FYP SUPERVISOR ALLOCATION AS STABLE MARRIAGES

GÖTZ PFEIFFER

1. Stable Marriages

From the wikipedia: The stable marriage problem has been stated as follows:

> Given n men and n women, where each person has ranked all members of the opposite sex in order of preference, marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners. When there are no such pairs of people, the set of marriages is deemed stable.

The Gale-Shapley Algorithm finds a solution to the stable marriage problem (and thus proves that such a solution exists) as follows (following Knuth):

1. Each men in turn, one by one, proposes to the woman on top of his list and crosses her off.
2. The women accepts the proposal if currently not engaged, or if this man ranks higher in her list than her current fiancé, who is then dropped.
3. Immediately after being dropped by his fiancée, this man proposes to the next women on his list, who either accepts or rejects him, and so on, until everyone is happily engaged.

This caricature of the complexities of love life in the real world is surprisingly easy to implement. Here is a version in the `ruby` language (which is used by `schoolmaster`):

```ruby
def stable_marriages(men, women)
  fiances = []    # woman i is engaged to man fiances[i]

  # 1. loop and propose
  men.each_index do |m|

    # 3. keep proposing
    while m
      w = men[m].shift;    # man m proposes to woman w

      # 2. accept or reject
      if not fiances[w] or women[w][m] < women[w][fiances[w]]
        m, fiances[w] = fiances[w], m  # w drops fiances[w] and accepts m
      end
    end
  end
  return fiances
end
```

The input to this function is two lists of n permutations of $\{0, \dots, n-1\}$, the n men's preferences and the n women's preferences. The output is a single permutation of $\{0, \dots, n-1\}$, describing which woman is engaged to which man.

With the resulting matching, there won't be a woman and a man who are both happier running off together than staying with their current partner. Of course, that does not mean everyone got their first choice. They only got a partner they cannot improve on in the presence of the other pairings. In terms of getting your first or second choice, it turns out that the men are fairing slightly better since they get to work on their lists starting from the top. The women, in contrast, will have to wait until someone high on their list comes around and proposes.

2. Supervising FYP

When assigning students to FYP supervisors, in the stable marriages scenario the students play the men and the supervisors play the women.

Students submit partial list of preferences and optionally nominate a partner. Teams of usually 2 students are formed by taking the nominations into account, and pairing up the remaining students randomly, but within their programs. A team's list of preferences is formed by combining the preference lists of its members. Supervisors and the number of projects they supervise are determined by the School, in order to achieve a fair distribution of the workload.

In order to apply the Gale-Shapley Algorithm, the following adjustments are made:

1. A supervisor supervising c projects is represented by c women. Each incarnation has a random permutation as its list of preferences, representing a specific way in which they are unbiased in relation to which students they will supervise.
2. Each team's partial list of preferences is first expanded to contain all incarnations of proposed supervisors, and then completed to a full permutation in a random fashion.
3. If necessary, some fictitious students are introduced to account for a possible oversupply of project supervisions provided by the School.

The Gale-Shapley Algorithm will generate a matching between student teams and supervisor incarnations, where no pair of a team and an incarnation could be happier than with their current match, given the expressed preferences.

Now, as some of the teams' preferences and *all* of the supervisors' preferences are random, most of this 'happiness' is quite meaningless. What does matter, however, is whether a student team is matched with one of their preferred supervisors.

Under the given constraints, it is impossible to match *every* team with a preferred supervisor. Some students list only one or two preferences, some none at all. Some potential supervisors are less visible, or known to the final year students and hence might be listed less often than the number of projects they are required to supervise.

The School's *main objective* in matching student teams with supervisors is to *provide as many teams as possible with a supervisor of their choice.*

With this year's student preferences, a single application of the Gale-Shapley Algorithm typically produces a matching where between 25 and 29 of the 45 teams get a supervisor of their choice. More than 98% of the possible outcomes have less than 32 preferences satisfied. Running the algorithm $10\,000$ times, with different random choices, produces a matching where 35 teams get a preferred supervisor. One such matching forms the basis of this year's allocation of supervisors to student teams.

In any solution to a stable marriage problem, it is absolutely possible that, say, two men can both improve on their preferences by swapping financées. However, none of the women concerned would benefit from swapping in terms of their preferences. In the FYP allocation problem, the supervisors' preferences are random choices, and should be ignored. This is reflected in the School's procedures by allowing student teams to swap supervisors, *provided that the supervisors overall workload does not change, and that all involved agree to swap.*