Weifan Jiang

Chongyi Xu

381 Project Proposal

First: The university’s situation

Second: Motivation

**Goal:**

To recommend a method that assign TA candidates of different courses, in such a way that respects the following considerations:

* Each candidate must be assigned to at most one course.
* Each course must be assigned an appropriate number of candidates
* Each candidate must be assigned only to the courses for which they are qualified
* Both professors and candidates’ preferences will be satisfied as much as possible

**Motivation:**

Undergraduate teaching assistant candidates in CSE get assigned to positions in different courses by the department. Even though candidates self-report their preferences to different courses, according to fellow CSE TAs, the situation that they got assigned to a course which they indicated “less-preferred”, is still common. Therefore, we would like to develop a method which put student candidate’s preference at maximum consideration, while not producing unpractical assignments.

**Community Partner:**

Hongtao Huang, [hongth@cs.washington.edu](mailto:hongth@cs.washington.edu), undergraduate teaching assistant at CSE.

Tejas Devanur, [tdevanur@uw.edu](mailto:tdevanur@uw.edu), graduate teaching fellow at Math Department

Conversations with professors were also conducted.

**Collecting input data:**

We will need a student candidate’s following quantified metric:

* Preference to each role (grader, teaching assistant, instructor) for each course.
* Qualification to each role for each course, which can be a linear combination of the candidate’s grade while in that course, experience with the course.

We will also need professor’s preference towards each student, which can simply be a ranking.

As for the department’s requirements, we would generally choose several representative courses (such as CSE 142, CSE 143, Math 394 etc) and get an averaged required number of teaching assistants, graders and instructors (if applicable) for different size of courses.

For this project, we will be randomly generating data of professor preferences for our method, and measure the “goodness” of produced assignments.

**Quantifying “goodness” of produced assignments:**

The hard constraints for an assignment:

* Each student may only be assigned to one role of one course.
* The required number of teaching assistants, graders and instructors must be met for each course.

The soft constraints for an assignment:

* Student candidate is assigned to a role as high prefered as possible.
* Student has strong qualification in the role one is assigned.
* Professor of the course (if possible) has strong preference for the assigned student.

The rating for an assignment should be based on these constraints: if any hard constraint fails, the rating should immediately fall to negative (indicating invalid). Else, it should be a linear combination of quantified ratings of soft constraints. Student candidate’s preference should be give a larger weight, since the goal is to let student get matched to top choices as much as possible.

**Possible Methods:**

The Metropolis-Hastings Algorithm:

We can treat all possible assignments of students to courses as a distribution, and the “goodness” score for each assignment being the weight of that assignment in the distribution. Imagine an undirected graph, with each vertex being a possible assignments, and two vertices are connected by an edge if and only if the two corresponding assignments differ by a small fraction. Pick a random starting point on the graph and walk randomly. In each step of random walk, pick a random neighbor of current vertex. Move to the neighbor if the neighbor has a better “goodness” score; else, move to the neighbor or stay at current vertex with a probability. The result of the random walk will be a sample from distribution.

Formulate a linear programming problem:

Similar to the Italian Volleyball League scheduling, we will formulate a linear programming question, with the objective function being a linear combination of soft constraints, and constraints being the strict/hard constraints defined above.

Stable Matching:

The problem can also be viewed as a matching problem. An unstable pair (in this case, say student A prefers course B, students B prefers course A, but student A and B are assigned to course B and A respectively) will cause students to complain that they are not assigned to top choices. Therefore, stable matching can be used to reduce the instability, therefore reduce the complaints. However, we must modify the algorithm to consider of other strict/soft constraints associated with this problem.

These three methods will be used to generate assignments, and the results will be compared to determine the best one.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Gale-Shapley paper

Compare student’s side and professor’s side with generating a lot data.

The randomness does not promise to climb the top of the mountain

Maximal matching (multiple positions) rank the preference

Hungerian algorithm (STUDY IT) property of the results

Maximum flow

One method before Friday

Survey to Tas to have a general data (sample data)

* Monte Carlo