Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

BUILDING EQUITABLE STUDENT PROJECT GROUPS - A SIMULATION STUDY TO ASSESS HEURISTIC ASSIGNMENT METHODS

Matthew Dabkowski[1], Stephen Gillespie[1], Ian Kloo[1], Devon Compeau[1], and Mai Tran[1]

[1]Dept. of Systems Eng. (DSE), United States Military Academy (USMA), West Point, NY, USA

ABSTRACT

This paper explores methods for forming equitable project groups in a class by minimizing the range in mean cumulative quality point averages (CQPAs) across the groups. Using simulation, it forms representative classes of various sizes from real-world CQPA data, and it compares the approximate solutions of seven heuristic assignment methods to the globally optimal solutions found with mixed integer linear programming. Finding a near optimal heuristic is useful as exact methods can be computationally expensive or require specialized knowledge or software. Simulation results indicate that the Alternating Convergence heuristic consistently outperforms the other six heuristics, generating near optimal solutions with low computational cost. This suggests the Alternating Convergence heuristic is a practical method for building project groups when equitability in scholastic achievement supports one's pedagogical philosophy or objectives.

1 INTRODUCTION AND BACKGROUND

In today's hyperconnected world, most professional settings involve groups of people working together to solve the world's most challenging problems. As these problems become increasingly complex and global in scale, the ability to work and function in a group becomes more crucial (Hung 2013). Educational institutions recognize this trend and shape their students for the realities of the modern workforce by actively fostering a collaborative learning environment (Barkley et al. 2005).

Definitionally, groups may be described as a collection of people "linked together by some type of relationship," interacting to accomplish a specific objective (Forsyth 2014, p. 2). For example, common reasons to form groups are to increase the cohesion within an organization and increase the exposure to new ideas between different sectors of a company. Once the purpose of a group or a collection of groups is established, they can be formed using a variety of algorithms, including random assignment, self-selection, and criteria-based methods. Research indicates that self-selection is preferable to random assignment for group dynamics (Chapman et al. 2006), while criteria-based approaches may be perceived as fairer (Cutshall et al. 2007).

Beyond measuring the effectiveness of assignment methods with group performance, researchers have developed metrics like the Jain's index to measure and optimize the fairness of the groups (Jain, Chiu, and Hawe 1984). Recently, Rezaeinia et al. (2022) continued to explore the notion of group equitability and the applications of the Jain's index when studying the allocation of graduate students to group projects. Pedagogically, building equitable groups – subsets of students with acceptably similar compositions in one or more dimensions of interest – is an important endeavor because it can enhance individual learning opportunities. Equitable groups can help combat biases and stereotypes that specific group members may have by introducing them to new ideas and experiences. They can also encourage group members to develop important teamwork and conflict resolution skills that are required throughout one's adult life.

This research paper focuses on a criterion-based method to form equitable groups. Specifically, we attempt to assign students to groups such that the difference in the largest and smallest group means for some desired metric (i.e., the range in means) is minimized. While one can use any situation-appropriate, quantifiable metric to define group equity, this paper focuses on student academic achievement as expressed in the cumulative quality point average (CQPA). The CQPA, which is synonymous with the grade point average (GPA), is a reasonable metric to use, as it is generally available, measured across a variety of domains by numerous observers (i.e., multiple classes with different instructors), and reasonably correlated with a student's academic interest, motivation, and capability. That stated, the methods discussed in this paper apply to any relevant measure.

For groups formed from populations of 25 or more people, finding the optimal solution becomes computationally expensive. Moreover, even if finding the optimal solution is tractable, there are many environments in which the knowledge or resources to do so is infeasible, such as for teachers who do not specialize in optimization or soldiers who are deployed in combat environments. As such, the primary purpose of this paper is to simulate and assess heuristics to identify methods that produce simple to understand, computationally efficient, near-exact solutions. The paper's remaining sections are organized as follows: Section 2 describes the method of determining group sizes and introduces exact and heuristic methods to assign students to groups; Section 3 assesses the assignment methods' anticipated theoretical performance using real-world CQPAs; Section 4 uses simulation to estimate their absolute and relative performance; Section 5 offers key findings and insights; and Section 6 summarizes the main ideas of the paper and proposes areas of future work.

2 BUILDING PROJECT GROUPS

2.1 Determining Group Size

Suppose we have a class of n students, and we want to build m equal-sized project groups. If n divides evenly into m, we can build m groups of n/m students. Otherwise, n is not divisible by m, and our m groups will have different sizes. For example, if we want to build five equal-sized project groups in a class of 18 students, the best we can do is three groups of four and two groups of three.

To determine the number and size of the groups, we can apply the well-known division algorithm (Rosen 2000). Specifically, if mod represents the modulo operator such that (n mod m) returns the remainder of n/m and $\lfloor\,\rfloor$ indicates the floor function such that $\lfloor n/m \rfloor$ returns the quotient, the distribution of groups is given by Algorithm 1.

Algorithm 1: The division algorithm

```
Data: n students, m project groups, n ≥ m
Result: distribution of project group size
if n mod m = 0 then
  | m groups of size n/m;
else
  | (n mod m) groups of size (⌊n/m⌋ + 1), and;
  | (m − n mod m) groups of size ⌊n/m⌋;
end
```

2.2 Assigning Group Members

With the distribution of group sizes established, the next task is to assign students to groups. If students cannot be assigned to multiple groups and all students must be assigned, the assignment process generates a partition – a mutually disjoint collection of groups (i.e., subsets) whose union equals the class (i.e., set). For instance, if we want to assign five students $\{a,b,c,d,e\}$ to two "equal-sized" groups, we can assign them as $\{a,b,c\}$ and $\{d,e\}$. Enumerating the possible assignments using the *setparts* function from R's *partitions* package, yields the 10 subsets seen in Table 1.

Table 1: Possible partitions of five students into a group of three (Group 1) and a group of two (Group 2), where the numbers in the table's interior denote the group assignments.

