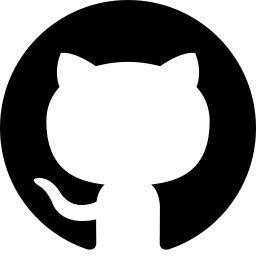
**Optimization of Assignments for**

**Teaching Assistants at UW**

*Visit for more details: https://github.com/weifanjiang/AssignTA*

*Chongyi Xu (*[*chongyix@uw.edu*](mailto:chongyix@uw.edu)*)*

*Weifan Jiang (*[*wfjiang@uw.edu*](mailto:wfjiang@uw.edu)

**Problem Description**

The assignment of different teaching positions is a complicated task. The word "teaching position" includes teaching assistants, graders and instructors here at UW. Each type of position has its unique qualifications and requirements. Some positions require teaching, while other do not. Most students are deemed to be certified to teach. Those whose first language is not English must pass the SPEAK test to be certified. The position qualifications do not only appear in different roles but also in different courses. For instance, the instructor positions would mostly be restricted to graduate students or faculty. Meanwhile, the teaching assistant positions could open to both undergraduate students and graduate students.

Our motivation for the project is from the interview with undergraduate TAs and graduate TAs about their teaching experience in early quarters. They noticed that many times, even though they self-report their preferences, they got assigned to a course which indicated as "less-preferred". Therefore, we would like to recommend a method that assigns candidates to courses, in such 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 candidates' and professors' preferences will be satisfied as much as possible.

**Mathematical Model**

Let represent student candidates, let represent courses. Let represent the number of teaching assistants required for course for

Let

* , if is qualified to teach
* , otherwise

The goal is to produce an assignment of candidates to courses, represented as

* , if is assigned to teach
* , otherwise

, subject to the following hard constraints:

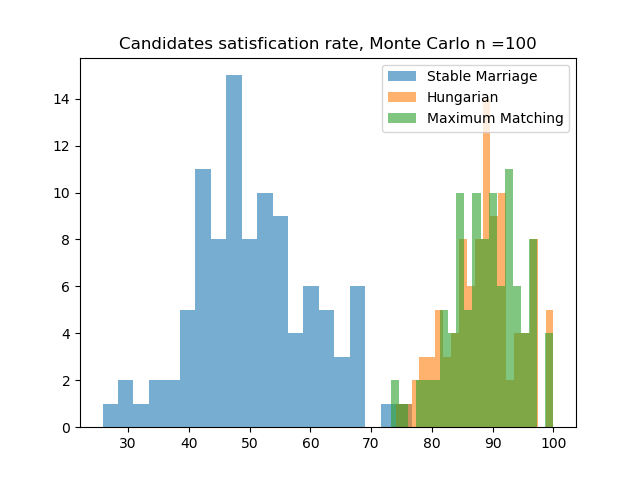
1. Each candidate must be assigned to at most one course.
2. Each course must be assigned an appropriate number of candidates.
3. Each candidate must be assigned only to the courses for which they are qualified.

