A Web Application for Instructors to Create Maximally Diverse Student Groups Utilizing Integer Linear Programming and Modern Web Frameworks

A Capstone Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Cal Hartzell

December 8, 2022

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Cal Hartzell

Capstone Advisor: Rosanne Vrugtman, Department of Computer Science

A Web Application for Instructors to Create Maximally Diverse Student Groups Utilizing Integer Linear Programming and Modern Web Frameworks

CS4991 Capstone Report, 2022
Cal Hartzell
Computer Science
The University of Virginia
School of Engineering and Applied Science Charlottesville, Virginia USA
crh3mn@virginia.edu

Abstract

Despite the increased prioritization of diversity across higher education, institutions lack robust tools enabling instructors to create maximally diverse student groups. I propose creating a web application that would perform this function automatically based on instructors' specific preferences and students' survey responses. The proposed application would use the Django web framework for the back-end, the Bootstrap web framework for the front-end, and the CVXPY convex optimization problem modeling language for determining maximally diverse groups through integer linear programming. Use of the latter to create student groups would allow for both a more computationally efficient and more diverse generation of groups compared to existing solutions. The use of modern web frameworks ensures efficient development, deployment, and maintenance of the created web application. In addition to use in higher education, future web applications could generate diverse groups for use in primary or secondary schools and workplaces.

1 Introduction

Within higher education, it has been repeatedly shown that increased student exposure to diversity leads to better student outcomes, namely increased social and academic growth, including student identity development, cognitive growth, intellectual engagement, leadership skills, and psychological well-being [1, 2]. As a result, higher education institutions strongly emphasize creating and maintaining a diverse student body, which includes admission efforts for the outreach and recruitment of underrepresented student groups, maintaining an inclusive campus environment, providing support services to students, and ensuring student access to a diverse group of faculty and mentors. While promoting a highly diverse student body is important in increasing student exposure to diversity, promoting frequent interpersonal diversity interactions within the student body is critical to achieving the better student outcomes [2].

Similar to student exposure to diversity, increased group-based learning within higher education has been repeatedly shown to foster better student outcomes, namely increased social, academic, and psychological growth, including fostering student critical thinking skills, cooperation ability, self-esteem, and classroom results [3, 4]. The traditional methods instructors utilize to form groups — random assignment, self-selection, and instructor assignment — all carry significant downsides in the creation of diverse groups. Specifically, random assignment and self-selection almost always lead to groups with low diversity, and instructor assignment to achieve high diversity is extremely time consuming to implement [5].

Given the aforementioned benefits of increased student exposure to diversity and group-based learning within higher education, enabling instructors to easily create diverse student groups would carry significant benefits to students by increasing interpersonal diversity interactions within the student body and encouraging instructors to implement group-based learning strategies, respectively. To enable instructors to easily create diverse student groups, I propose creating a web application to automatically generate maximally diverse student groups based on instructors' specific preferences and students' survey responses. The proposed application would use the Django web framework for the back-end, the Bootstrap web framework for the front-end, and the CVXPY convex optimization problem modeling language for determining maximally diverse groups through integer linear programming. Use of the latter to create student groups would allow for both a more computationally efficient and more diverse generation of groups compared to existing solutions. The use of modern web frameworks ensures efficient development, deployment, and maintenance of the created web application.

2 Related Works

There are a few existing tools that enable instructors to automatically create diverse student groups. The most popular tool available is CATME's (Comprehensive Assessment of Team-Member Effectiveness) web-based software Team-Maker [6]. CATME's research shows that the automatic creation of diverse student groups based on various factors, including students' disciplines, GPAs, prerequisite course grades, and gender, leads to better student outcomes. The proposed web application determines maximally diverse groups through integer linear programming, which compared to Team-Maker's hill-climbing approach would yield more diverse generated groups while being more computationally efficient (i.e., faster). Further, the proposed application makes use of a couple key popular modern web frameworks, which should significantly reduce development, hosting, and maintenance cost of the proposed web application compared to Team-Maker; as a result of this, the proposed web application would also be able to be provided to instructors at a significantly lower price compared to Team-Maker.

An extremely relevant work by Canale et al., (2021) provides an integer linear programming formulation of the maximally diverse grouping problem [7]. The formulation needs to simplified and adapted to work with the proposed web application, but a very strong foundation is provided for implementing a linear programming solution. Where Canale et al. produced a private application only available to MBA Directors within the University of Montevideo in Uruguay, the proposed application would expand upon their research by providing a web application publicly available to instructors implementing an integer linear programming solution.

Further important related work concerns the choice of tools for the proposed application: Django, Bootstrap, and CVXPY. Plekhanova (2009) shows that Django was the best of all back-end web frameworks tested, offering a more powerful underlying architecture, faster application development, and more automatically generated elements when compared to competing frameworks [8]. Gaikwad and Adkar (2019) show that Bootstrap as a front-end framework enables extremely fast and effective web application development [9]. Diamond and Boyd (2016) show that CVXPY is a great convex optimization problem modeling language embedded in Python, allowing programmers to easily express linear formulations in a natural syntax [10].

