Teaching Creative Problem Solving in a MOOC

Pascal Van Hentenryck
NICTA and The University of Melbourne
Lvl 2 / Bldg 193, The University of Melbourne
Melbourne, VIC, Australia, 3010
pvh@nicta.com.au

Carleton Coffrin
NICTA
Lvl 2 / Bldg 193, The University of Melbourne
Melbourne, VIC, Australia, 3010
carleton.coffrin@nicta.com.au

ABSTRACT

The practice of discrete optimization involves modeling and solving complex combinatorial problems which have never been encountered before and for which no universal computational paradigm exists. Teaching such skills is challenging: Students must learn, not only the core technical skills, but also an ability to think creatively in order to select and adapt a paradigm to solve the problem at hand. This paper explores the question of whether the teaching of such creative skills translates to massive open online courses (MOOCs). It first describes a methodology for teaching discrete optimization that has been successful on campus over fifteen years. It then discusses how to adapt the campus format to a MOOC version. The success of the approach is evaluated through extensive data analytics enabled by the wealth of information produced by MOOCs.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education—*MOOC*

Keywords

Problem Solving, MOOC, Artificial Intelligence, Inquiry-Based Learning, Experience Report

1. INTRODUCTION

Discrete optimization is a subfield of artificial intelligence and operations research focusing on the task of solving NP-hard optimization problems, such as the knapsack and the traveling salesman problems. Due to the computational complexity of these applications, the practice of discrete optimization has remained more an art than a science: practitioners are constantly confronted with novel problems and must determine which computational paradigm, modeling, and searching techniques to apply to the problem at hand. As a consequence, the teaching of discrete optimization must not only convey the core concepts of the field but also develop the intuition and creative thinking necessary to apply these skills in novel situations. Teaching such skills is a challenge for instructors who must present students with complex problem-solving tasks and keep them motivated over a long period of time.

These challenges are further exacerbated in Massively Open Online Classes (MOOCs), which have generated substantial interest since the success of Stanford's computer science MOOCs [3, 4, 8, 5]. There are now several start-up companies (e.g., Coursera, Udasity, and edX) offering MOOC platforms and numerous leading universities have partnered with these providers. MOOCs present unique teaching challenges, from the material presentation to the design of assignments and student motivation without traditional interaction patterns. It is thus important to determine whether successful frameworks for teaching problem-solving skills in traditional university settings translate, or can be adapted, to teaching and learning within a MOOC. This question is largely unaddressed in the literature. Existing publications and technical reports on MOOCs [6, 7] focus primarily on what happened in a course: They report various cohort statistics [7] and aim at understanding key performance indicators such as completion rates [6]. Few papers discuss the design and effectiveness of MOOC assessments [9] and how to motivate students to succeed in MOOCs.

This paper is an attempt to shed some light on how to teach problem-solving skills in a MOOC. Section 2 documents the design of a discrete optimization class that has been honed for over fifteen years at a leading U.S. institution and was highly popular among senior undergraduate and graduate students, Section 3 then presents how the teaching philosophy behind this class was adapted to a MOOC setting. This is followed in Section 4 by extensive data analytics, enabled by the wealth of information collected in a MOOC, to validate the design decisions in both the traditional and MOOC versions of the class. In particular, the data analytics results indicate how gamification, community development, and an open course structure were key ingredients to the success of the MOOC. Finally, Section 5 concludes the paper.

2. THE DISCRETE OPTIMIZATION CLASS

Discrete Optimization is an introductory course designed to teach students how to solve NP-Hard problems in practice. The prerequisites are strong programming skills and familiarity with classic computer science algorithms (e.g. sorting, depth first search, minimum spanning tree) and basic linear algebra. It covers an introduction to three core topics in the

discrete optimization field, Constraint Programming, Local Search, and Mixed Integer Programming. Each of the topic areas is relatively independent and any presentation order is admissible. The pedagogical philosophy of the course is that inquiry-based learning is effective in teaching creative problem solving skills.

