Genetic Algorithm Optimization of Teams for Heterogeneity

P. K. Imbrie
Department of Engineering Education,
University of Cincinnati,
Cincinnati, USA
imbriepk@ucmail.uc.edu

Jutshi Agarwal
Department of Engineering Education,
University of Cincinnati,
Cincinnati, USA
agarwaji@mail.uc.edu

Gibin Raju
School of Education,
University of Cincinnati,
Cincinnati, USA
rajugm@mail.uc.edu

Abstract— **This work in progress study aims at developing a system to designate teams to relatively large groups of students in a classroom setting. It is motivated by recent changes in the ABET Criterion 3 accreditation guideline that requires students to demonstrate "an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives." Outside of accreditation guidelines, team-based learning is becoming the prescribed pedagogical tool to enhance student collaboration and prepare them for professional work environments upon graduation. One important factor in effective team-based learning is to ensure teams are balanced in their abilities and characteristics across team members. Demographical memberships of students can also play an important role in team dynamics and impact the overall success in content learning. Instructors usually assign teams to students by manually looking into their abilities from previous grades and other performance factors. This can then become tedious when aiming to maintain uniform heterogeneity in classrooms of students coming from diverse backgrounds of knowledge, skills, ethnicity, and gender.**

While some studies have presented the use of computer-aided tools, visualization categorization, and other Artificial Intelligence techniques, this work proposes the use of the Genetic Algorithm (GA) to form teams optimized for heterogeneity. The Genetic Algorithm presented uses a discrete integer-based chromosome representation and groups alleles to represent each team. Standard GA operators enable the algorithm to be adapted to any optimizing criteria deemed fit by the instructor. The algorithm presented in this work uses self-reported competency data on 3 different computational skills for 1300 students enrolled in a first-year engineering design thinking course divided into over 20-course sections of strength 40-60 each. Net team scores for each computational tool are calculated and the variation across each skill minimized by the fitness function for individual sections. Constraints based on gender and ethnicity are applied to minimize demographical imbalance between teams. The algorithm is tested for all sections in the Fall 2019 cohort to give consistent team configuration. Discussions on the stability and validity of configurations generated are discussed. Future work include methods that use a fuzzy logic decision-making tool for multiple criteria optimization.

Keywords—genetic algorithm, team formation, engineering education

I. INTRODUCTION

Recent changes in the ABET Criterion 3 [1] accreditation guideline require students to demonstrate "an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives." Outside of accreditation guidelines, team-based learning is becoming the prescribed pedagogical tool to enhance student collaboration and prepare them for professional work environments upon graduation**.** More and more teachers are using small group learning in the classroom for project-based learning pedagogy. There is evidence to show that collaborative learning exhibits improved performance from individual learning in terms of problem-solving, conceptual understanding, and mental retention [2]. Collaborative learning involves dividing students into groups to work together on classroom activities to facilitate common learning goals [3].

An important issue that comes with implementing collaborative learning is a team assignment. Instructors commonly use self-selected teams, randomly-assigned teams, and instructor-assigned teams in classrooms. While some research suggests that students have the best team experiences using self-assigned teams [4], [5], [6], self-selection can also lead to excessive homogeneity [7] such that the teams lack diversity [6], [8] and necessary skills required for the team's task [5]. Random assignment may not improve issues with self-assignment and also increase concerns about fairness [6]. Instructors can implement criteria based team assignment which can improve student outcomes in instructor-assigned teams [9]. Despite clear advantages, logistical challenges inhibit teachers from implementing criteria-based team assignments [6], [10]. Several computer-aided team formation methods have been implemented to assist instructors in assigning teams using course-relevant criteria [11], [12], [13], [14], [15]. The present work proposes the use of a genetic algorithm to form teams in a first-year engineering course at a large R1 university. It is to be noted that this work provides an algorithm to optimize the formation of teams under certain constraints and is not intended to measure the success of teams formed under those constraints over others. Previous work using similar tools has been discussed followed by the theory of a genetic algorithm. The method section discusses the classroom set-up and implementation of the algorithm before preliminary results are presented.

