Math 480 - Course Project

Creating a schedule for "Chaos Week"

Derek Rhodeham
Justin Zecak
Davis Zvejnieks

1 Description

Dr. Jonah Ostroff runs MathILy, a 6-week math camp for high school students. Between the the first three weeks and the last two weeks is "chaos week." During "chaos week" a new set of classes are voted on by students, and they are enrolled in a week long program containing their top five classes.

In 2015, there were around 30 available topics, from which students would vote for most interesting. These classes were narrowed down to 10 classes, which the three instructors could teach. Each student could then vote 0 to 3 on the remaining classes, 0 representing no interest, 3 representing most interest. Each student was then assigned five classes, which they would participate in daily for the week. Students were assigned classes not only by their most preferred class, but also by instructor overrides based on their knowledge of students. There were 5 time slots available, so 2 classes were taught at the same time.

Dr. Ostroff would like a schedule for 2016 so that there are no time conflicts between classes an instructor could teach while maximizing the preferences of students. This year the schedule should accommodate 24 students, 4 to 5 instructors, and 15 classes. This would mean three classes will be taught at each of the 5 time slots.

2 Impact

Dr. Ostroff had used a method for creating the schedule by searching for a bipartite graph to represent classes which would share a time slot. This method does not easily scale to finding a 3-matching in a hyper-graph. More complexity arises if the schedule were to accommodate 3 or more simultaneous classes.

This method in 2015 could be solved by hand or by a computer in polynomial time. This method would represent the schedule of 10 classes as a K_{10} graph. Edges would be removed from the graph, if a time conflict occurred with an instructor or student. Then a bipartite graph was found, with each pair of vertices representing classes to occur simultaneous. Additionally, the staff would initially assign about 30% of the classes before using this method, and only after the schedule was made were the rest of the classes assigned.

When scaling this problem to three simultaneous problems, this method becomes more time consuming. We propose a method to automate the process of finding a 3-matching using a computer program to eliminate the time needed to find a working schedule. By translating the problem into a linear programming (LP) problem the schedule would not only be easier to compute, but can be scaled to larger schedules using various LP relaxation methods.

3 Methods

We intend to use LP solving algorithms the issue of 3-matching. By finding sets of 3 connected vertices, we can determine which classes can occur in the same time slot. We will examine relaxation methods and their respective accuracies and time complexities. This will be useful for establishing a method to use on for larger schedules if needed for MathILy in the future. In addition, heuristic approaches will also be examined to reduce computation time for larger schedules.

4 Constraints

Below are the constraints for the schedule. These will be formulated into mathematical equations in the LP, if able. If not the data can be prepared ahead of time to incorporate these restrictions in the solution.

- Hard
  - No student is scheduled for two classes at the same time period.
  - No instructor is schedule for two classes at the same time period.
  - Each class is taught by a qualified instructor.
  - If a student's preference is overwritten by an instructor, he or she must be in the class.

- Soft
  - Maximize preference of each student. Look at average rating of classes, and compare to the preference of classes they are actually assigned.
  - There is an even distribution among classes taught for instructors.
  - Each instructor has a class with every student.
  - Each class has a female student.

5 References

Yuk Hei Chan, a PhD candidate at the University of Maryland, wrote his masters thesis on linear programming relaxations of matching problems in hyper-graphs[1]. He discusses in his in his thesis the time complexity of finding k-matchings, integrality gap of various LP relaxations, and methods to improve the LP relaxations. Chan mentions a polynomial time approximation of k-matching, which could be applied for larger data sets. He also compares k-matching to other problems, such as set packing. This resource will be invaluable in searching for ways to improve schedule creation and adapt it to larger schedules.

[1] https://www.cs.umd.edu/~yhchan/thesis.pdf