The class assignments consist of designing algorithms to solve five NP-Hard problems of increasing complexity, knapsack, graph coloring, traveling salesman (TSP), warehouse location, and capacitated vehicle routing (CVRP). The approach to solving these problems attempts to emulate a real-world discrete optimization experience, which is, your boss tells you *"solve this problem, I don't care how"*. This situation is simulated by providing the students with a problem specification and their task is to produce high quality solutions to the problem by any means necessary. The lectures contain the necessary ideas to solve the problems, but the best technique to apply is left for the students to discover [2]. This assignment design not only prepares students for how optimization is conducted in the real-world but is also pedagogically well-founded under the guise of guided inquiry-based learning [1].

Like many design tasks in STEM subjects, NP-Hard problems are open ended since there is no known "ideal" solution. This makes the algorithm design task a very creative process. This is the *art* of discrete optimization. The class fosters this creativity in several ways: all paths to high quality solutions are rewarded equally; students are encouraged to work in pair programming teams; all students are encouraged to collaborate and share ideas outside of class to complete the assignments, although the implementation details should be their own work.

2.1 Assignment Structure

Discrete optimization tasks have a general declarative specification in the form of an objective function f and a system of constraints g_i ($i \in C$). This allows the instructor to specify the task at hand rigorously. Let $\mathbf{x}$ be a vector of decision variables and $\mathbf{w}$ a vector of input data, an optimization problem seeks to minimize $f(\mathbf{w}, \mathbf{x})$ subject to $g_i(\mathbf{w}, \mathbf{x})$ ($\forall i \in C$). Due to the nature of NP-Hard problems, the assignments can be incredibly difficult to solve well, but verifying the quality of a solution is very easy. Furthermore, the objective function provides a simple quantitative measure for marking and comparing student solutions. To make the assignments more concrete, we discus the first assignment in detail.

$$\begin{aligned} \max \quad & \sum_{i \in I} v_i x_i \\ \text{s.t.} \quad & \sum_{i \in I} w_i x_i \leq K \\ & x_i \in \{0,1\} \ (i \in I) \end{aligned} \tag{KS}$$

The knapsack problem (KS) is the most common NP-Hard problem covered in undergraduate computer science curriculum and hence it makes an excellent warm-up assignment in a discrete optimization class. The problem is defined as a container with capacity K and a set of items I. Each item $i \in I$ has a value v_i and a weight w_i. The goal is to find a subset of the items $I^* \subseteq I$ that has the highest value and does not exceed the capacity K. By introducing a decision variable $x_i \in \{0,1\}$ ($\forall i \in I$) to indicate whether an item is placed in the container or not, this problem can be modeled as the system of linear equations in (KS). The student's task is to design an algorithm that can find a high-quality assignment of the decision variables $\mathbf{x} = \{x_1, x_2, \ldots, x_{|I|}\}$ given any problem input $\mathbf{w} = \{K, v_1, w_1, v_2, w_2, \ldots, v_{|I|}, w_{|I|}\}$.

2.2 Assignment Grading

There are many possibilities for the grading of algorithm design tasks. In the spirit of the "by any means necessary" theme of the assignments, we focus on a very quantitative metric: the solution quality on six to eight problem inputs for each assignment, T. Each input data $i \in T$ is graded on a 10 point scale using four basic categories: 0 points, 3 points, 7 points, and 10 points. Two thresholds (h_i^1, h_i^2) are defined. These thresholds correspond to the objective value of a good and a great solution respectively, where "good" is qualitatively defined as a solution that can be achieved by using the material provided in the lectures and "great" requires the student to go beyond the lecture material. With these values defined, the grading is done by the following algorithm: (1) submissions not conforming to the output specification or infeasible are given 0 points; (2) any feasible solution receives at least 3 points; (3) feasible solutions exceeding the "good" threshold (i.e., $f(\mathbf{w}_i, \mathbf{x}) \leq h_i^1$) receive at least 7 points; and (4) feasible solutions exceeding the "great" threshold (i.e., $f(\mathbf{w}_i, \mathbf{x}) \leq h_i^2$) receive a full 10 points. This point based system is carefully designed to reward two general approaches to the material equally: (1) a high quality approach that does not scale to the largest input values; and (2) a medium quality approach that solves all of the inputs similarly. Notice that $10\frac{|T|}{2} + 3\frac{|T|}{2} \approx 7|T|$. This allows students two viable meta-strategies for approaching the assignments and rewards both similarly.

