Title
Experiments in curriculum development, collaboration technology and social dynamics to streamline learning.

Author
Hosein, Nicholas

Publication Date
2021

Peer reviewed|Thesis/dissertation

Experiments in curriculum development, collaboration technology
and social dynamics to streamline learning.

By

NICHOLAS HOSEIN

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____

Andre Knoesen, Chair

____

Rajeevan Amirtharajah

____

Lee Michael Martin

Committee in Charge

2021

Contents

Dissertation Abstract

The aim of this research was to: 1) Explore how we can decentralize teaching, reducing dependency on the oracle at the blackboard and instead structure around peer-peer teaching. 2) Determine the pros / cons of this decentralized approach through experiments. 3) If effective, propose how to scale these methods so they can be used outside of a controlled experimental setting. In part 1 we discuss some initial work on developing an in-person curriculum centered around peer-peer learning. In part 2 we look at modifications and improvements as they apply to remote learning. In part 3 a communication tool called KarmaCollab is developed to automate the collaborative processes. In part 4 we look beyond the classroom at using technology to create learning networks that grow beyond the boundaries of the university.

1 Fostering Entrepreneurship Through Targeted Adversity: A Senior Design Case Study

1.1 Abstract

Engineering education has the potential for significant social and economic impacts through entrepreneurship. In this regard, most engineering curriculum falls short in three critical areas, thereby limiting its effectiveness. Firstly, students are often indoctrinated into the "specific approach to solve a problem" mindset when in fact real world technical problems are dynamic, nuanced and most importantly, can be solved many ways depending on the resources and expertise at hand. Secondly, aversion to risk is a side effect of the university grading system as thinking outside the textbook, trying new things and failing are typically not rewarded. These limitations exist in large part due to the difficulty of grading or ranking intellectual work outside of established boundaries. Finally, and most importantly, social intelligence is taught less, both implicitly and explicitly, than technical knowledge, especially in STEM fields. The ability to communicate, arbitrate and resolve tense social situation with empathy is as or more important than book knowledge when it comes to success in both entrepreneurship and industry as a whole. This paper outlines one methodology for entrepreneurial focused courses in engineering with the end goal of boosting students success in an existing company or with their own startup. This is accomplished using a highly social course format with gradually increasing assignment ambiguity, adversity and complexity while having fall backs and redundancy for predictable progression of the class as a whole. In our case study course, students design, assemble and test from-scratch IoT electronic products which are then entered into a university wide startup competition. A survey is created to determine students confidence in various areas related to success post graduation, either working in industry or starting their own venture. On average, 57% of students responded that the new format has advantages over other courses they are currently taking, with 28% reporting no difference and 15% indicating the opposite.

1.2 Introduction

Engineering professions face challenges requiring competence in uncertainty, problem solving creativity and the ability to balance conflicting demands with high level perspectives in mind. Rather than train students to 'know' things, they should be trained to 'understand' things[1]. To be effective in their industry, this requires not just a solid technical foundation but also skill in human relations, enabling them to function, both autonomously and simultaneously as part of a large team. Unfortunately, the standard method of teaching is centered around traditional lecture style formats with minimal to no social interaction, creative thinking or open ended challenges. Research into instructional strategy show that when courses are designed around actively engaging students in the material, levels of understanding, retention and transfer knowledge are increased compared to lecture centric formats[2]. The popular flipped learning format, which advocates at home learning via online media with in class group exercises, has demonstrated benefits such as increased learning gain, flexibility, increased interaction, improved professional skills, and increased student engagement[3]. Despite these findings, mainstream engineering educators are hesitant to try new formats, partly as resistance to change but mostly due to the ambiguity and uncertainty of how to create an effective course with limited resources and personnel[1 4]. There is also concern about scalability, as more dynamic and interactive assignments often require significantly more setup and faculty workload as enrollment increases[5].

Another step further, beyond simply being part of an existing organization or company, is the need for fostering new ventures through creative design combined with understanding market needs and opportunities. It is well established that economic growth of a country lays heavily on the entrepreneurial ability of its people[6,7,8]. Standard educational practices are designed to be scalable, yes, however at the risk of not exposing students to the ambiguity and risk necessary for an entrepreneur to be successful. It is not enough that engineering students can solve problems they have seen before, they must know how to approach new problems that are to become relevant as societal and technological landscapes change over time. The majority of entrepreneurship education revolves around case studies[9] or guided role play[10]. While both important as part of a learning experience, it is secondary to preparing students socially and emotionally. In this paper we start by reviewing previous work, explaining the new course structure followed by presenting survey results and concluding remarks.

1.3 Related Work

Increasing the efficacy of teaching methods, with focus on generating more self sufficient, self engaged students has been a hot topic of recent decades as these traits are linked to a students future success in the workplace. Lord et al.[2] split two large biology classes into groups, one following the traditional teacher-centered format, the other a student-centered constructivist format[2]. The latter format, which aims to coherently embed new information into a students existing map of knowledge, was shown to improve average exam scores by more than 5% when compared to the former. The flipped method, also known as inverted learning, has been gaining popularity in STEM (science, technology, engineering, math) fields with 29% of higher education faculty reported in progress of implementing it[11]. Flipped learning aims to shift direct instruction from the classroom to home, thereby opening class time to dynamic, interactive learning experiences[12]. Karabulut-Ilgu et al.[3] flipped a transportation engineering course and used questionnaires and class video recordings to show students had a positive view toward the change. The more broadly defined, blended learning method combines face-to-face interaction with online tools in a general sense. In order to better teach entrepreneurial skills to students, Sidhu et al.[10] incorporated a mock startup company course which takes students from concept to low tech demo. By shifting focus away from the time consuming technical details, more teamwork, self-reflection, and inductive learning could be taught. In a very different approach Weaver et al.[9] used a series of case studies of existing startups to give students a more holistic view of what it takes to bring an innovation to market. These case studies were given over many courses as a supplement to the existing curriculum. Somewhat in between the two former methods, Jarrar et al.[13] formulated a case studies based course which also includes an elevator pitch and product

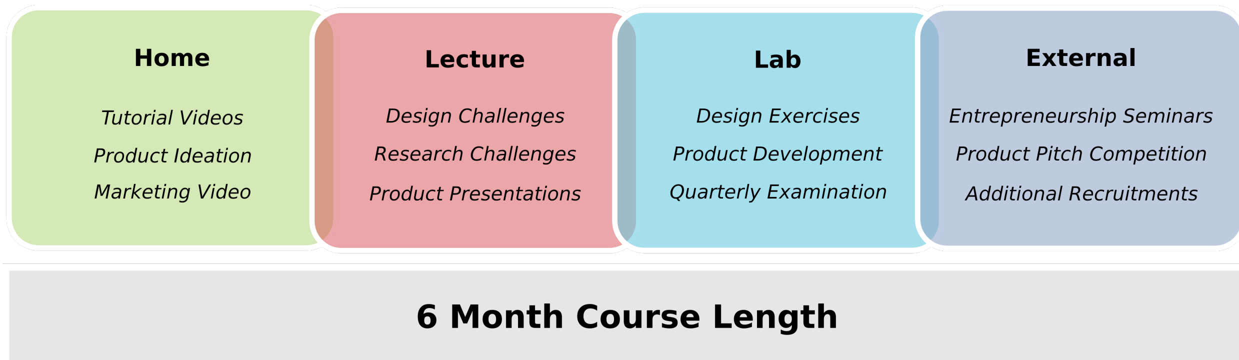


Figure 1: EEC 136 course overview after implementing the proposed changes. The course is subdivided into 4 distinct time periods, at home, in lecture, in the laboratory and at externally hosted events.

formulation but no prototyping.

1.4 Course Structure

The teaching approach explored in this research deviates from the traditional delivery of STEM courses. Instead of making small incremental changes year by year, a complete redesign is performed and the aggregate result collected. Most changes are based on prior research in education, and hence, have independent support. In this research the overarching motifs used when developing the curriculum are as follows:

1. *Exploration*: Encourage risk and failure as this is part of working on real world problems. Make a safe environment for testing new ideas and solutions to problems.
2. *Socialization*: Give students opportunities to engage socially, with their peers, future users of their product, professors in the field, and industry professionals.
3. *Independence*: Increase the difficulty and level of independence students / teams are expected to handle. By the end teams should be fairly self sufficient.

An embedded systems senior design course at the University of California, Davis is selected for the entrepreneurship focused redesign. The goal is to guide students through the creation of a real internet of things (IoT) product. This includes designing a printed circuit board, assembling and testing by hand, modeling and 3D printing the enclosure and programming the firmware.

Course time is divided into four periods, at-home, in-lecture, in-lab and external to the course. At home, students are given links to tutorial videos designed specifically for the course. These are intended to replace traditional lecture material. They also brainstorm product ideas to pursue as a class project and when their startup prototype is completed, a marketing video is created to showcase their company. Lecture periods are designed to be interactive and students work together to complete various group technical challenges. This time is also used for presenting progress on their startup project. Lab time consists, at first of more controlled design exercises, and later full product development. A final examination is administered in lab time at the mid and endpoint of the course. Finally, a set of entrepreneurship seminars, partnerships and resources are provided externally to the course itself. In the following sections, our three themes of exploration, socialization, and independence are used to expand on how exactly course time and activities are designed around fostering skills that help students perform in industry or as an entrepreneur.

1.4.1 Exploration

For students to feel comfortable making mistakes and trusting their creative capacities they need to be allowed to fail and learn from those failures. Often times lab exercises and homeworks imply specific solutions and punish for not following the expected sequence. Other times problems are defined in a closed, highly constrained form which does not lend much to exploration.

To balance the need for creative risk without complicating grading, in-lecture challenges and in-lab design exercises are made to have clear and simple visual indicators of success. Examples of visual indicators are light, sound, console text, waveforms and debugger output. This not only allows for consistent grading but also gives students the creative freedom to approach problem any way they would like.

In addition, in-lab design exercises are constructed to have cross-use between teams. Cross-use of designs is implemented as a marketplace by which teams can use past implementations from other teams in current lab exercises. Given each lab builds and connects to the previous, students can worry less about taking a risk and having it fail. In addition, it also requires that teams understand how to deconstruct others implementation for incorporation.

1.4.2 Socialization

Creating a sense of unity within the class is important for the free communication of ideas, leading students to ask each other for help while reserving teaching staff time for more pressing issues. By the end of the course, each student will have been part of 25+ randomly assigned teams for a variety of short and long term tasks. The types of tasks that use randomized teams are in-lecture challenges and presentations, in-lab exercises, and at-home product ideation. The only assignments which do not use randomized teams are final examinations and final project teams. Final project teams are determined by student interest in a certain startup project idea.

Since lecture material is delivered in the form of online video tutorials, lecture time can be spent on two types of in-lecture challenges. A research challenge involves teams searching for answers to a mix of open and close ended technical questions. One individual in the group is then chosen at random to represent the entire group. This promotes both learning and teaching as everyone in the group has a vested interest in the entire team knowing the answers. A design challenge involves the team programming a solution to a hypothetical problem. The first teams to finish are given the highest score after which they separate and help other teams complete before time runs out. This gives students the opportunity to learn socially and take on both roles as mentor and mentee. In-lab design exercises are week long and more in depth compared to the one hour in-lecture design challenges. Students are encouraged to exchange ideas and brainstorm across teams, however work submitted must be original.

Once the class has performed product ideation and voted on final projects ideas, teams are allowed to expand outside of the course boundaries. This includes partnering with a computer science department senior design team to co-found the company, taking on partnerships with existing companies, or working in coordination with a researcher at the university. Students may also recruit individuals via startup mixers hosted by the university or incorporate acquaintances that share a passion for the product idea.

1.4.3 Independence

Until now most students have not experienced the level of independence and responsibility that will be required of them. To ease the process, the intensity is increased over the duration of the class. At stage one, students complete the at-home video tutorials. Each video is kept brief and focused so it can be easily referenced in the future. At stage two, research and design in-lecture challenges are incorporated. At stage 3, they are given in-lab design exercises which describe a final result and students are expected to use the resources at hand (video tutorials and internet) to complete. At stage 4, a three part individually taken midterms is administered. Unlike a traditional midterm there are very little instructions, students are given a high level design objective and it is their responsibility to use the resources at hand to solve it. At stage 5, they are officially assigned to a startup project and begin to engage with future customers (which they seek out), and modify their product design accordingly. They also start to build and pitch their design to judges at a startup competition. All teams are offered the opportunity to continue work on their product after completion of the course for university credit. The expectation is that at this point teams are fairly self sufficient.

1.4.4 Boost / Penalty

A new concept is introduced called the boost. Unlike extra credit, boosts, if not achieved, can lower a students overall final grade. Boosts are a measure of students performance above and beyond and creates an competitive environment. There are two situations in which boosts are used. First is to reward for finishing design exercises one or two days early for a 20% or 40% boost respectively (and an equal penalty for one and two days late). This pushes students to start early as well as helps trickle down design tips to struggling groups. On average, 26% and 28% of students submitted their assignment one and two days early with the incentive. Boosts are earned for taking business seminars, recruiting additional members to their team and making progress through a university wide startup competition. Of the 7 teams, 5 received boosts for participating in a university startup competition with one team placing.

