CMPSCI 711: “Really Advanced Algorithms”

Lecture 3 – Principle of Deferred Decisions & Stable Matchings

Andrew McGregor

Last Compiled: February 3, 2009

Outline

Recall Last Week's Puzzle

- Take a standard pack of 52 cards that is randomly shuffled.
- Split into 13 piles of 4 and label piles $\{$A,2,. . . ,10,J,Q,K$\}$.
- Take first card from "K" pile.
- Take next card from "X" pile where X is the face value of the previous card taken.
- Repeat until either all cards are removed (you win) or we get stuck (you lose).

What's the probability you win?

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Proof.

- Suppose at iteration j we draw card X but pile "X" is empty.

□

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Proof.

- Suppose at iteration j we draw card X but pile "X" is empty.
- If pile 'X" is empty and X$\neq$K then we have already drawn 4 copies of card X prior to iteration j. Contradiction!

$\square$

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Proof.

- Suppose at iteration j we draw card X but pile "X" is empty.
- If pile 'X" is empty and X$\neq$K then we have already drawn 4 copies of card X prior to iteration j. Contradiction!

$\square$

Lemma

We win iff the fourth K is the 52nd card.

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Proof.

- Suppose at iteration j we draw card X but pile "X" is empty.
- If pile 'X" is empty and X≠K then we have already drawn 4 copies of card X prior to iteration j. Contradiction!

□

Lemma

We win iff the fourth K is the 52nd card.

Proof.

- When 1st, 2nd, or 3rd K is seen we don't terminate because "K" pile is non-empty.

□

Structural Observations

Lemma

The last card before we terminate (either winning or loosing) is K.

Proof.

- Suppose at iteration j we draw card X but pile "X" is empty.
- If pile 'X" is empty and X$\neq$K then we have already drawn 4 copies of card X prior to iteration j. Contradiction!

□

Lemma

We win iff the fourth K is the 52nd card.

Proof.

- When 1st, 2nd, or 3rd K is seen we don't terminate because "K" pile is non-empty.
- Terminate when 4th K is seen: we win iff it's the 52nd card.

□

Principle of Deferred Decisions

- How do we compute the probability that the fourth K is the 52nd card? $\mathbb{P}$[fourth K is 52nd card] equals:

$$\frac{\#\text{ game configurations such that K is 52nd card revealed}}{\#\text{ game configurations}}$$

Principle of Deferred Decisions

- How do we compute the probability that the fourth K is the 52nd card? $\mathbb{P}\,[\text{fourth K is 52nd card}]$ equals:

$$\frac{\#\text{ game configurations such that K is 52nd card revealed}}{\#\text{ game configurations}}$$

- Principle of Deferred Decisions: Let the random choices unfold with the progress of the analysis rather than fixing random events upfront.

Principle of Deferred Decisions

- How do we compute the probability that the fourth K is the 52nd card? $\mathbb{P}$[fourth K is 52nd card] equals:

$$\frac{\#\text{ game configurations such that K is 52nd card revealed}}{\#\text{ game configurations}}$$

- Principle of Deferred Decisions: Let the random choices unfold with the progress of the analysis rather than fixing random events upfront.
- For clock solitaire this means we may assume that at each draw, any unseen card is equally unlikely.

Principle of Deferred Decisions

- How do we compute the probability that the fourth K is the 52nd card? $\mathbb{P}$[fourth K is 52nd card] equals:

$$\frac{\#\text{ game configurations such that K is 52nd card revealed}}{\#\text{ game configurations}}$$

- Principle of Deferred Decisions: Let the random choices unfold with the progress of the analysis rather than fixing random events upfront.
- For clock solitaire this means we may assume that at each draw, any unseen card is equally unlikely.

Theorem

The probability we win clock solitaire is $1/13$.

Outline

Stable Matching Problem

Consider a society in which there are n women $(w_1, \ldots, w_n)$ and n men $(m_1, \ldots, m_n)$.

- A matching is a $1-1$ correspondence between the men and the women (we assume a monogamous, heterosexual society).

Stable Matching Problem

Consider a society in which there are n women $(w_1, \ldots, w_n)$ and n men $(m_1, \ldots, m_n)$.

- A matching is a $1-1$ correspondence between the men and the women (we assume a monogamous, heterosexual society).
- Each female has a (strict) preference list for the males and each male has a (strict) preference list for the females.

Stable Matching Problem

Consider a society in which there are n women $(w_1, \ldots, w_n)$ and n men $(m_1, \ldots, m_n)$.