Student performance on the assignments is evaluated weekly and delivered in class. However, due to the open ended nature of these algorithm design tasks and the independence of the course topics, students are often inspired later in the course to revise their solutions to earlier assignments. Pursuing these inspirations is encouraged by allowing students to return to all previous assignments at any point and improve their grades. The final grade is based on the solution quality on the last day of class.

2.3 Motivation and Gamification

Given the challenging nature of the assignments in discrete optimization, the ability of students to maintain a high level of motivation is critical to their successful completion of the course. Employing a liberal collaboration policy in discrete optimization has the effect of creating a *buzz* about the assignments. Outside the classroom, students are constantly sharing information about the best solutions they have found and how they found them. This informal feedback has an effect of keeping the students engaged and motivated to complete the assignments. However, two additional class activities were used to amplify the effects of this feedback: (1) a friendly class competition was created via a *leader board*, which directly compares student performances at the start of each week; (2) students had to prepare *show-and-tell*-style presentations discussing how they approached the assignments. The most successful algorithms were disseminated through these peer education sessions, bridging the gap between the lecture material and the assignments. The leader

Student	Input 1	Input 2	Input 3
A	$f(\mathbf{w}_1, \mathbf{x}_1^a)$	$f(\mathbf{w}_2, \mathbf{x}_2^a)$	$f(\mathbf{w}_3, \mathbf{x}_3^a)$
B	$f(\mathbf{w}_1, \mathbf{x}_1^b)$	$f(\mathbf{w}_2, \mathbf{x}_2^b)$	$f(\mathbf{w}_3, \mathbf{x}_3^b)$
C	$f(\mathbf{w}_1, \mathbf{x}_1^c)$	$f(\mathbf{w}_2, \mathbf{x}_2^c)$	$f(\mathbf{w}_3, \mathbf{x}_3^c)$
...	...	...	...

Table 1: A Leader Board for Sharing Solution Quality.

board is an essential and fairly unique part of the discrete optimization class and warrants a detailed explanation.

It is interesting to notice that the objective function (f) allows students to share information about the quality of their algorithm without revealing the details. This unique way of sharing information can be used to build a table comparing students performance. Table 1 illustrates how a *leader board* can be constructed. In the table, the leader board indicates the solution quality of three students $\{A, B, C\}$ on three problem inputs $\{1, 2, 3\}$, where $\mathbf{w}_1$ is the input data for problem 1 and $\mathbf{x}_1^a$ is student A's solution to problem 1. The class competition generated by the leader board, not only increases discussion among the students, but also has an additional benefit of pushing the best students to go beyond full credit (i.e., 10 points) and improve their solution to be the best in the class.

2.4 Success on Campus

Discrete optimization is traditionally offered as an advanced undergraduate and introductory graduate level class. The typical enrollment is between twenty and thirty students annually. Based on classroom critical review surveys, the students indicated that: they worked around 15 hours per week on average; the course material was challenging; they learned a lot; and the overall experience was very positive. Common statements captured via students open ended comments include: *"[praise for] the professor's consistent encouragement of student participation"* — *"[the] rigorous assignments were found to be most instrumental to success in this class"* — *"the majority of learning occurred when students worked on the projects."* — *"[the projects] were crazy hard, but crazy fun!"*. As we will see in the next Section, many of these aspects were successfully recreated in the MOOC version.

3. THE MOOC

Adaptation of discrete optimization from the classroom to a MOOC poses three core challenges: (1) how to scale assessments to a huge number of students; (2) how to accommodate the heterogeneity of the student body; (3) how to recreate the community dimension, which fueled the students' motivation to complete the course, and naturally developed through co-location of students on a campus. This section discusses how these three challenges were addressed in the adaption of discrete optimization to the MOOC setting.

Assessment Scaleability. Based on the assignment specification, it is relatively simple to make the grading process fully automated. This makes these assignments ideal for a MOOC setting, where scalability is paramount. Furthermore, the leader board also scales to MOOC delivery. Hence, the assignments and leader board in the MOOC were identical to the classroom version.

Accommodating Heterogeneity. The student body in a MOOC is much less homogenous than in a typical classroom. Many of the students are seeking continuing education. These students have many unpredictable scheduling requirements such as, work, family, and vacation. The age of the students is very broad and the skills of the student body can vary widely. Some may have a degree in the subject area, while others have no exposure at all (including high-school students) [7]. One major advantage of the MOOC format is that this heterogeneity can be at least partially accommodated by making the content always available. Discrete optimization provides the entire course material on the first day of class. This allows students to design a study plan around their life constraints and to go at their own pace, accommodating their varying degrees of proficiency in the subject area.