1.5 Course Results

As an experiment, the boost was removed from in-lab design exercises during the second half of the course. Without the boost it was observed that students would not start the lab, often times, until the day before. As a result more students underestimates how long the lab would take, resulting in late submissions (26% late vs 14% with boosts). When the boost was present, it was noticed that some students would start as early as 7 days before the due date, right after the previous lab was submitted. There were also large degrees of self organization in that students, without the instruction of teaching staff, created and organized out of class meet ups to work. A facebook chat was put together with (according to the students) everyone in the course so multiple teams could meet and work together, as well as for exchanging ideas about design exercises.

In order to understand students sentiments towards the unique course structure, a survey was administered. Eight questions are selected to provide feedback on a number of desirable learning attributes. The questions are listed below and prefaced with “Compared to other courses taken this quarter, senior design helped you ...”. Students rate each item from strongly agree to strongly disagree with other courses they are currently taking as baseline.

1. learn skills relevant to your future job in industry
2. learn conflict resolution in your team and otherwise
3. learn to troubleshooting complex systems
4. learn resources management (time, people, money)
5. tackle ambiguous problems more confidently
6. improve your general communication skills

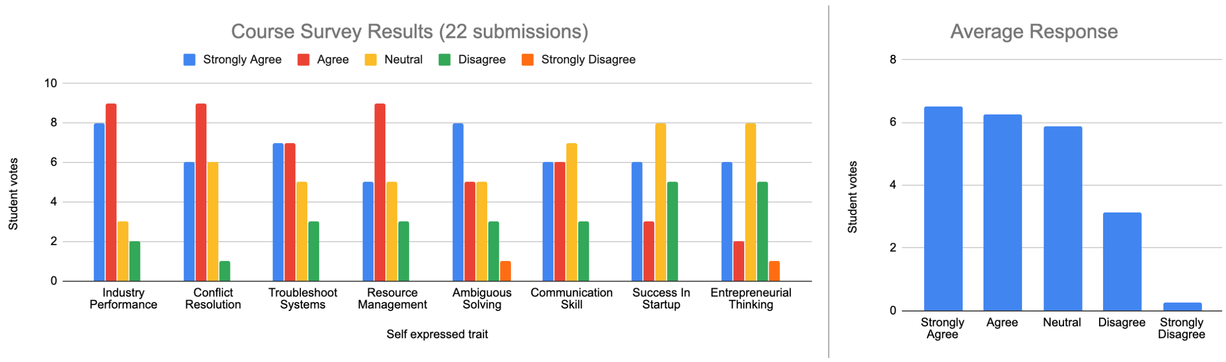


Figure 2: EEC 136 students are surveyed at the conclusion of the course to determine if the curriculum changes resulted in a net positive or negative effect on various skills.

7. have skills that will help you succeed in a startup

8. become more confident with entrepreneurial thinking

The response to all questions are net positive with average response showing strong agreement that students perceived the new course format as having benefits. The plots are sorted from highest (left) to lowest (right) net response. Overall, 58% of students reported strong agree or agree over all 8 questions with 15% reporting strong disagree or disagree, indicating that the majority of the class found the new curriculum advantageous compared to other courses currently taken. 77% (vs 9%) of students reported that their experience with senior design provided skills more relevant to their future jobs when compared to other courses currently taken and 68% (vs 5%) of students thought the course gave them better conflict resolution skills. 64% (vs 14%) found they were better at troubleshooting complex systems and managing resources, 59% (vs 18%) felt they could tackle ambiguous problems more confidently, 55% (vs 14%) reported better communication skills, 41% (vs 23%) felt more likely to succeed in a startup, and 36% (vs 27%) thought they had better entrepreneurial thinking.

1.6 Development and Execution

The course is executed with one professor and three assistants for 28 students over 6 months. The design of the course took significantly more time than the execution. A total of 13 video tutorials are recorded, each covering specific technical aspects of the course. Videos are designed to be concise, stream of consciousness, with both overt and sub communicated ideas. For example, when teaching how to design a circuit board the video will go through the motions, click by click without discontinuities, sometimes being unsure and visiting google or explicitly referencing commonly used documentation. Lack of discontinuities means the student can be assured that everything will work if followed exactly, putting them at ease. By acting less like an oracle and more like a student, the videos convey technical information as much as they do how to use the internet. Each video is kept about 10 min so it is easy to reference. Between brainstorming, scripting, re-recording and video editing, the commitment was about 6hrs/video. Once the course is in progress, however, time investment significantly reduces compared to the average senior design course. Strong community bond results in questions being directed to peers, this along with lack of formal “chalk and talk” lecture shifts the image of the teacher away from information source and towards mentor. This allows staff to spend more time conveying nuanced technical points and wisdom on inter-personal skills, and less time repeating basic information. The overall effort put into the course, to our estimate, is not more than any other with the exception that the effort is front loaded. The re-use of the tutorial videos lend to a mid to long term benefits depending how quickly underlying technical software becomes obsolete.

1.7 Conclusion

This research has demonstrated that applying fundamental changes to the way students interact with each other as well as with the teaching staff can have an impact on future job or entrepreneurial skills. Incorporating a strong social environment, with room for mistakes and messy exploration gives students the opportunity to work on being ok pushing forward without knowing the answers. The motivation of the boost points adds a new dynamic to the classroom and is an additional tool in the toolbox for educators. Changes in education are desperately needed for the future of society. Here a novel curriculum was proposed and shown to have a positive impact on a range of skill sets that are often neglected in a traditional course format.

2 Concurrent-Asynchronous Communication Patterns for Remote Learning Classes

2.1 Introduction

To aid in the transition to a virtual classroom environment, foundational updates to communication patterns are tested. Case study and data presented demonstrate support from students, reduced workload for staff and encouraging learning outcomes. The base assumption, of which this work relies on, is that reducing the frequency and quality of communication between students and teaching staff slows the learning process. The discussion revolves around three concepts; concurrency, asynchrony and vetted communications. (1) Concurrency describes the presence of independent groups of students collaborating simultaneously. (2) Asynchrony refers to the ability of a student to get assistance without idle waiting due to fixed-time availability of teaching staff. (3) Vetted communication is the establishment of standards by which mentor-mentee relationships result in net positive progress towards a learning goal. These three concepts are borrowed from the computer science field. Concurrency and asynchrony are means to increase the bandwidth or 'amount of data' communicated at a given time over a certain transportation medium like fiber optic or wireless. Vetting is more akin to error correction, ensuring the message sent was the message received.

First, let us look at the traditional in-person approach and how applying that model to a virtual environment changes patterns of communication. (1) From a concurrency standpoint, in-person naturally fosters befriending and self organized study groups, asking questions to neighbors during lecture, discussion cross talk and impromptu group collaboration. The transition to online video based learning limits these kinds of organic social exchanges. (2) Any teacher centered approach will be by its nature synchronous. Students must wait for office hours or lab times to get help. Even then, they wait in line for assistance as teaching staff time is divided among students. Moving this to a virtual environment is worsened given the awkwardness of the platform itself, with dropped audio and unintuitive setup of virtual rooms. (3) Vetting is automatic in the traditional model as teachers are the gold standard of knowledge.

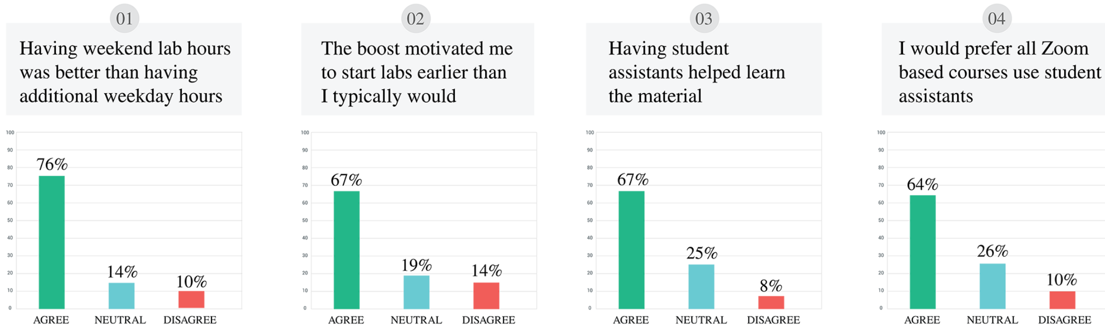


Figure 3: Student survey responses. (Zoom refers to the video conference software used)

2.2 Methods

Students are introduced to the concept of Boost, which is a reward for going above and beyond what is required in the class. In this case study course, students are told that 20 boost points raise their final grade by one percentage point. In other regards, traditional course components remain unchanged, including lectures, lab hours, office hours, written midterms and a final project. Course modifications and additions are summarized below:

1. Early Submission Boost - to motivate students to start early, they are offered 2 boost point per day up to 5 days for checking off labs before the official due date.

2. Student Assistant Boost - if a student gets a lab verified by teaching staff early, they qualify to be a student assistant at 5 boost points per 3 hour lab section.

3. Weekend Lab Sections - to give more consistent, ongoing help, lab sections are held 7 days a week. Boost-qualified student assistants can help mentor up to 2 lab sections per week.

4. Mentorship Priority - Students needing help are divided across virtual rooms, each with a student assistant. Student assistants engage teaching staff directly if stuck.

(1) The introduction of student assistants increases concurrency as the number of teaching staff is effectively expanded. (2) Daily lab times reduces the maximum delay to get help from 72 hours to 24. While this is not asynchronous, it is an improvement. (3) Labs are vetted as student assistants must successfully complete the lab before assuming the role. In the event a student assistant

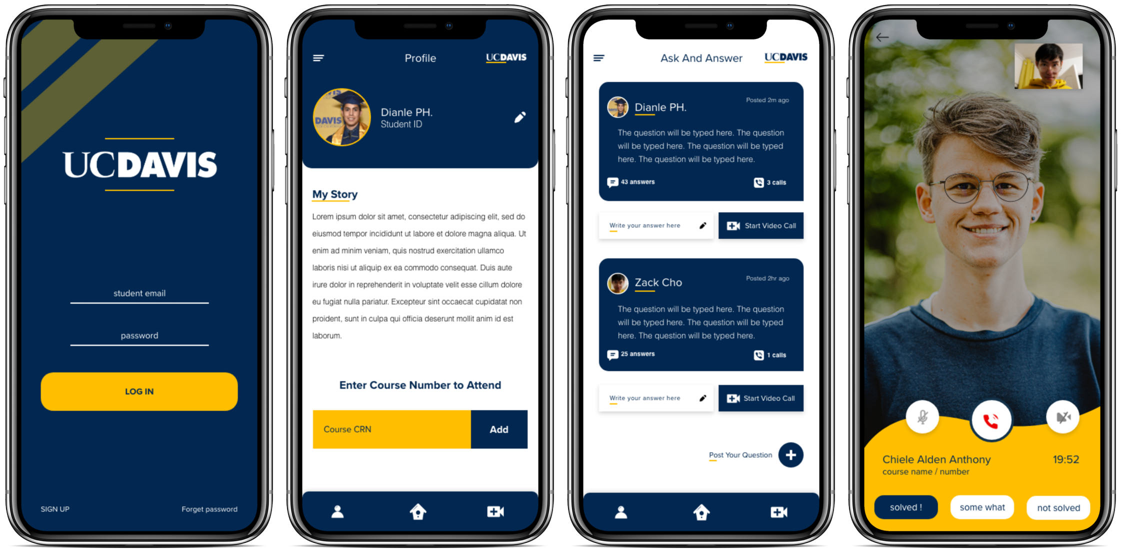


Figure 4: The UC Davis Boost App (visual prototype).

cannot help, teaching staff are always on hand to support. In addition to these changes, hints and tips video guides are recorded for labs as a convenience for students.

2.3 Results

An anonymous survey was administered to determine student sentiment towards the changes. Of the 65 students in the course, 63 participated. The main results are summarized in Figure 3. Of the students in the course, 25% were student assistants at least once during the quarter, with 8% participating weekly. The median number of early credit accumulated over the course across students is 10 days. From the standpoint of the teaching staff, it is agreed that having student assistants helped significantly in managing the high volume of questions, most of which are repetitive or have a simple, straight forward answers. During the final lab section, there was an interesting case of a student assistant coming in to help, knowing that they had already participated twice for that week (the maximum). They expressed that they enjoyed helping their classmates so much that the points were not necessary. While not explicitly tested, there is anecdotal evidence that the boost system (early submission and student assistant) had a positive impact on student self esteem and community.

2.4 Next Steps

While successful, the boost method, as implemented has certain limitations. Firstly, students cannot get help instantly. They must wait for a lab time to engage with a student assistant or teaching staff. Secondly, teaching staff must expend time to manage the boost system itself in terms of assigning assistants to student groups and tracking points. Thirdly, there is no way to connect students to mentorship outside of the course itself. The Boost App by UC Davis creates a closed social network consisting of students enrolled in a course, alumni of the course and teaching staff. The following user case demonstrates how the app could work.

1. Myriam is stuck. This is her first circuit design course and she can't figure out why Lab 3 Part B is not flashing the light. She posts to the app with 3 potential outcomes:
    (a) In under 15 minutes Myriam is contacted by her classmate Damion who happily helps her via a face to face video call. Damion gets 35 minutes of boost credit.
    (b) Nobody in her class has reached that far in the lab. Chou, an alumni of the course, volunteers to help Myriam for boost points towards his advanced circuits class.
    (c) It has been 1 hour and nobody has responded or been able to successfully answer the question. The app notifies Dinah, who is on the teaching staff, to help.

