Aaron Lichtblau

LabTA Matching Writeup

**Keywords**

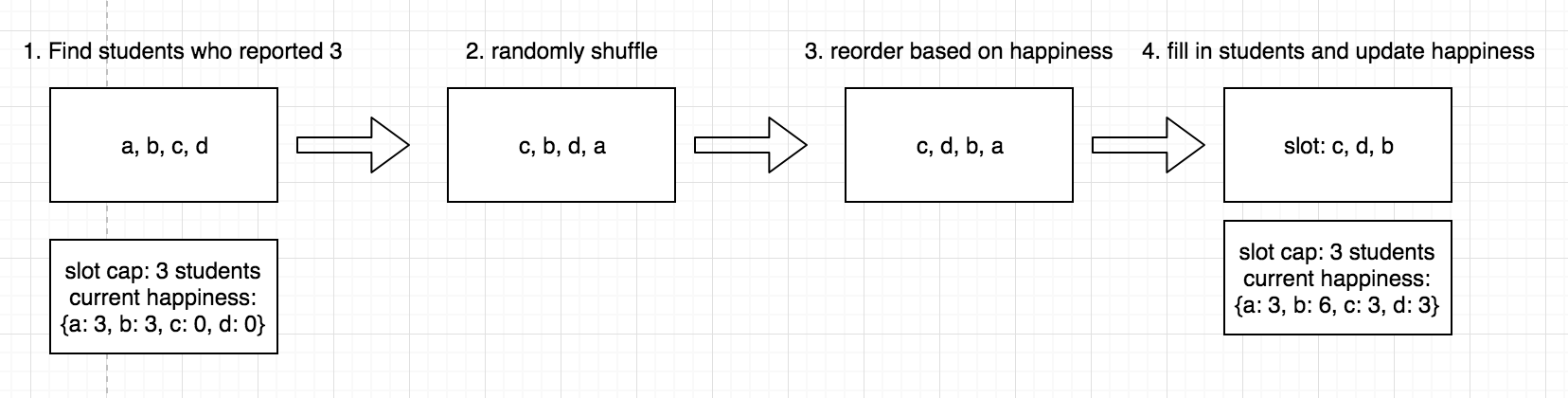
Happiness: this is a measure of how “happy” a student is with the current match. It is calculated as: sum of their preference score (0-3) for each slot they are given.

Availability: this is a measure of how “available” a student is. It is calculated as: number of slots the student marked as 2 or 3.

Experience: a measure of how experienced the student is. Calculated as number of semesters worked as a TA. (currently this is 0 or 1 based on if the TA was on the new hires sheet or not).

**Discussion of matching algorithm**

The algorithm goes slot by slot. For each slot, it finds the students who reported a 3. It randomly shuffles these students, then ranks the shuffled list of students according to their happiness (how happy are they already). The happiest students are put at the back of the list. Then the matcher puts students into the slot from the front of the list until the slot is full. If the list runs out and the slot is still not full, the algorithm searches for the students who reported a 2 and repeats from there.



Please note that the 2hr vs 4hr time slot was not considered. After looking at peoples’ preferences again, their preference between 2hr and 4hr was reflected in how they reported their scores on slots. Thus, it would be unlikely for someone who said that they want a 2hr slot to rank back-to-back slots as 3’s. Furthermore, I thought it of less importance to give people their desired shift type compared to maximizing overall happiness. My reasoning is that if you got a slot that you marked as a ‘3’, it is unlikely that you would complain about that. For example, if someone put back-to-back 3’s and got both, but their preference was for a 2hr shift.

**Important/Useful features:**

1. By looking at how happy students already are, the algorithm takes fairness into account. For example, consider when there are two students who report 3’s on Monday 7-9 and Monday 9-11. If student 1 gets placed into Monday 7-9 and student 2 does not, then student 2 will be ahead of student 1 in consideration for the Monday 9-11 slot.
2. Stats- the happiness, availability and experience statistics give demonstrable properties of the algorithm that allow for comparison between matches.
3. Randomness- by incorporating randomness into the algorithm, each time it is run it produces a different matching. This has the added benefit of being flexible in case a TA needs to be moved out of a slot. The algorithm can be run multiple times to generate different possibilities for filling the slot. Then, the head TA can choose which matching they like best.

**Results**

In total happiness, the sum of all students’ happiness scores at the end of the matching, the algorithm gets: 257. The real matching got a score of 233.

In correlation between availability and happiness, the algorithm reports a correlation coefficient of around .25. The real matching reported -0.055.

In experience per slot, the algorithm reports an average of 0.5, with an average of 2 slots which are all new hires and 2 slots that are all TA’s with past experience. The real matching reports an average of 0.55, with 1 slot that was all new hires and 1 slot that was all TA’s with past experience.

**My Thoughts**

I was happy that the algorithm outperformed the past year’s match in terms of overall happiness. Furthermore, only 1 “2” was given out which means that everyone is very happy with the slots they were given. This puts much greater importance on “fairness”.