II. Previous Work

In addition to the computer-aided team formation tools developed by [11], [12], [13], [14] literature points to several methodologies of building teams using AI tools like Genetic Algorithms [15], Fuzzy Logic, Genetic Fuzzy Systems, and Group Genetic Algorithms. Furthermore, Paredes et al. [2] developed the Faraway-so-close Algorithm and the TOGETHER tool to use visualization techniques to form teams that are uniformly heterogeneous using instructor based criteria. Uniform heterogeneity aims at creating groups that are equally heterogeneous to support the diversity of competence and skills with minimum inter-group variance in net skills. Wang et al. created DIANA, a genetic algorithm-based system that showed evidence of higher satisfaction levels and group outcomes than traditionally assigned groups [16]. In DIANA, authors use similarities in shapes (representing groups formed on a 2D space) to generate a report for instructors based on psychological variables. In a chapter dedicated to forming student teams in classrooms, Mutingi and Mbohwa used grouping genetic algorithms [17] where groups within each potential solution were manipulated, unlike a traditional genetic algorithm which has been explained below.

A. *Genetic Algorithms*

Genetic Algorithms are aimed at solving optimization problems using concepts of natural evolution. Theories of evolution like natural selection, mutation, "survival of the fittest" are interpreted mathematically and implemented to find near–optimal solutions in a vast search space.

In optimization problems, potential solutions are populated to form a generation in a one-time step called *initialization*. Each potential solution is represented by a series of "alleles" or "genes" to form a "chromosome." This is explained in Figure 1. The "fitness" of each solution is calculated by an objective function relevant to the problem at hand.

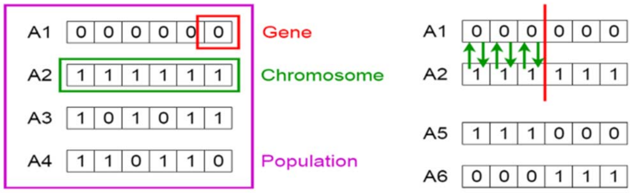

Fig. 1. Basic structure of a genetic algorithm solution space (Left fig. shows gene, chromosome, and population, right fig. shows the process of recombination happening at the red line)

Following are the typical operators used in the genetic algorithm:

Selection

Selection is a stochastic parameter that defines which parents are selected to contribute genes towards the reproductive progression. Using their fitness values towards the optimization function, each parent is assigned a certain probability position of selection for gene propagation. Then a random number is chosen to select the parent that lies above that random threshold.

Recombination

Recombination refers to the operator that enables parent genes to combine in a specific pattern to generate a new combination of genes for a child chromosome. In figure 1, Chromosome A1 and A2 recombine to exchange genes at the red line and result in A5 and A6.

Mutation

A mutation is a stochastic operator that implements the reproductive phenomenon when a child gene undergoes a random change with new traits different from either parent. See figure 2.

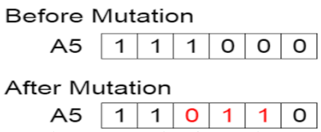

Fig. 2. Example of Mutation

Elitism

Elitism refers to the implementation of the *survival of the fittest* theory in evolution. The best chromosomes from both parents and child population are selected and propagated to the next generation, rather than using the weaker genes from either of the generations.

Genetic algorithms have been used to find solutions to NP-hard problems like the traveling salesman problem where the best solution is hard to find but a "good enough" problem is acceptable. As such, there is no ideal solution in an NP-hard problem because the ideal solution has not been found yet. The genetic algorithm just provides a good enough solution. The following section describes the implementation of the algorithm in our case which is also an NP-hard problem.

III. Methodology

In this paper, a genetic algorithm, inspired by grouping genetic algorithms, is used to assign teams to students in a first-year engineering course. MATLAB was used to implement the algorithm.