3 Proposed Project Design

The proposed web application consists of three primary components: a front-end, a back-end, and a maximally diverse group solver. The front-end consists of Django HTML template files utilizing the Bootstrap framework to easily and effectively maintain a quality unified interface throughout the entire web application. The back-end consists of a Django project which could be hosted on a vast variety of cloud services to effectively serve the web application to users through the internet. The maximally diverse group solver consists of a Python program utilizing the CVXPY convex optimization problem modeling language to determine maximally diverse group assignments by modeling and solving integer linear programming problems.

3.1 Front-End Design

Utilizing Django HTML template files for the front-end ensures that the Django back-end can easily and effectively interact with the front-end, which includes populating the web application interface with user specific output and receiving user specific input. The front-end is divided into two major components: the instructor front-end and the student front-end. With the instructor front-end, instructors can create classes to assign groups, assign weighted attributes to classes to be used in determining diversity, send out questionnaires to students to receive student specific responses for each of the assigned class attributes, and have the web application automatically generate group assignments for their classes. With the student front-end, students can submit diversity questionnaire responses for classes assigned by instructors.

3.1.1 Instructor Front-End Design

Unlike students, instructors would need to make an account and login to use the web application, as information would need to persist e.g., instructors would need to be able to always access classes to view questionnaire response progress and to generate group assignments. For each class, the instructor would need to assign various attributes to be used in determining a maximally diverse group assignment, which in turn would be used to automatically generate a student questionnaire. Attribute types include select one from list, select many from list, quantitative continuous, and quantitative discrete. While instructors could input their own custom attributes, the web application would include commonly utilized attributes e.g., graduation year, GPA, gender, race, major, and prerequisite course grades.

After all attributes are determined by the instructor, the web application generates a questionnaire URL which is to be distributed to all students in the class. Before group assignments are generated, the instructor must weight the importance of each attribute to group diversity, which can be done qualitatively (e.g., high, medium, or low) or numerically. After having the web application generate maximally diverse groups, the instructor can reassign weights before regenerating group assignments and see the effect on group diversity by manually modifying any undesirable individual group assignments.

3.1.2 Student Front-End Design

The student front-end is much simpler than the instructor front-end, consisting only of the generated class questionnaire accessible through the instructor distributed URL. The student is presented with the relevant class's questionnaire, and provides a response for each of the included attributes before submitting their response. It's also important the student submits relevant identifying information set by the class instructor e.g., a name or student ID, so that the instructor can easily determine which students are assigned to each group and track individual student questionnaire responses.

3.2 Back-End Design

By developing the front-end utilizing Django HTML templates, creating a Django project for the back-end is extremely simplified, making it easy for the back-end to populate output on the front-end and receive input from the front-end for the web application. The back-end follows Django's standard model-template-views (MTV) architecture, enabling features critical to efficient development including automatic object-oriented database interaction and automatic web application administration. The other critical advantage of utilizing Django for the back-end is that it enables simple and scalable deployment of the web application; for example, Heroku can automatically deploy and host Django projects, automatically configuring necessary resources such as the database with practically no developer input required.

3.3 Maximally Diverse Group Solver Design

The maximally diverse group solver is implemented in Python utilizing the CVXPY convex optimization problem modeling language. Initially, as the Django back-end is also implemented in Python, the solver is implemented directly into the Django project requiring no developer configuration and connection of the solver; later, if the web application reaches significant scale, the solver would likely need to be implemented as its own microservice, callable by the Django back-end. The solver determines the maximally diverse group assignments for a given class by modeling and solving with CVXPY an integer linear programming problem formulation.

3.3.1 Integer Linear Programming Problem Formulation

The following is adapted and simplified from Canale et al.'s (2021) integer linear programming problem formulation [7]. Consider the following variables:

- N: the number of students;
- G: the number of teams;
- M_{min}: the minimum number of students per team (inclusive);
- M_{max}: the maximum number of students per team (inclusive);
- d_{ij}: the distance between students i and j, determined using the normalized and weighted Euclidean distance in regards to the instructor chosen class attributes;
- x_{ij}: the binary decision variable such that $x_{ig} = 1$ if and only if student i is assigned to group g, $x_{ig} = 0$ otherwise.

Consider the following linear programming problem formulation:

$$\max_{x_{ig}} \sum_{g=1}^{G} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} x_{ig} x_{jg} \quad (1)$$

$$\text{s.t.} \sum_{g=1}^{G} x_{ig} = 1, \forall\, i \in \{1, \ldots, N\} \quad (2)$$

$$M_{min} \leq \sum_{i=1}^{N} x_{ig} \leq M_{max}, \forall\, g \in \{1, \ldots, G\} \quad (3)$$

$$x_{ig} \in \{0, 1\}, \forall\, (i, g) \in \{1, \ldots, N\} \times \{1, \ldots, G\} \quad (4)$$

The goal objective function (1), is to maximize the sum of intragroup diversity of all teams, which for each team is simply the distance-sum among all student pairs in that team. Constraint (2) states that each student is assigned to exactly one group, constraint (3) states that each group has between M_{min} and M_{max} students, and constraint (4) defines the binary domain for the decision variable set x.