The advantage of a virtual course is that communication can be fully asynchronous and on demand, there is no need to be in a physical room at a specified time. Video calls are best with 2-4 people not 30+, hence breaking up interactions is desirable and good for concurrency. By supplementing a virtual course with the Boost App, student feedback can be tracked so vetting can be performed, and current students can be connected with qualified mentors in a timely manner. The app as described here can be easily incorporated into any course to foster a sense of community in the virtual classroom, while also creating positive learning outcomes for students and reducing cost and complexity for staff.

3 Promoting socialization and collaboration through a course management application.

3.1 Introduction

The COVID-19 pandemic has forced universities to transition to a fully online format, resulting in a renewed interest into how technology can aid learning while physically apart. While many courses can easily transition to video streaming, others such as STEM laboratory classes, require hands-on training, and as a result, experienced significant hurdles with the remote learning switch. In this paper, the impact of an internally developed smartphone application called KarmaCollab is evaluated alongside the incorporation of socialized teaching and course gamification. We will look at UC Davis Electrical and Computer Engineering laboratory courses and the impact KarmaCollab had on the online course format. The relationships between course grades, KarmaCollab app engagement, student self-reported sentiment via an end-of-quarter survey, and teaching staff interviews are presented to showcase interesting remote learning insights.

At the start of 2020, university students, staff, and faculty faced the unforeseen challenge of transitioning to a fully online curriculum due to the COVID-19 shelter in place order. Although fully online course formats are nothing new, university courses are traditionally built around an in-person experience. One area that thrives from an in-person format is STEM laboratory courses. From chemical mixtures in a controlled lab setting to constructing circuits with the assistance of a laboratory Teaching Assistant (TA), STEM laboratory courses teach hands-on experience that students may not obtain elsewhere. Along with the lost opportunity to learn in-person, fully online courses have requirements such as reliable internet access, a suitable studying environment, and strong self-motivational skills. These factors make online learning particularly challenging.

Historically, online courses are a more affordable option for students looking to further their education. Before taking an online course, however, students must assess whether their skills are on par with the requirements of an online course. Such skills range from strong self-motivation to comfort navigating a computer. Transforming an in-person course to fit the standard of an online one can provide a jarring experience for students who are not mentally prepared for the transition as online lacks the level of socialization and intimacy compared to its in-person counterpart. To help navigate these challenges, online tutoring platforms (i.e. Chegg or Quizlet), chat group applications (i.e. Discord or Slack), and instructional platforms (i.e. Khan Academy) are all useful for supporting a remote learning environment. KarmaCollab was developed at the University of California, Davis (UC Davis) as an experimental platform to test new ways of utilizing technology to streamline social learning and advance the remote experience for students and staff in STEM courses.

3.2 Related Work

Research is plentiful on the topic of integrating technology into the classroom. Pilgrim et al.[14] discussed that using technology such as smartphone or web apps, provides educators the ability to engage students, foster higher-level thinking and develop problem-solving skills that align with today's technological society. Brindley et al.[15] furthers this discussion with work on creating collaborative learning groups in an online environment. Their findings show a correlation between participation in small group collaboration and deeper learning, development of learning, and teamwork skills. Collaboration was found to create an increased sense of community for the learner, thus increasing satisfaction and retention. Sanders et al.[16] brings up an interesting point that students often may not be equipped with an adequate level of readiness to collaborate in an online format and often see colleagues as rivals. Building community is essential to establish, and with the help of online resources and instructional guidance, a healthy dynamic can be created in the remote classroom. The flipped method, also known as inverted learning, has been increasingly researched in STEM fields. Flipped learning aims to shift direct instruction from the classroom to home, opening class time to flexible, interactive learning experiences. Karabulut-Ilgu et al.[3] applied the flipped method to a transportation engineering course. Students expressed a positive sentiment to the change, saying they enjoyed the flexibility, pacing and felt like they understood the material more at the end of the course. Blended learning is another method that combines the

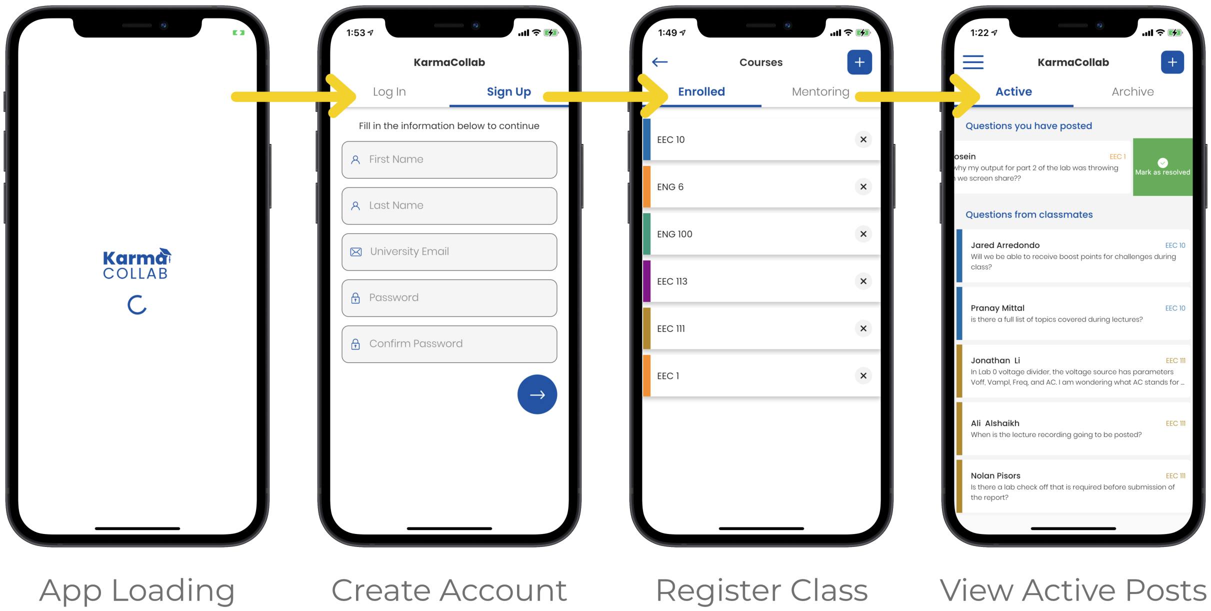


Figure 5: KarmaCollab onboarding of students.

strength of face-to-face learning with online tools. Garrison and Kanuka et al.[17] proposes ways that blended learning can be used in higher education to enhance in-person learning with technology. The KarmaCollab research presented here is fundamentally different than the prior work in that by creating our platform, from scratch, in-house, we afford a higher degree of flexibility on what collaborative learning theories can be tested in the classroom.

3.3 KarmaCollab Overview

University students have access to a smartphone and high-speed internet as part of an assumed requirement to participate at a university. Apps have entered the education space that helps students and teaching staff more efficiently disseminate information and streamline communication. This overlay of technology and education brings fresh opportunities for students to learn and faculty to teach. KarmaCollab is an experimental platform that allows us to analyze technology's effect on student learning and behavior. Figure 5 shows the onboarding process for students on KarmaCollab. They download the app, create an account using their university email, register for their classes, and contribute to the discussion.

Posting and Answering

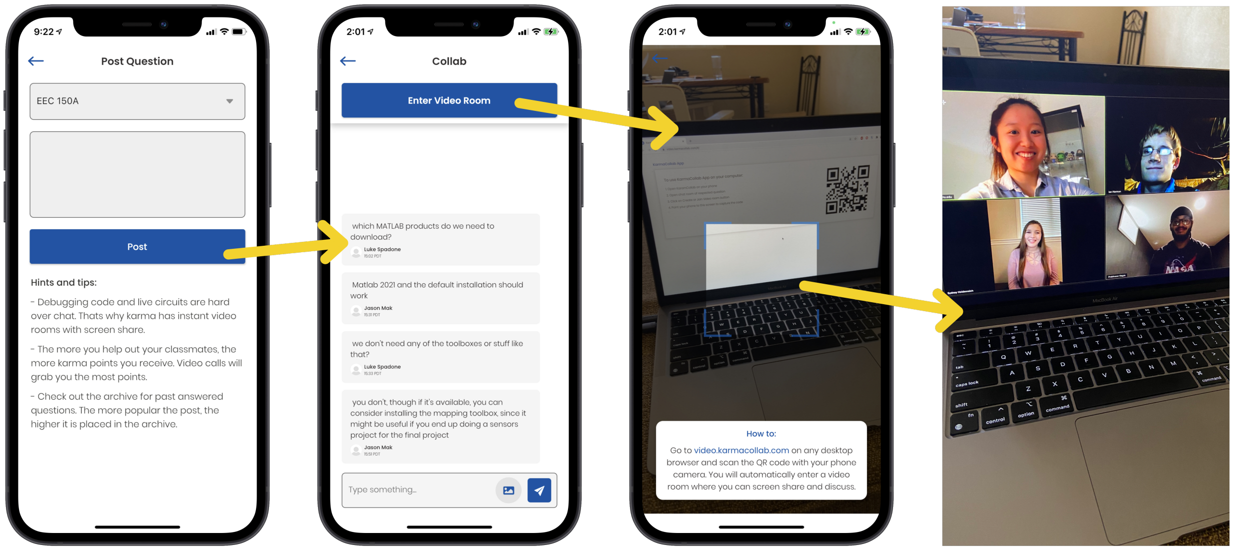
KarmaCollab is intended to resemble a social media platform like Twitter and less similar to a group chat platform like Slack. The app keeps questions to a short life cycle, being posted, answered, and then migrated to the archive section of the app, where they expire after not being viewed for an extended time. Figure 5 shows the active post screen where questions and discussions from different courses are color-coded. Each post on KarmaCollab is its own dedicated space with text chat, image posting, and video room. Posts can be marked resolved by the question poster or expired if left untouched for a long time.

Instant Screen Share

Video chat has become a useful replacement for face to face discussion during remote learning. One of the issues faced with many video chat services (like Zoom) is that jumping on spontaneous calls can be overly complicated. It usually involves creating a call, sharing a link with participants via email, and scheduling a time to talk. With KarmaCollab, the expectation is that students that want to screen share or talk face to face over video could jump into a room in a matter of seconds. Figure 6 shows the flow of posting a question and joining the video room to screen share or talk. At the top of every post in the active tab of KarmaCollab is a button that will launch a QR code scanner. The video chat automatically launches once the companion web app is scanned from any browser (no login required). KarmaCollab video rooms have no capacity limits and allow for screenshare, which is used extensively in project courses involving simulations and coding.

Self Managed Archive

KarmaCollab tries a new approach to archiving posts. Other platforms such as Slack, Piazza, and Blackboard allow for an infinitely long archive of all questions posted, sometimes even from past instances of the course. KarmaCollab uses a model more like Twitter, where trending archived topics bubble to the top of the archive, and posts that have lost relevance are expired and deleted.



Post A Question   Chat Over Text   Launch A Room   Talk/Screenshare

Figure 6: KarmaCollab getting questions answered.

The expectation is that reduced clutter and more intelligent sorting will make the archive more useful to most students. If a question is expired, a student can always ask again as an active question, although it is expected that this will be infrequent.

Leaderboard and Tracking

The app automatically tracks three metrics that are used to determine a students ‘karma score’ for a class. This karma score places students in a class-wide leaderboard which gives recognition to students who are interacting with their peers. The first metric tracked is passive engagement in the app as a viewer of posts. The second is engagement interacting on different posts through text (over a minimum character length) or uploaded images. Finally, how many live video calls they have participated in with teaching staff or students. Students that are actively helping vs. being helped also get a slight advantage when it comes to karma points in general. The more a student engages with their peers, the more karma points they receive, the higher up they see themselves moving in the leaderboard. A screenshot of the leaderboard is shown in Figure 7.

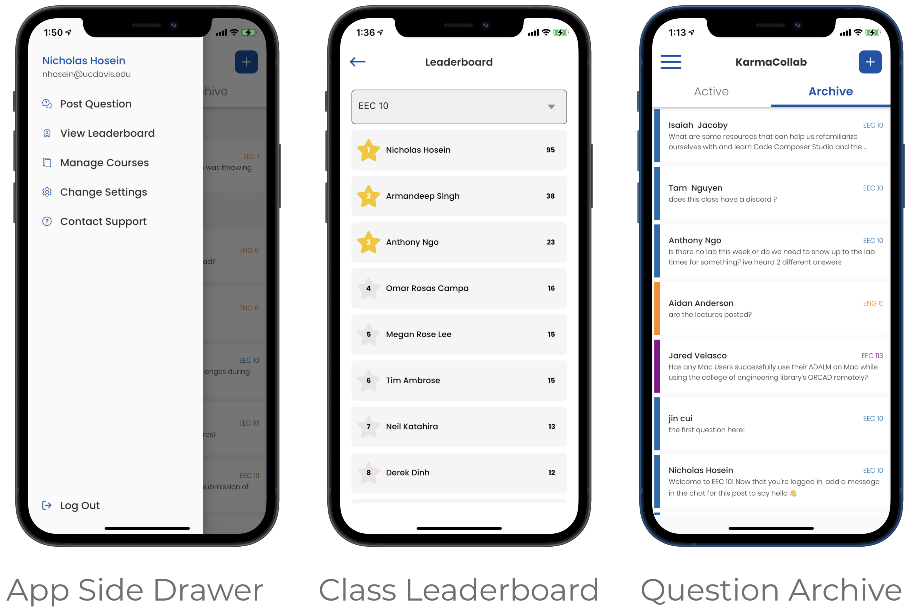


Figure 7: KarmaCollab special features.