	Partition Number									
Student	1	2	3	4	5	6	7	8	9	10
a	1	1	1	1	1	1	2	2	2	2
b	1	1	1	2	2	2	1	1	1	2
c	1	2	2	1	1	2	2	1	1	1
d	2	1	2	1	2	1	1	2	1	1
e	2	2	1	2	1	1	1	1	2	1

To calculate the number of possible partitions $p(n,m)$ directly, we can apply Theorem 13.2 from Andrews (1976) as seen in Equation (1) below:

$$p(n,m) = \begin{cases} \dfrac{n!}{((n/m)!)^m \, m!} & \text{if } r = 0, \\ \dfrac{n!}{((q+1)!)^r \, (q!)^{m-r} \, r! \, (m-r)!} & \text{otherwise,} \end{cases} \tag{1}$$

where $r = n \bmod m$ and $q = \lfloor n/m \rfloor$. Generally speaking, as n or m increases, the number of possible partitions increases, and the growth can be dramatic. For example, if we want to build five equal-sized groups in a class of 20 students, there are over 2.5 billion possible partitions. If we add a 21st student to the class, one of the groups would have to increase its membership from four to five. This seemingly minor growth increases the number of possible partitions to approximately 53 billion. Moreover, if we want to limit our groups to less than five students, we would have to add a sixth group, increasing the number of possible partitions to roughly 475 billion.

2.2.1 Exact Assignment Method - Mixed Integer Linear Programming

As discussed in Section 1, this paper aims to build equitable groups that minimize the range in mean CQPAs. Mathematically, this can be modeled by the following binary program.

Index and set use: Let $N = \{1,\ldots,n\}$ denote the set of students, and $M = \{1,\ldots,m\}$ denote the set of groups.

Data: Let c_i denote the CQPA of student i, and s_j denote the size of group j, as determined by the division algorithm.

Decision variables: Let $x_{i,j}$ be binary, where $x_{i,j} = 1$ implies student i is assigned to group j; otherwise, $x_{i,j} = 0$.

Formulation:

$$\min \quad \left(\max_{j \in M} \left\{ \frac{1}{s_j} \sum_{i=1}^{n} c_i x_{i,j} \right\} - \min_{j \in M} \left\{ \frac{1}{s_j} \sum_{i=1}^{n} c_i x_{i,j} \right\} \right) \tag{2a}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{i,j} = 1 \qquad \forall\, i \in N \tag{2b}$$

$$\sum_{i=1}^{n} x_{i,j} = s_j \qquad \forall\, j \in M \tag{2c}$$

$$x_{i,j} \in \{0,1\} \quad \forall\, i \in N,\ j \in M \tag{2d}$$

Discussion: In the objective function (2a), $\frac{1}{s_j}\sum_{i=1}^{n} c_i x_{i,j}$ is the mean CQPA of the students assigned to group j. The maximum and minimum of the m group means are given by $\max_{j \in M}\left\{\frac{1}{s_j}\sum_{i=1}^{n} c_i x_{i,j}\right\}$ and

$\min_{j \in M}\left\{\frac{1}{s_j}\sum_{i=1}^{n} c_i x_{i,j}\right\}$, respectively, and their difference represents the range. Constraint set (2b) ensures each student is assigned to one group, and constraint set (2c) requires each group to have the requisite number of students.

As written, the above binary program is non-linear due to the $\max_{j\in M}\{\cdot\}$ and $\min_{j\in M}\{\cdot\}$ terms in (2a). In general, equivalent linear formulations of non-linear mathematical programs are faster to solve, and their solutions are guaranteed to be globally optimal. To this end, we can linearize the maximum and minimum operators in (2a) by: (a) replacing the $\max_{j\in M}\{\cdot\}$ and $\min_{j\in M}\{\cdot\}$ terms with the continuous variables y and z, respectfully (see (3a)); (b) introducing m additional binary variables b_j (see (3i)); and (c) adding appropriate constraint sets on the new variables (see (3d) through (3g)) (Asghari et al. 2022). These adjustments transform the original binary non-linear program into the equivalent mixed integer linear program (MILP) seen below, where B represents a suitably "big" number (e.g., 1000).

$$
\begin{aligned}
\min \quad & y - z && && \text{(3a)}\\
\text{s.t.} \quad & \sum_{j=1}^{m} x_{i,j} = 1 && \forall\ i \in N && \text{(3b)}\\
& \sum_{i=1}^{n} x_{i,j} = s_j && \forall\ j \in M && \text{(3c)}\\
& y \geq \frac{1}{s_j}\sum_{j=1}^{m} c_i x_{i,j} && \forall\ j \in M && \text{(3d)}\\
& z \geq \frac{1}{s_j}\sum_{j=1}^{m} c_i x_{i,j} - B b_j && \forall\ j \in M && \text{(3e)}\\
& z \leq \frac{1}{s_j}\sum_{j=1}^{m} c_i x_{i,j} && \forall\ j \in M && \text{(3f)}\\
& \sum_{j=1}^{m} b_j \leq m-1 && \forall\ j \in M && \text{(3g)}\\
& x_{i,j} \in \{0,1\} && \forall\ i \in N,\ j \in M && \text{(3h)}\\
& b_j \in \{0,1\} && \forall\ j \in M && \text{(3i)}
\end{aligned}
$$

The MILP defined by equations (3a) through (3i) can be solved exactly using implicit enumeration methods such as the branch and cut method (Bertsimas and Tsitsiklis 1997), as implemented in modern solvers such as the International Business Machines' Complex Linear Programming Expert (CPLEX) (IBM Corporation 2019).

2.2.2 Heuristic Assignment Methods

Heuristics are often used to solve problems when optimal methods are either unknown or resource prohibitive. In the case of assigning students to equitable project groups, we assess five deterministic (Linear Draft, Snake Draft, Feed the Poor, Tax the Rich, and Alternating Convergence) and two stochastic (Random and Stratified Random) heuristics. Of these, the Linear Draft, Snake Draft, Random, and Stratified Random are existing heuristics as cited below. The authors developed the Feed the Poor, Tax the Rich, and Alternating Convergence heuristics. While other alternatives abound, these seven heuristics represent a reasonable set of well-known or intuitive ways to methodically build project groups.

