STABLE HUSBANDS

D E KNUTH

Rapporteurs: Luiz E Buzato and Paul D Ezhilchelvan

Stable Husbands (extended abstract)

The purpose of this lecture is to demonstrate the use of three techniques for analyzing the behavior of combinatorial algorithms on random data. The techniques are called "late binding," "tail inequalities," and "negligible perturbation."

The example problem. Suppose n boys and n girls have each ranked the members of the opposite sex. A *stable matching* is a way to pair them up so that no boy and girl prefer each other to the partners they have been assigned. For example, if the preferences are

$$\begin{array}{llcll} A: & Y > X > Z > W & \qquad & W: & A > B > D > C \\ B: & X > W > Y > Z & & X: & C > A > D > B \\ C: & W > Y > X > Z & & Y: & B > D > A > C \\ D: & X > W > Z > Y & & Z: & B > A > C > D \end{array}$$

then (AW, BX, CY, DZ) is unstable because A prefers X to W and X prefers A to B. But (AY, BW, CX, DZ) and (AZ, BW, CX, DY) turn out to be stable, and these are in fact the only stable matchings for the stated preferences.

The *stable husbands* of a girl are the boys she can be paired with in a stable matching. Thus, A has two stable husbands (namely Y and Z) when the preferences are as shown, while B has only one (namely W).

The algorithm. Here is an algorithm that outputs all the stable husbands of a given girl G, for a given set of preferences. The basic idea is to maintain partial matchings in which each boy who currently has a partner is paired with his best possible choice, among all partial matchings not yet ruled out, for which G is paired with somebody not yet output. One of the boys who doesn't have a current partner is temporarily called P; he will *propose* to one of the girls, and she will decide whether to accept or reject his proposal (at least for the time being). The role of P passes from boy to boy according to the following rules:

A0. Initially all boys and girls are unpaired.

A1. If at least one boy has no current partner, let P be one such boy and proceed to A2. Otherwise let P be the current partner S of the special girl G. Output S and remove the pair GS from the current matching. (The matching was stable, so S was one of G's stable husbands.)

A2. If P has already proposed to all the girls, terminate the algorithm. Otherwise let H be the girl P likes best among all those he hasn't approached so far; P now proposes to H.

A3. If H has already received a proposal from a boy she prefers to P, she rejects P's offer. Otherwise she accepts, tentatively; the pair HP enters the matching. If H has no current partner when she accepts P, the algorithm continues at A1; otherwise the algorithm continues at A2, with P equal to the boy just rejected by H. ☺

Late binding. Our goal is to estimate the number of stable husbands output by Algorithm A when the preferences of boys and girls are independently and uniformly random. The technique of "late binding" replaces Algorithm A by a related procedure, Algorithm B, which lets the preferences unfold dynamically to whatever extent the algorithm needs them as it runs. Whenever a boy is asked to propose in Algorithm B, he proposes to a random girl, who rejects him if he has already asked her at least once before. When a girl receives her kth nonredundant proposal, she accepts with probability $1/k$. A stochastic process with these characteristics has the same number of outputs as Algorithm A would have on random preferences, because it has the same transition probabilities between states.

The modified algorithm maintains the following variables, where j ranges from 1 to n:

A_j = set of girls proposed to so far by boy j.
l = number of boys who have played the role of proposer.
p = the boy who is currently proposing.
h = the girl who is being proposed to.
x_j = the boy who has made the best offer to girl j, or 0.
k_j = number of nonredundant proposals to girl j.

B0. Set $A_j \leftarrow \emptyset$, $x_j \leftarrow 0$, $k_j \leftarrow 0$ for $1 \le j \le n$, and set $l \leftarrow 0$.

B1. If $l < n$, increase l by 1 and set $p \leftarrow l$. Otherwise output x_g (where g represents the special girl G) and set $p \leftarrow x_g$, $x_g \leftarrow 0$.

B2. Let h be a random number, uniform in $[1 \mathinner{..} n]$. If $h \in A_p$, repeat step B2 (a proposal by p to h is redundant). Otherwise replace A_p by $A_p \cup \{h\}$.

B3. Increase k_h by 1. With probability $1 - 1/k_h$, return to B2 (the girl rejects the proposal). Otherwise interchange $p \leftrightarrow x_h$ (she accepts); if now $p = 0$, return to B1, otherwise return to B2. ☺

Notice that Algorithm B never terminates. This simplifies the analysis of the number of outputs produced.

Tail inequalities. Although Algorithm B is rather complicated, we can prove that it will usually have a fairly simple behavior. The proof is based on showing that the "tails" of certain probability distributions are small; i.e., that certain random variables rarely assume values far from their mean.