Community Development. The greatest challenge in the MOOC version of discrete optimization is to replace the community aspect of the class. We hypothesize that the buzz and community, which motivates students in the classroom version, is fostered by a combination of challenging problems, encouragement of teamwork, and iterative, continuos assignment feedback. In an attempt to build a similar sense of community in the MOOC, several feedback mechanism were employed.

Grading is the first feedback provided to the students. In the campus version, grading feedback was provided every two weeks. However, students would implicitly discuss solution quality between these weekly reports. To compensate for this lack of social interactions, we allow the assignments to have an unlimited number of submissions. The students can then seek feedback as often as they wish.

The leader board is the second feedback mechanism. In the classroom version, the leader board was updated weekly and reviewed during the lecture period. To improve feedback in the MOOC, the leader board is continuously updated. This encourages students to stay engaged with their peer's solutions regularly, rather than once a week. Once again, this replaces an activity that happened in the classroom by face-to-face contact and word of mouth.

A very liberal collaboration policy for sharing information on the forums is the third feedback mechanism. Students are encouraged to share their experience on the forums and can provide very detailed accounts (e.g., even pseudo code) of how they achieved good solutions to the assignments. This was complemented with very active participation by the teaching staff in the forum, allowing the staff to guide the class discussion and providing a replacement for campus-based informal instructor feedback, such as office hours.

A weekly *mailbag* video is the final feedback mechanism. These videos include weekly announcements, an official response to popular discussion threads in the forums, and weekly statistics on the class's progress and demographics. Once again, revealing aggregate class statistics to the students provides a feeling for the boarder community they are a part of. Some of the most important weekly statistics, such as the class points plot (see Figure 1), were presented live on the course page so that students could actively engage with the status of the class in real time. Table 2 summarizes the differences between the classroom and MOOC versions of

discrete optimization.

Topic	Classroom	MOOC
Content Presentation	linear	open
Lectures	live in class	recorded video
Assignments Feedback	weekly	real-time
Leader Board	weekly	real-time
Student Discussion	face-to-face	forum
Team Work	pair programming	pseudo code
Professor Feedback	class discussion	weekly mailbag

Table 2: Classroom versus MOOC Course.

4. UNDERSTANDING THE MOOC

Section 2 introduced the pedagogical philosophy of discrete optimization and Section 3 discussed the adaptation to a MOOC including a hypothesis about community development and student motivation. Both sections postulate how the design of discrete optimization affects the student learning experience. This section leverages the huge amount of data produced by a MOOC to provide some evidence of the claims in previous sections. This data does not include rigorous experiments to make these claims irrefutable. However, we believe the evidence provided here is immensely informative for the design of such rigorous experiments and is valuable to the community in its current form. Before discussing the details of the student experience in the discrete optimization MOOC, we will provide some perspective by reviewing the basic class statistics.

4.1 Inaugural Session Overview

The inaugural session of the discrete optimization MOOC ran over a period of nine weeks. During the nine months between the first announcement of discrete optimization and the course launch, 50,000 individuals showed an interest in the class. As is typical of a MOOC, less than 50% (17,000) of interested students went on to attend class and view at least one video lecture [6, 7]. Around 6,500 students experimented with the assignments and around 4,000 of those students made a non-trivial attempt at solving one of the algorithm design tasks.

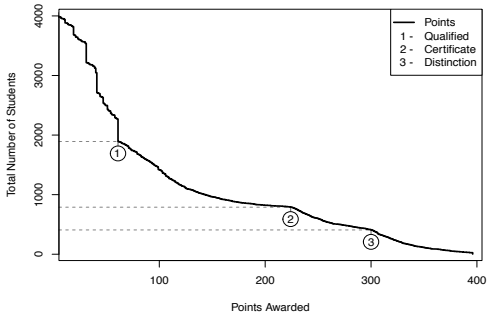

Figure 1: The Cumulative Distribution of Grades.