Linear Draft The Linear Draft heuristic sorts n students in descending order by CQPA and subsequently assigns them to m groups in $\lceil n/m \rceil$ draft rounds using the same draft order in each round. For example, if there are seven students to be assigned to three groups, the sorted students (labeled a though g) would be assigned to groups (designated 1 to 3) as follows: Round 1 | $a \rightarrow 1$, $b \rightarrow 2$, $c \rightarrow 3$; Round 2 | $d \rightarrow 1$, $e \rightarrow 2$, $f \rightarrow 3$; Round 3 | $g \rightarrow 1$.

This is approximately similar to how the American National Football League (NFL) allocates players during the draft to each of 32 teams (NFL 2024), although the actual NFL draft allows for nuances such as trades. This also assumes that every team ranks all players in the same order. Regardless, the Linear Draft is a highly common and easily applied heuristic.

Snake Draft The Snake Draft heuristic is identical to the Linear Draft except for the draft order of the project groups, which flips at the beginning of each draft round. Continuing with the example from above, if there are seven students to be assigned to three teams, they would be assigned as follows: Round 1 | $a \rightarrow 1$, $b \rightarrow 2$, $c \rightarrow 3$; Round 2 | $d \rightarrow 3$, $e \rightarrow 2$, $f \rightarrow 1$; Round 3 | $g \rightarrow 1$. This heuristic is often employed in fantasy sports leagues to assign players to teams (Lee and Liu 2022).

Feed the Poor The Feed the Poor heuristic initially assigns each group one of the m highest ranked students by CQPA. It then assigns the remaining student with the next highest CQPA to the partially filled group with the lowest mean CQPA. This process repeats itself until all students are assigned.

Tax the Rich The Tax the Rich heuristic is identical to Feed the Poor, except it assigns the remaining student with the lowest CQPA to the partially filled group with the highest mean CQPA. Notably, this approach always assigns the students with the highest and lowest CQPAs to the same team.

Alternating Convergence The Alternating Convergence heuristic combines the Feed the Poor and Tax the Rich heuristics. As before, it initially assigns each group one of the m highest ranked students by CQPA. It then alternates between "taxing the rich" and "feeding the poor" on every assignment until all students are assigned.

Random In random assignment, students are assigned to the m groups using any standard random number generator. This is effectively a standard random sample from all possible permutations of group formation (Lohr 2010).

Stratified Random In stratified random assignment, students are sorted in descending order by CQPA and divided into strata of size m or $m-1$, similar to a stratified sample (Lohr 2010). Within each strata, students are randomly assigned to one of the m teams. For strata of size $m-1$ (and without loss of generality), students are only assigned to the first $m-1$ teams.

3 ANTICIPATED THEORETICAL PERFORMANCE BY ASSIGNMENT METHOD

3.1 Real-world CQPA Data

We can assess the anticipated performance of the assignment methods in Section 2 using real-world CQPAs. Among the data sources available, the High School Longitudinal Study of 2009 (HSLS09) is an attractive choice for its relevance, curation, and size. Specifically, in 2009 the National Center for Education Statistics (NCES) began following a representative sample of over 23,000 high school freshmen from nearly 950 schools (National Center for Education Statistics 2020a). Between 2017 and 2018 NCES augmented HSLS09 by collecting postsecondary academic transcripts from the study's participants (National Center for Education Statistics 2020b), and the merged public use dataset includes the participants' cumulative college GPAs normalized to a 4-point scale (i.e., CQPAs). Given our focus on building project groups for upper division courses, we restricted our attention to HSLS09 participants with at least 24 months of undergraduate enrollment who were currently enrolled in a 4-year program or had already attained a bachelor's degree. Applying these filters yielded a subsample of 6,130 CQPAs.

3.2 Distribution Fitting

Mathematically, CQPA is the weighted average of the quality points a student has earned in their courses, where the weights are the courses' corresponding number of credit hours. For example, imagine a student took two courses in their first semester of college – AB123 and CD456. If AB123 was 3 credit hours and the student earned an A (4.0 quality points) and CD456 was 3.5 credit hours and the student earned an A- (3.7

quality points), the student's CQPA would be $(4(3)+3.7(3.5))/(3+3.5) = 3.8385$. This simple illustration highlights two points. First, although CQPA is a function of a finite number of discrete variables that can assume small subsets of similar values, it can safely be treated as a continuous variable for analytical purposes. Second, CQPA is bounded below and above by the quality points of the lowest and highest grades a student has earned, respectively, which are contained in the closed interval $[0,4]$. In short, CQPA is bounded and continuous.

Interestingly, although NCES acknowledges the HSLS09 variable for CQPA (denoted *X5GPAALL*) is continuous, it limits *X5GPAALL* to one decimal place (Duprey et al. 2020), which restricts CQPA to 41 possible values (i.e., from 0.0 to 4.0 in increments of 0.1) and essentially makes it discrete. Beyond sacrificing fidelity, this discretization is problematic for assessing the performance of Section 2's assignment methods. Specifically, if we want to build representative classes of n students and we fit a discrete distribution to *X5GPAALL* or resample it with replacement, the likelihood of obtaining duplicate CQPAs will be dramatically higher than reality. Accordingly, we fit a four-parameter Beta distribution (Vose 2008) to our subsample of CQPAs using the `eBeta_ab` function from R's `ExtDist` package (Wu et al. 2023).