A. *Course structure and data*

The first-year engineering course, ENED 1100 – Engineering Design Thinking – I at the University of Cincinnati is the first of a two-part course with a student enrollment of about 1300-1400 students. Students are split into 25-28 sections with a strength of 40-60 students in each section. Students engage in a flipped classroom set-up to engage in basic design concepts and engineering fundamentals through active learning. A flipped classroom set-up implements non-traditional learning elements in that recorded lectures, instructional videos, and other remotely-accessible pedagogical resources are used before class to introduce fundamentals of a new concept. Subsequently, class-time is spent on activities involving applied-learning through complex problem solving with peer interaction [18]. Team-based learning is an integral part of this course, where students primarily spend class-time problem-solving on multiple computational tools like LabVIEW, Excel, and Python, with their teams, facilitated by instructors and four teaching assistants. The instructor-to-student ratio is 1:12. Other topics covered are design thinking, project management, spatial visualization. The teams also engage in two design-centered projects including a robot development using Lego EV3- Mindstorm systems. The course is computational tool intensive and so it is imperative that teams are organized with fairness in the context of computational skill diversity.

At the commencement of the course, students answer an 80-item inventory that reports self-assessed competency of basic programming knowledge (SPK), Excel Knowledge(EK), and Computer Knowledge(CK). Each inventory item is based on a Likert scale of 0-9. The skills are isolated for each student and normalized on a 0-1 scale. In addition, gender and ethnicity data are combined with the normalized competencies. Responses from the Fall 2019 cohort was selected post-facto as a test-case for this study. The data is de-identified and students are numbered from 1 to N_i, where N_i is the strength of section *i*. The objective is to split N_i students into teams of 3-4 such that:

- each team has diverse levels of skillset
- inter-team variation is minimal
- women and students belonging to underrepresented minorities are not isolated in any team

B. Clustering and data visualization

The data is observed using a *fuzzy c-means* algorithm [19] [20] where each skill is represented on an axis of a 3-D frame. Since the maximum strength of a team is 4, 4 clusters are formed to visualize the diversity of skills in each section. Using this visualization, it can be determined if the spread of skills are spread out in a way that each point in a cluster is a representative member of the cluster in a team containing 4 cluster representatives each. Using this tool is optional and independent of the genetic algorithm.

C. Genetic Algorithm Set-up

As mentioned earlier, the student data is deidentified and students are numbered 1, 2, 3, ……. N_i for a class-strength of N_i. A chromosome contains alleles representing each student so that N_i alleles form a chromosome. The sequence of the alleles is important because the first 4 alleles represent team 1, the next 4 alleles represent team 2, and so on.

Sample Allele of 20 students in 5 teams (students placed randomly for demonstration):
13 7 10 18 6 2 12 16 15 9 11 4 14 3 19 17 8 1 5 20

TABLE 1: TEAM CONFIGURATION OF A SAMPLE ALLELE OF STUDENTS CHOSEN RANDOMLY

Team no	**Team members**
Team 1	13 7 10 18
Team 2	6 2 12 16
Team 3	15 9 11 4
Team 4	14 3 19 17
Team 5	8 1 5 20

After visualization, the system begins by identifying number of the 4 people teams and 3 people teams based on the number of students = N_i. Based on the modulus of the total number of students with 4, maximum teams will have 4 members, rest 3. For example, in a class of 55 people, 13 teams of 4 (n_4), 1 team of 3(n_3) are formed for N_{tot} = 14. This can be represented by the following:

$$N_{tot} = n_4+n_3 \tag{1}$$

$$N_i = 4*n_4 + 3*n_3 \tag{2}$$

The genetic algorithm begins with initialization of population and generating M random permutations between 1 and N_i, where M is the population strength of the genetic algorithm

Fitness Function – Skills Distribution

The fitness function is based on the uniformly heterogeneous team formation concepts. The net skill of each team is calculated by summing up-skill scores for all 4 (or 3) members. The standard deviation between the inter-team net skill score is calculated for each skill separately. The standard deviations for each skill are added to form the fitness function. This fitness function is minimized by the GA to find the most uniformly distributed team configuration.