However, by allowing TA’s to work up to 8 hours, there was high variance in happiness. When I lowered to 4 hrs cap, the variance was lower than Shirley’s and total happiness is just over Shirley’s (+4 on average). So there is a tradeoff here between keeping everyone similarly happy and maximizing total happiness. If we keep everyone similarly happy by reducing the hours worked cap to 4, the algorithm is comparable to Shirley. However, the algorithm gave out ~5 1’s when the cap was reduced to 4.

The correlation between availability and happiness is promising because it shows that my attempt to consider the slots a student was already given yielded a more “fair” matching. However, I believe that a much higher correlation than .25 is possible. Any thoughts about how to improve the fairness of the matching will be very helpful.

The experience is currently not considered in the matching process. I believe that the large number of potential matchings means that if a matching produces an undesirable pairing of TA’s in a slot, it can easily be run again to get a different pairing. Furthermore, if the data on how many semesters a TA has worked is available, this is ready to be incorporated into the outputted stats on experience. This data would give a more accurate look at how many times the algorithm needs to be run to produce a matching without problematic pairings of TA’s.

Update (6-10):

When I compared the variance of happiness, Shirley's matching had low variance (<2), whereas the algorithm's had ~7. This is because no one received more than 2 slots in Shirley's matching. To lower the variance of happiness, I dropped the possible number of slots a TA could get to 2. It had the desired effect of lowering variance to <2. However, it greatly lowered total happiness to ~235 (just above Shirley's) and gave out ~five 1's (Shirley's gave out one '1'). Thus, the algorithm seems to give a comparable performance to Shirley's manual selection with the additional bonuses of 1. having many parameters to play around with 2. being instantaneous 3. giving different matchings with statistics to compare them by.

Update (6-11):

Added envy-free and incorrect stats: mine does much better than Shirley’s in these regards.

Patched up overlapping slots bug

Update (6-15):

When availability was changed to being sum instead of count, correlation to happiness jumped up. I tried changing the calculation of “happiness” to being normalized (divide by availability) but this showed worse performance (greater variance, more envy, lower correlation)

From Matt Weinberg to Everyone: (10:19 AM)

For every round, make a bipartite graph with a node for every student on the left and a node for every slot on the right Edge between a student and a slot if that student is "eligible" for that slot in this round

Find a maximum matching in this graph, where every student uses \leq 2 (or whatever the constraint is) total slots, and every slot has at most the right # students A round is "successful" if every time slot is filled, and unsuccessful otherwise

Full algorithm:

Round 1: all 3s are eligible Round 2: all 3s + a random 2 from each student Round 2': all 3s + a random 2 from every student who didn't give > 5 3s Terminate at the first round which is successful

all of the weights are between [10, 19] all of the weights are between [100, 102] 3s --> 102, 2s --> 101,

There is some set of k slots such that # total edges going into them is < k ^^ constricting set Hall's Theorem says: if no perfect matching, --> constricting set

<https://lumbroso.youcanbook.me>

6-17

The “smart” ordering combined with corrective function greatly improved the performance of the algorithm. No one is given a 1. The only statistic that is lowered is the correlation between availability and happiness (~.15). This is because the order was decided based on summing the preferences so that harder to fill slots went first.

|  |  |  |
| --- | --- | --- |
|  | Max Weight | Greedy |
| Pros | 1. Easily adaptable weighing function 2. Cleaner code: will be easier to integrate 3. Easier for reach goals: playing around with soft constraints | 1. Outputs many schedules 2. Not too difficult to adapt how slots are given |
| Cons | 1. Outputs 1 schedule 2. No control over variance (everything is assigned at once) 3. Total happiness is slightly lower 4. “black box” matching function makes it harder to deal with constraints like overlaps | 1. A bit messy with inputs and overall code 2. Currently, the correlation of availability to happiness is low |

Stats Comparison:

Shirley Max Weight Greedy

|  |  |  |
| --- | --- | --- |
| Total Happiness: 0.9031007751937985  Availability to happiness correlation: -0.0173767262022999  Variance of happiness: 1.9222222222222216  Envy stats: 148  Incorrect stats: 63 | Total Happiness: 0.9534883720930233  Availability to happiness correlation: 0.31499823942380256  Variance of happiness: 1.2545454545454546  Envy stats: 72  Incorrect stats: 17 | Total Happiness: 0.9767441860465116  Availability to happiness correlation: 0.19461912771629342  Variance of happiness: 0.7909090909090911  Envy stats: 43  Incorrect stats: 16 |

1. Quick recap of statistics used to evaluate a schedule

2. Review of the main goals:

a. maximize happiness

b. ensure incentive compatibility

c. minimize variance of happiness

d. meet specific work experience, slot type (2hr vs 4hr) and other requirements

e. maximize flexibility and ease of use for adding future features (TA rating based on efficiency and other metrics)

3. Explain a problem in my implementation of max weight matching algorithm

4. Explain the current edge weighing function

a. discuss how it aligns with main goals

b. thoughts on how to improve based on psychological intuition of mini survey data

5. Pros and Cons of max weight matching and “Aaron’s” algorithms

a. my brief list of thoughts

b. comparison of statistics

c. discussion of more statistics to use to evaluate

6. Next steps

a. working on presentation for showing Shirley and Maia

b. thoughts on how to implement “soft” constraints

c. “taking it across the finish line”

d. how to elicit student feedback