Algorithm Design

Arden Rasmussen

September 29, 2017

Contents

1 Chapter 1

1.1 A First Problem Stable Matching

1.1.1 The Question

The Stable Matching Problem originated in 1962 from Davis Gale and Lloyd Shapley. Gale and Shapley asked: Given a set of preferences among employers and applicants, can we assign applicants to employers so that for every employer E, and every applicant A who is not scheduled to work for E, at least one of the following two things in the case?

(i) E prefers everyone of its accepted applicants to A; or

(ii) A prefers her current situation over working for employer E.

If this hold, the outcome is stable: individual self-interest will prevent any applicant/employer deal from being made behind the scenes.

Formulating the Problem A "bare-bones" version of the problem can be useful for a basic solution: each of n applicants applies to each of n companies, and each company wants to accept a *single* applicant. This preserves the fundamental issues of the original problem.

Gale and Shapley, observed that this problem can be viewed as devising a system that n men and n women can end up getting married, and everyone is seeking to be paired with exactly one individual of the opposite gender.

So consider a set $M = \{m_1, \ldots, m_n\}$ of n men, and a set $W = \{w_1, \ldots, w_n\}$ of N women. Let $M \times W$ denote the set of all possible ordered pairs of the form (m, w), where $m \in M$ and $w \in W$, a *matching* S is a *set* of ordered pairs, each of $M \times W$, with the property that each member of M and each member of W appears in at most one pair in S. A *perfect matching* S' is a matching with the property that each member of M and each member of W appears in *exactly* one pair in S'.

Matching and perfect matchings are objects that will recur frequently; they arise naturally in modeling a wide range of algorithmic problems.

Now we can add the notion of *preferences* to this setting. Each man $m \in M$ *ranks* all the women; we will say that $mprefers wtow'$ if m ranks w higher than w'. We will refer to the ordered ranking of m as his preference list. We will not allow ties in the ranking. Each woman, analogously, ranks all the men.

Given a perfect matching S, what can go wrong? There is the situation where: There are two pairs (m, w) and (m', w') in S with the property that m prefers w' to w, and w' prefers m to m'. In this canse, there's nothing to stop m and w' from abandoning their current partners and heading off together; the set of marriages is not self enforcing. We'll say that such a pair (m, w') is an *instability* with respect to S: (m, w') does not belong to S, but each of m and w' prefers the otehr to their partner in S.

Our goal, then, is a set of marriages with no instabilities. We'll say that a matching S is *stable* if

(i) it is perfect, and

(ii) there is no instability with respect to S.

Two questions spring immediately to mind:

- Does there exist a stable matching for every set of preference lists?
- Given a set of preference lists, can we efficiently construct a stable match if there is one?

It is important to note that it is possible for an instance to have more than one stable matching.

1.1.2 Designing the Algorithm

Let us consider some of the basic ideas that motivate the algorithm.

- Initially, everyone is unmarried. Suppose an unmarried man m choses the woman w who ranks highest on his preference list and *proposes* to her. Can we declare immediately that (m, w) will be one of the pairs in our final stable matching? Not necessarily: at some point in the future, a man m' whom w prefers may propose to her. On the other hand, it would be dangerous for w to reject m right away; she may never receive a proposal from someone she ranks as highly as m. So a natural idea would be to have the pair (m, w) enter an intermediate state — *engagement*.
- Suppose we are now at a state in which some men and women are *free* — not engaged — and some are engaged. The next step could look like this. An arbitrary free man m chooses the highest-ranked woman w to whom he has not yet proposed, and he proposes to her. If w is also free, them m and w become engaged. Otherwise, w is already engaged to some other man m'. In this case, she determines which of m or m' ranks higher on her preference list; this man becomes engaged to w and the other becomes free.
- Finally, the algorithm will terminate when no one is free; at this moment, all engagements are declared final, and the resulting perfect matching is returned.

Algorithm 1.1.A Gale-Shapley Algorithm

```
Initially all m ∈ M and w ∈ W are free
while there is a man m who is free and hasn't proposed to every woman do
    Choose such a man m
    Let w be the highest-ranked woman in m's preference list to whom m has not yet proposed
    if w is free then
        (m, w) become engaged
    else w is currently engaged to m'
        if w prefers m' to m then
            m remains free
        else w prefers m to m'
            (m, w) become engaged
            m' becomes free
        end if
    end if
end while return the set S of engaged pairs
```