Random Matching Markets

(joint work with Simon Mauras and Adrian Vetta)

Paweł Prałat

Updated: 2024/02/13

Department of Mathematics, Toronto Metropolitan University

Toronto Metropolitan University

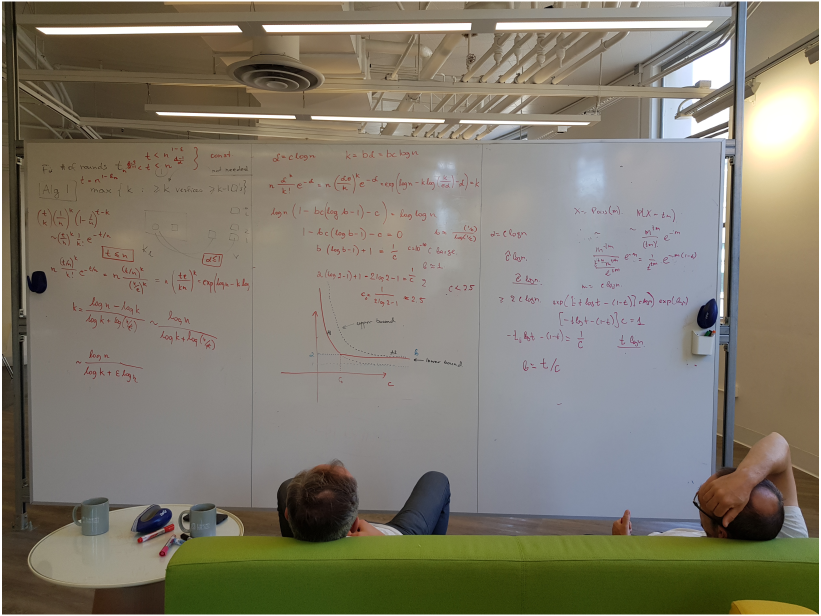
Recent Visits to UC Berkeley

Simons Institute for the Theory of Computing, Fall 2022

– Graph Limits and Processes on Networks: From Epidemics to Misinformation

– Data-Driven Decision Processes

Working hard....or hardly working?



Recent Visits to UC Berkeley

Simons Laufer Mathematical Sciences Institute, Fall 2023

– Algorithms, Fairness, and Equity

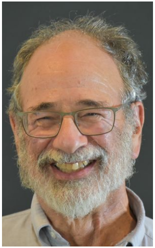
– Market and Mechanism Design (?)

Market and Mechanism Design

Many economists, game theorists, including:

– Paul Milgrom (Stanford), 2020 Nobel Memorial Prize in Economic Sciences “for improvements to auction theory and inventions of new auction formats.”

– Alvin Roth (Stanford), 2012 Nobel Memorial Prize in Economic Sciences “for the theory of stable allocations and the practice of market design."

I started three projects with this group:

– Maximizing Trades in Random Markets
(Nick Arnosti, Alan Frieze)

– Zero-intelligence Traders on a Random Network
(Nick Arnosti, Bogumil Kaminski, Mateusz Zawisza)

– Random Matching Markets
(Simon Mauras and Adrian Vetta)

Stable Matchings

Definition – Matching

– Collection $\mathscr{D}$ of "doctors" and $\mathscr{H}$ of "hospitals".

Definition – Matching

– Collection $\mathscr{D}$ of "doctors" and $\mathscr{H}$ of "hospitals".

– Each doctor $d \in \mathscr{D}$ has a set of preferences over all hospitals $\mathscr{H}$, represented (partial) list ordered from most preferred to least preferred, and vice-versa.

Definition – Matching

– Collection $\mathscr{D}$ of “doctors” and $\mathscr{H}$ of “hospitals”.

– Each doctor $d \in \mathscr{D}$ has a set of preferences over all hospitals $\mathscr{H}$, represented (partial) list ordered from most preferred to least preferred, and vice-versa.
(Partial as some hospitals are “unacceptable” match to d.)