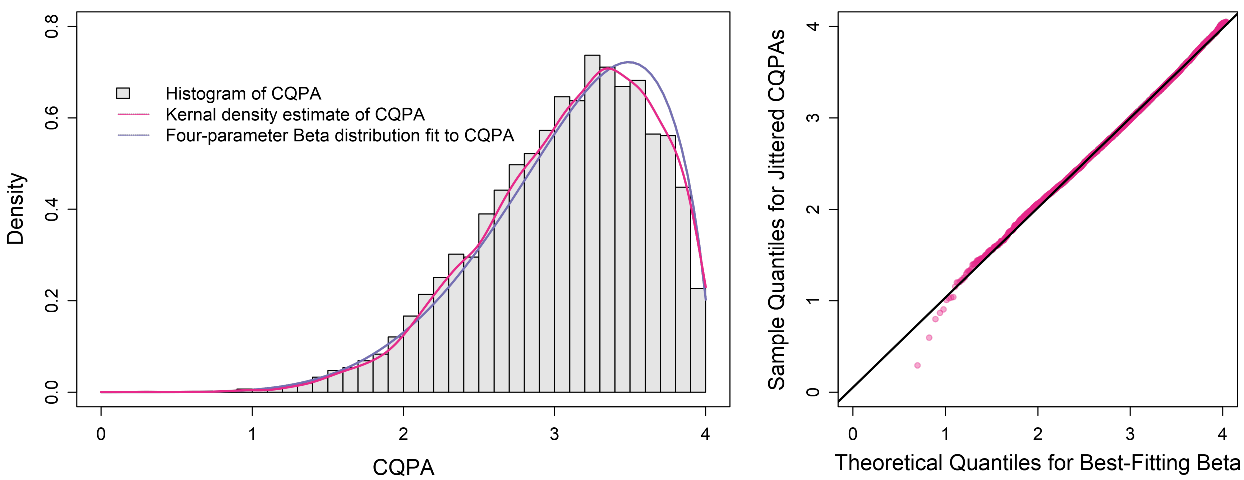

Figure 1: Goodness-of-fit plots for the CQPAs of the HSLS09 subsample described in Section 3.1. The left panel displays the best-fitting four-parameter Beta distribution (blue line) superimposed on the histogram (gray bars) and kernel density estimate (pink line) of HSLS09's *X5GPAALL* variable. The right panel is the Q-Q plot of the quantiles for the best-fitting four-parameter Beta distribution and the subsample of CQPAs, where a small amount of random noise ($\varepsilon \sim \text{Uniform}(0, 0.05)$) has been added to the CQPAs to fill in the distributional gaps caused by discretization.

As seen in the left panel of Figure 1, the best-fitting four-parameter Beta distribution $(\text{Beta}(\alpha = 5.61, \beta = 1.76, l = 0.13, u = 4.04))$ generally follows the histogram and kernel density estimate of the CQPA subsample, especially over the interval [1.0, 3.3]. The Q-Q plot in the right panel reinforces this and confirms the adequacy of the fit, as all but 9 of the 6,130 points closely mirror the line of equality.

3.3 Order Statistics

Armed with a suitable distribution for CQPA, we can estimate the expected CQPAs in a class of n students using order statistics, where $\mathbb{E}[C_{(i)}]$ represents the expected CQPA of the i^{th} ranked student. To calculate $\mathbb{E}[C_{(i)}]$, we first transform the best-fitting four-parameter Beta random variable C into the standard two-parameter Beta random variable $C' \sim \text{Beta}(\alpha = 5.61, \beta = 1.76, l = 0, u = 1)$ using the relation $C' = (C-l)/(u-l)$ (Johnson et al. 1994-1995). Next, we use the `beta.dist` function from R's `binGroup` package (Zhang et al. 2018) to calculate the expected values of the order statistics for a sample of size n drawn from the standard two-parameter Beta distribution ($\mathbb{E}[C'_{(i)}]$). Finally, we invert the previous transformation to recover $\mathbb{E}[C_{(i)}]$ as follows: $\mathbb{E}[C_{(i)}] = (u-l)\mathbb{E}[C'_{(i)}] + l$ (Johnson et al. 1994-1995). For illustrative purposes, if we have a class of 20 students, their expected CQPAs can be calculated using the R command: `(4.04 - 0.13) * beta.dist(alpha = 5.61, beta = 1.76, grp.sz = 20) + 0.13`, which yields the $\mathbb{E}[C_{(i)}]$ seen in Table 2

Table 2: Expected CQPAs in a class of 20 students.

i	1	2	3	4	5	6	7	8	9	10
$\mathbb{E}[C_{(i)}]$	1.8588	2.1985	2.4081	2.5658	2.6953	2.8069	2.9062	2.9966	3.0804	3.1590
i	11	12	13	14	15	16	17	18	19	20
$\mathbb{E}[C_{(i)}]$	3.2337	3.3055	3.3752	3.4436	3.5114	3.5796	3.6492	3.7220	3.8013	3.8947

3.4 Expected Range of Group CQPAs

The approach described in Section 3.3 provides a convenient way to assess the anticipated performance of the deterministic assignment methods in Section 2. Specifically, suppose we have a class of $N = \{1,\ldots,n\}$ students, and we want to assign them to $M = \{1,\ldots,m\}$ project groups. The expected CQPA of project group j (denoted g_j) is $\mathbb{E}[\frac{1}{s_j}\sum_{i\in g_j} C_i]$, where s_j is the number of students assigned to project group j. By the linearity of expectations, the expectation of a sum of random variables is equal to the sum of their expectations (Lehman et al. 2015), implying $\mathbb{E}[\frac{1}{s_j}\sum_{i\in g_j} C_i] = \frac{1}{s_j}\sum_{i\in g_j}\mathbb{E}[C_i]$. Order statistics are random variables in their own right. Accordingly, once n is known, the expected values of the students' ordered CQPAs (i.e., $\mathbb{E}[C_{(i)}]$) can be calculated, the students can be assigned to project groups via the deterministic assignment methods in Section 2, and the expected CQPAs of the project groups can be estimated using $\frac{1}{s_j}\sum_{(i)\in g_j}\mathbb{E}[C_{(i)}]\ \forall\ j \in M$. Applying this approach to the class of 20 students seen in Table 2 yields the anticipated performance in Figure 2 on the next page.

