**CS 787: Advanced Algorithms**

# Introduction

Instructor: Dieter van Melkebeek

In this lecture we give a brief overview of the topics to be covered in the course, as well as get a taste of the kinds of arguments we will be using later on by looking at the *Stable Marriage Problem*.

## 1 Course Overview

The first several lectures will cover topics in algorithm design, many of which are covered at greater length in undergraduate algorithms courses. If you are completely unfamiliar with any of these topics, it might be advisable to take a look at one of the books on reserve or another source before the appropriate lecture.

1. Greedy Algorithms
2. Divide and Conquer Approaches
3. Dynamic Programming
4. Flow Problems
5. NP-Completeness

A large part of the course will address ways of "dealing" with NP-hard optimization problems. Recall that these are problems that (unless $\mathsf{P} = \mathsf{NP}$) do not have polynomial time algorithms. One example is VERTEX COVER. A *vertex cover* is a subset of vertices in a graph such that every edge in the graph is incident on at least one of the vertices. The associated optimization problem, VERTEX COVER, is to find a *minimal* vertex cover— i.e., one with the smallest possible number of vertices.

Here are some ways we will address NP-hard optimization problems:

1. Approximation Algorithms: These are efficient algorithms that return solutions guaranteed to be within a certain factor of optimal. In the example of VERTEX COVER, there is a simple algorithm that will always return a vertex cover that is no larger than twice the size of the minimal vertex cover.

2. Exact Algorithms: These are algorithms that find an optimal solution, but run in exponential time. Within exponential time, however, often the most straightforward algorithm is not the most efficient.

3. Heuristics: These are algorithms that do not have proven guarantees on their behavior, but seem to perform well in practice. They are generally a last resort when approximation algorithms aren't good enough and exact algorithms take too long.

Tools that we will bring to bear in these approaches include basic Combinatorics, Linear Programming, Semidefinite Programming, and Randomness.

Time permitting, we may cover other advanced topics.

## 2 The Stable Marriage Problem

For the rest of the lecture, we discuss the *Stable Marriage Problem*. The notes here are given for completeness and clarity, but the powerpoint presentation from lecture can be found on the course webpage.

The problem is, given a set of $n$ boys and $n$ girls, each of whom has a ranked preference list of all members of the opposite sex (with no ties), how should we pair them off?

The first question is, what do we want out of a pairing? A good criterion is that there be no *rogue couples*. We say that if a boy and a girl who are not paired prefer each other to the ones with whom they are paired, they are a rogue couple. Since the members of a rogue couple have an incentive to break up with their assigned partners, we call a pairing with one or more rogue couples *unstable*.

### 2.1 The Traditional Marriage Algorithm

Let's try out the "traditional" marriage algorithm (TMA). We'll see that there is always a stable pairing, and that the traditional marriage algorithm always finds one.

Traditional Marriage Algorithm()
For each day that some boy gets a "no", do:
—Morning
Each girl stands on her balcony.
Each boy proposes under the balcony of the best girl whom he has not yet crossed off.
—Afternoon (for girls with at least one suitor)
To today's best suitor: "Maybe, come back tomorrow."
To any others: "No, I will never marry you."
—Evening
Any rejected boy crosses the girl off his list.
Finally, each girl marries the girl to whom she last said "maybe."

We will argue that the TMA is always well defined, and is guaranteed to terminate. Then, we will show that it always produces a (particular) stable pairing (and a fortiori, that such a pairing always exists). First, some lemmas:

**Lemma 1** (Improvement Lemma)**.** *If a girl ever tells a boy "maybe," then she will always have a suitor that she likes at least as well (or a husband).*

*Proof.* The proof is by induction. For the base case, the statment of the lemma gives us that at some time $t_0$, the girl has a boy $b$ on a string. As an induction hypothesis, we assume that at time $t-1$, she has a boy $b^*$ on a string that she likes at least as well as $b$. Finally, if this is true, then at time $t$ she will only let go of $b^*$ to replace him with someone she likes even better. The lemma follows by the transitivity of "at least as good". □

**Lemma 2.** *No boy can be rejected by all the girls.*

*Proof.* By contradiction. Suppose some boy $b$ is rejected by all the girls. At that point, every girl must have a boy on a string (by Lemma 1). Moreover, all these boys are different (girls don't share balconies) and different from $b$. That is, the $n$ girls have $n$ different suitors, none of whom is $b$. But then there must be at least $n+1$ boys, which is a contradiction! □

Now we show that the TMA terminates:

**Theorem 1.** *The TMA terminates in at most $n^2$ days.*

*Proof.* A "master list" of all $n$ boys' lists starts with a total of $n^2$ girls on it (one entry per boy per girl). Each day that at least one boy gets a "no," at least one girl gets crossed off the master list. Therefore, the number of days is bounded by the original size of the master list. □

We can say more: since no list ever becomes empty (Lemma 2), the number of days is bounded by $n(n-1) \leq n^2$. As an exercise, you can investigate how tight the upper bound $n(n-1)$ is.