Definition – Matching

– Collection $\mathscr{D}$ of "doctors" and $\mathscr{H}$ of "hospitals".

– Each doctor $d \in \mathscr{D}$ has a set of preferences over all hospitals $\mathscr{H}$, represented (partial) list ordered from most preferred to least preferred, and vice-versa.
(Partial as some hospitals are "unacceptable" match to d.)

– A matching is a set of vertex disjoint edges in the complete bipartite graph on $\mathscr{D} \cup \mathscr{H}$:

$$\mu : \mathscr{D} \cup \mathscr{H} \to \mathscr{D} \cup \mathscr{H} \cup \{\emptyset\};$$

$\mu(x) = \emptyset$ if agent x is unmatched.

Definition – Stable Matching

– For a fixed set of preferences, matching μ is stable if there does not exist a pair $(d, h) \in \mathscr{D} \times \mathscr{H}$ such that:

a) doctor d is not matched to hospital h,

b) doctor d prefers h more than $\mu(d)$,

c) hospital h prefers d more than $\mu(h)$.

Definition – Stable Matching

– For a fixed set of preferences, matching μ is stable if there does not exist a pair $(d, h) \in \mathscr{D} \times \mathscr{H}$ such that:

a) doctor d is not matched to hospital h,
b) doctor d prefers h more than $\mu(d)$,
c) hospital h prefers d more than $\mu(h)$.

– A pair (d, h) is called stable for a fixed set of preferences if $\mu(d) = h$ in some stable matching.

Definition – Stable Matching

– For a fixed set of preferences, matching μ is stable if there does not exist a pair $(d, h) \in \mathscr{D} \times \mathscr{H}$ such that:

a) doctor d is not matched to hospital h,
b) doctor d prefers h more than $\mu(d)$,
c) hospital h prefers d more than $\mu(h)$.

– A pair (d, h) is called stable for a fixed set of preferences if $\mu(d) = h$ in some stable matching.

– The canonical method of finding some stable matching is the one-side-proposing deferred acceptance algorithm.

Definition – Stable Matching

– For a fixed set of preferences, matching μ is stable if there does not exist a pair $(d, h) \in \mathscr{D} \times \mathscr{H}$ such that:

a) doctor d is not matched to hospital h,
b) doctor d prefers h more than $\mu(d)$,
c) hospital h prefers d more than $\mu(h)$.

– A pair (d, h) is called stable for a fixed set of preferences if $\mu(d) = h$ in some stable matching.

– The canonical method of finding some stable matching is the one-side-proposing deferred acceptance algorithm.

– DPDA — Doctor-proposing deferred acceptance algorithm.
– HPDA — Hospital-proposing deferred acceptance algorithm.

Definition – Doctor-proposing Algorithm (DPDA)

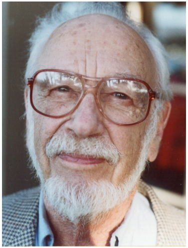
Do the following as long as there are some unmatched doctors and at least one of them has not proposed to every hospital:

– pick any such doctor d,

– d "proposes" to their favourite hospital h which they have not yet proposed to,

– if h likes d more than $\mu(h)$, then h "accepts" d's proposal ($\mu(h) = d$ and $\mu(d) = h$).

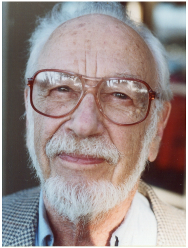
Definition – Doctor-proposing Algorithm



Theorem (Gale, Shapley, 1962)

– *DPDA always computes a stable matching* μ_0.

Definition – Doctor-proposing Algorithm

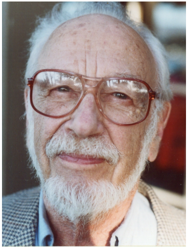


Theorem (Gale, Shapley, 1962)

– *DPDA always computes a stable matching* μ_0.

