combos[totteam+i]) d=d+1;
}
if(d==0) alike=1;
count++;
/*end while loop*/

printf("number of switches= %i \n",count);
for(i=0; i<N_VALUE; i++){  
printf(" %.0f",optimal[i]);
}
printf("\n");
printf("ultmin= %.4f\n",ultmin);
printf("ultsum= %.4f\n",ultsum);
return 0;
} /*end main */

boolean enumeratecombs( int n, int k, int j[] ) {  
int i;
int m;
if( j[0] < 0 ) {  
for( i = 0; i < k; i++ ) {  
j[i] = i;
}
return TRUE;
}
else {  
for( i = k-1; i >= 0 & & j[i] >= n - k + i; i-- ) {}
if( i >= 0 ) {
    j[i]++;
    for( m = i + 1; m < k; m++ ) {
        j[m] = j[m - 1] + 1;
    }
    return TRUE;
}
else return FALSE;
}

float maxmin(int k, float a[], float b[][N_VALUE], int r) {
int u, v, w, x, y, z, i;
float fmin;
float tempval[10];
int index[3];

if( k == 3 ) u=3;
else if(k ==4) u=6;
else if(k==5) u=10;

for(i=0; i<k; i++){
    index[i]=a[r]-1;
    r++;
}
    r=r-u;
x=0;
y=1;
for(i=0; i<u; i++){
    tempval[i]=b[index[x]][index[y]];
y++;
    if(y>(k-1)){
        x++;
        y=x+1;
    }
}

fmin = 5;
for( i = 0; i < u; i++ ) {
    if( tempval[i] < fmin ){
        fmin = tempval[i];
        worst=i;
    }
}

sums[t-1] = 0;
for( i = 0; i < u; i++ ) {
    sums[t-1] += tempval[i];
}
return fmin;
}

void translate( int a[], int b, int t ) {
    int f=0;
    int g=0;
    for( f = 0; f < N_VALUE; f++ ) availstud[f]=1;

    for( f = 0; f < N_VALUE; f++ ) { /*fill in which studs avail */
        for( g = totteam; g < b; g++ ) {
            if( availstud[f] != 0 ) {
                if( combos[g] - 1 == f ) availstud[f] = 0;
                else availstud[f] = 1;
            }
        }
    }

    if( t != totteam ) {
        g = 0; /* put avail students into new matrix */
        for( f = 0; f < N_VALUE; f++ ) {
            if( availstud[f] == 1 ) {
                allavail[g] = (f + 1);
                g++;
            }
        }
        for( f = 0; f < K_VALUE; f++ ) {
            for( g = 0; g < N_VALUE - K_VALUE * (t-1); g++ ) {
                if( a[f] == g ) combos[b+f] = allavail[g];
            }
        }
    }

else {
    g = 0;
    for( f = 0; f < N_VALUE; f++ ) {
        if( availstud[f] == 1 ) {
            combos[b+g] = f+1;
            g++;
        }
    }
}
}
void maketeam() {
    while( enumeratecombos(N_VALUE-K_VALUE*(t-1), K_VALUE, comb) ) {
        loop++;
        if( loop > 8912730) break; /*Comment this out to run only Enumerate. To run only pairwise comparison, change the number to 0. */

        if( lastct == 1 ) {
            for( i = totteam-1; i >= 0; i-- ) {
                if( combos[i] == setmin ) teamgo = i+1;
            }
        }

        if( lastct==1 && t > teamgo && combos[totteam-1] != setmin ) {
            d=K_VALUE;
            for( i = 0; i < K_VALUE; i++ ) {
                comb[i] = N_VALUE - K_VALUE * (t-1) - d;
                d--;
            }
        }
    }

    else {
        if( lastct == 1 ) lastct = 0;
        b = totteam + K_VALUE * (t-1);
        if( t == 1 ) {
            count++;
            for( i = 0; i <= K_VALUE - 1; i++ ) {
                combos[b+i] = comb[i]+1;
            }
        }
        for( i = 0; i < K_VALUE; i++ ) {
            allcomb[(t-1) * K_VALUE + i] = comb[i];
        }