Primary Case Study Setup

Practitioners in the STEM field must possess competent problem-solving skills. There may not always be a single step-by-step solution to a problem, thus it is essential to guide learners in STEM to understanding a topic rather than just memorize answers. The path to understanding does not only require analytical skills but social skills. Laboratories, traditionally used in the STEM fields, lean towards in-person, hands-on learning in a lab setting to provide easier collaboration among students. Learning through teaching others, or socialized teaching, sparks deeper learning of a subject as an individual brainstorms ways of teaching others rather than only focusing on understanding the material. This case study involves comparing KarmaCollab to its baseline. Both accomplish the same fundamental dynamic of social teaching, but KarmaCollab is more automated and technology-driven.

EEC 10 - Winter Quarter (Baseline)

EEC 10 Winter Quarter 2021 was a hands-on introductory course to analog and digital circuits where 64 students built a sound following robot using a microcontroller and some basic analog circuits. Students are introduced to the concept of 'boost' which is a reward for proactively participating in the class. In the baseline case study course, students are told that for every 20 boost points earned, they raise their final course grade by one percentage point. If they receive all available boost points available, they get a +10% total grade boost. Course modifications and additions are summarized below.

1. Early Submission Boost - to motivate students to start early, they are offered two boost points per day for up to 5 days to check off labs before the official due date.
2. Student Assistant Boost - if a student gets a lab-verified by teaching staff early, they qualify to be a student assistant at five boost points per 3-hour lab section.
3. Weekend Office Hours - to give more consistent, ongoing help, additional virtual lab hours are held on the weekend.
4. Mentorship Priority - Students that need assistance are divided across virtual rooms, each with a student assistant. Student assistants engage teaching staff directly if they are not able to help a student under their mentorship. This structure provides a benefit to the students getting help and reinforces the concepts learned by the student assistant.

In other regards, traditional course components remain unchanged, including lectures, lab hours, office hours, written midterms, and a final project.

EEC 10 - Spring Quarter (KarmaCollab)

EEC 10 Winter Quarter 2021 was a hands-on introductory course to analog and digital circuits where 70 students built a sound following robot using a microcontroller and some basic analog circuits. Like the baseline course, students are introduced to the boost concept for being proactive. They are given the same +10% final grade shift incentive. The course was run identically to the baseline course other than the exceptions noted below.

1. Student Assistant Boost - At the end of the quarter, the highest-ranking student on the KarmaCollab leaderboard received 25 boost points, the most available. The second highest contributor received 24, so on and so forth.

2. Flexible Lab Section - The first 30 minutes of the lab section are held as usual, with the remaining 3 hours 30 minutes continuing on KarmaCollab. TA's perform a checkoff of a lab exercise after the student is done. Checkoffs are scheduled via KarmaCollab, on-demand when students are ready as opposed to during fixed periods like lab time or office hours.

3. On-Demand Office Hours - Two TA's are dedicated to monitoring KarmaCollab outside of lab section times. Students can schedule times to video chat or checkoff via the app.

4. TA Specialization - Instead of each TA performing all tasks (labs, office hours, grading etc) in Spring, each TA took up a specialty. One TA did all of the grading, two took on-demand office hours, one helped run the course, and one created lecture challenges.

5. Group Challenges - Instead of having a lecture during the lecture period, students were randomly broken into groups for challenging group assignments. Lectures are pre-recorded and viewed at home. A challenge could be to create a circuit in a group and show the output or to research an engineering concept and prepare the group to answer questions.

Foundational Theory Overview

The current social and technological landscape is observed, from Facebook, Twitter, Zoom, Piazza, TikTok, etc. The KarmaCollab platform is designed around what currently engages the 18-22 year undergraduate demographic. The preference for using commercial applications as a foundation is due to the lack of custom-developed platforms in academia available to base this work on. In addition, many of the best academic works on the topic are published in the moderate

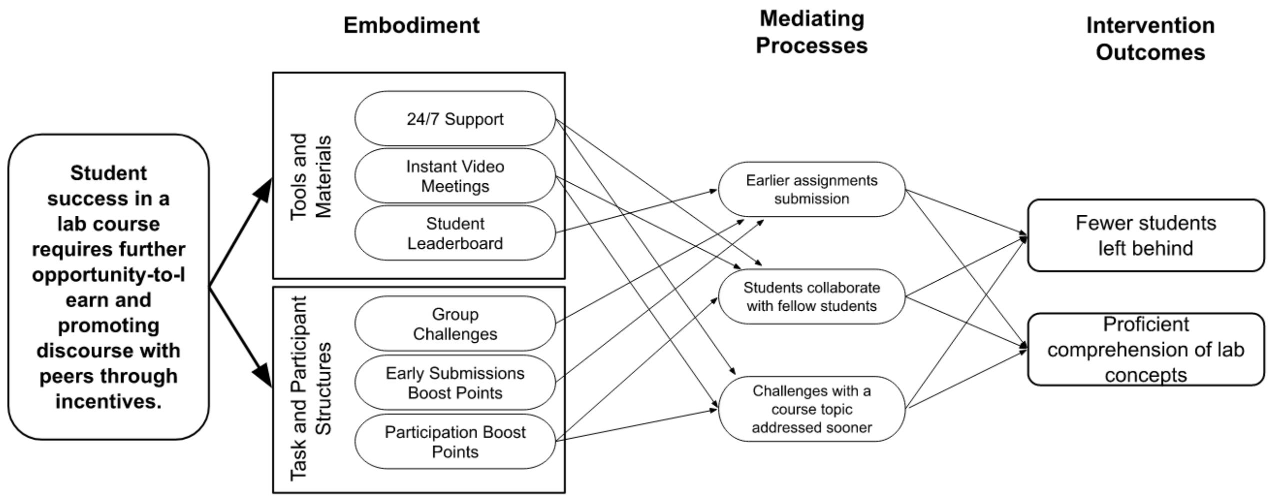


Figure 8: This conjecture map shows from left to right, the assumed theory of aiding learning, the tangible embodiment of how the course was altered in response, the mediating processes expected to result from that embodiment, and finally the measurable outcomes for students.

to distant past, making them less relevant when looking at technology-driven by modern day social trends. Instead, academic theory is used to influence how these current-day ‘engaging’ features can be utilized to support education. This research is based on work by del Rosario et al[18] and the underlying principle that when communication bottlenecks exist it stifles peer-to-peer based learning. The solution is to increase communication concurrency by eliminating teaching staff as the intermediary, increasing asynchrony of communication by allowing discussion outside of official class hours, and using teaching staff as moderators and less as all-knowing oracles. These principles drive the KarmaCollab app experiment. The EEC 10 Winter 2021 class (referred to as the baseline) used the suggested curriculum format proposed by del Rosario et al[18]. This current work extends these practices through the KarmaCollab app platform during the subsequent run of EEC 10 course in the Spring of 2021.

Study Conjecture Map

To formalize the design of the learning environment, a conjecture map is used as developed by Sandoval et al[19]. The conjecture map visualized in Figure 8 is broken into four elements. The high-level conjuncture describes how to support the kind of learning we are interested in supporting in the specific context. That conjecture is then realized through the embodiment of the specific design. The embodiment is assumed to create mediating processes which in turn result in our desired measurable outcomes. As will be discussed further in the following sections, fewer students were left behind with 7.8% in the baseline and 4.3% in the KarmaCollab based course, which supports the first of our desired outcomes. Here, 'left behind' is defined as receiving below a 50% average grade on their lab assignments. Secondly, we consider that in both the baseline and KarmaCollab course, the average lab scores were both 87%, implying that students in each cohort learned equally as well. This consistency supports our second outcome.

In the following data analysis sections, the grade information between EEC 10 Winter 2021 (baseline) and EEC 10 Spring 2021 (KarmaCollab) are compared with app engagement data considered alongside. Student administered surveys are then presented with insights into student sentiment about the course changes. Teaching assistant interviews then give a perspective of how the teaching staff viewed the use of KarmaCollab in the classroom.

Analysis of Grade Data

Course grade data for Winter 2021 (baseline) and Spring 2021 (KarmaCollab) were anonymized and studied. Only lab scores are evaluated as an indicator of comprehension as other aspects of the course, such as quizzes, were not parallel comparisons. Any student who scored 0 on all assignments during the entire quarter course was assumed to not be an active participant in the class and, hence, removed from the evaluation. Spring 2021 boost points were out of 100 total, with Winter 2021 out of 150. Winter 2021 boost points are adjusted to a scale 0 to 100 using the following min-max with

$$z_i = \frac{x_i - \text{min}(x)}{\text{max}(x) - \text{min}(x)} \times 100$$

Where z_i is the i_{th} normalized value in the Winter quarter boost point dataset, x_i is the i^{th} value in

Table 1: Bivariate correlation between lab scores and boost points for Winter and Spring Quarter. *Correlation is significant at the 0.05 level (2-tailed). **Correlation is significant at the 0.01 level (2-tailed)

	EEC 10 Winter	EEC 10 Spring
Pearson Correlation	0.319*	0.448**
Sig (2-tailed)	0.01	0
Students in study	64	70

the Winter quarter dataset, min(x) is the minimum value in the Winter quarter boost point dataset, and max(x) is the maximum value in the Winter quarter boost point dataset.

Both courses had the same 10% incentive for getting boost points. Average lab grades are evaluated against boost points to determine the bivariate correlation using Pearson coefficients. The results can be seen in table 1 with a plot of average lab scores vs boost points shown in Figure 9. The correlation between boost points and lab score is 0.319 for Winter and 0.448 for Spring. Thus correlation is significant at the 0.05 level for Winter and 0.01 for Spring with a sample size of 64 and 70, respectively. The positive correlation in both the Winter (baseline) and Spring (KarmaCollab) courses demonstrate that students with higher total boost points had a better comprehension of the material. The scatter plot shows the baseline course had more students struggling, with 7.8% under the 50% lab score mark. In the KarmaCollab version only 4.3% of students were under the 50% mark. We expect this could be due to the ease of getting help with the app. The average lab score in both courses were 87% indicating that comprehension was equal across both courses.

Analysis of App Data

Data is extracted from the KarmaCollab database and analyzed to gain insights from student chat, video, and question post engagement. Only aggregate lab scores and boost points are considered for the rest of this analysis. The lab grade data is cross-referenced with app data to add additional dimension to the analysis.

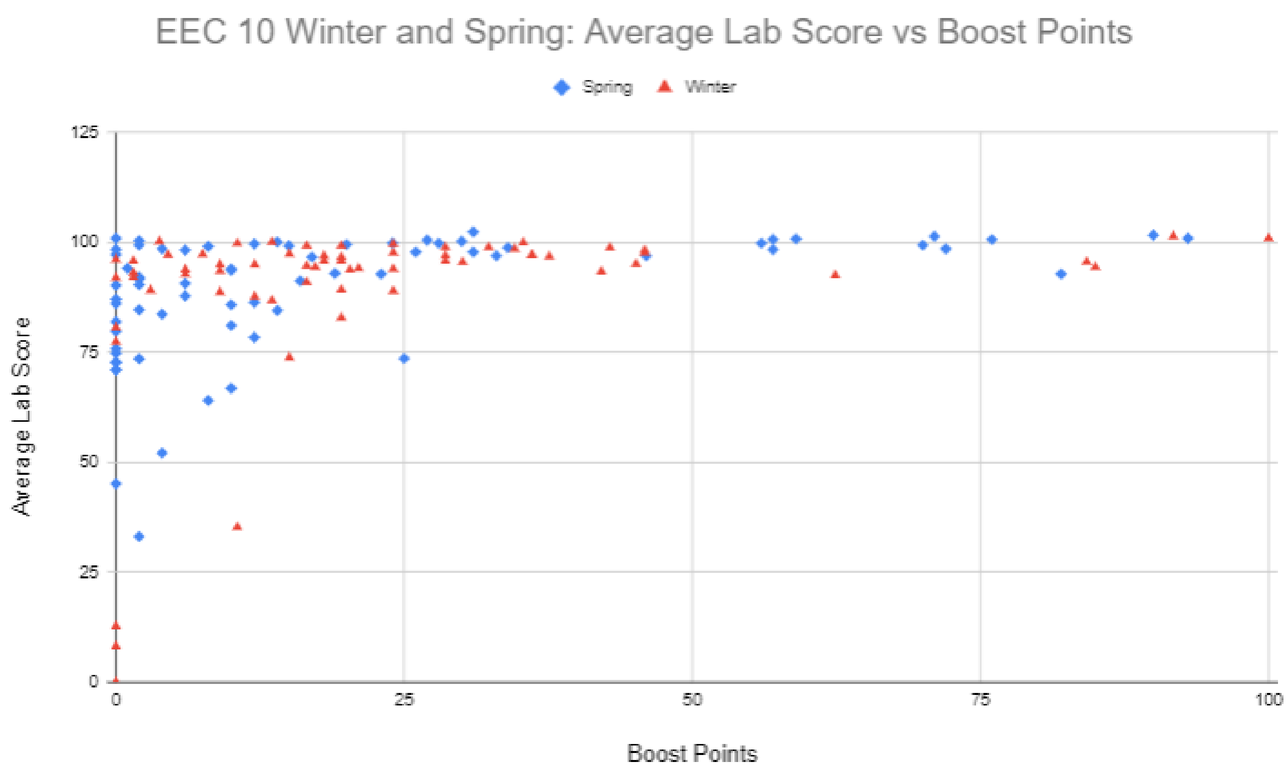


Figure 9: Scatterplot of average lab scores vs boost points for Winter and Spring Quarter.