– *Moreover, this is the doctor-optimal stable outcome (that is, every doctor is matched in* μ_0 *to their favourite stable partner).*

Definition – Doctor-proposing Algorithm



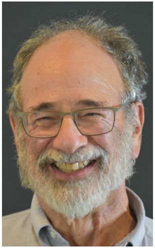
Theorem (Gale, Shapley, 1962)

– DPDA always computes a stable matching μ_0.

– Moreover, this is the doctor-optimal stable outcome (that is, every doctor is matched in μ_0 *to their favourite stable partner).*

– In particular, the resulting matching is independent of the execution order.

Definition – Doctor-proposing Algorithm



Theorem (Roth, 1986 (Rural Hospital Theorem))

For any set of preferences, the set of unmatched agents is the same across every stable outcome.

Applications

Applications in a variety of real-world situations:

– assignment of graduating medical students to their first hospital appointments (best known),

– . . .

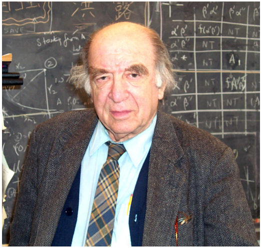
– assigning users to servers in a large distributed internet service.

Recognition

– In 2012, the Nobel Memorial Prize in Economic Sciences was awarded to Shapley and Roth "for the theory of stable allocations and the practice of market design."

– ...

– In 2007, the Nobel Memorial Prize in Economic Sciences was awarded to Hurwicz "for having laid the foundations of mechanism design theory."



Random Preference Lists

– In this talk, we are interested in uniformly random complete preference lists.

– That is, each doctor $d \in \mathcal{D}$ has one of the $|\mathcal{H}|!$ possible preference rankings of all the hospitals, chosen uniformly at random.

– Similar property holds for each hospital $h \in \mathcal{H}$.

Random Preference Lists

– In this talk, we are interested in uniformly random complete preference lists.

– That is, each doctor $d \in \mathcal{D}$ has one of the $|\mathcal{H}|!$ possible preference rankings of all the hospitals, chosen uniformly at random.

– Similar property holds for each hospital $h \in \mathcal{H}$.

– Sounds not so realistic? Surprisingly, it applies to many important scenarios.

Balanced Case

Definition – Balanced Case

– n hospitals;
each hospital h has a random preference of doctors.

– n doctors;
each doctor d has a random preference of hospitals.

Definition – Balanced Case

– n hospitals;
each hospital h has a random preference of doctors.

– n doctors;
each doctor d has a random preference of hospitals.

– rank(d): the rank of a doctor d (matched to a hospital h) is the index of h on d's preference list (where lower is better).

– rank(h): the rank of a hospital h (matched to a doctor d) is defined analogously.

Results – Balanced Case

Theorem (Wilson, 1972)

In doctor-optimal stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = O(\log n) \qquad \mathbb{E}[rank(h)] = \Omega(n/\log n).$$

Results – Balanced Case

Theorem (Wilson, 1972)

In doctor-optimal stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = O(\log n) \qquad \mathbb{E}[rank(h)] = \Omega(n/\log n).$$

– consider doctor-proposing algorithm (producing doctor-optimal stable outcome).

– the algorithm behaves essentially as the well-known "coupon collector" problem.

– the doctors have amnesia.

Unbalanced Case

Definition – Unbalanced Case

The "Effect of Competition".

– n hospitals;
each hospital h has a random preference of doctors.

– $(n+1)$ doctors;
each doctor d has a random preference of hospitals.