By the end of discrete optimization, 795 students earned a certificate of completion. This was truly remarkable: less than 500 students graduated from the classroom version in fifteen years of teaching. The typical completion rate calculation of $795/17000 = 4.68\%$ could be discouraging. However, a detailed inspection of the number of points earned by the students is very revealing. Figure 1 presents the total number of students achieving a particular point threshold (i.e., a cumulative distribution of student points). Within the range of 0 and 60 points, there are several sheer cliffs in this distribution. These correspond to students abandoning the assignments as they get stuck on parts of the warm-up knapsack assignment (students meeting the prerequisites should find this assignment easy). At the 60 point mark (*1* in Figure 1), 47% of the students (i.e., 1884) remain. We consider these students to be *qualified* to complete the course material, as they have successfully completed the first assignment. The remainder of the point curve is a smooth distribution indicating that the assignments are challenging and well calibrated. Two small humps occur at locations indicated by *2* and *3*: These correspond to the certificate threshold values. The shape indicates that students who are near a threshold put in some extra effort to pass it. However, the most important result from this figure is that if we only consider the population of active students who attempted the assignments and were qualified, the completion rate is $795/1884 = 42.2\%$.

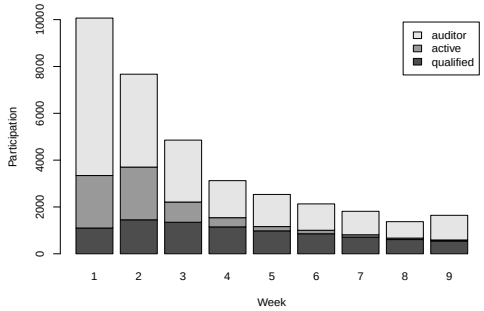

Figure 2: Weekly Student Activity by Auditors, Active Students, and Qualified Students.

Due to the open nature of MOOCs, it is interesting to understand the student body over time. Figure 2 indicates the number of students who were active in the class over the nine-week period. The active students are broken into three categories: *"auditors"* those who only watch videos; *"active"* students who worked on the assignments; and *"qualified"* active students who passed the qualification mark in Figure 1. The steady decline in total participation is consistent with other MOOCs [7], but the breakdown of students into the active and qualified subpopulations is uncommon and revealing. In fact, the retention rate of the qualified students is very good and differs from other student groups.

4.2 Gamification

This section presents results suggesting the positive effect of the leader board in motivating students to push their solutions beyond a grade of 100%. Indeed, there is, in general, a significant gap between a grade of 100% and the best-known solutions on some of the hardest assignments. To illustrate the changes in the leader board over time, Figure 3 provides a detailed example of the submission behavior of a few of the best students on a representative assignment. Each line in the figure represents the submission quality over time of a single student and the solid dots indicate new solution submissions. The horizontal dashed line indicates the quality threshold the student must achieve to get a perfect grade (i.e., h^2). The plot indicates that some students choose to stop working on the problem after achieving 10 points (e.g., the single dot at the center of the figure), while others continue to work on the problem after achieving full credit (e.g.,

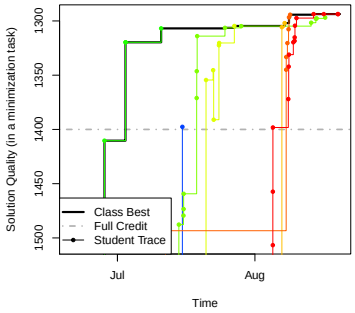

Figure 3: The Submission Behavior of the Best Students on a Particular Assignment over Time.

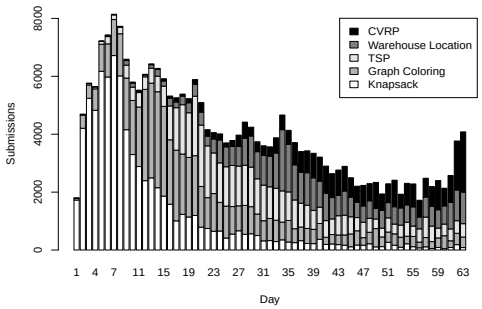

Figure 4: Submissions over Time by Assignment.