Chat, Video, Question Engagement

In Figure 10, chat, video, and question post engagement are shown with respect to the average lab score in just the first two weeks of the course. Figure 11 shows the same data but aggregated over the entire quarter, not just the beginning. Chat engagement consists of how many times a student posts a message (over a specific minimum character limit) or an image in one of the discussion rooms. Video engagement involves participating in a video call with at least one other student or teaching assistant present. Question engagement consists of any question posted by the student asking for help or requesting a check-off on a lab. In the first two weeks of the course, there seems to be a disproportionate number of questions being asked relative to actual communication via chat or video on the lowest end of the grade spectrum (30-39% grade level). The students on the highest end of the spectrum (100+ % grade level) were asking many questions but also engaging on chat and video during those first two weeks. As the quarter progressed, the mid-range achievers (60-89% grade level) caught up in engagement compared to the first two weeks. In general, the ratio of chat, video, and posts seem to be about equal across grade ranges except at the very lowest end. This range is somewhat unexpected as one would think that the high performers would consist of the naturally talented students who would ask very few questions and primarily help others. With high performers, they not only chat more than most of the other groups, but they also asked the most questions throughout the class.

Analysis of Survey Responses

In addition to our primary case study course, EEC 10, surveys were administered to three additional courses using the surveying platform Qualtrics. The four courses during the Spring 2021 quarter that were surveyed were EEC 10 (70 enrolled), EEC 150A (35 enrolled), ENG 6 (204 enrolled), and ENG 100 (74 enrolled). Each course utilized KarmaCollab differently based on the instructor's preferences. Incentives and utilization for courses are summarized here.

- EEC 10 (Intro to Digital and Analog Systems) Incentives offered and participation required

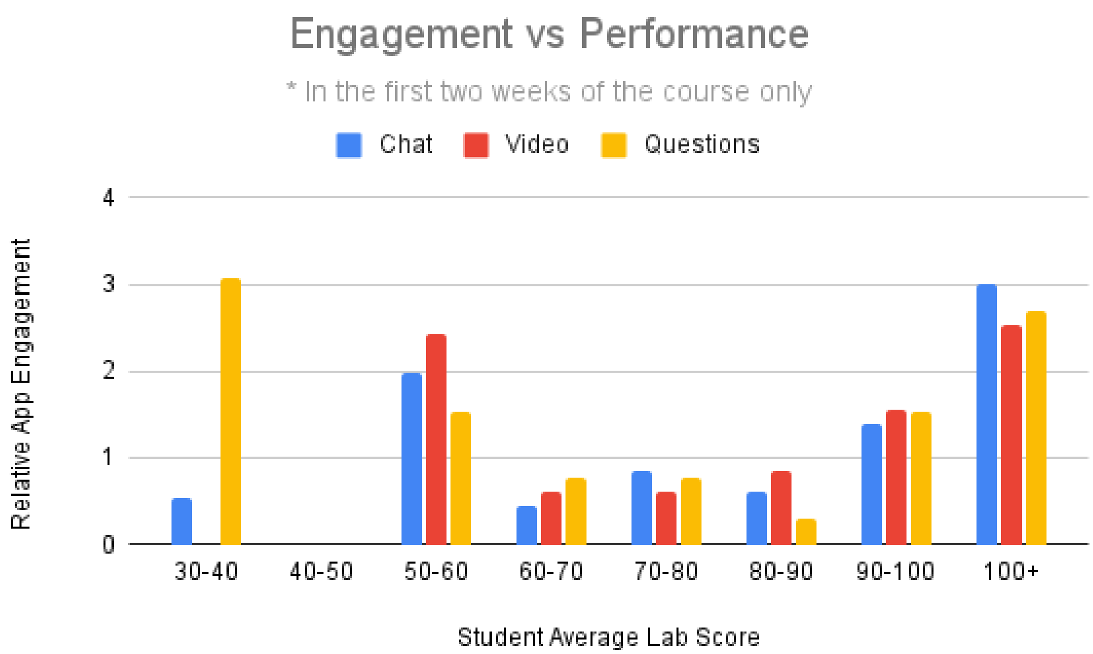


Figure 10: Chat, video, and question engagement for the first two weeks of the course evaluated by average lab score. The 100+ category is due bonus questions on labs that push lab grades over 100%.

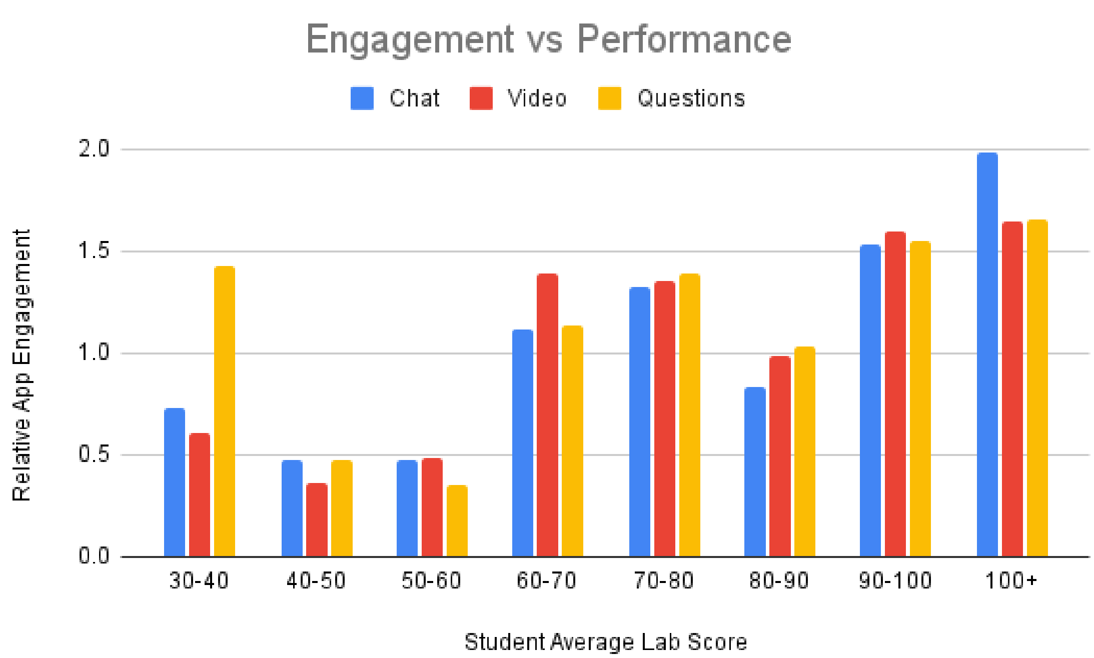


Figure 11: Chat, video, and question engagement for the entire quarter evaluated by average lab score.

Figure 12: Student participation on KarmaCollab across courses.

- EEC 150A (Signals and Systems) Incentives offered but participation optional
- ENG 6 (Engineering Problem Solving) Incentives not offered and participation optional
- ENG 100 (Electronic Circuits Systems) Incentives not offered and participation optional

Due to the differences in implementation, a full set of survey questions were sent to EEC 10 with only relevant questions were sent to the other three courses. For this paper, EEC 10 is considered the primary subject of the case study, given it was the only class that used KarmaCollab at its full capacity. The other classes are used as a value-added for comparison to EEC 10.

Figure 12 shows us that EEC 10 and EEC 150A had the most participation on KarmaCollab, probably due to the fact there were incentives to use the app. ENG 6 and ENG 100 show less participation in KarmaCollab and more participation in other platforms (i.e. Slack, Discord, Facebook). There is an inherent deterrent for students to use official platforms as they know teaching staff will be viewing what is posted. Backchannels such as a private group chats allow students to trade answers and cheat on assignments without repercussions. Students overwhelmingly liked the idea of getting boost points as a reward for not procrastinating on lab assignments, as is shown in Figure 13, however, there seems to be no consensus if having a boost point linked leaderboard on the app motivated students to engage with their classmates.

Teaching Staff Interviews

An interview was conducted with the 6 EEC 10 Spring 2021 teaching assistants (TAs). Overall, TAs agreed that KarmaCollab provided more flexibility for students and themselves. Students were able to turn in their assignments and checkoff their lab with a TA at any time over video chat. The TAs observed that students had an initial shock to the new class format but adjusted quickly. When students experienced glitches in the app it caused them to question the technology. There were expectations students had from using more established platforms like Slack and Piazza which were not met by KarmaCollab at the time of running the test. TAs received student feedback that, although they would like to assist others on the app, their life outside of class was too busy. Many students wanted to schedule KarmaCollab time instead of jumping on in the moment, showing that for some, more structure is preferred. TAs preferred the "specialized" format in which they were able to focus on one aspect of the course (i.e. grading lab assignments, monitoring KarmaCollab, course logistics, etc) as opposed to dividing their time on many different tasks. Having students answering most of the questions on the platform before they even got there was a big help in reducing TA workload.

Conclusion

At the beginning of the COVID-19 shelter in place order, many universities frantically transitioned to an online format from their traditional in-person format. The focus was originally on making resources and course material available online followed by exploring and testing different technologies to streamline remote learning. KarmaCollab was an experimental platform, run from any smartphone, that allowed students and teachers to recreating some of the sparks of in-person learning while operating remotely. KarmaCollab utilizes concepts such as gamification via the leaderboard, students teaching students, and quick lab support through instant video chat - all done in a seamless and automated way. Although KarmaCollab is not favored (and was not intended) to replace the traditional in-person lab course, it does have the potential for enhancing

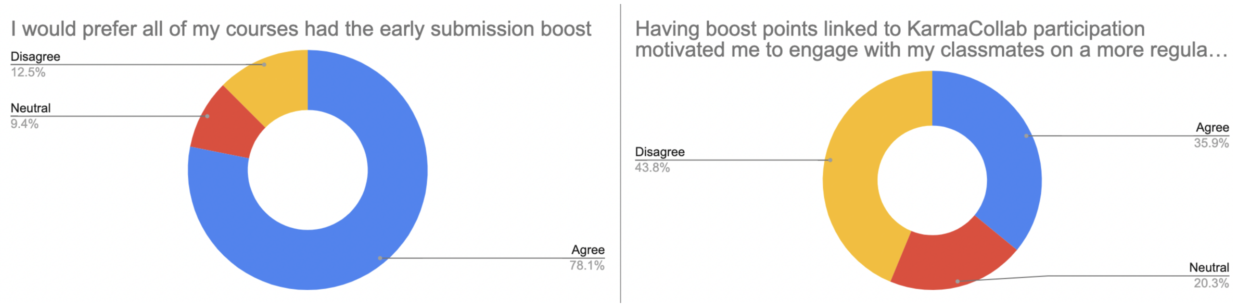


Figure 13: As reported for the primary case study course EEC 10 Spring 2021. [Left] Response to question, "I would prefer all of my courses had the early submission boost". [Right] Response to question, "Having boost points linked to KarmaCollab participation motivated me to engage with my classmates on a more regular basis"

the in-person or fully remote classroom experience.

4 Dynamic Group Formation with Suitability Constraints in large Social Networks

4.1 Introduction

For people seeking to form a team for a specific purpose, like a side project or study group, challenges quickly arise once they have exhausted their social circle in the search for teammates. In the wake of the current pandemic, meeting new people that are right for a specific team is even more difficult than before due to the lack of in-person events. On social media platforms, users often have large networks of friends but have very few close personal relationships in them. So, posting on those to look for people that are compatible, share the same goal and are interested in the niche group, is being hopeful at best. We present a scalable framework for establishing small online groups that balances two objectives, making the best group recommendations to users and guiding group hosts to the best users for their group. We illustrate this framework using three use cases and evaluate a server-less implementation using a large social media dataset to simulate a production environment.

The ongoing pandemic has forced many events and gatherings to be cancelled or moved online. However, many social aspects of in person gatherings cannot be easily facilitated by digital media, and networking is among them. Attending relevant events, classes or conferences is a frequent approach to building a network of compatible and like-minded people. So without real chances to network, a social task that immediately suffers would be the need to form a group of a common purpose. For example, finding suitable co-founders for start up ventures based on casual conversation at mixers or academic events, or establishing rapport with classmates during practical exercises to form study groups. These are situations where real gatherings allow for personal connections as a by-product of its main purpose and can greatly help with meeting new people if you need a group.

While online platforms can create an acceptable stage for the main event, they do not provide the same level of interpersonal possibility. We consider the problem of a user seeking to form a small purposeful group of the best and most compatible people. On existing social networks, users often have a large number of friends but very few of them are close personal relationships. The large network of loose and estranged relationships is not very useful for finding group members that share the same specific interest, purpose and timing.

Consider a student forming a study group. Studies by Dolmans et al.[20] and Springer et al.[21] show that there are a number of cognitive and motivational benefits to small group learning. There are also the benefits to presentation, communication and team-building skills. Study groups allow students to take responsibility and benefit from small group learning outside the classroom. The issue is forming the right study group. Without preexisting relationships forming a group in a class is close to a random assortment.