Results – Unbalanced Case

Theorem (Wilson, 1972)

In doctor-optimal stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = O(\log n) \qquad \mathbb{E}[rank(h)] = \Omega(n/\log n).$$

Results – Unbalanced Case

Theorem (Wilson, 1972)

In doctor-optimal stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = O(\log n) \qquad \mathbb{E}[rank(h)] = \Omega(n/\log n).$$

Theorem (Ashlagi, Kanoria, Leshno, 2017 — Unbalanced)

In every stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = \Omega(n/\log n) \qquad \mathbb{E}[rank(h)] = O(\log n).$$

Results – Unbalanced Case

Theorem (Ashlagi, Kanoria, Leshno, 2017 — Unbalanced)

In every stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = \Omega(n/\log n) \qquad \mathbb{E}[rank(h)] = O(\log n).$$

– short proof provided by Cai and Thomas, 2021; hospital-proposing algorithm with one doctor rejecting all proposals ("list truncation" technique).

Results – Unbalanced Case

Theorem (Ashlagi, Kanoria, Leshno, 2017 — Unbalanced)

In every stable matching, for any d and h,

$$\mathbb{E}[rank(d)] = \Omega(n/\log n) \qquad \mathbb{E}[rank(h)] = O(\log n).$$

– short proof provided by Cai and Thomas, 2021; hospital-proposing algorithm with one doctor rejecting all proposals ("list truncation" technique).

– doctor-proposing algorithm (producing doctor-optimal stable outcome) is more natural but "*unfortunately, this random process is fairly difficult to analyze (for instance, to get a useful analysis, we'd need to keep track of which doctor is currently proposing, which hospitals they have already proposed to, and how likely each hospital is to accept a new proposal).*"

Proofs

Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.

Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.

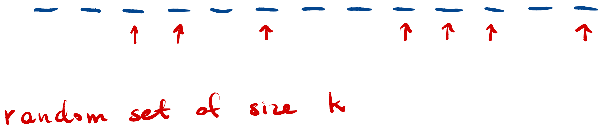
Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.

Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.



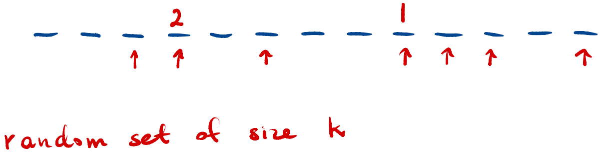
Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.

random set of size k

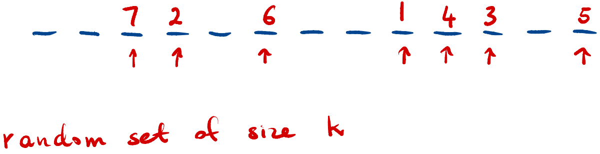
Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.



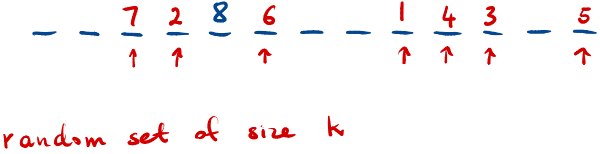
Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.



Principle of Deferred Decision

We may defer exposing information about the random preferences for a given hospital h until a new proposal is made to h.



Principle of Deferred Decision

– Consider an algorithm in which doctors propose.

Principle of Deferred Decision

- Consider an algorithm in which doctors propose.
- Hospital h is popular if it received at least k proposals, where

$$k = \left\lfloor \frac{n}{5c \log n} \right\rfloor \qquad (c \text{ is a constant, large enough}).$$

Principle of Deferred Decision

– Consider an algorithm in which doctors propose.

– Hospital h is popular if it received at least k proposals, where

$$k = \left\lfloor \frac{n}{5c \log n} \right\rfloor \qquad (c \text{ is a constant, large enough}).$$

– We will stop the algorithm prematurely at time T when there are exactly $\lfloor c \log n / 4 \rfloor$ popular hospitals.

Principle of Deferred Decision

– Consider an algorithm in which doctors propose.

– Hospital h is popular if it received at least k proposals, where

$$k = \left\lfloor \frac{n}{5c \log n} \right\rfloor \qquad (c \text{ is a constant, large enough}).$$

– We will stop the algorithm prematurely at time T when there are exactly $\lfloor c \log n / 4 \rfloor$ popular hospitals.