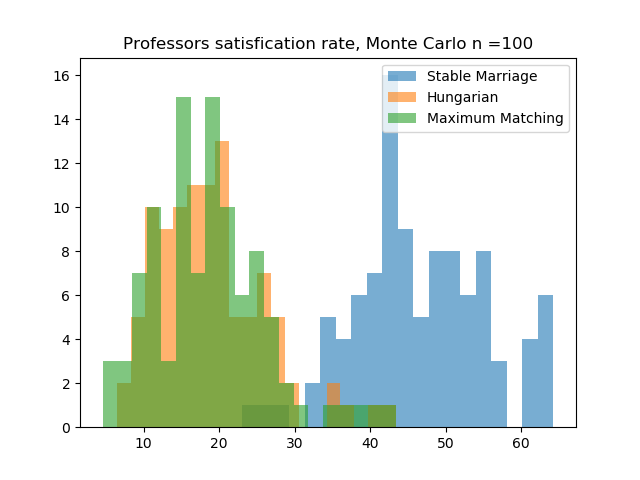
**Conclusion**

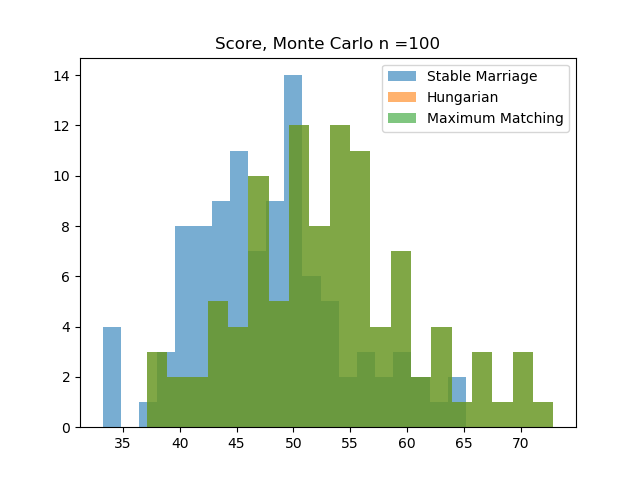
Considering the case with 500 candidates and 20 courses (relatively close to the real-world case), we can see that Maximum Matching produced a best satisfaction rate for candidates, which is around 88%. Hungarian Algorithm has a really close performance, which has an averaged candidates’ satisfaction rate approximately at 85%. Stable Marriage Algorithm does not give back an ideal satisfaction rate for candidates. However, it performs better than other two methods when we are considering professors’ satisfaction rate.

And there is an extremely interesting finding that Hungarian Algorithm always gives back the same score as Maximum Matching does although they had completely different numbers of satisfaction rate. First, we thought that was a marginal case. Meanwhile, as we tested our model using different numbers of candidates and courses, this phenomenon always showed up. After rethinking about our model, we realized it might be caused by the algorithm implementation that both Hungarian and Maximum Matching were looking for a “max” preference score. In the other hand, Stable Marriage focused more on generating a “stable” assignment.

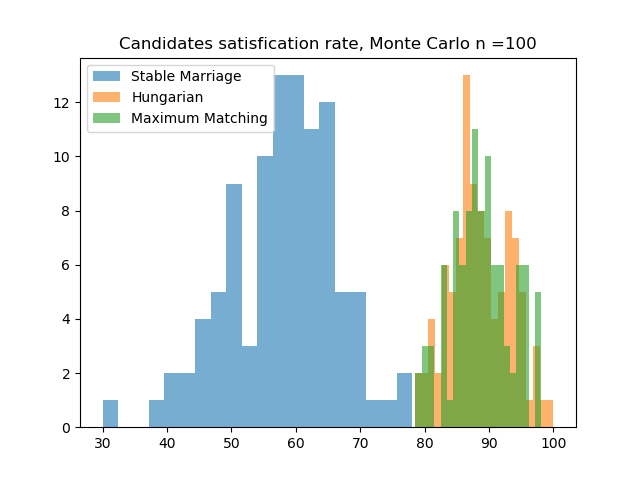
**100 Candidates & 10 Courses**

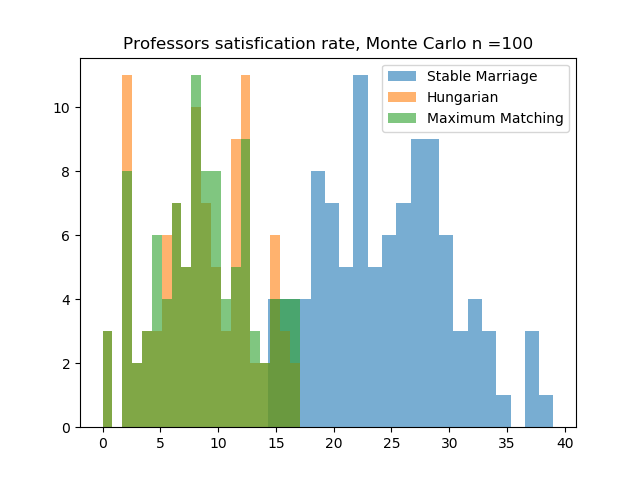


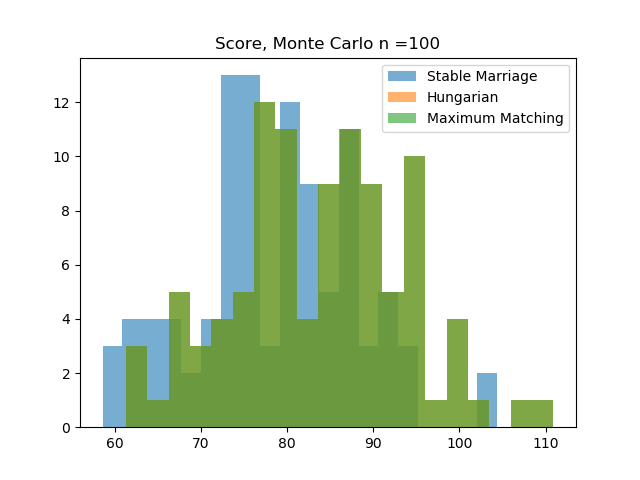




**300 Candidates & 15 Courses**







**500 Candidates & 20 Courses**