Fitness function calculation algorithm:

```
For each skill,  i = 1 to 3
        For each team, j = 1 to $N_4$
                For each member, k = 1 to 4
        $Sum\_skill_{i,j} = Sum\_skill_{i,j} + member\_skill_{i,k}$
                End
        End
        For each team j = $N_4$ + 1 to $N_{tot}$
                For each member k = 1 to 3
                $Sum\_skill_{i,j} = Sum\_skill_{i,j} + member\_skill_{i,k}$
                End
        End
                $Std\_dev_i = sqrt( □ Sum\_skill_{i,j} /(N_{tot}-1))$
End
        Fitness function = □$Std\_dev_i$
```

Constraints of Genetic Algorithm: Race and Gender

The ideal team formation includes teams where underrepresented groups are not isolated however mixed-gender teams are maximized. This is to assist in a diverse learning environment. Research suggests that while mixed-gender groups promote learning for all involved [21], a female member in a team can feel dominated when placed singularly against all men [22]. Similarly, for ethnicity, members of marginalized communities can feel singled out against peers of a non-minority group [23] [24]. Team configurations, generated by the GA are penalized by adding higher value to the fitness function, whenever a team consists of isolated underrepresented groups. The genetic algorithm is then iterated over several generations using selection, recombination, mutation, and elitism operators. For each generation, a roulette-wheel implementation selects whether a parent participates in recombination [25]. In recombination, exchanges take place between two members of adjacent teams while in mutation, a single individual is randomly swapped with another team. Elitism combines parent and child population to sort and select the best candidates in both. Thresholds for each process are determined on a trial-and-error basis as is typical of a traditional genetic algorithm.

IV. RESULTS

A. *Visualization*

After looking at visualization results for several sections of the class (Figure 3), it was noticed that the students were not distributed into clusters with clear boundaries. The strength of each cluster was also not uniform. Any algorithmic manipulation using results of clustering would be tedious and as inefficient as random initialization.

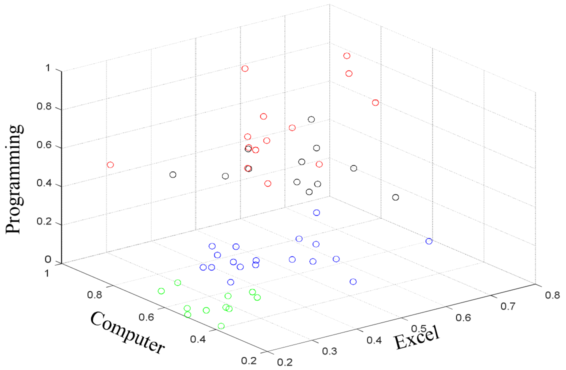

Fig. 3. Sample results of clustering visualization for one section (each cluster represented by a different color)

B. *Genetic Algorithm*

As a test case, the genetic algorithm of section 002 with a class strength of 47 students was run. Out of 47 students, 14 were female and 3 were non-white. Figure 4 below shows the fitness function history over 200 generations and it is seen that the genetic algorithm converges to a near-optimum solution halfway and then continues to optimize. The standard deviation across teams for excel knowledge was 0.4326, across computer knowledge was 0.2847, and across programming knowledge was 0.4739. Each skill standard deviation was less than 0.5 on a 0-1 scale which is quite satisfactory. $N_4 = 11$; $N_3 = 1$ which implies that the last 3 students are in one team while all other teams have 4 students.

The final chromosome generated by the algorithm was:

29 32 47 20 11 26 5 24 35 25 9 21 16 33 37 1
45 23 8 41 46 40 10 17 30 36 31 12 38 22 13 18
2 15 34 14 27 43 28 7 39 6 4 42 44 3 19

The chromosome above denotes the configuration of teams in section 002 where students (29,32, 47,20) should be in team 1, (11, 26, 5, 24) in team 2 and so on. The genetic algorithm takes between 90-140 secs to run for 200 generations and about 270 secs for 500 generations.