        min = maxmin( K_VALUE, combos, utility, b );
        combos[t-1] = min;
        setmin = min;
    }

    else {
        if( t == 2 ) setmin=combos[0];
        else {
            setmin = 5;
            for( i = 0; i < (t-1); i++ ) {
                if( combos[i] < setmin ) setmin = combos[i];
            }
        }
    }
}
translate( comb, b, t );
for( i = 0; i < K_VALUE; i++ ) {
    allcomb[(t-1) * K_VALUE + i] = comb[i];
}
min = maxmin( K_VALUE, combos, utility, b );
combos[t-1] = min;
if( min < setmin ) setmin = min;
}

if( min > ultmin ) {
    if( t < (totteam - 1) ) {
        t++;
        comb[0] = 1; /* resets for first loop of next while */
        maketeam( b, t );
    } else { /* last team */
        combosct++; lastct++;
        t++;
        b = totteam + K_VALUE * (t-1);
        translate( comb, b, t );
        min = maxmin( K_VALUE, combos, utility, b );
        combos[t-1] = min;
        sum = 0;
        for( i = 0; i < totteam; i++ ) {
            sum += sums[i];
        }
    }
    /* setmin is lowest of all teams min */
    if( min < setmin ) setmin = min;
}

if( setmin > ultmin ) {
    ultmin = setmin;
    ultsum = sum;

    printf("%4.4f, %4.4f, %i, %i\n", ultmin, ultsum, loop, combosct);
    fflush( stdout );

    for( i = totteam; i < totteam + K_VALUE * t; i++ ) {
        optimal[i - totteam] = combos[i];
    }
}
else if( setmin == ultmin && sum > ultsum ) {
    ultsum = sum;

    fflush( stdout );
}
for( i = totteam; i < totteam + K_VALUE * t; i++ ) {
    optimal[i-totteam]=combos[i];
}

} /* for(i=0; i<N_VALUE; i++){
    printf(" %0f", optimal[i]);
    printf("\n"); */

} /* end of if min > ultmin */
} /* end of else for lastct=1 */
} /* end while loop */

for( i = 0; i < K_VALUE; i++ ) {
    comb[i] = allcomb[K_VALUE * (t-2) + i];
}

t--; /* end function definition */

C. Random-Only Code – this C code carries out the algorithm described in Section IV for generating many sets of random teams. This code also keeps a running average of the minimum compatibility and puts the minimums into buckets to approximate a distribution of min. compatibility scores.

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

typedef struct {
    char util1[1];
    char pad1[1];
    char util2[1];
    char pad2[1];
    char util3[1];
    char pad3[1];
    char util4[1];
    char pad4[1];
    char util5[1];
    char pad5[1];
    char util6[1];
    char pad6[1];
    char util7[1];
}
char pad7[1];
char util8[1];
char pad8[1];
char util9[1];
char pad9[1];
char util10[1];
char pad10[1];
char util11[1];
char nil_null[2]; /*newline and '0' */
} util_matrix;