This randomness results in a variety of personality types, learning styles and interests among the members. This variety has a negative effect not only on the group's compatibility, and therefore comfort in engaging discussion, but also the effectiveness of the small group learning since its members learn best in different ways. Even if the students do not know their best learning method and style, work by Chamorro et al.[22] shows the correlation between personality and preference for learning methods.

By creating an online means to form these groups we can allow students to find a group with high compatibility both as people and as learners. It allows us to consider a wider set of potential group mates and a number of key variables such as learning style, personality traits[23], subject comfort and general common interests. We can then form the ideal online study group in an efficient manner. For example a German student in an English speaking University can find a study group with similar interests and the same native language.

The issue that needs to be addressed is the conflicting needs of the student creating the group and those being presented with groups to join. The seekers want to see groups that are closest to their own preferences. However if they apply to those, there might be users even closer to the host's target attributes. The outcome is that the group host has a large number of applications to review and the seeker is likely to be rejected from the group in comparison to better fits. Therefore the problem is finding the right combination of load balancing and matchmaking to meet both needs. We must balance the happiness of group hosts and group seekers, by showing seekers groups they are likely to get into and enjoy while helping hosts find the most suitable users for their group in the least applications.

We present a framework for establishing small groups that solves the problem described. We focus on scalability through an event-driven architecture and a simulated annealing style approach to maintaining an optimal system state. Illustrative scenarios are used to describe the usefulness of the framework and an implementation of the study groups example is done using server-less cloud functions. We evaluate the implementation by using social media data to simulate a production environment with a large number of users.

4.2 Related Work

The problem described can be viewed as an extension to the matchmaking algorithm of a dating application. Work done in Hitsch et al.[24], Brozovsky et al.[25], Li et al.[26],[27], and et al.[28] describe attempts to find solutions to the one-to-one matchmaking problem. The methods include using the Gale-Shapley algorithm by Dubins et al[29] as well as recommender systems and a combination of techniques for producing additional user attributes. These methods accomplish a means of determining high quality two-sided matching. However, the use case does not need to consider the load of matches or applications faced by a user as much as it has to focus on producing the most likely matches. Nor do they explicitly address the over time scalability of the approaches used.

Online gaming lobby creation is a popular group focused matchmaking use case so their approaches are worth considering. In Boron et al.[30], Agarwal et al.[31], Myslak et al.[32], and Manweiler et al.[33] approaches to producing balanced and well matched groups of gamers while addressing the time constraints of their use case are described. The methods place users in groups that best match their case specific attributes but these applications are shaped by the time sensitive nature of their purpose. To ensure users spend the little time waiting to play, lobby creation is done automatically and so solutions are not concerned with the constraints of our described problem. In trying to emulate the social activity of forming a small meaningful group, granting both hosts and group seekers control and choice is paramount. With managing the load of applications bared by group hosts, the problem expands out of only matchmaking and into the area of load balancing applications for the host's review.

Load balancing algorithms receive user requests as input and distribute those requests among available resources based on their capacity. Variations of such algorithms including FIFO, fair scheduling and capacity scheduling are presented in Ghomi et al.[34] and Zaharia et al.[35]. These are strong solutions to load balancing problems in general. However, as applied to our problem, once an allocation is made, these methods do not address maintaining a steady state system while the sets of groups, users and their attributes change over time. Nor do they concentrate on matching attributes that can shape the way the load is distributed. Maintaining steady state can be perceived as consistently moving towards an optimal solution in the face of sub-optimal changes. Simulated Annealing (SA),[36], uses the analogy of the annealing of solids to solve optimization problems. At each stage of the SA algorithm, with some probability one stays in the present state else moves to a new state. This results in eventually moving to lower states of energy, gradually approaching an approximate global optimum[37].

For maintaining the best allocation of users to small groups while considering a system with many users and many groups, finding the optimal solution is a very computationally intense task. One possible approach, is over night batch processing that re-stabilizes the system. However this is still a huge computation and not the most timely solution. Works by Attiya et al.[38] and Saadatpour et al.[39] show cases of SA's use in load allocation problems. While none of these are a complete solution to our problem, they present evidence that a scalable approach to solving the problem described and maintaining a close to optimal state over time can be achieved with Simulated Annealing.

4.3 Problem Statement and Proposed Approach

Users within the matchmaking system are shown a set number of group recommendations at all times and they can apply or belong to as many groups as they wish. The number of users recommended a group is dynamically determined relative to the capacity of that group and population size of users in the system.

Our framework manages such a system in the face of events, for example a newly joined user, and is defined by an event-driven architecture. It comprises of considerations for five of the major events that occur in a small group matchmaking system while satisfying the following goals:

- Balancing Happiness - Both group seekers and group hosts want matches where user preferences and group attributes are the most similar. However, too many group options or group applications would overwhelm both types of users and must be limited. Therefore we must find a balance between the needs of group seekers and hosts.
- Adaptability - As a framework for matchmaking, it must be easily adaptable to different applications.

- Maintaining Steady State - Sub-optimal changes like new groups or users will occur and the system must be able to rectify them.

4.3.1 Ideal State

The measurement of the system's state is called the entropy value. In a system of users and groups, the ideal state is an entropy of 0. In this state groups are recommended to a reasonable number of the most appropriate users, and users are presented with a set of groups they are most likely to want and be acceptable for. Therefore the entropy value evaluates both the number of users per group and the differences between group features and user preferences for those groups. The entropy value is composed of two metrics, preference differences and users per group.

Preference Differences is a measure of the dissimilarity between a user and a group. The cost functions for this measure can vary with the attributes and desired matching outcome per application of the framework. In calculating the system-wide entropy we take the average preference difference between users and groups.

Users per Group is a measure of how many users are recommended a group, relative to the group's capacity. We cannot allocate the exact capacity because the allocations are only presenting the group to users, not guaranteeing they would apply or that the host would accept them. The measure of users per group is called the alpha value where u is the number of users in a group, c is its capacity and $\alpha = \frac{u}{c}$. Ideally we would want all groups to have equal alpha values, so a system wide measure of users per group would be the variance of alpha values.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (\alpha_i - \tilde{\alpha})^2 \tag{1}$$

Where u is the number of users in a group, c is its capacity, α_i is the alpha of the ith group, $\tilde{\alpha}$ is the mean alpha and n is the number of groups, variance, σ^2. The system entropy value combines the above two metrics. Resulting in a formula for entropy, e,

$$e = \sigma^2(A) + \frac{\sum_{i=1}^{n}(d_i)}{n} \tag{2}$$

Where A is the set of α values for all groups, d_i is the preference difference value of the ith user in the system and n is the total number of users.

4.3.2 Events

The framework is built around the structure of the following main system use cases:

New User: As a new user joins they would need to be given groups they can choose to apply for based on their defined preferences. One option is to do this greedily and give them the best groups that match their desires and then let the system eventually resolve itself in terms of balance. However if we are assigning groups that we know are likely to be swapped out, we can instead assign the lowest alpha groups. This way we use the opportunity to help balance group sizes and the quality of matches will be improved soon after by running a rectifying method called the bubble function.

New Group: There is a set limit to the number of groups that can be presented to a user at any given time, so a newly created group cannot simply be given to whichever users closest match its features. We take a similar approach to the new user case and focus on balancing allotments, therefore managing a reasonable number of applications faced by group hosts without giving them too little. A new group takes the worst matching users from the groups with the highest alpha values. Doing this improves the alpha variance while sacrificing the least matching quality.

Remove User: When a user chooses to delete their account, they need to be removed from the groups they belong to. The vacancy created will be addressed by the alpha balancing mechanisms present in the other cases.

Remove Group: If a group host or creator decides to delete their group, firstly the users belonging to it need to be updated. However unlike the Remove User case, the vacancy created here in the user's group slots cannot be left to the system to fix. It must be dealt with immediately for the sake of ensuring users have a list of groups presented to them at all times. Therefore once removed from the deleted group, the users are spread out among the lowest alpha groups.

Close Group: Once a group is closed from applications, it enters a state where allocated users that have not been accepted to the group, are considered rejected. Both accepted and rejected users are seen as having a free group slot at this point and are spread among the lowest alpha groups to fill it. Accepted users are included in order to maintain a consistent number of options, and to be mindful of applications where the users may intend to join more than one group and want further options despite being accepted to one.

4.3.3 Bubble Function

While the aforementioned cases make immediate considerations for balancing group α values, they do not improve the quality of matches. This function focuses on rectifying that and is ran following any sub-optimal changes to a user's groups. We accomplish this by searching a user's recommended groups for any swaps with users shown other groups that would decrease the system entropy. Users involved in swaps will see updated group recommendations.

Searching for the best swap of a user's group on the entire set of groups can quickly become too heavy of a computation. So we filter into a reasonably sized subset based on some significant feature, for example in the case of the study group, the closest 100 groups can be the subset for the sake of valuing similar time-zones, nationality and university. The choice of filtering is dependent on the application and user preference.

We take a greedy approach of finding the best swaps, where for each group a user has, we search the entire set of other groups' members for the best swap. This approach is simple and allows for the possibility of stopping after finding a number of good swaps, that way the system gets to an approximate ideal state local to the user. A swap is considered the 'best' if the difference between the sum of the preference difference costs of the original two user-group pairs and the sum of the costs of the newly swapped pairs is at its greatest.

4.4 Framework Applications

The events of the framework architecture remain generally the same across applications. Adapting to an application focuses on the definition of preference differences that measure the matching quality. The following examples illustrate this as well as the usefulness of the framework through different applications.

4.4.1 Study Groups

For a university student, members of the same class might not be the best candidates to form a study group with. Learning styles and personality traits can vary greatly and can negatively affect the compatibility needed for a well functioning study group. Even if potentially suitable candidates exist in the same class, within a large university class students barely know one another. The rapport and discussion required to determine compatibility is time consuming, and for some, extremely difficult.

Online study group matchmaking increases the efficiency and likelihood of forming an ideal study group by considering widening the scope of candidates and considering key attributes. These key attributes also define our preference difference cost functions and include learning styles, personality traits as per the Big Five[23], subject comfort, distance between members and general interest tags.

4.4.2 Online Gaming Teams

Many people play online video games alone, and depending on the game, the competitive nature or need for a larger group can often lead to users needing to find a group or team even if they play with friends. This is often the case for Multiplayer Online Battle Arena (MOBA) games where skill varies greatly among players and finding a team of five equally capable players that are also compatible as people, can be difficult. Role-playing games like Dungeons and Dragons are often entirely impossible for those without a gaming social circle and trending social deduction games such as Among Us or Mafia require also require large groups.

Online lobbies are a possible means but those only on player skill and not social compatibility which is paramount for the most enjoyable experience. Ideal gaming teams can be created by devising cost functions around ensuring similar player skill, a variety of preferred roles, similar intention whether its competitive improvement or casual fun and general interest tags for building team rapport.

4.4.3 Startup Team of Founders

A startup venture is a daunting journey and it is rare for a single person to be capable of a successful trek on their own. For this reason, founding teams are usually formed. Aside from the distribution of work, a diverse team of founders can foster a more innovative environment through differing perspectives and experiences. The challenge is in finding a group of people that are diverse in terms of skills and knowledge but still similar in interests and personality traits in order to be compatible. Factors cost functions would consider here include ensuring a variety of primary roles among members, a variety of skills, similar preferences for business ideas and general interest tags.

4.5 Implementation

The study groups application of the framework was implemented using a set of cloud services and datasets that closely resembled the actual scenario.

4.5.1 Cost Function

We used the following cost function definition as the means of measure preference differences between a user and group. The cost function c of the study group application is defined as the average of comfort level (c_c) and distance (c_d).

$$c = \frac{c_c + c_d}{2}$$

A user defined measure of subject content confidence as an integer. User values do not need to match a group's set target comfort level but the closer they are, the lower the cost. Where difference between user comfort level and target group comfort level score is v, comfort level cost is given by

$$c_c = \frac{v}{v+1} \tag{3}$$

For example, consider a group with a target comfort level of 5 out of 5. A user whose comfort is 1 would have a comfort cost of 0.8 while another with a comfort of 4 would have a much lower cost of 0.2 because they are closer to the group's value. Distance cost function. Similar to comfort level, the closer the value the lower the cost. With distance between user and group host, d, distance cost is given by

$$c_d = \frac{d}{d+1} \tag{4}$$

4.5.2 Datasets

The Facebook100[40] dataset was used to represent users in the system as it provides real social media users with educational attributes. The 'Student Major' feature was used to represent a value for user subject comfort level because its distribution was the closest to a bell curve. The dataset did not contain location data for users which was needed for measuring distance costs. We used the tweet-geolocation-5m[41] dataset and assigned location information to each pseudo user. This resulted in a dataset of approximately 1.2 million users with subject comfort level and location attributes.

4.5.3 Architecture

An dummy android app was developed using Flutter to drive simulated user actions such as new users joining and new groups being created. Google Cloud Platform services were used for data storage and hosting function logic. Specifically, Cloud Functions were used for hosting server-less functions for each of the event use cases and functions described in the previous section. Firestore was used for storing user data sent from the application. Fig. 14 provides an illustration of the solution's system architecture.