V. DISCUSSION AND FUTURE WORK

The genetic algorithm results for 1 out of 25 sections have been shown in Figure 4. It was observed that the GA was able to converge to acceptable values of inter-team skill variation easily and kept optimizing the fitness criteria over several generations. In this case, any value that drops below 50 was considered acceptable because it showed evidence that no constraint (of race and gender) had been violated. With such results, this work-in-progress paper was able to show the preliminary results of an automated tool that can systematically assign teams to a cohort of 1300 students in less than an hour. Without an algorithm-generated configuration, each of the 25 instructors would manually assign teams to about 60 students requiring several man-hours to accomplish a task every semester. This tool shows promising results in creating a flexible system that is robust enough to assign teams based on several instructor selected criteria. The use of standard GA operators and simple logical steps in coding allows for the implementation of algorithms in multiple situations and programming platforms with minor adjustments.

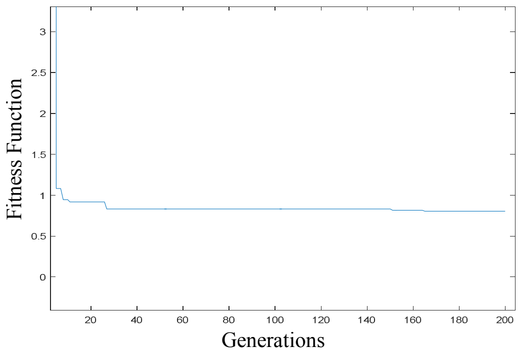

Fig. 4. Optimization of Fitness Function over GA generations

The tool however comes with a few limitations which will be addressed in future work. One of the challenges of developing team formation algorithms is that there are no single optimizing criteria. This makes validation of results challenging [16]. In this case, another group of students in section 011 was used and compared against manual assignment by faculty who used similar constraints and scores. A comparison of the sum of the standard deviations of the three skills between the two methods showed a difference of 0.018, a fairly close level of optimization considering the manual assignment took hours while the GA converges in less than 5 minutes.

TABLE 2: COMPARISON OF GA AGAINST MANUAL ASSIGNMENT

Skill	Standard deviation between the net skill of teams assigned manually	Standard deviation between the net skill of teams using GA
Excel	0.249862	0.2601
Computer	0.267991	0.2648
Programming	0.364576	0.3757
Sum	**0.882428**	**0.9006**

A second challenge with using the Genetic Algorithm is results are dependent on the initial condition of initialization and prone to get stuck in local minima where constraint violation cannot be overcome, or fitness function does not minimize to near-optimal conditions. A robust genetic algorithm can be developed using initialization conditions from the clustering algorithm. Additionally, in this research, a single optimizing criterion was used to minimize the standard deviation of 3 skills by a linear combination. A multi-optimization criterion can be implemented by a heuristic classification using Fuzzy Logic in the fitness function. Future work will implement these characteristics.

VI. References

[1] Engineering Accreditation Commission of ABET, "Criteria for Accreditation," ABET, Baltimore, MD, 1997.

[2] P. Paredes, A. Ortigosa and R. Rodriguez, "A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization," *Journal of Universal Computer Science,* vol. 16, no. 19, pp. 2882-2901, 2010.

[3] S. A. Kalaian, R. M. Kasim and J. K. Nims, "Effectiveness of Small-Group Learning Pedagogies in Engineering and Technology Education: A Meta-Analysis," *Journal of Technology Education,* vol. 29, no. 2, 2018.

[4] D. J. Neal, Group Competitiveness and Cohesion in a Business Simulation, 1997.

[5] J. A. Mello, Improving Individual Member Accountability in Small Work Group Settings, 1993.

[6] D. R. Bacon, K. A. Stewart and W. S. Silver, Lessons from the Best and Worst Student Team Experiences: How a Teacher Can Make the Difference, 1999.