– Of course, the algorithm might converge to a stable matching before it happens; however, we will show that a.a.s. it will not happen.

Principle of Deferred Decision

Lemma

For each hospital h, we independently generate a subset $A_h \subseteq [n+1]$ of cardinality $k = \lfloor n/(5c \log n) \rfloor$. We run the algorithm until it stops prematurely or a stable matching is created. Let D_h be the set of doctors that proposed to hospital h.

Then, the following property holds: for each unpopular hospital h, doctors in D_h have ranks from A_h on the list of preferences of h.

Liking

– If a stable matching is created, then some poor doctor *d* must have proposed to every single hospital but is still unemployed.

Liking

– If a stable matching is created, then some poor doctor d must have proposed to every single hospital but is still unemployed.

– Most hospitals have d not so high on their preference lists so it is not too surprising that they did hire d.

Liking

– If a stable matching is created, then some poor doctor d must have proposed to every single hospital but is still unemployed.

– Most hospitals have d not so high on their preference lists so it is not too surprising that they did hire d.

– But there are still many hospitals that have d quite high on their respective preference lists; it is unlikely that all of them, especially unpopular ones, found a better match.

Liking

A hospital h likes a doctor d if d is on one of the top $\lfloor c \log n \rfloor$ places on the corresponding list of preferences.

Liking

A hospital h likes a doctor d if d is on one of the top $\lfloor c \log n \rfloor$ places on the corresponding list of preferences.

Lemma

A.a.s., every doctor is liked by at least $c \log n / 2$ hospitals.

Liking

A hospital h likes a doctor d if d is on one of the top $\lfloor c \log n \rfloor$ places on the corresponding list of preferences.

Lemma

A.a.s., every doctor is liked by at least $c \log n/2$ *hospitals.*

– The number of hospitals that like d is the binomial random variable $X \in \mathrm{Bin}(n, \lfloor c \log n \rfloor/(n+1))$ with expectation asymptotic to $c \log n$.

– The lemma follows immediately from Chernoff's bound and the union bound over all doctors.

Does the Algorithm Stop Prematurely?

Does the Algorithm Stop Prematurely?

– Regardless whether we stop prematurely or not, there are at most $\lfloor c \log n/4 \rfloor$ popular hospitals.

Does the Algorithm Stop Prematurely?

– Regardless whether we stop prematurely or not, there are at most $\lfloor c \log n/4 \rfloor$ popular hospitals.

– Fix any doctor d and suppose that d proposed to all hospitals.

– There are at least $c \log n/2$ hospitals that like d so at least $\lfloor c \log n/4 \rfloor$ unpopular hospitals like d.

Does the Algorithm Stop Prematurely?

– Regardless whether we stop prematurely or not, there are at most $\lfloor c \log n/4 \rfloor$ popular hospitals.

– Fix any doctor d and suppose that d proposed to all hospitals.

– There are at least $c \log n/2$ hospitals that like d so at least $\lfloor c \log n/4 \rfloor$ unpopular hospitals like d.

– Which of the hospitals that like d become unpopular depends on many other events so we need to take the union bound over all possible selections of $\lfloor c \log n/4 \rfloor$ hospitals out of $\lceil c \log n/2 \rceil$.

Does the Algorithm Stop Prematurely?

– If an unpopular hospital h that likes d is matched with someone better than d, then h likes someone from $D_h \setminus \{d\}$.

Does the Algorithm Stop Prematurely?

– If an unpopular hospital h that likes d is matched with someone better than d, then h likes someone from $D_h \setminus \{d\}$.

– h does *not* like anyone from $D_h \setminus \{d\}$ is at least

$$\prod_{i=1}^{k}\left(1 - \frac{\lfloor c \log n \rfloor}{n-i}\right) = \left(1 - (1+o(1))\frac{c \log n}{n}\right)^{k} = (1+o(1))e^{-1/5}.$$

so the probability that we aimed to estimate is at most $1 - (1+o(1))e^{-1/5} < 1/5$.