#define boolean int
#define TRUE 1
#define FALSE 0

#define N_VALUE 36
#define K_VALUE 3
#define totteam 12
#define ROWSIZE 25

float maxmin(int k, float a[], float b[][N_VALUE], int r);
float combos[N_VALUE+totteam];
float testcom[N_VALUE+totteam];
int availstud[N_VALUE]; /* 1 if student avail, 0 if not. */
int allavail[N_VALUE-K_VALUE];
float studmat[N_VALUE][11];
int i;
int d;
int b=0; /* where to begin writing a team sequence */
int j;
float k,dem,p,v;
int teamgo=0; /* The team that needs to be worked on */
int h=0;
int t=0;
float tempval[K_VALUE];
float min;
float setmin=5; /* minimum min. in set of teams. */
float ultmin=0; /* minimum utility w/in best set of teams - we want to maximize this! */
float optimal[N_VALUE]; /* optimal is where best teams are stored within pair comparison */
float utility[N_VALUE][N_VALUE];
float sums[totteam];
float sum=0;
float ultsum=0; /* will maximize this as secondary goal */
# C Program

```c
int m=0;
int f=0;
int ttype=2;
int alike=0;
int count=0;
int array1[N_VALUE];
int final[N_VALUE];
int maxroll;
int roller;
int g;
int r;
int temp1;
int temp2;
int arraynew[2][N_VALUE];
int b1=0; /* these are the buckets we will use to approx. the distribution */
int b2=0;
int b3=0;
int b4=0;
int b5=0;
int b6=0;
int b7=0;
int b8=0;
int b9=0;
int b10=0;
float ave=0; /* This is an average of the min. for each random set. */

int main() {
    util_matrix um;
    FILE* matrix_file;
    char util_1[1+1];
    char util_2[1+1];
    char util_3[1+1];
    char util_4[1+1];
    char util_5[1+1];
    char util_6[1+1];
    char util_7[1+1];
    char util_8[1+1];
    char util_9[1+1];
    char util_10[1+1];
    char util_11[1+1];
    float ut1, ut2, ut3, ut4, ut5, ut6, ut7, ut8, ut9, ut10, ut11;

    memset( (char*)(&um), '0', ROWSIZE);
    memset( util_1, '0', 2);
    memset( util_2, '0', 2);
    memset( util_3, '0', 2);
    ```
memset( util_4, '0', 2);
memset( util_5, '0', 2);
memset( util_6, '0', 2);
memset( util_7, '0', 2);
memset( util_8, '0', 2);
memset( util_9, '0', 2);
memset( util_10, '0', 2);
memset( util_11, '0', 2);

matrix_file=fopen("test536.csv","r");
if( matrix_file == NULL ) {
    printf("File does not exist. \n");
    exit(1);
}

while( fgets((char*)(&um), ROWSIZE, matrix_file) ) {
    h++;
    memcpy(util_1, um.util1, 1);
    memcpy(util_2, um.util2, 1);
    memcpy(util_3, um.util3, 1);
    memcpy(util_4, um.util4, 1);
    memcpy(util_5, um.util5, 1);
    memcpy(util_6, um.util6, 1);
    memcpy(util_7, um.util7, 1);
    memcpy(util_8, um.util8, 1);
    memcpy(util_9, um.util9, 1);
    memcpy(util_10, um.util10, 1);
    memcpy(util_11, um.util11, 1);
    ut1 = atof( util_1);
    ut2 = atof( util_2);
    ut3 = atof( util_3);
    ut4 = atof( util_4);
    ut5 = atof( util_5);
    ut6 = atof( util_6);
    ut7 = atof( util_7);
    ut8 = atof( util_8);
    ut9 = atof( util_9);
    ut10 = atof( util_10);
    ut11 = atof( util_11);

    studmat[h-1][0]=ut1;
    studmat[h-1][1]=ut2;
    studmat[h-1][2]=ut3;
    studmat[h-1][3]=ut4;
    studmat[h-1][4]=ut5;
    studmat[h-1][5]=ut6;
studmat[h-1][6]=ut7;
studmat[h-1][7]=ut8;
studmat[h-1][8]=ut9;
studmat[h-1][9]=ut10;
studmat[h-1][10]=ut11;
}
fclose(matrix_file);

if( ttype == 1 ) /*city block distance */
for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
        utility[i][j]=0;
        for( h = 0; h<11; h++ ) {
            utility[i][j] += abs(studmat[i][h] - studmat[j][h]);
        }
    }
}

else if( ttype == 2 ) /* 15.279 */
for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
        k = (1.0/6.0) *
            ( abs(studmat[i][4] - studmat[j][4]) +
            abs(studmat[i][5] - studmat[j][5]) +
            abs(studmat[i][6] - studmat[j][6]) );
        
        dem = (1.0/3.0) *
            ( abs(studmat[i][0] - studmat[j][0]) +
            abs(studmat[i][1] - studmat[j][1]) +
            .25*(abs(studmat[i][2] - studmat[j][2])) );
        p = (-.25) *
            ( abs(studmat[i][7] - studmat[j][7]) +
            abs(studmat[i][8] - studmat[j][8]) +
            abs(studmat[i][9] - studmat[j][9]) +
            abs(studmat[i][10] - studmat[j][10]) - 4 );