As seen in Figure 2, the anticipated performance of the deterministic assignment methods varies substantially, as the worst performing heuristic's range (Feed the Poor at 0.887) is nearly 14 times wider than the best performing heuristic's range (Alternating Convergence at 0.065), and the best performing heuristic's range is nearly 22 times wider than the globally optimal solution (MILP at 0.003). Notably, these differences were realized despite the consistent assignment of the top four students to different groups, and with the exception of the Linear Draft heuristic, the assignment of the highest and lowest performing students to the same group. In short, when it comes to the anticipated performance of the deterministic assignment methods, the action appears to be in the interior of the ordering. Nonetheless, in a relative, expected sense, Alternating Convergence appears to be the best heuristic, followed by Snake Draft, Tax the Rich, Linear Draft, and Feed the Poor, in that order. The degree to which this relationship holds in general is explored in Section 4.

4 OBSERVED SIMULATED PERFORMANCE BY ASSIGNMENT METHOD

To test the anticipated theoretical performance of the deterministic heuristics in Section 3.4 against the MILP-generated optimal solutions, as well as assess the performance of the two stochastic approaches, we performed a simulation of each assignment method across 1,000 hypothetical classes for class sizes of 10, 15, 20, and 25 students each (a total of 4,000 simulation replicates). First, we created the hypothetical classes by drawing the appropriate number of student CQPAs from the distribution fit to actual student data, as described in Section 3.2. Next, we found each class's smallest possible difference between the average CQPAs of the highest and lowest performing groups after assignment by solving the MILP described in Section 2.2.1 with CPLEX. Last, we estimated the performance of the heuristics by generating the heuristic group assignments for each class, calculating the ranges of the project groups' mean CQPAs, and subtracting the respective MILP-generated globally optimal solutions from these values.

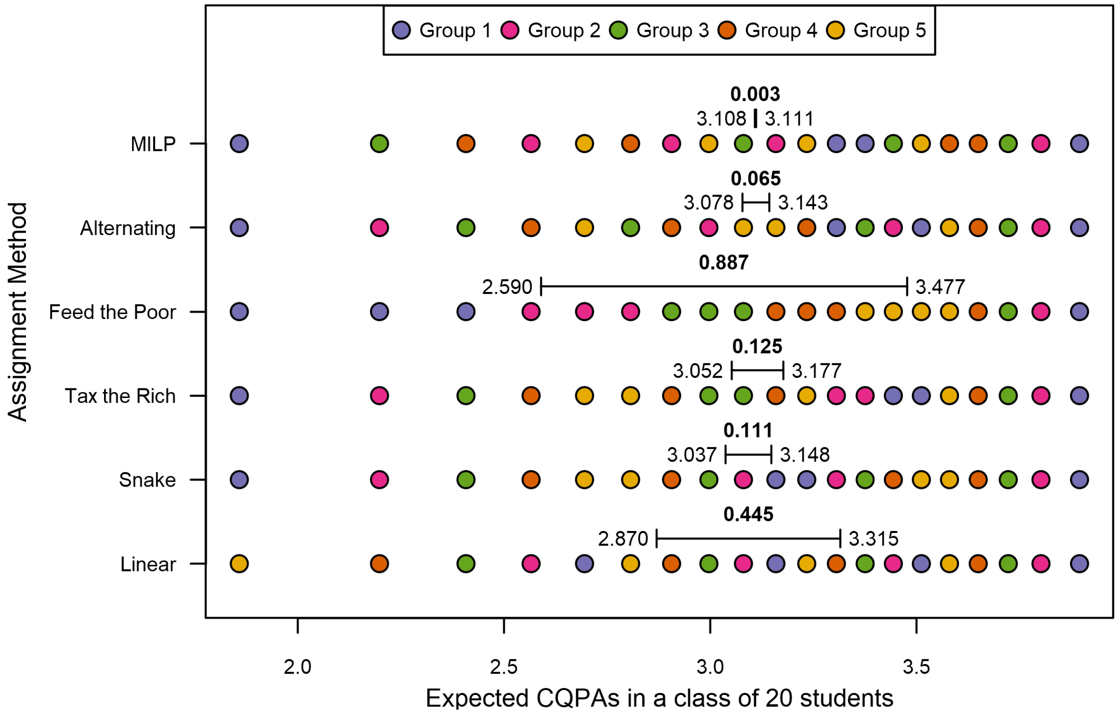

Figure 2: Anticipated performance of Section 2's deterministic assignment methods applied to a class of $n = 20$ students divided into $m = 5$ equal-sized project groups, where circles represent students, locations on the horizontal axis reflect expected CQPAs, and colors denote project groups. For each assignment method, the minimum and maximum expected project group CQPAs are to the left and right of the range bar immediately above the students, and the range of project group CQPAs is given in bold.

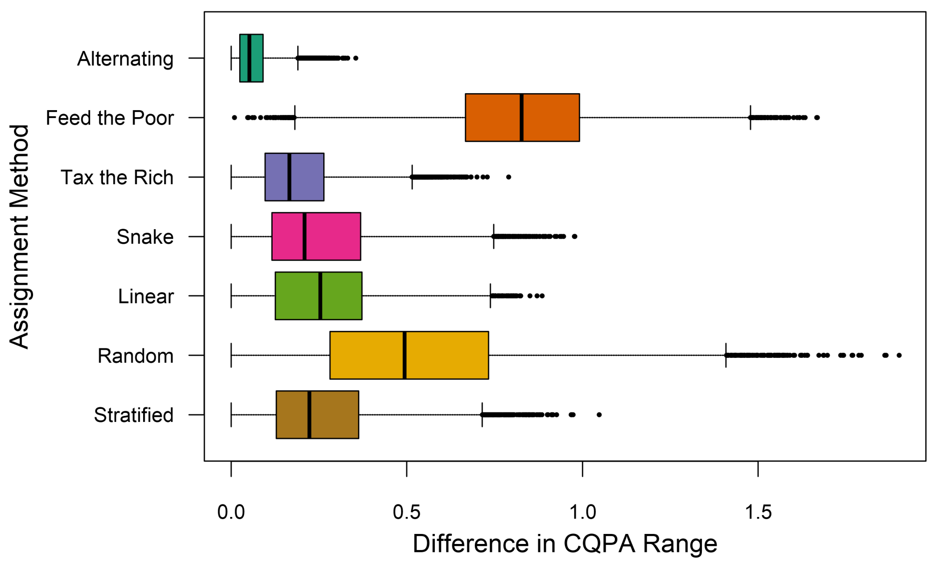

Figure 3: Boxplots of the differences between the CQPA ranges of heuristically assigned project groups and their respective MILP-generated globally optimal solutions.