4 Anticipated Outcomes

While publicly releasing this web application would marginally increase the diversity of student groups under instructors already utilizing tools to automatically create diverse student groups e.g., CATME, the real benefit would come from instructors adopting this web application who currently lack access to such tools. As a specific example, CATME currently charges ~$2.75 per student per year [11], and as a result of this prohibitive cost at UVA fewer than 20% of instructors have access to CATME. With the web application's utilization of modern web frameworks and an efficient group assignment algorithm, costs for development, hosting, and maintenance should be significantly lower compared to existing tools, which should greatly improve instructor access as it can be provided at a comparatively significantly lower cost. Given the aforementioned link between student exposure to diverse groups and better student outcomes, publicly releasing this web application and significantly increasing instructors' ability to create diverse student groups would significantly better student outcomes overall.

5 Conclusion

Given the evidence of increased student exposure to diversity and group-based learning bettering student outcomes within higher education, as well as the lack of available robust instructor tools for diverse student group creation, I propose a web application enabling instructors to easily and automatically create maximally diverse student groups. By utilizing modern web frameworks and an efficient integer linear programming group assignment algorithm, the proposed web application would carry significantly lower development, hosting, and maintenance costs compared to existing instructor diverse student group creation tools. These lower costs should greatly improve instructor access to automatic diverse student group creation tools as soon as the proposed web application is publicly released. This would significantly increase instructors' ability to create diverse student groups and significantly improve student outcomes overall.

6 Future Work

The most impactful future work would involve fully developing and publicly releasing the proposed web application. Work could also be done to expand application use outside of higher education, including use in primary or secondary schools and workplaces. The proposed web application could also be expanded with additional functionality: a schedule attribute could be included, allowing students to indicate what times of the week they are available, only allowing students into the same group if their schedules have an instructor-determined amount of overlapping availability. Attributes could also be expanded to include those in which instructors favor grouping of alike student characteristics (e.g., instructors preferring that students from specific groups or organizations be grouped together).

Additionally, the proposed web application could be integrated with existing instructor management tools (e.g., Canvas, Google Classroom, and Blackboard Learn) to increase instructor access and ease-of-use. Upon the full development and public release of the proposed web application, further work could be done analyzing the effects of creating diverse student groups emphasizing specific student attributes.

7 Acknowledgments

A special thanks is extended to Wenxi Huang and Ethan Steere for helping in the development of the idea for this research project, who as a team could possibly in the future fully develop and publicly release a web application enabling instructors to easily and automatically create maximally diverse student groups, drawing from some of the research presented here.

References

[1] Patricia Gurin, Eric Dey, Sylvia Hurtado, and Gerald Gurin. 2002. Diversity and higher education: Theory and impact on educational outcomes. *Harvard educational review* 72, 3 (2002), 330–367. https://doi.org/10.17763/haer.72.3.01151786u134n051

[2] Nicholas A. Bowman. 2013. How Much Diversity is Enough? The Curvilinear Relationship Between College Diversity Interactions and First-Year Student Outcomes. *Research in Higher Education* 54, 8 (2013), 874–894. https://doi.org/10.1007/s11162-013-9300-0

[3] R.B. Hilborn. 1994. Team learning for engineering students. *IEEE Transactions on Education* 37, 2 (May 1994), 207–211. https://doi.org/10.1109/13.284996

[4] Marjan Laal and Seyed Mohammad Ghodsi. 2012. Benefits of collaborative learning. *Procedia - Social and Behavioral Sciences* 31 (2012), 486–490. https://doi.org/10.1016/j.sbspro.2011.12.091

[5] Donald R. Bacon, Kim A. Stewart, and Elizabeth Scott Anderson. 2001. Methods of Assigning Players to Teams: A Review and Novel Approach. *Simulation & Gaming* 32, 1 (2001), 6–17. https://doi.org/10.1177/104687810103200102

[6] Richard A Layton, Misty L Loughry, Matthew W Ohland, and George D Ricco. 2010. Design and validation of a web-based system for assigning members to teams using instructor-specified criteria. *Advances in Engineering Education* 2, 1, Article 1 (2010), 28 pages.

[7] Eduardo Canale, Franco Robledo, Pablo Sartor, and Luis Stábile. 2022. Solving the Max-Diversity Orthogonal Regrouping Problem by an Integer Linear Programming Model and a GRASP/VND with Path-Relinking Approach. *Symmetry* 14, 1, Article 18 (2022), 13 pages. https://doi.org/10.3390/sym14010018

[8] Julia Plekhanova. 2009. Evaluating web development frameworks: Django, Ruby on Rails and CakePHP. Institute for Business and Information Technology (2009), 20 pages.

[9] Suraj Shahu Gaikwad and Pratibha Adkar. 2019. A review paper on bootstrap framework. *IRE Journals* 2, 10 (2019), 349–351.

[10] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research* 17, 1 (2016), 2909–2913.

[11] Purdue University. 2022. How much does CATME cost? Retrieved from https://info.catme.org/instructor-faq/how-much-does-catme-cost