        v = .5 * ( abs(studmat[i][3] - studmat[j][3]) );
        
        utility[i][j] = (.5* k + .15* dem + .15* p + .2* v);
    }
}

/*ensures that the compatibility between person and himself is 0. */
for( i = 0; i < N_VALUE; i++ ) {
    for( j = 0; j < N_VALUE; j++ ) {
        if( i == j ) utility[i][j]=0;
    }
}

srand((unsigned int) time(0)); /*randomize seed*/
ultmin=0;
ultsum=0;

for(f=0; f<100; f++) {/*loops through each random combo*/

    for(i=0; i<N_VALUE; i++){
        arraynew[1][i]=i+1;

        maxroll=100;
        for(i=0; i<N_VALUE; i++){
            arraynew[2][i]=genrand(maxroll);
        }

        for(i=0; i<N_VALUE; i++){
            for(h=i+1; h<N_VALUE; h++){
                if(arraynew[2][h]>arraynew[2][i]){
                    temp1=arraynew[2][h];
                    temp2=arraynew[1][h];
                    arraynew[2][h]=arraynew[2][i];
                    arraynew[1][h]=arraynew[1][i];
                    arraynew[2][i]=temp1;
                    arraynew[1][i]=temp2;
                }
            }
        }
    }

    /* for(i=0; i<N_VALUE; i++){
        printf("%i ",arraynew[1][i]);
    }
    printf("\n");
    for(i=0; i<N_VALUE; i++){
        printf("%i ",arraynew[2][i]);
    }
    printf("\n"); */

    for(i=0; i<N_VALUE; i++){
        combos[i+totteam]=arraynew[1][i];
    }

    /*find min for each team, put in first totteam slots of combos*/
for(t=1; t<=totteam; t++){
    b=totteam+K_VALUE*(t-1);
    min = maxmin(K_VALUE, combos, utility, b);
    combos[t-1]=min;
}

    sum=0;
    for(h=0; h<totteam; h++){
        sum+=sums[h];
    }
    setmin=5;
    for(i=0; i<totteam; i++){
        if(combos[i]<setmin) setmin=combos[i];
    }

    /* printf("setmin= %.4f\n",setmin); */

    if(f==0) ultmin=setmin;

    if(setmin<.05) b1++;
    else if(setmin <.1) b2++;
        else if(setmin <.15) b3++;
        else if(setmin <.2) b4++;
        else if(setmin <.25) b5++;
        else if(setmin <.3) b6++;
        else if(setmin <.4) b7++;
        else if(setmin <.5) b8++;
        else if(setmin <.6) b9++;
        else b10++;

    if(setmin>ultmin){
        ultmin=setmin;
        ultsum=sum;
        for(h=0; h<N_VALUE; h++){
            optimal[h]=combos[totteam+h];
        }
    }
    else if(setmin==ultmin & & sum >ultsum){
        ultsum=sum;
        for(h=0; h<N_VALUE; h++){
            optimal[h]=combos[totteam+h];
        }
    }

}/* end iteration of f loop */
for(i=0; i<N_VALUE; i++){
    printf(" %.0f",optimal[i]);
}
printf("\n");
printf("ultmin= %.4f\n",ultmin);
printf("ultsum= %.4f\n",ultsum);
printf("buckets are %i %i %i %i %i %i %i %i %i %i \n", b1, b2, b3, b4, b5, b6, b7, b8, b9, b10);

printf("ave is %.4f \n", ave);
} /*end main */

float maxmin(int k, float a[], float b[][N_VALUE], int r) {
    int u, v, w, x, y, z, i;
    float fmin;
    float tempval[10];
    int index[3];

    if( k == 3 ) u=3;
    else if(k == 4) u=6;
    else if(k==5) u=10;

    for(i=0; i<k; i++){
        index[i]=a[r]-1;
        r++;
    }

    x=r-u;
    y=0;
    for(i=0; i<u; i++){
        tempval[i]=b[index[x]][index[y]];
        y++;
        if(y>(k-1)){
            x++;
            y=x+1;
        }
    }

    fmin = 5;
    for( i = 0; i < u; i++ ) {
        if( tempval[i] < fmin ){
            fmin = tempval[i];
        }
    }
    sums[t-1] = 0;
    for( i = 0; i < u; i++ ) {
        sum
SUMS[t-1] += tempval[i];
}
return fmin;

int genrand(int maxroll){
    int roll;
    roll = rand() % maxroll + 1;
    return roll;
}
Appendix II: Improvements of heuristic methods over time

A. Maximize-the-minimum charts
The following 10 figures depict the change in the results for the five heuristic methods for different time periods - <1 minute to 60 minutes.
It is clear that, in all cases, the best method was random and pairswitch. Note how level the line for random-pairswitch is, indicating that it very quickly reaches a good solution and makes only small improvements with additional time. Also note that in the 100-student instances (the right-hand column) the random only heuristic lagged farther behind than it did for the 36-student instances, and in general there is a greater difference between the heuristics for 100 students than from 36. As noted in Section V, this is more likely a result of the larger teams (5 students/team vs. 3 students/team) rather than the larger classes.
