Automated features for improving the existing First-Year Seminar Assignment Program

John Chu

Submitted in extended fulfillment of student-faculty collaborative research for the Department of Computer Science and Mathematics Dickinson College, 2022-2023

Professor Dick Forrester, Supervisor

17 May 2023

Abstract

Every first-year student at Dickinson College is required to take a first-year seminar. The summer before those students arrive, they each select six seminars from all seminar choices. Each student is then assigned to a seminar from their list. It used to be manually processed, and Dr. Richard Forrester and Thanh T. To devised mathematical techniques to assign first-year students to seminars. These mathematical techniques contributed to balancing the gender ratio and student-type ratio of the students in the classroom, and they were implemented via a Mosel program that Dr. Forrester created. This research aims to enhance the performance of the current program by generating various versions and evaluating each upgraded version to identify the most effective program that can successfully achieve its initial research objectives.

Acknowledgments

I wish to convey my utmost gratitude to Professor Dick Forrester for his exceptional knowledge and invaluable guidance in the field of operations research and computational methods which was crucial in the completion of my research. Additionally, I would like to express my gratitude to Than T. To for initiating the research and collaborating with Professor Forrester to develop mathematical techniques that ensure a balanced gender and student-type ratio of Dickinson students in the classroom. Lastly, I am grateful to Dickinson College for granting me access to the First-Year Seminar and the relevant student data.

Conclusion

We have developed three versions of Python programs that improved the original program developed by Dr. Forrester for assigning Dickinson first-year students to First-Year Seminars. The first version converted an original Mosel assignment program to a Python program using the Gurobi solver, and we discovered that the Python Gurobi solver provided better optimal and utopian points, smaller local optimal points. We then enhanced this feature by successfully normalizing the scale of objectives precisely by including nadir points in computing normalized objectives as the original program did not include nadir points. As Thanh To's Honors research with Dr. Forrester indicated, nadir points were necessary to normalize the scales between three objective functions, which were rank, gender, and citizenship. The results displayed slightly more balanced outcomes in terms of gender and student-type distribution in the classroom. However, both of those implementations were hindered by long program run times. Therefore, we tackled this problem by linearizing the nonlinear gender and citizenship objectives. We accomplished this by replacing the quadratic sum-squared differences method to the absolute value of the differences between males and females and U.S. and international students in the classroom to construct gender and citizenship objective functions. To linearize the nonlinear absolute value function, we used a fair allocation technique introduced in operations research studies. Due to the linearized gender and citizenship objectives, the empirical analysis of these linearized objectives result in taking approximately one second to run which is a significant improvement compared to the first two versions of the program which took around five minutes to execute.