The Discrete Optimization MOOC:
An Exploration in Discovery-Based Learning

Introduction

Discrete optimization is a subfield of computer science and mathematics focusing on the task of solving real-world optimization problems, such as the travelling salesman problem. Due to the computational complexity of optimization tasks, the practice of discrete optimization feels more like an art than a science: practitioners are constantly confronted with novel problems and must determine which computational 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 challenging for instructors who must present students with complex problem-solving tasks and keep them motivated to complete those tasks.

Over fifteen years, a classroom-based introduction to discrete optimization was developed and honed at a leading U.S. institution. The class assessments were designed around the ideas of discovery-based learning (Bruner 1961) to provide the students with a simulation of the real-world practice of discrete optimization. The course design was successful, and was highly popular among senior undergraduate and graduate students. The recent surge of interest in Massive Open Online Courses (MOOCs) and readily available platforms (e.g., Coursera, Udacity, and edX) makes a MOOC version of discrete optimization technically possible. But it raises an interesting question: will the discovery-based learning methodology of discrete optimization translate and be successful on an e-learning platform such as a MOOC? Technical reports on large-scale MOOCs are fairly recent (Kizilcec 2013, Edinburgh 2013) and have primarily focused on course demographics and key performance indicators such as completion rates. Few papers discuss the effectiveness of different pedagogical and assessment designs in MOOCs.
This paper is an attempt to shed some light on the effectiveness of teaching problem-solving skills in a MOOC by use of discovery-based learning. It begins with some background about the subject area and the class design motivations. It then turns to data analytics to understand what happened in the inaugural session of the Discrete Optimization MOOC, and concludes with a brief discussion of the success and potential improvements of the MOOC version of the class.

The Discrete Optimization Class

Discrete Optimization is an introductory course designed to expose students to how optimization problems are solved in practice. It is typically offered to senior undergraduate and junior graduate students in computer science curriculums. The prerequisites are strong programming skills, familiarity with classic computer science algorithms, and basic linear algebra. The pedagogical philosophy of the course is that inquiry-based learning is effective in teaching creative problem solving skills.

The course begins with a quick review of Dynamic Programming (DP) and Branch and Bound (B&B), two topics that are often covered in an undergraduate computer science curriculum. It then moves on to an introduction to three core topics in the discrete optimization field, Constraint Programming (CP), Local Search (LS), and Mixed Integer Programming (MIP). The students’ understanding of the course topics is tested through programming assignments. The assignments consist of designing algorithms to solve five optimization problems of increasing difficulty, knapsack, graph coloring, travelling salesman (TSP), warehouse location, and capacitated vehicle routing (CVRP). These algorithm design tasks attempt to emulate a real-world discrete optimization experience, which is, your boss tells you “solve this problem, I don’t care how”. The lectures contain the necessary ideas to solve the problems, but the best technique to apply (DP, B&B, CP, LS, MIP) is left for the students to discover. 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 (Banchi 2008). These assessments are complex monolithic design tasks, a sharp contrast to the quiz-based assessments common to many MOOCs.

The complexity and open-ended nature of these algorithm design tasks allows the students to have many successful solutions. In the classroom version, students are often inspired later in the course to revise their solutions to earlier assignments, based on the knowledge they acquired throughout the course. Inspired by this classroom behavior, the MOOC version of the class adopts an open format. The students are allowed, and encouraged, to revise the assignments during the course. The final grade is based on their solution quality on the last day of class.

Understanding the MOOC

The previous section discussed the basic design of the Discrete Optimization class and the pedagogical philosophy behind it. This section uses the vast amount of data produced by a MOOC to provide some evidence that the MOOC adaptation of Discrete Optimization was successful and the use of discovery-based assessment design can also be effective in an e-learning context. Before discussing the details of the students’ experience in Discrete Optimization, we first review the basic class statistics to provide some context.