- A matching is a $1-1$ correspondence between the men and the women (we assume a monogamous, heterosexual society).
- Each female has a (strict) preference list for the males and each male has a (strict) preference list for the females.
- The matching is unstable if there exists w_i and m_j such that
  - w_i and m_j are not matched to each other.
  - w_i prefers m_j to her match.
  - m_j prefers w_i to his match.

Stable Matching Problem

Consider a society in which there are n women $(w_1, \ldots, w_n)$ and n men $(m_1, \ldots, m_n)$.

- A matching is a $1-1$ correspondence between the men and the women (we assume a monogamous, heterosexual society).
- Each female has a (strict) preference list for the males and each male has a (strict) preference list for the females.
- The matching is unstable if there exists w_i and m_j such that
  - w_i and m_j are not matched to each other.
  - w_i prefers m_j to her match.
  - m_j prefers w_i to his match.
- A configuration that is not unstable is stable.

Stable Matching Problem

Consider a society in which there are n women $(w_1, \ldots, w_n)$ and n men $(m_1, \ldots, m_n)$.

- A matching is a $1-1$ correspondence between the men and the women (we assume a monogamous, heterosexual society).
- Each female has a (strict) preference list for the males and each male has a (strict) preference list for the females.
- The matching is unstable if there exists w_i and m_j such that
  - w_i and m_j are not matched to each other.
  - w_i prefers m_j to her match.
  - m_j prefers w_i to his match.
- A configuration that is not unstable is stable.

Does a stable matching always exist? Can we find one in polynomial time?

The Gale-Shapley "Proposal" Algorithm

- Let i be the smallest value such that m_i is unmatched.

The Gale-Shapley “Proposal” Algorithm

- Let i be the smallest value such that m_i is unmatched.
- m_i proposes to the most desirable woman (according to his list) that hasn't already rejected him.

The Gale-Shapley "Proposal" Algorithm

- Let i be the smallest value such that m_i is unmatched.
- m_i proposes to the most desirable woman (according to his list) that hasn't already rejected him.
- She accepts if either a) she is currently unmatched, or b) she finds m_i more desirable than her current match (in which case her current match becomes unmatched.)

The Gale-Shapley "Proposal" Algorithm

- Let i be the smallest value such that m_i is unmatched.
- m_i proposes to the most desirable woman (according to his list) that hasn't already rejected him.
- She accepts if either a) she is currently unmatched, or b) she finds m_i more desirable than her current match (in which case her current match becomes unmatched.)
- Repeat until there are no unmatched men left.

The Gale-Shapley "Proposal" Algorithm

- Let i be the smallest value such that m_i is unmatched.
- m_i proposes to the most desirable woman (according to his list) that hasn't already rejected him.
- She accepts if either a) she is currently unmatched, or b) she finds m_i more desirable than her current match (in which case her current match becomes unmatched.)
- Repeat until there are no unmatched men left.

Does the algorithm terminate? Is the resulting matching stable?

Algorithm is Well-Defined

Lemma

Whenever there's an unmatched man m_i, there is someone he hasn't proposed to.

Algorithm is Well-Defined

Lemma

Whenever there's an unmatched man m_i, there is someone he hasn't proposed to.

Proof.

- Once a woman becomes matched she doesn't become unmatched (although she may change her match.)

□

Algorithm is Well-Defined

Lemma

Whenever there's an unmatched man m_i, there is someone he hasn't proposed to.

Proof.

- Once a woman becomes matched she doesn't become unmatched (although she may change her match.)
- All the woman to which m_i proposed are already matched.

□

Algorithm is Well-Defined

Lemma

Whenever there's an unmatched man m_i, there is someone he hasn't proposed to.

Proof.

- Once a woman becomes matched she doesn't become unmatched (although she may change her match.)
- All the woman to which m_i proposed are already matched.
- If m_i has proposed to everyone, all the women are matched, hence all the men are matched. Contradiction!

$\square$

Algorithm is Efficient

Theorem

The algorithm terminates after $O(n^2)$ repeats.

Algorithm is Efficient

Theorem

The algorithm terminates after $O(n^2)$ repeats.

Proof.

- At each stage of the algorithm, let t_i be the number of women to which m_i could still potentially propose.

$\square$

Algorithm is Efficient

Theorem
The algorithm terminates after $O(n^2)$ repeats.

Proof.

- At each stage of the algorithm, let t_i be the number of women to which m_i could still potentially propose.
- At each step $\sum_{i\in[n]} t_i$ decreases by 1.

$\square$

Algorithm is Efficient

Theorem

The algorithm terminates after $O(n^2)$ repeats.

Proof.

- At each stage of the algorithm, let t_i be the number of women to which m_i could still potentially propose.
- At each step $\sum_{i \in [n]} t_i$ decreases by 1.
- Initially $\sum_{i \in [n]} t_i = n^2$ so there can be at most n^2 steps.