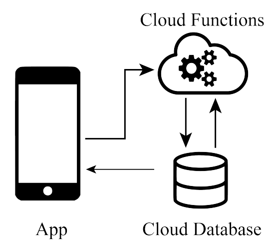


Figure 14: System Architecture

4.6 Evaluating Functions

The bubble function is the heart of the framework, it swaps users to rectify sub-optimal changes in a way that makes both users and group hosts happy. The first experiment was done with the intention of testing the ability of the bubble function, on a still set of users and groups, to move the system towards an ideal state of low entropy. A sample of 100 users and 20 groups were created in such a way that all groups had the same α value. Users and groups only held a single feature of interest, comfort level, for the sake of simpler calculation verification and simpler visualization of results. Next, the bubble function was triggered on random users 350 times. The function was triggered manually since no sub-optimal changes will occur to trigger it automatically.

The overall system entropy was tracked as the sum of the variance of group α values and the average preference difference between groups and users. This was tracked on update for any group or user by an additional cloud function. All trace values were accompanied by a timestamp and the entropy over time was plotted. Fig. 15 indicates that our results are as expected, the system strictly decreases in entropy until eventually it plateaus. Therefore we can conclude that the bubble function successfully improves the state of matches.

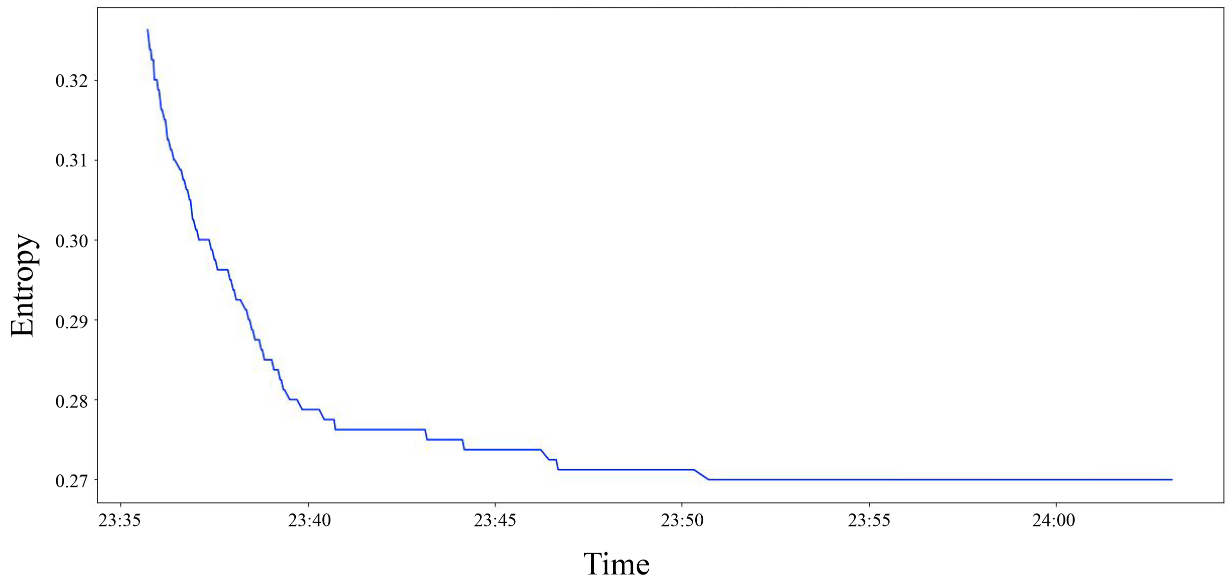


Figure 15: System entropy over bubble evaluations.

The resulting low entropy state system was used as the basis for the second experiment. Here the goal was to determine the ability of the system to maintain an ideal state following sub-optimal changes. A handful of new users and groups were added one at a time, with the bubble function automatically following these events as it should. The aforementioned entropy trace continued in the same way. The entropy over time is provided in Fig. 16. Instead of strictly decreasing, at the points of simulated events there were small spikes of entropy due to sub-optimal changes. The bubble function that followed in each case immediately brought that entropy back down. It should also be noted that the plot is still tending downwards because of the further improved matches that were found during the changes.

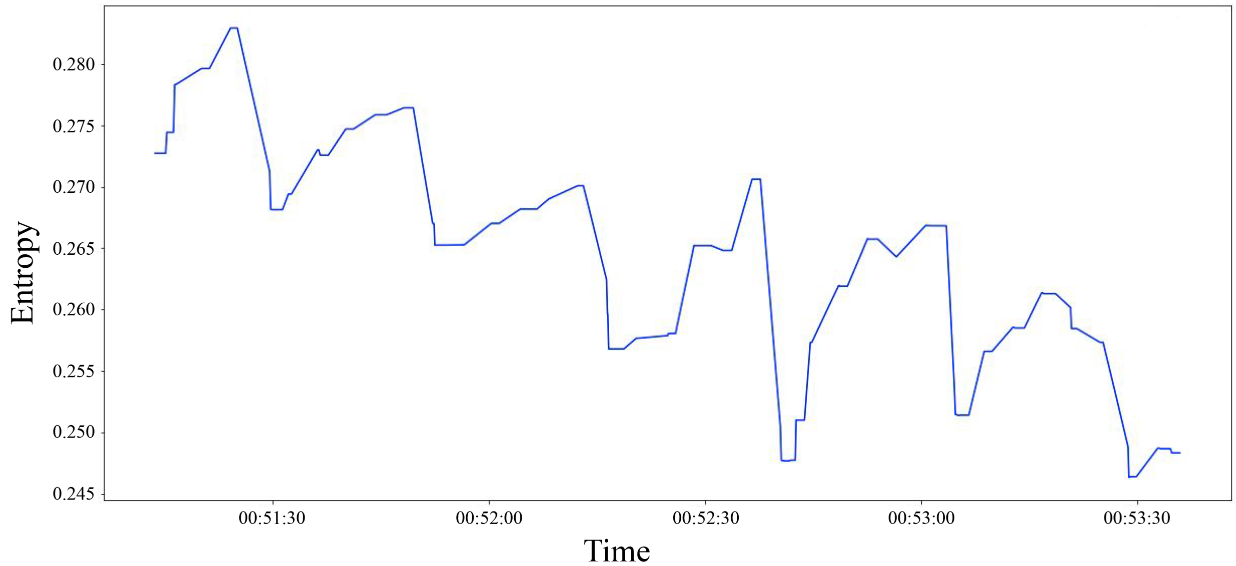


Figure 16: System entropy over steadying state evaluation.

4.6.1 Visualization of Results

Through the two experiments we showed that the framework can successfully improve the state of the system and maintain the stability continuously. However this is based on our own definitions of entropy and cost formulae. Therefore we also produced two data visualization plots that help display the relationship between our system results and derived solution. We repeated the experiment but with 1000 users and 20 groups holding 50 users each. Before and after it was ran, we plotted each user's value of the feature of interest against their group's corresponding value. As more points overlap they merge, and their shade of copper darkens as density increases.

Fig. 17 shows the initial state of groupings with the initial entropy values. The arrangement is random and represents a poor allocation of users. Each group has a wide spanning variety of user feature values shown by the spread of points on the plot. The final state of groupings after the framework functions were ran is shown in Fig. 18. A linear positive correlation between user and group values for the feature of interest can be seen. This shows that users move closer to one another and towards groups with comfort level values close to their own. These are the groups

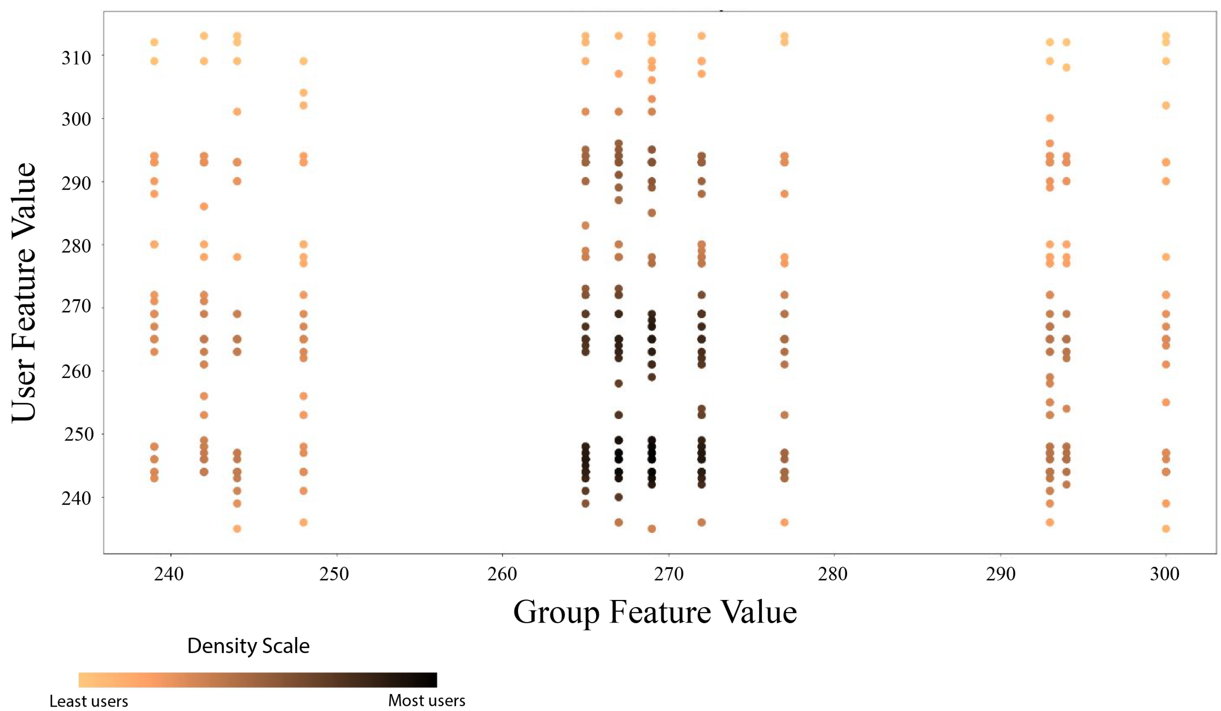


Figure 17: Initial state of user feature values vs. group feature values.

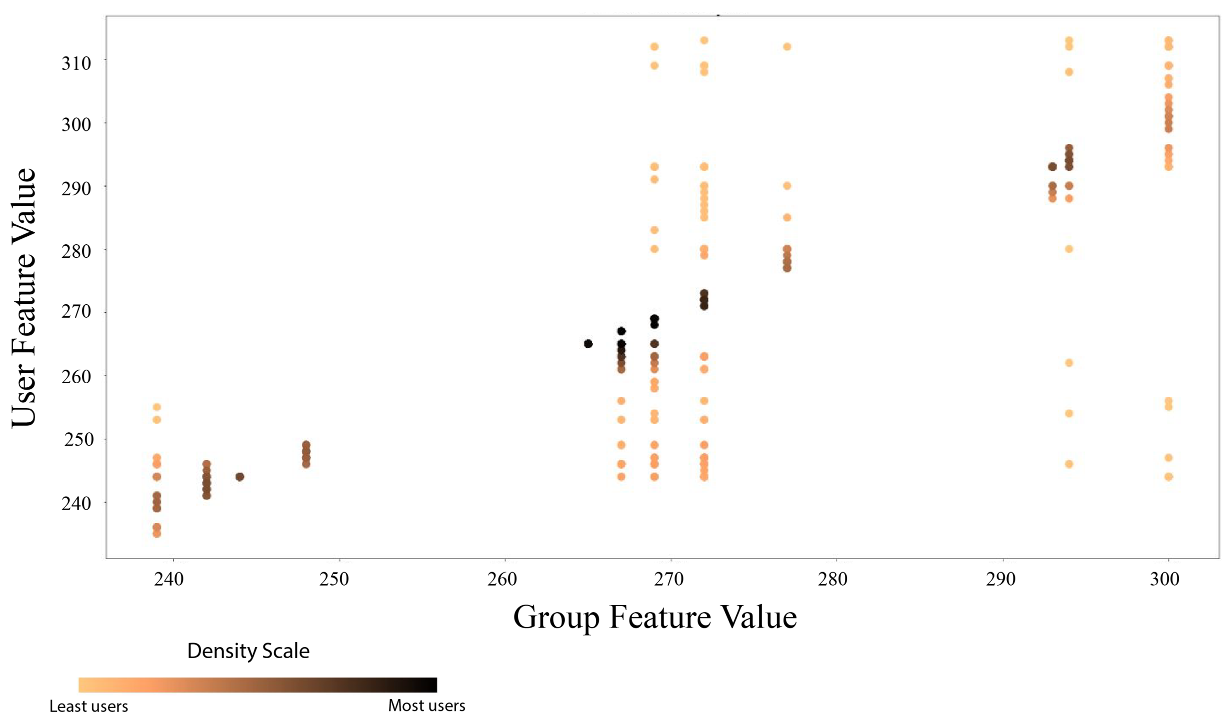


Figure 18: Final state of user feature values vs. group feature values.

they would be happiest with and are most likely to get accepted into. Small outlying clumps of users are still present due cases where the best matching group is already filled with better matches.

4.7 Conclusion

In the case of forming small online groups, existing options are too impersonal because of their size and do not consider the happiness of both group seekers and creators. We provide a solution to this in the form of a framework based on simulated annealing. The framework helps group hosts get the best members in the least application reviews and shows users the best groups they are likely to want but also to be accepted by, because they are among the best users for it. We then elaborated on the framework's usefulness through examples and validated it through implementation and testing. While the proposed solution addresses the problem, there is still room for further improvement.