Figure 3 confirms the theoretical expectation that the Alternating Convergence method is the most performant heuristic method. On average, this method resulted in a CQPA gap between the best and worst groups of only 0.064 points. The Tax the Rich, Snake Draft, Stratified Random, and Linear Draft methods were the next best and were similarly performant on average. Random allocation performed worse than this group, and the overall worst assignment method was the Feed the Poor heuristic.

Looking beyond the overall performance in Figure 3, it is even more clear that the Alternating Convergence method is the best of the heuristics. The left panel of Figure 4 shows that most of the methods performed worse when forced to create groups with small numbers of students and that increasing the group size decreased the distance from the MILP-generated globally optimal solutions (i.e., 0). This finding is intuitive, as increasing the number of students in the groups creates more opportunities to balance the groups' mean CQPAs. In all cases, the Alternating Convergence method is the preferred heuristic. In the right panel of Figure 4, we can see the performance advantage from using the Alternating Convergence method is constant across class sizes. Interestingly, the Feed the Poor, Tax the Rich, and both random methods performed worse as class size increased, while the other methods remained constant. Tax the Rich, however, was the second-best method given the class sizes tested in this study.

Figure 5 shows that the class size and number of group combinations that allowed for even groupings (e.g., a class size of 20 with groups of 4) generally resulted in heuristic performance closer to the MILP-generated globally optimal solutions. For the Alternating Convergence method, there was a statistically significant (but practically insignificant) difference of 0.003 CQPA points when using even versus uneven groups. Interestingly, the relationship is reversed for the Linear Draft, which performed closer to the MILP optimal values when the groups were uneven. This performance, however, is still much worse than the Alternating Convergence method for uneven groups.

In addition to performance, the runtime of each algorithm is also important to consider. Figure 6 shows the mean runtime of each algorithm over the different class sizes. All experiments were run in R on an Apple M1 Pro processor with 32 GB of RAM. The left panel demonstrates that the MILP runtime increases exponentially as class size increases, while the right panel indicates that the Alternating Convergence, Tax the Rich, and Feed the Poor methods scale linearly. The remaining methods exhibited constant runtimes over the class sizes tested. While the heuristic methods' runtime scaling differs, the differences in the actual runtimes are practically insignificant, as they are measured in hundredths of a second.

The exponential increase in runtime for the MILP suggests that it may be impractical to run for larger class sizes. Moreover, while the average MILP runtime for 25-student classes was 3 seconds, 5 samples out of 1,000 took longer than 10 minutes to solve. For comparison, the maximum runtime for Alternating Convergence was 0.29 seconds. Additionally, when attempting experiments with class sizes of 50, we encountered hours-long MILP runtimes.

5 DISCUSSION

Forming groups of students that minimize the maximal difference in group means of CQPAs can be solved exactly using the MILP as described in Section 2.2.1. There are at least seven heuristics that can approximate the exact solution as described in Section 2.2.2. The team assessed the performance of these methods by simulating 1,000 hypothetical cases of classes of size 10, 15, 20, and 25 based on a real-world distribution of CQPAs from the HSLS09 dataset. From this work, there are two key findings: resource constraints and heuristic performance.

For relatively small classes (e.g., 25 or fewer), it is feasible to solve the assignment problem exactly. Depending on precise computing resources, however, this becomes challenging when the class size increases, as the number of possible groups that can be formed from a class of size n split into m groups grows combinatorially. For example, there are approximately 1.4×10^{37} ways to form 10 groups of five from 50 students. This growth, and the additional computation required, inhibit finding the exact solution in an acceptable amount of time for most common cases. Heuristic solutions can resolve this resource constraint, assuming they are sufficiently close to optimal.

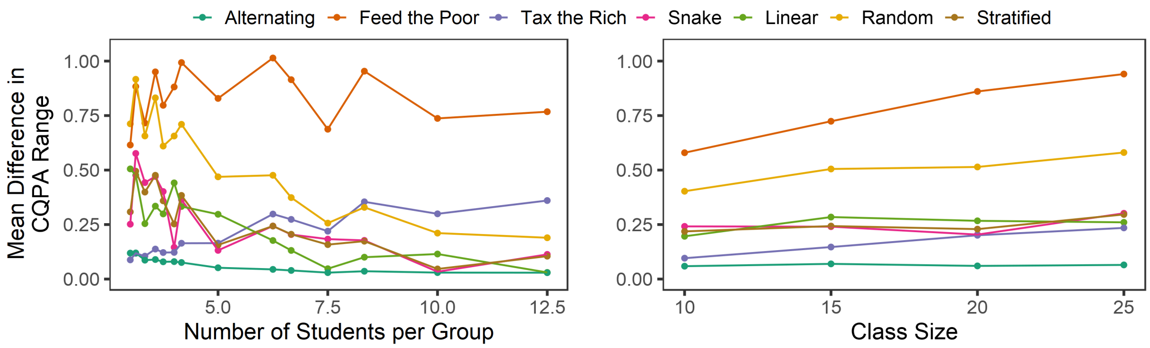

Figure 4: Mean differences between the CQPA ranges of heuristically assigned project groups and their respective MILP-generated globally optimal solutions by the number of students in each group (left panel) and class size (right panel).