Inaugural Session Overview

The inaugural session of Discrete Optimization 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. 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.
By the end of Discrete Optimization, 795 students earned a certificate of completion. This was truly remarkable as 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 1a 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 assessments 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 (mark A in Figure 1a), 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 mark B and mark C: These correspond to the two 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 students who attempted the assignments and were qualified, the completion rate is $795/1884 = 42.2\%$.

Due to the free and open nature of MOOCs, it is interesting to understand the student body over time. Figure 1b indicates the number of students who were active in the class over the nine-week period. The active students were broken into three categories: auditors, those who only watched videos; active, students who worked on the assignments; and qualified, active students who passed the qualification mark in Figure 1a. The steady decline in total participation is consistent with other MOOCs (Edinburgh Group - 2013), 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.

**Discovery-Based Learning**

The use of discovery-based assignments was effective in the classroom version of Discrete Optimization, but it is unclear if it will translate to the MOOC format. It is difficult to measure precisely if the students learned creative problem solving skills, but we can look at their exploration of the course material for an indication.
In a post-course survey of Discrete Optimization, the students were asked to identify which optimization techniques they tried on each assignment. Figure 3 summarizes the students’ responses and Table 1 compares those responses to the best optimization techniques for each problem. Looking at Figure 3, we can see that there is a great diversity among the techniques applied to each problem. This suggests that students took advantage of the discovery process and tried several approaches on each problem. Second by comparing Table 1 and Figure 3, we can see that there is a strong correspondence between the best techniques for a given problem and the ones that most students explored. This suggests that students are picking up on the intuition of how to solve novel optimization problems and applying the correct techniques.

However, the most telling evidence for the success of the discovery-based learning appears in the free form text responses that student produced when asked the open ended question, “My favorite part of this course is…” Many aspects of the course were discussed. However, looking at the frequencies of various words in their responses (see Table 2), indicates that the programming assignments were one of the most discussed elements of the course. Even on par with the lectures. This positive response to the assignments is consistent with student reviews of the classroom version of Discrete Optimization, and further suggests that the discovery-based learning approach was successfully translated to the e-learning platform.
Success of the MOOC

Awarding 795 certificates of completion was a great success in itself, but there are many other ways to measure a class’ 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) indicates that this goal was 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 (Figure 3a). 71.9% of students found the course to be challenging while only 6.11% thought that it was too difficult (Figure 3b). 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 moderate amount from the course (Figure 3c) and 74.9% feel confident in their ability to use the course material in real-world applications.

Lessons Learned

Despite the success of Discrete Optimization, there is significant room for improvement in the course design. The vast number of students in a MOOC has the effect of shining light on all of the problems in the course design, no matter how small. For example one forum thread entitled, “Somewhat torn, don’t feel like I’m learning anything”, discusses some of the challenges students face with discovery-based learning. It is clear that some students found the discovery processes disturbing and would prefer a more structured learning experience. In another thread, “if you’re looking for a new challenge: Find a way to remotivate me!” a student explains how he became discouraged with the discovery-based learning approach after trying several ideas without success. These comments, among others, have inspired us to improve the class by making the exploration process easier. This will be achieved in two ways: (1) revising the introductory course material to include some guidance on how to explore optimization problems and (2) provide supplementary “quick-start” videos on how to get started exploring a particular optimization technique. We hope, by lowering the burden of exploration, more students will get the benefits of discovery-based learning without the frustrations.
Conclusion

Teaching the creative problem solving skills required by discrete optimization practitioners is a challenging task. This paper has presented initial evidence that teaching such creative skills is possible in a MOOC. The essential idea was to use assignments inspired by discovery-based learning, so the students not only learn the core technical skills but how to apply them to unfamiliar tasks. The success of the course design was demonstrated through data analytics, enabled by the wealth of information produced in MOOCs. We believe the significant resource investment required to make the custom discovery-based learning assignments was a great investment in the course, and we hope our experience will inspire other MOOC practitioners to put in the additional effort try discovery-based learning tasks in their classes.

References


MOOCs@Edinburgh Group. MOOCs @ Edinburgh 2013: Report #1. Online: http://hdl.handle.net/1842/6683, accessed on May 2013.