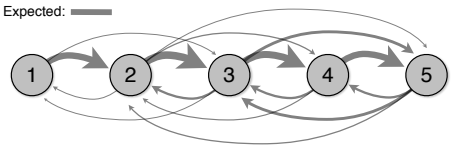

Figure 5: Transition Activity (without self-loops and edges below 10%).

the concentration of points in the bottom right). Although the true motivations of these student is unknown, the submission behaviors are consistent with a competition for the best position on the leader board.

Figure 3 is somewhat anecdotal and it is important to consider the class as a whole. Of the 4,000 students who participated seriously in the assignments, 89.4% received at least one full-credit score and 61.6% went on to make additional submissions beyond the 10 point mark. Focusing on the 61.6% of students that made additional submissions, the majority of those students submitted twice, but over 500 students submitted 3, 4, or 5 times after achieving a perfect score. The most active students submitted over 20 times after getting full credit. These aggregate results indicate that many students are submitting after receiving full credit and beg a more formal study of the leader board to precisely understand student motivations.

4.3 Open Course Structure

Section 3 suggested that the open course structure helped accommodate the heterogeneous student body. This section analyzes the assignment submission behavior of the students to better understand how the open course structure affected assignments. Figure 4 shows the submissions of the assignments on each day, over the 9 weeks (63 days) the course was open. Observe that the number of submissions on a given day greatly exceeds the number of students active in the assignments. This indicates that students are submitting their results for feedback often. The submissions are broken down by assignment to indicate which assignment students are working on. In a classic rolling assignment deadline design, one would expect a series of peeks as students focused on finishing the assignments before the deadline. In this figure, we see only two peaks: on the knapsack assignment in the first two weeks of class and on the last CVRP assignment at the end of class when students put in extra effort just before the final deadline. The lack of peaks in the assignment submissions between the start and end of the course could be explained by two student behaviors: (1) students are taking advantage of new insights and bouncing between assignments to improve their solutions; (2) the students are working at very different rates on the assignments. The next analysis supports these two points and focuses on the 800 students who attempted every assignment at least once.

To understand how students switch between assignments we can model their submission behavior as a state transition diagram, as in Figure 5. States in the diagram are assignments and edges represent the average number of transitions between assignments for the student body. In a classic design of rolling assignment deadlines, this transition diagram would have strong self-loops (i.e., from assignment x to assignment x) where students repeatedly work on the same assignment and strong transitions to the next assignments (i.e., from assignment x to assignment x+1). All other transitions are only possible in an open assignment design. Figure 5 presents the assignment transition behavior in the discrete optimization MOOC. On average, we would expect each student to make one transition between each assignment. This is indicated by an expected line thickness in the diagram. The figure reveals that there is a strong trend for students to move in a serial manner between the assignments. However, a significant number of students move between the assignments. The strongest trend is between assignments 3 (TSP) and 5 (CVRP), which is particularly revealing as there is a strong mathematical connection between those two problems.

The transitions between assignments alone cannot explain the submission distribution in Figure 4. The other explanation is that students are working in a serial manner but the rate at which they complete the assignments is very different. To understand the rate that students complete the assignments, we can look at the total number of days between their first and last submissions. The median value reveals that half of the students took more than 50 days while the other half took less then 50 days to complete the assignments. Several students even completed the course in just two weeks. This wide variety of completion rates helps to explain Figure 4 and indicates the variety of skill levels and time investments of the student body.

4.4 Community Development

The success of the community development goals are the most challenging to demonstrate, but some insight is provided by behavioral statistics. The students took great advantage of the unlimited assignment feedback. Among the 800 students who attempted all five assignments, each student submitted a solution 5.9 times on average. Students

viewed the leader board often, leading to a total of 61,000 views over the class duration. In a survey where students were asked if they enjoyed the leader board, 66% responded "yes", 3% responded "no" and 31% were indifferent. It was observed that the leader board inspired some students to ask direct questions to the best students of the class in the forums. This led to excellent threads where the best students shared their experience on the assignments. Students enjoyed the live analytics of Figure 1, which accumulated 13,000 views over the seven weeks it was available. The mailbags were very popular with number of unique views ranging from 9,000 to 2,000, which represents the vast majority of students active in the course on the week the mailbag was delivered.

The most telling evidence for the successful community development appears in the thousands of free form text that student produced via the class forums and surveys, which are difficult to summarize in this paper. Hundreds of students shared similar sentiments to this representative comment *"the continuous TAs presence in the forums and the mailbags make [the course] feel alive, instead of being simply a pre-recorded set of information pills fed to us."* Despite the apparent benefits discussed here, this topic could be expanded significantly with further study and analysis.