Let $P(z) = p_0 + p_1 z + p_2 z^2 + \cdots$ be the probability generating function for a random variable X that assumes nonnegative integer values. The *tail inequalities* assert that

$$\begin{aligned} \Pr(X \le r) &\le x^{-r} P(x) \qquad \text{when } 0 < x \le 1; \\ \Pr(X \ge r) &\le x^{-r} P(x) \qquad \text{when } x \ge 1. \end{aligned}$$

These inequalities follow immediately from the observation that $p_k \le x^{-r} p_r x^k$ when $0 < x \le 1$ and $k \le r$, and also when $x \ge 1$ and $k \ge r$.

We can often obtain excellent bounds on the probability that X is large or small by choosing x so that the right-hand side of a tail inequality is minimized. For example, let X be the number of successes in N independent trials when there is probability $1/n$ of success on each trial. (This is the number of times a particular girl h is selected during N executions of step B2.) The probability generating function for X is $((n-1+z)/n)^N$, and the mean value of X is is N/n. The probability that X is at most half the mean is

$$\Pr\left(X \le \frac{N}{2n}\right) \le \left(\frac{1}{2}\right)^{-r} P\left(\frac{1}{2}\right) = 2^r\left(1-\frac{1}{2n}\right)^{2nr} \le 2^r e^{-r} = \left(\frac{2}{e}\right)^r,$$

using the first tail inequality with $r = \frac{1}{2}N/n$ and $x = \frac{1}{2}$, because $1+s \le e^s$ for all real values of s. This quantity $(2/e)^r$ goes to zero exponentially fast as $r \to \infty$. Similarly, the probability the X is at least twice the mean is

$$\Pr\left(X \ge \frac{2N}{n}\right) \le 2^{-r}P(2) = 2^{-r}\left(1+\frac{1}{n}\right)^{nr/2} \le 2^{-r}e^{r/2} = \left(\frac{\sqrt{e}}{2}\right)^r.$$

(Here we have used the second tail inequality with $r = 2N/n$ and $x = 2$.) It follows that X will be between $\frac{1}{2}N/n$ and $2N/n$ except with exponentially small probability.

Negligible perturbation. A stochastic process like Algorithm B can be viewed as an infinite tree, with branches that correspond to random transitions between states. In the particular case of Algorithm B we can let each node α of the tree represent a computation path to the beginning of step B2, with $2n$ branches leading to subsequent nodes $\alpha_1^a, \ldots, \alpha_n^a, \alpha_1^r, \ldots, \alpha_n^r$, where α_j^a means that the next value of h was j and a proposal was accepted, while α_j^r means that the next value was j and a proposal was redundant or rejected.

The probability of going from α to α_j^a is 0 if $j \in A_p(\alpha)$, or $1/(k+1)n$ if $j \notin A_p(\alpha)$ and $k = k_j(\alpha)$, where $A_p(\alpha)$ and $k_j(\alpha)$ are the values of A_p and k_j defined by Algorithm B's path from the root of the tree to node α. The probability of going from α to α_j^r is, similarly, $1/n$ if $j \in A_p(\alpha)$, or $k/(k+1)n$ if $j \notin A_p(\alpha)$ and $k = k_j(\alpha)$. These probabilities define the behavior of Algorithm B. We write $\Pr(\alpha)$ for the probability of reaching node α.

Suppose we perturb some of the branching probabilities in the tree, changing Pr to another probability distribution Pr$'$ that is easier to deal with. Let C be the set of all nodes at level N of the tree that lie beneath a perturbed probability, and let $\Pr(C)$ be sum of $\Pr(\alpha)$ for all $\alpha \in C$. Then

$$\Pr(\text{not } C) = \sum_{\alpha \notin C} \Pr(\alpha) = \sum_{\alpha \notin C} \Pr'(\alpha) = \Pr'(\text{not } C),$$

so it must be true that

$$\Pr(C) = \Pr'(C).$$

If $\Pr(C)$ is small, the perturbation will have a negligible effect on the probability of an arbitrary event E at time N, i.e., at an arbitrary set of nodes at level N. For we have

$$\begin{aligned}|\Pr(E) - \Pr'(E)| &= \left|\sum_{\alpha\in E} \Pr(\alpha) - \sum_{\alpha\in E} \Pr'(\alpha)\right| \\ &= \left|\sum_{\alpha\in E\cap C} (\Pr(\alpha) - \Pr'(\alpha))\right| \\ &\le \sum_{\alpha\in C} |\Pr(\alpha) - \Pr'(\alpha)| \le \sum_{\alpha\in C} (\Pr(\alpha) + \Pr'(\alpha)) = 2\Pr(C).\end{aligned}$$

Thus we can conclude that the original algorithm will have essentially the same behavior as the easier-to-analyze one.