$\square$

Algorithm is Correct

Theorem

The matching found by the Gale-Shapley algorithm is stable.

Algorithm is Correct

Theorem

The matching found by the Gale-Shapley algorithm is stable.

Proof.

- Proof by contradiction: Suppose matching includes m_i-w_j and m_k-w_l but m_i and w_l prefer to be matched to each other.

□

Algorithm is Correct

Theorem

The matching found by the Gale-Shapley algorithm is stable.

Proof.

- Proof by contradiction: Suppose matching includes m_i-w_j and m_k-w_l but m_i and w_l prefer to be matched to each other.
- Since m_i prefers w_l to w_j, he must have proposed to w_l before he proposed to w_j.

$\square$

Algorithm is Correct

Theorem

The matching found by the Gale-Shapley algorithm is stable.

Proof.

- Proof by contradiction: Suppose matching includes m_i-w_j and m_k-w_l but m_i and w_l prefer to be matched to each other.
- Since m_i prefers w_l to w_j, he must have proposed to w_l before he proposed to w_j.
- But then, w_l must prefer her current match to m_i: either she already had a better match when m_i proposed or she matched m_i initially and then got a better proposal. Contradiction!

$\square$

Outline

Probabilistic Analysis

- In randomized algorithms, we consider algorithms that make random choices and investigate what happens when they process a "fixed input." E.g., the min-cut algorithm from lecture 1.

Probabilistic Analysis

- In randomized algorithms, we consider algorithms that make random choices and investigate what happens when they process a "fixed input." E.g., the min-cut algorithm from lecture 1.
- In probabilistic analysis, we consider random input and investigate what happens when it's processed by a fixed algorithm. E.g., the Gale-Shapley algorithm when the preference lists are random.

Probabilistic Analysis

- In randomized algorithms, we consider algorithms that make random choices and investigate what happens when they process a "fixed input." E.g., the min-cut algorithm from lecture 1.
- In probabilistic analysis, we consider random input and investigate what happens when it's processed by a fixed algorithm. E.g., the Gale-Shapley algorithm when the preference lists are random.

Theorem

If the preference lists are random, the expected number of iterations of Gale-Shapley is $\leq nH_n$.

Probabilistic Analysis of Gale-Shapley Algorithm (1/2)

- Principle of deferred decision: we may assume that at each step m_i proposes to a woman chosen uniformly at random from the women that have no yet rejected him.

Probabilistic Analysis of Gale-Shapley Algorithm (1/2)

- Principle of deferred decision: we may assume that at each step m_i proposes to a woman chosen uniformly at random from the women that have no yet rejected him.
- To simplify things, use a modification of the Gale-Shapley algorithm, the "amnesiac" algorithm.

Probabilistic Analysis of Gale-Shapley Algorithm (1/2)

- Principle of deferred decision: we may assume that at each step m_i proposes to a woman chosen uniformly at random from the women that have no yet rejected him.
- To simplify things, use a modification of the Gale-Shapley algorithm, the "amnesiac" algorithm.
  - At each step, m_i proposes to a woman uniformly at random from the set of all n women.

Probabilistic Analysis of Gale-Shapley Algorithm (1/2)

- Principle of deferred decision: we may assume that at each step m_i proposes to a woman chosen uniformly at random from the women that have no yet rejected him.
- To simplify things, use a modification of the Gale-Shapley algorithm, the "amnesiac" algorithm.
  - At each step, m_i proposes to a woman uniformly at random from the set of all n women.
  - This doesn't change the outcome of the algorithm since, if m_i was rejected by w_j before, he'll be rejected again.

Probabilistic Analysis of Gale-Shapley Algorithm (1/2)

- Principle of deferred decision: we may assume that at each step m_i proposes to a woman chosen uniformly at random from the women that have no yet rejected him.
- To simplify things, use a modification of the Gale-Shapley algorithm, the "amnesiac" algorithm.
  - At each step, m_i proposes to a woman uniformly at random from the set of all n women.
  - This doesn't change the outcome of the algorithm since, if m_i was rejected by w_j before, he'll be rejected again.
  - The expected running time of the modified algorithm is an upper bound for the running time of original algorithm.

Probabilistic Analysis of Gale-Shapley Algorithm (2/2)

Theorem

If the preference lists are random, the expected number of iterations of Gale-Shapley is at most nH_n.

Proof.

Since the algorithm terminates once all women have received at least one proposal, the random process is analogous to the coupon collector problem. $\square$

Outline

Readings

For next time, please make sure you've read:

- Appendix C: Basic Probability Theory (9 pages)
- Chapter 1: Introduction up to section 1.4 (14 pages)