Future work includes obtaining feedback on the usefulness of example applications and improving the implementation and simulation. We held the online study groups example in primary focus, but also provided two other examples to demonstrate the usefulness of the framework. In each example the target audience is evident. We can validate their usefulness by designing and administering appropriate surveys to judge potential users' thoughts on the current state of the problem and how much they value a solution.

Furthermore, our implementation featured two cost functions for the sake of simplicity and ease of understanding. Assessing a combination of more cost functions would be a closer representation of a real application. Moreover the simulations by which we assess the implementation can be improved by modelling the experiments to closer resemble a real system. We can do this by emulating a realistic user growth curve and churn rate. Improving the quality and variety of how we evaluate the framework can create chances for beneficial discoveries that would lead to more successful production use.

Finally we plan to investigate the formulation of this problem as a transshipment problem in which the groups are intermediate nodes which all feed into a supersink node. Upper and lower bounds on the flow between each group node and the supersink node can be used to satisfy the requirements. The upper bound can be used to limit the size of the group and lower bound can be used to balance membership across groups. Suitability costs are assigned on the edges from the source node to the group nodes. In order to achieve scalability, a distributed solution would be required and in order to continuously adapt to changes, an asynchronous approach can be used. We plan to evaluate the applicability of the distributed, asynchronous algorithm of Bertsekas[42].

5 Dissertation Conclusion

It can be argued that the most pressing challenges facing modern society revolve around the inefficient dissemination of knowledge, the lack thereof manifesting as low quality decision making at all levels of leadership, and the dampening of every individuals ability to offer their unique gifts and perspectives for the common good. At the core of the issue is a lack of scalability of a teacher-centric, 'chalk and talk', oracle at the podium method of education. This research has shown potential for a peer-peer centered model of education where teaching staff enter the roll of moderator and mentor. This not only eases the burden on teaching staff but provides a well needed social outlet for students who find themselves increasingly isolated in an 'everything online' world. As was shown in the EEC 136 entrepreneurship course redesign study, risk taking, discourse, and experimentation, can be re-introduced into a complex project course curriculum without increasing the workload of teaching staff. The multi-year EEC 10 case study demonstrated that a combination of technology and curriculum redesign can create an environment of teacher moderated collaboration, replacing many roles of the traditional teacher, while making getting help easier and elevating struggling students. Moving into the future, much more can be done in terms of matchmaking and expanding peer-peer learning networks. The same social media innovations pioneered for entertainment and online networking could be the starting place for new, modern day tools for use by educators. The application of social dynamics driven technology in the classroom has the potential to both supplement or fully replace many traditional curricula. More interesting, however, is how some of these findings could be adapted to developing nations where students are intensely motivated to learn, but due to financial and logistical constraints, teachers and physical schools simply cannot exist. This work should be seen an a stepping stone to more advanced peer based learning models built on modern technology with our human, social nature at the core of the design.

References

[1] Andy M Connor, Sangeeta Karmokar, and Chris Whittington. From stem to steam: Strategies for enhancing engineering & technology education. 2015.

[2] Thomas R Lord. A comparison between traditional and constructivist teaching in college biology. *Innovative Higher Education*, 21(3):197–216, 1997.

[3] Aliye Karabulut-Ilgu, S Yao, Peter Tarmo Savolainen, and C Jahren. A flipped classroom approach to teaching transportation engineering. In *Proceedings of the 123rd ASEE Annual Conference and Exposition*, 2016.

[4] Vimal Viswanathan and John T Solomon. A study on the student success in a blended-model engineering classroom. ASEE, 2018.

[5] Reid Bailey and Michael C Smith. Implementation and assessment of a blended learning environment as an approach to better engage students in a large systems design class. In *2013 ASEE Annual Conference & Exposition*, pages 23–692, 2013.

[6] C Mirjam Van Praag and Peter H Versloot. What is the value of entrepreneurship? a review of recent research. *Small business economics*, 29(4):351–382, 2007.

[7] Tom Byers, Tina Seelig, Sheri Sheppard, and Phil Weilerstein. Its role in engineering education. *The Bridge*, 43 (2), 2013.

[8] Nathalie Duval-Couetil, Angela Shartrand, and Teri Reed. The role of entrepreneurship program models and experiential activities on engineering student outcomes. *Advances in Engineering Education*, 5(1):n1, 2016.

[9] Jonathan Weaver and Nassif Rayess. Developing entrepreneurially minded engineers by incorporating technical entrepreneurship case studies. *The Journal of engineering entrepreneurship*, 2(1):10–27, 2010.

[10] Ikhlaq Sidhu, Alexander Fred-Ojala, Sana Iqbal, and Charlotta Johnsson. Applying entrepreneurial teaching methods to advanced technical stem courses. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–7. IEEE, 2018.

[11] Aliye Karabulut-Ilgu, Nadia Jaramillo Cherrez, and Charles T Jahren. A systematic review of research on the flipped learning method in engineering education. *British Journal of Educational Technology*, 49(3):398–411, 2018.

[12] Flipped Learning Network. What is flipped learning? the four pillars of flip. flipped learning network, 2014.

[13] Majed Jarrar and Hanan Anis. The impact of entrepreneurship on engineering education. *Proceedings of the Canadian Engineering Education Association (CEEA)*, 2016.

[14] Jodi Pilgrim, Christie Bledsoe, and Susan Reily. New technologies in the classroom. *Delta Kappa Gamma Bulletin*, 78(4), 2012.

[15] Jane E Brindley, Lisa M Blaschke, and Christine Walti. Creating effective collaborative learning groups in an online environment. *International Review of Research in Open and Distributed Learning*, 10(3), 2009.

[16] Martin Sanders. A failure to collaborate. *Chronicle of Higher Education*, 54(24), 2008.

[17] D Randy Garrison and Heather Kanuka. Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, 7(2):95–105, 2004.

[18] Mason del Rosario, Nicholas Hosein, Timothy Ambrose, Rajeevan Amirtharajah, André Knoesen, and Hooman Rashtian. Enabling student success in an online lab-based circuits course. *Advances in Engineering Education*, 8(4):n4, 2020.

[19] William Sandoval. Conjecture mapping: An approach to systematic educational design research. *Journal of the learning sciences*, 23(1):18–36, 2014.

[20] Diana HJM Dolmans and Henk G Schmidt. What do we know about cognitive and motivational effects of small group tutorials in problem-based learning? *Advances in health sciences education*, 11(4):321, 2006.

[21] Leonard Springer, Mary Elizabeth Stanne, and Samuel S Donovan. Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis. *Review of educational research*, 69(1):21–51, 1999.

[22] Tomas Chamorro-Premuzic, Adrian Furnham, and Martin Lewis. Personality and approaches to learning predict preference for different teaching methods. *Learning and individual differences*, 17(3):241–250, 2007.

[23] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.

[24] Gunter J Hitsch, Ali Hortaçsu, and Dan Ariely. Matching and sorting in online dating. *American Economic Review*, 100(1):130–63, 2010.

[25] Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. *arXiv preprint cs/0703042*, 2007.

[26] Xiaoyong Li, Jie Yuan, Erxia Li, Wenbin Yao, and Junping Du. Trust-aware and fast resource matchmaking for

personalized collaboration cloud service. *IEEE Transactions on Network and Service Management*, 16(3): 1240–1254, 2019.

[27] Kun Tu, Bruno Ribeiro, David Jensen, Don Towsley, Benyuan Liu, Hua Jiang, and Xiaodong Wang. Online dating recommendations: matching markets and learning preferences. In *Proceedings of the 23rd international conference on world wide web*, pages 787–792, 2014.

[28] Lee Mackinnon. Love's algorithm. *Algorithmic Life: Calculative Devices in the Age of Big Data*, page 161, 2015.

[29] Lester E Dubins and David A Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.

[30] Michał Boroń, Jerzy Brzeziński, and Anna Kobusińska. P2p matchmaking solution for online games. *Peer-to-peer networking and applications*, 13(1):137–150, 2020.

[31] Sharad Agarwal and Jacob R Lorch. Matchmaking for online games and other latency-sensitive p2p systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 315–326, 2009.

[32] Mateusz Myślak and Dominik Deja. Developing game-structure sensitive matchmaking system for massive-multiplayer online games. In *International Conference on Social Informatics*, pages 200–208. Springer, 2014.

[33] Justin Manweiler, Sharad Agarwal, Ming Zhang, Romit Roy Choudhury, and Paramvir Bahl. Switchboard: a matchmaking system for multiplayer mobile games. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 71–84, 2011.

[34] Einollah Jafarnejad Ghomi, Amir Masoud Rahmani, and Nooruldeen Nasih Qader. Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 88:50–71, 2017.

[35] Matei Zaharia. Job scheduling with the fair and capacity schedulers. *Hadoop Summit*, 9, 2009.

[36] Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

[37] Daniel Delahaye, Supatcha Chaimatanan, and Marcel Mongeau. Simulated annealing: From basics to applications. In *Handbook of Metaheuristics*, pages 1–35. Springer, 2019.

[38] Ibrahim Attiya, Mohamed Abd Elaziz, and Shengwu Xiong. Job scheduling in cloud computing using a

modified harris hawks optimization and simulated annealing algorithm. *Computational Intelligence and Neuroscience*, 2020, 2020.

[39] Motahareh Saadatpour, Abbas Afshar, and Helaleh Khoshkam. Multi-objective multi-pollutant waste load allocation model for rivers using coupled archived simulated annealing algorithm with qual2kw. *Journal of Hydroinformatics*, 21(3):397–410, 2019.

[40] Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3):526–543, 2011.

[41] Arkaitz Zubiaga, Alex Voss, Rob Procter, Maria Liakata, Bo Wang, and Adam Tsakalidis. Towards real-time, country-level location classification of worldwide tweets. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):2053–2066, 2017.

[42] D. Bertsekas. A distributed asynchronous relaxation algorithm for the assignment problem. *1985 24th IEEE Conference on Decision and Control*, pages 1703–1704, 1985.

[43] Julie E Mills, David F Treagust, et al. Engineering education—is problem-based or project-based learning the answer. *Australasian journal of engineering education*, 3(2):2–16, 2003.

[44] Lee Martin. The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research (J-PEER)*, 5(1):4, 2015.

[45] Guy A Boy. From stem to steam: toward a human-centred education, creativity & learning thinking. In *Proceedings of the 31st European conference on cognitive ergonomics*, page 3. ACM, 2013.

[46] Yunglung Chen, Yuping Wang, Nian-Shing Chen, et al. Is flip enough? or should we use the flipped model instead? *Computers & Education*, 79:16–27, 2014.

[47] Eun Man Choi. Applying inverted classroom to software engineering education. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 3(2):121, 2013.

[48] Randall S Davies, Douglas L Dean, and Nick Ball. Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course. *Educational Technology Research and Development*, 61(4):563–580, 2013.

[49] Khosrw Ghadiri, Mohammad H Qayoumi, Ellen Junn, Ping Hsu, and Sutee Sujitparapitaya. Stratagems in stem. *age*, 24:1.

[50] Gloria J Kim, Erin E Patrick, Ramakant Srivastava, and Mark E Law. Perspective on flipping circuits i. *IEEE Transactions on Education*, 57(3):188–192, 2014.

[51] Jerzy Rutkowski. Flipped classroom-from experiment to practice. In *Proceedings of 1st International KES Conference on Smart Technology Based Education and Training, Chania, Greece*, 2014.

[52] Bjarne Schmidt. Improving motivation and learning outcome in a flipped classroom environment. In *2014 International Conference on Interactive Collaborative Learning (ICL)*, pages 689–690. IEEE, 2014.

[53] Daniel Levi. *Group dynamics for teams*. Sage Publications, 2015.

[54] Alex Pentland. The new science of building great teams. *Harvard Business Review*, 90(4):60–69, 2012.

[55] Marwa Yousif Hassan, Abdi O Shuriye, A Abdallah, Momoh JE Salam, and Othman O Khalifa. Integrating entrepreneurship into engineering education. *International Journal of u-and e-Service, Science and Technology*, 10(2):45–52, 2017.

[56] Hala W Hattab. Impact of entrepreneurship education on entrepreneurial intentions of university students in egypt. *The Journal of Entrepreneurship*, 23(1):1–18, 2014.

[57] OKCupid: The Math Behind Online Dating. URL `https://blogs.ams.org/mathgradblog/2016/06/08/okcupid-math-online-dating/`.

[58] Harshit Gupta and Kalicharan Sahu. Honey bee behavior based load balancing of tasks in cloud computing. *International journal of Science and Research*, 3(6), 2014.

[59] Vittorio V Busato, Frans J Prins, Jan J Elshout, and Christiaan Hamaker. The relation between learning styles, the big five personality traits and achievement motivation in higher education. *Personality and individual differences*, 26(1):129–140, 1998.

[60] B Ahadi and M Narimani. Study of relationships between personality traits and education. *Trakia Journal of Sciences*, 8(3):53–60, 2010.

[61] Reshawn Ramjattan, Nicholas Hosein, Patrick Hosein, and Andre Knoesen. Dynamic group formation with suitability constraints in large social networks. In *2020 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pages 1–6. IEEE, 2020.

[62] Asmaa Benbaba. Handbook of distance education: edited by michael grahame moore and william c. diehl, 2021.