4.5 Success as a MOOC

Although awarding 795 certificates of completion was a great success in itself, there are many other ways to measure a class's success. The goal of discrete optimization was to provide a challenging course where dedicated students would learn a lot. The following statistics from a post-course survey of the students ($n = 622$) indicate those goals were achieved. 94.5% of students said they had a positive overall experience in the course with 40.7% of students marking their experience as excellent. 71.9% of students found the course to be challenging while only 6.11% thought that it was too difficult. The students were very dedicated to the challenging material with 56.6% working more than 10 hours per week. Despite the significant time investment, the vast majority, 93.7%, of students, felt that the assignment grading was fair. 94.5% of students said that they learned a significant amount from the course and 74.9% feel confident in their ability to apply the course material to real-world applications. It is also interesting to compare the MOOC post-course survey data to those collected in the classroom critical review. Table 3 compares four of the key survey questions across both version of the class. Despite the vastly different delivery system, the results are remarkably similar.

Topic	Scale	Classroom $n = 22$	MOOC $n = 622$
Hours per Week	1 to 20	14.5	12.2
Overall Experience	1-pos., 5-neg.	1.50	1.85
Difficulty	1-hard, 5-easy	1.64	1.14
Learned	1-lot, 5-little	1.41	2.04

Table 3: Classroom vs MOOC Experience.

When the students were asked the open ended question, *"My favorite part of this this course is ..."*, many aspects of the course were discussed. However, looking at the top five most common words in their responses, $\langle assignment, 222\rangle$, $\langle programming, 196\rangle$, $\langle lectures, 141\rangle$, $\langle time, 124\rangle$, $\langle search, 84\rangle$, it is clear that the programming assignments were one of the most popular parts of the class. Which is also consistent with the critical review of the classroom version of discrete optimization (see quotes from Section 2.4).

5. CONCLUSIONS

Teaching the creative problem skills required by discrete optimization practitioners is a challenging task. This paper has reviewed our experience in teaching such skills in the classroom and in a MOOC. Both course designs were motivated by discovery-based learning, continuous feedback, and gamification. Surprisingly, the students from both classes reported similar learning experiences in terms of time commitment, overall experience, difficulty, and overall learning, which suggests that the learning experience of the classroom was effectively replicated in the MOOC version. A first look at the data provided by the MOOC suggests that the course design choices, had a positive effect on student motivation and online learning. However, these findings are still anecdotal and we hope this work will support and inspire new, rigorous studies on how these course design choices can create effective online learning experiences at the MOOC scale.

6. ACKNOWLEDGMENTS

The authors would like to thank Victor Pillac and Gregor Kennedy for their criticism and insight in the preparation of this paper, the Learning and Teaching initiative at the University of Melbourne for supporting this work, and the students and teaching assistants of the classroom version. This work was conducted in part at NICTA and is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

7. REFERENCES

[1] H. Banchi and R. Bell. The many levels of inquiry. *Science and Children*, 46(2):26–29, 2008.

[2] J. S. Bruner. The act of discovery. *Harvard Educational Review*, 31:21–32, 1961.

[3] S. Cooper and M. Sahami. Reflections on stanford's moocs. *Commun. ACM*, 56(2):28–30, Feb. 2013.

[4] P. Hyman. In the year of disruptive education. *Commun. ACM*, 55(12):20–22, Dec. 2012.

[5] J. Kay, P. Reimann, E. Diebold, and B. Kummerfeld. Moocs: So many learners, so much potential ... *Intelligent Systems, IEEE*, 28(3):70–77, 2013.

[6] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, pages 170–179, New York, NY, USA, 2013. ACM.

[7] MOOCs@Edinburgh Group. Moocs @ edinburgh 2013: Report #1, may 2013.

[8] C. Severance. Teaching the world: Daphne koller and coursera. *Computer*, 45(8):8–9, 2012.

[9] A. Vihavainen, M. Luukkainen, and J. Kurhila. Multi-faceted support for mooc in programming. In *Proceedings of the 13th annual conference on Information technology education*, SIGITE '12, pages 171–176, New York, NY, USA, 2012. ACM.