B. Improvement of heuristics over time, Maximize-the-average objective function
Again, the random and pairwise method is superior, and relatively flat over time. While in the maximize-the-minimum case, the worst method was generally random, in this case the worst is often enumerate only, with random only placing second-worst.
Appendix III: Team Surveys from Classroom Experiment

A. Team Formation Survey

Code number (as given to you by your professor): ________________

Class Section (circle one): A (Prof. Dunphy) B(Prof. Heagney) C(Prof. Breslow)

Gender (circle one): Male       Female

Race (circle one): White       Non-white

Age ___________

Please rate the following on a level of 1 to 3, with 1 being the lowest and three being the highest:

How good are your writing skills? __

How good are your oral presentation skills? __

How good are your interpersonal communication skills? __

How motivated are you to do well in this class? __

Please go to http://www.humanmetrics.com/cgi-win/JTypes1.htm and take the Myers-Briggs type indicator test (press the “Do it!” button and answer the 72 questions that appear.) Record the results below:

First letter of type (E/I): ___
Strength of preference: ___%

Second letter of type(S/N): ___
Strength of preference: ___%

Third letter of type(T/F): ___
Strength of preference: ___%

Fourth letter of type(J/P): ___
Strength of preference: ___%
B. Team Satisfaction Survey

Your professor (circle one): Dunphy  Heagney  Breslow
Your code number: __________

**Please rate your team’s characteristics**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Excellent</th>
<th>Good</th>
<th>Fair</th>
<th>Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear understanding of team goals</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Commitment to team goals</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Preparedness for meetings</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Efficiency of meetings</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Quality of group participation in discussions</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Ability to reach consensus on decisions</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Equal sharing of workload</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Motivation</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Support for each other</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Overall team performance</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Please indicate the extent to which you agree or disagree with the following statements:**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual team members brought valuable qualities to the team</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>I am enjoying working with the members of my team</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**As a result of your work with this team, to what extent did your ability to do the following improve?**

<table>
<thead>
<tr>
<th>Ability</th>
<th>A great deal</th>
<th>A fair amount</th>
<th>Somewhat</th>
<th>Not at all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communicate effectively with teammates</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Resolve conflict and reach agreement within team</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Pay attention to the feelings of all team members</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Listen to the ideas of others with an open mind</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Exercise my leadership skills</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Motivate other team members</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Work collaboratively with students of different backgrounds, cultures, and levels of expertise</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>