[7] D. S. Jalajas and R. I. Sutton, Feuds in Student Groups: Coping with Whiners, Martyrs, Saboteurs, Bullies, and Deadbeats, 1984.

[8] C. Kirchmeyer, Multicultural Task Groups: An Account of the Low Contribution Level of Minorities, 1993.

[9] B. Oakley, R. M. Felder, R. Brent and I. Elhajj, "Turning student groups into effective teams," *Journal of Student Centered Learning,* vol. 2, no. 1, pp. 9-34, 2004.

[10] R. Decker, "Management team formation for large scale simulations," in *Developments in Business Simulation & Experentìal Exercises*, J. D. Overby and A. L. Patz. Statesboro, Eds., Statesboro, GA, Association for Business Simulation and Experiential Learning, 1995.

[11] D. R. Bacon, K. A. Stewart and E. S. Anderson, Methods of Assigning Players to Teams: A Review and Novel Approach, 2001.

[12] M. A. Redmond, "A computer program to aid assignment of student project groups," *SIGSCE Bulletin,* vol. 33, no. 1, p. 134–138, 2001.

[13] R. Cavanaugh, M. Ellis, R. Layton and M. Ardis, "Automating the process of assigning students to cooperative-learning teams," in *American Society for Engineering Education Annual Conference & Exposition*, Salt Lake City, Utah, 2004.

[14] R. A. Layton, M. L. Loughry, M. W. Ohland and G. D. Ricco, "Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria," *Advances in Engineering Education,* vol. 2, no. 1, pp. 1-28, 2010.

[15] T. R. Henry, "Creating effective student groups: An introduction to groupformation.org," in *Proceeding of the 44th ACM technical symposium on Computer science education*, Denver, 2013.

[16] D.-Y. Wang, S. S. Lin and C.-T. Sun, "DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups," *Computers in Human Behavior,* vol. 23, pp. 1997-2010, 2007.

[17] M. Mutingi and C. Mbohwa, "Multi-Criterion Team Formation Using Fuzzy Grouping Genetic Algorithm," in *Grouping Genetic Algorithms: Advances and Applications*, J. Kacprzyk, Ed., Springer International Publishing, 2017.

[18] N. Goedhart, N. B.-v. Westrhenen, C. Moser and M. B. M. Zweekhorst, "The flipped classroom: supporting a diverse group of students in their learning," *Learning Environments Research,* vol. 22, pp. 297-310, July 2019.

[19] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences,* vol. 10, no. 2-3, pp. 191-203, 1984.

[20] S. Panda, S. Sahu, P. Jena and S. Chattopadhyay, "Comparing Fuzzy-C Means and K-Means Clustering Techniques: A Comprehensive Study," *Advances in Computer Science, Engineering & Applications,* vol. 1, no. 166, pp. 451-460, 2012.

[21] J. Schnittka and C. Schnittka, ""Can I drop it this time?" Gender and Collaborative Group Dynamics in an ‘‘Can I drop it this time?’’ Gender and Collaborative Group Dynamics in an Engineering Design-Based Afterschool Program Engineering Design-Based Afterschool Program," *Journal of Pre-College Engineering Education Research,* vol. 6, no. 2, 2016.

[22] Academy of Management, "How Adding a Female's Voice Can Improve All-Male Teams," *Summary - Topics - Diversity,* pp. 1-2, 2019.

[23] United Nations Department of Economic and Social Affairs, "Chapter VII : Indigenous peoples and ethnic minorities: marginalization is the norm," in *Report on the World Social Situation 2018*, New York, United Nations, 2018, pp. 97-108.

[24] B. Souza and K. Neubeck, "Marginalized Voices in Eugene: Report on Focus Groupsin Eugene's Communities of Color, Muslim and LGBTQ Communities," Eugene Human Rights Commission, Eugene, Oregon, 2017.

[25] S. Mirjalili, Evolutionary Algorithms and Neural Networks, Brisbane, Australia: SpringerLink, 2018.