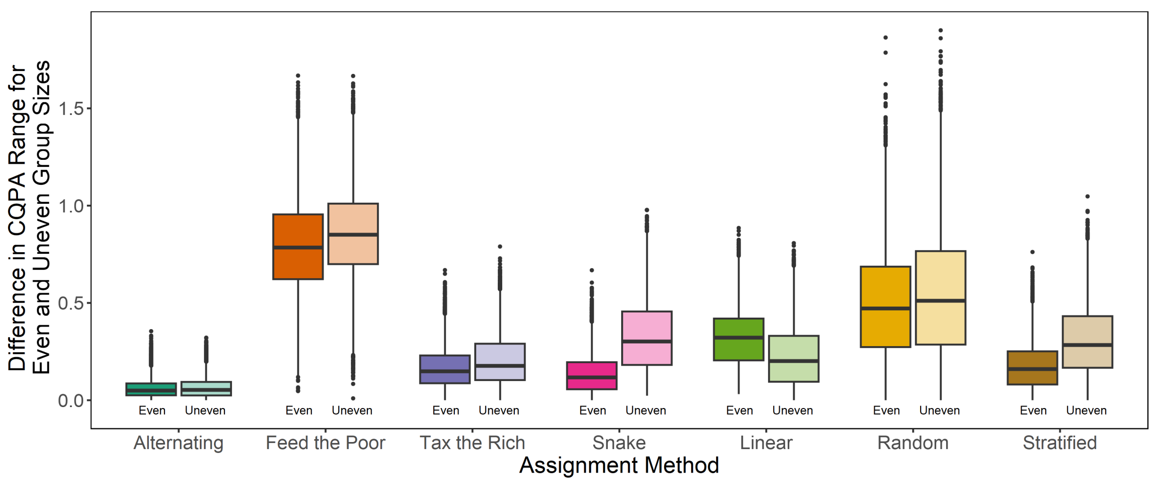

Figure 5: Differences between the CQPA ranges of heuristically assigned project groups and their respective MILP-generated globally optimal solutions for classes with even and uneven group sizes.

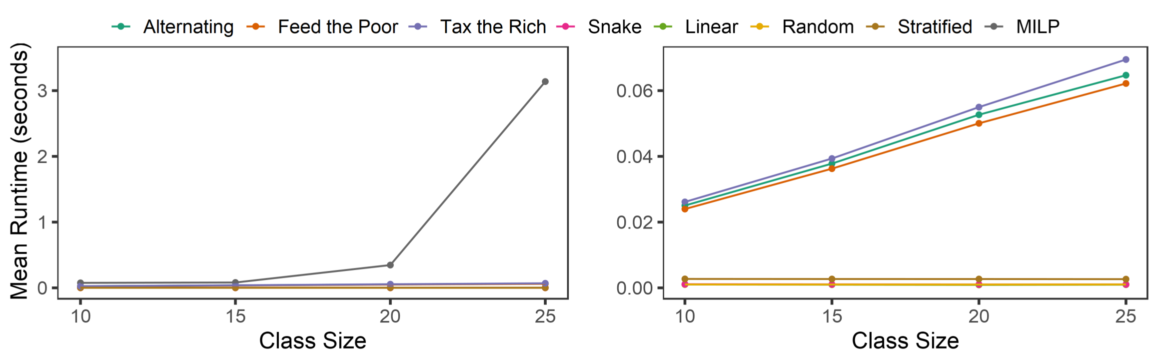

Figure 6: Mean runtime (in seconds) of heuristically assigned project groups and MILP-generated globally optimal solutions by class size for all methods (left panel) and heuristics only (right panel). Notably, the maximum y-axis limits of the plots differ by two orders of magnitude.

The heuristic solutions are resource efficient with most of the computation devoted to ordering the students according to CQPA. The natural question, then, is which one to choose. Simulation reveals that Alternating Convergence is not only the most performant heuristic explored in this paper, but it is also reasonably accurate relative to the exact solution. In the context of forming equitable groups, the mean difference between the CQPA ranges of groups assigned with the Alternating Convergence heuristic and their respective MILP-generated globally optimal solutions is only 0.064 on a 4.0 scale, which represents an average error of 1.60%. It is unlikely that students or instructors could truly discern such a fine difference across groups. The performance of the Alternating Convergence heuristic is also consistent, as the standard deviation of its error was the least across all studied at 0.051 on a 4.0 scale (1.27%) – less than half the next closest heuristic's standard deviation. This means a teacher forming groups can be highly confident that using the Alternating Convergence heuristic will lead to a near-exact solution.

While this paper focused on creating student project groups with equitable potential for academic achievement, its methods have broad applicability, especially in situations when its heuristics must be implemented manually. For example, in the military forming equitable teams of soldiers is often desirable. Moreover, when operating in a field environment, a military unit's access to computers or software may be limited. Illustratively, imagine a forward deployed military unit wants to balance the mean weight of its personnel and cargo across helicopters for an upcoming air assault operation. The Alternating Convergence heuristic could be applied by-hand to obtain a good, likely excellent solution.

6 CONCLUSION AND FUTURE WORK

This paper explored different methods for forming equitable project groups based on students' CQPAs. We compared the performance of seven easy-to-implement heuristic assignment methods to MILP, an exact method based on mathematical programming. While the MILP solutions are globally optimal, solving the optimization problems became resource prohibitive as the number of students being assigned became large. In such circumstances, heuristics provide relief, but there is a trade-off between the heuristics' resource efficiency and the closeness of their solutions to the global optima. Of the heuristics examined in this paper, Alternating Convergence yielded the group assignments that most closely matched the performance of the exact solutions across different class sizes and numbers of groups.

Overall, this research provides a simple, intuitive way for instructors to build equitable project groups with nearly equal average CQPAs. Future work may include additional testing to examine our hypothesis that the Alternating Convergence heuristic's performance remains superior and constant as class sizes increase. Additionally, we can assess the heuristics in more detail to determine if different distributions of student CQPA impact the performance of the heuristics. Finally, we intend to develop an easily accessible application that quickly and effectively assigns students to project groups via exact or heuristic methods.