Application. The three basic ideas (late binding, tail inequalities, negligible perturbation) can now be brought together as follows. We say that an event E_N occurs a.s. ("almost surely") if $\Pr(\text{not } E_N) \to 0$ as $N \to \infty$. And we say that E_N occurs q.s. ("quite surely") if $\Pr(\text{not } E_N) \to 0$ superpolynomially fast as $N \to \infty$; this means that $\Pr(\text{not } E_N) = O(N^{-m})$ for all fixed exponents m.

If M events $E_N^{(1)}, E_N^{(2)}, \ldots, E_N^{(M)}$ each occurs q.s., and if M is bounded by any polynomial in N, then the combined event

$$E_N^{(1)} \text{ and } E_N^{(2)} \text{ and } \cdots \text{ and } E_N^{(M)}$$

also occurs q.s.

Let $N = n^{1+\delta}$ where δ is a constant, $0 \le \delta \le \frac{1}{2}$. We will study the first N levels of the tree that corresponds to Algorithm B, i.e., the first N proposals (including redundant ones).The tail inequalities prove

Lemma 1. Each girl q.s. receives at least $\frac{1}{2}n^\delta$ proposals and at most $2n^\delta$ proposals (including redundancy). Indeed, we proved earlier that the probability she doesn't is at most

$$\left(\frac{2}{e}\right)^{n^\delta/2} + \left(\frac{\sqrt{e}}{2}\right)^{2n^\delta}.$$

We can perturb the probabilities at nodes of the tree where a girl has received fewer than $\frac{1}{2}n^\delta$ or more than $2n^\delta$ proposals, without affecting the overall behavior significantly. Thus we can assume that Lemma 1 holds always, not only q.s. This leads to

Lemma 2. Each boy q.s. begins at most $2n^\delta$ runs of proposals (i.e., sequences of proposals before he is accepted).

Lemma 2 in turn allows us to make further perturbations and we can prove the following sequence of lemmas as we zero in on the algorithm's probable behavior:

Lemma 3. Each run q.s. contains at most $n^\delta(\log n)^2$ nonredundant proposals.

Lemma 4. Each boy q.s. proposes to at most $2n^{2\delta}(\log n)^2$ girls.

Lemma 5. Each run q.s. contains at most $n^\delta(\log n)^2$ proposals.

Lemma 6. Each boy q.s. makes at most $2n^{2\delta}(\log n)^2$ proposals.

Lemma 7. Each boy q.s. proposes to a given girl at most $\log n$ times.

Lemma 8. Each girl q.s. receives at least $\frac{1}{2}n^\delta/\log n$ nonredundant proposals.

Lemma 9. A girl who receives m nonredundant proposals a.s. accepts at least $(1-\epsilon)\ln m$ and at most $(1+\epsilon)\ln m$ of them.

Theorem. *Algorithm B a.s. produces at least* $\frac{1-\epsilon}{2}\ln n$ *and at most* $(1+\epsilon)\ln n$ *outputs.*

Corollary. *A girl almost surely has between* $\frac{1}{2}\ln n$ *and* $\ln n$ *stable husbands, when preferences are random.*

Complete details appear in the paper "Stable Husbands" by Knuth, Motwani, and Pittel, *Discrete Structures and Algorithms* **1** (1990), 1–19.

—*Donald E. Knuth*
Stanford University
March 31, 1992

DISCUSSION

Rapporteur: Luiz E Buzato and Paul D Ezhilchelvan

Lecture One

During the talk, Professor Andrew Tanenbaum sought clarification over the speaker's notion of stable matching. The speaker explained citing an example in which a boy gets his favourite girls while the girls do not. He confirmed that such asymmetry is permitted within his definition of a stable match.

Professor John McCarthy wanted to know whether the proposed solution will be correct, if girls, rather than boys, are allowed to propose. The author said that the solutions will still stand correct except that boys will get stable wives.

During the after-talk discussion, Professor McCarthy asked whether the speaker considered more realistic situations where the priorities of the girls were not assigned randomly but based on some pre-defined criteria. The speaker answered saying that such situations have been considered to some extent, he also cited the extreme case where all girls have the same ranking for the boys, in this case the matching problem becomes a hashing problem.

Professor Michael Rabin continued asking about the practical relevance of the algorithm, by citing the problem of the assignment of internal students to hospitals. He wondered whether hospitals or students should have the priority of choice (as boys had the choice over girls in the algorithm described). The author remarked that the hospitals should have the right to assign students, given that there is a high level competition among students for better hospitals. As a consequence, hospitals will get the optimal choice.