**Corollary 1.** *Each girl will marry her absolute favorite of the boys who visit her during the TMA.*

Finally, we show that the result produced by the TMA is a stable pairing (and, since the TMA is guaranteed to terminate with a pairing, this implies that a stable pairing must exist).

**Theorem 2.** *Let $T$ be the pairing produced by the TMA. $T$ is stable.*

*Proof.* Let $b$ and $g$ be any couple in $T$. Suppose $b$ prefers $g^*$ to $g$. In this case, we can argue that $g^*$ prefers her husband to $b$.

During the TMA, $b$ proposed to $g^*$ before he proposed to $g$. Therefore, $g^*$ rejected $b$ for someone she liked better. By Lemma 1, the person she married was also someone she liked better than $b$. □

## 2.2 A Little Fun

Who does better in the TMA, boys or girls? We'll show that in a very strong sense, the boys are better off. In fact, the boys get their best imaginable pairings, and the girls get their worst!

We say that a boy's *optimal girl* is the highest ranked girl for whom there is some stable pairing in which the boy marries her. A boy's *pessimal girl* is the lowest ranked girl for whom there is a stable pairing in which the boy marries her. We say that a pairing is *male-optimal* if every boy marries his optimal girl, and *male-pessimal* if every boy marries his pessimal girl.

**Theorem 3** (The Naked Mathematical Truth!)**.** *The Traditional Marriage Algorithm always produces a male-optimal, female-pessimal pairing.*

*Proof.* Male-optimality:
Let $T$ be the pairing produced by the TMA. Suppose, by way of contradiction, that some boy gets rejected by his optimal girl during the TMA. Let $t$ be the earliest time at which this happens. In particular, at time $t$, some boy $b$ gets rejected by his optimal girl $g$ because she says "maybe" to a preferred $b^*$. Since $b$ is the first boy to be rejected by his optimal girl, $b^*$ has not yet been rejected by his optimal girl. Therefore, $b^*$ likes $g$ at least as well as his optimal girl. On the other hand, since $g$ is $b$'s optimal girl, there must exist some stable pairing $S$ in which they are married. Let $g^*$ be $b^*$'s wife in $S$. Figure 1 shows the picture as it involves $b$, $b^*$, $g$, and $g^*$, where the numbers indicate the relative rankings.

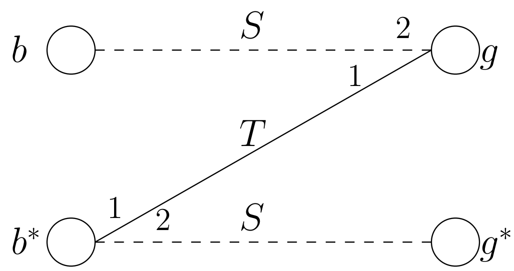


Figure 1: $b^*$ and $g$ are a rogue couple under $S$, so $S$ can't be stable!

As the picture shows, we've already reached a contradiction to the stability of $S$, since under our assumptions $b^*$ and $g$ would prefer each other to their pairings in $S$. Therefore, the TMA pairing must be male-optimal.

Female-pessimality:

We now know that the TMA pairing is male-optimal. Now suppose that it is not female-pessimal. That is, there is a stable pairing $S$ where some girl $g$ does worse than in $T$. Let $b$ be her husband in $T$. Let $b^*$ be her husband in $S$. By assumption, $g$ likes $b$ better than her husband in $S$. We also know that $b$ likes $g$ better than his wife in $S$, since $g$ is his optimal girl. Therefore, $b$ and $g$ are a rogue couple under $S$ and $S$ cannot be stable (see Figure 2). This is a contradiction, and the theorem is proved.

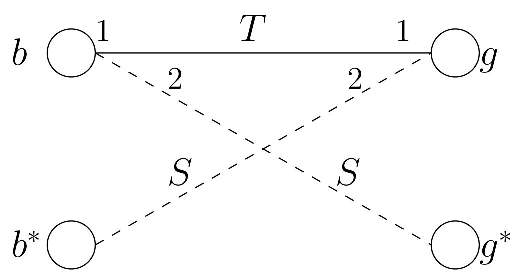


Figure 2: $b$ and $g$ are a rogue couple under $S$, so $S$ can't be stable!

□