ACKNOWLEDGMENTS

The views expressed herein are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, or the Department of Defense.

REFERENCES

Andrews, G. E. 1976. "The Theory of Partitions". In *Encyclopedia of Mathematics and Its Applications*, edited by G.-C. Rota, Volume 2. Reading, Massachusetts: Addison-Wesley.

Asghari, M., A. M. Fathollahi-Fard, S. Mirzapour Al-E-Hashem, and M. A. Dulebenets. 2022. "Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey". *Mathematics* 10(283):1–26.

Barkley, E. F., C. H. Major, and K. P. Cross. 2005. *Collaborative Learning Techniques: A Handbook for College Faculty*. 1st ed. San Francisco, California: John Wiley & Sons.

Bertsimas, D. and J. N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Belmont, Massachusetts: Athena Scientific.

Chapman, K. J., M. Meuter, D. Toy, and L. Wright. 2006. "Can't We Pick Our Own Groups? The Influence of Group Selection Method on Group Dynamics and Outcomes". *Journal of Management Education* 30(4):557–569.

Cutshall, R., S. Gavirneni, and K. Schultz. 2007. "Indiana University's Kelley School of Business Uses Integer Programming to Form Equitable, Cohesive Student Teams". *Interfaces* 37(3):265–276.

Duprey, M. A., D. J. Pratt, D. H. Wilson, D. M. Jewell, D. S. Brown, L. R. Caves *et al*. 2020. "High School Longitudinal Study of 2009 (HSLS:09) Postsecondary Education Transcript Study and Student Financial Aid Records Collection - Data File Documentation - Appendices. NCES 2020-004". Technical report, National Center for Education Statistics.

Forsyth, D. R. 2014. *Group Dynamics*. 6th ed. Belmont, California: Wadsworth Cengage Learning.

Hung, W. 2013. "Team-Based Complex Problem Solving: A Collective Cognition Perspective". *Educational Technology Research and Development* 61:365–384.

IBM Corporation 2019. "Solving Mixed Integer Programming Problems (MIP)". https://www.ibm.com/docs/en/icos/12.10.0?topic=optimization-solving-mixed-integer-programming-problems-mip, accessed 27th February.

Jain, R. K., D.-M. W. Chiu, and W. R. Hawe. 1984. "A Quantitative Measure of Fairness and Discrimination". Technical report, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, Massachusetts.

Johnson, N. L., S. Kotz, and N. Balakrishnan. 1994-1995. *Continuous Univariate Distributions*. 2nd ed, Volume 1-2. John Wiley & Sons.

Lee, M. D. and S. Liu. 2022. "Drafting Strategies in Fantasy Football: A Study of Competitive Sequential Human Decision Making". *Judgment and Decision Making* 17(4):691–719.

Lehman, E., F. T. Leighton, and A. R. Meyer. 2015. *Mathematics for Computer Science*. Cambridge, Massachusetts: Massachusetts Institute of Technology OpenCourseWare.

Lohr, S. L. 2010. *Sampling: Design and Analysis*. 2nd ed. Boston, Massachusetts: Brooks/Cole.

National Center for Education Statistics 2020a. "High School Longitudinal Study of 2009". https://nces.ed.gov/surveys/hsls09/, accessed 27th February.

National Center for Education Statistics 2020b. "Postsecondary Education Transcript Studies". https://nces.ed.gov/surveys/pets/, accessed 27th February.

NFL 2024. "The Rules of the NFL Draft". https://operations.nfl.com/journey-to-the-nfl/the-nfl-draft/the-rules-of-the-draft/, accessed 27th April.

Rezaeinia, N., J. C. Góez, and M. Guajardo. 2022. "Efficiency and Fairness Criteria in the Assignment of Students to Projects". *Annals of Operations Research* 319(2):1717–1735.

Rosen, K. H. 2000. *Elementary Number Theory and Its Applications*. 4th ed. Reading, Massachusetts: Addison-Wesley.

Vose, D. 2008. *Risk Analysis: A Quantitative Guide*. 3rd ed. Chichester, England: John Wiley & Sons.

Wu, H., A. J. R. Godfrey, K. Govindaraju, and S. Pirikahu. 2023. *ExtDist: Extending the Range of Functions for Probability Distributions*. R package version 0.7-2.

Zhang, B., C. Bilder, B. Biggerstaff, F. Schaarschmidt and B. Hitt. 2018. *binGroup: Evaluation and Experimental Design for Binomial Group Testing*. R package version 2.2-1.

AUTHOR BIOGRAPHIES

MATTHEW DABKOWSKI is an Associate Professor in USMA's DSE, currently serving as the Deputy Department Head. He holds a BS in Operations Research from USMA, an MS in Systems Engineering from the University of Arizona (UA), and a PhD in Systems and Industrial Engineering from the UA. His email address is matthew.dabkowski@westpoint.edu.

STEPHEN GILLESPIE is an Assistant Professor in USMA's DSE, currently serving as the System Engineering Program Director. He holds a BA and an MA in Mathematics from Boston University and a PhD in Systems Engineering from the Naval Postgraduate School. His email address is stephen.gillespie@westpoint.edu.

IAN KLOO is an Assistant Professor in USMA's DSE and a PhD student in Carnegie Mellon University's (CMU's) Societal Computing Program. He earned a Master in Policy Analytics from CMU and a Bachelor of Business Administration from The College of William and Mary. His email address is ian.kloo@westpoint.edu.

DEVON COMPEAU is an Instructor in USMA's DSE, currently teaching deterministic and stochastic modeling. He holds a BS in Systems Engineering from USMA and an MS in Industrial and Operations Engineering from the University of Michigan. His email address is devon.compeau@westpoint.edu.

MAI TRAN is an Assistant Professor in USMA's DSE, currently teaching decision analysis and directing the EXCEL Scholars Program. She holds a BS in Mathematics from Binghamton University and a PhD in Mathematics from the University of Albany. Her email address is mai.tran@westpoint.edu.