Does the Algorithm Stop Prematurely?

The probability that all unpopular hospitals that like d are matched with someone better than d is at most

$$
\begin{aligned}
\binom{\lceil c \log n/2 \rceil}{\lfloor c \log n/4 \rfloor} \left(\frac{1}{5}\right)^{\lfloor c \log n/4 \rfloor} &\leq 2^{\lceil c \log n/2 \rceil} \left(\frac{1}{5}\right)^{\lfloor c \log n/4 \rfloor} \\
&= O(1) \cdot \left(\frac{4}{5}\right)^{c \log n/4} \\
&= O(1) \cdot \exp\left(-\frac{c \log(5/4)}{4} \log n\right) \\
&= o(1/n),
\end{aligned}
$$

provided that c is large enough.

Does the Algorithm Stop Prematurely?

By the union bound over all doctors, we get the following.

Lemma

A.a.s., for every doctor there exists at least one unpopular hospital that is not matched with a better candidate.

Does the Algorithm Stop Prematurely?

By the union bound over all doctors, we get the following.

Lemma

A.a.s., for every doctor there exists at least one unpopular hospital that is not matched with a better candidate.

Corollary

A.a.s., the algorithm stops prematurely when there are exactly $\lfloor c \log n/4 \rfloor$ popular hospitals.

Does the Algorithm Stop Prematurely?

By the union bound over all doctors, we get the following.

Lemma

A.a.s., for every doctor there exists at least one unpopular hospital that is not matched with a better candidate.

Corollary

A.a.s., the algorithm stops prematurely when there are exactly $\lfloor c \log n/4 \rfloor$ popular hospitals.

Our final task is to show that it takes at least $\ell = \lfloor n^2/(ac \log n) \rfloor$ proposals, in total, to reach this situation (a is a constant, large enough).

Final Touch

– Suppose that a doctor d proposed x_d times for a total of $\sum_d x_d = \ell = \lfloor n^2/(ac \log n) \rfloor$ proposals.

Final Touch

– Suppose that a doctor d proposed x_d times for a total of $\sum_d x_d = \ell = \lfloor n^2/(ac\log n)\rfloor$ proposals.

– The number of scenarios:

$$\begin{aligned}\binom{\ell+n-1}{n-1} &\leq \left(\frac{e\ell(1+o(1))}{n}\right)^n = \left(\frac{en(1+o(1))}{ac\log n}\right)^n \\ &\leq n^n = \exp(n\log n),\end{aligned}$$

slightly too much to apply the union bound over.

Final Touch

– Suppose that a doctor d proposed x_d times for a total of $\sum_d x_d = \ell = \lfloor n^2/(ac \log n) \rfloor$ proposals.

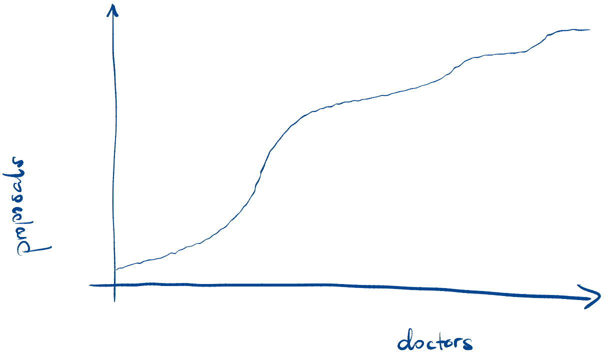
– The number of scenarios:

$$\begin{aligned}\binom{\ell + n - 1}{n - 1} &\leq \left(\frac{e\ell(1 + o(1))}{n}\right)^n = \left(\frac{en(1 + o(1))}{ac \log n}\right)^n \\ &\leq n^n = \exp(n \log n),\end{aligned}$$

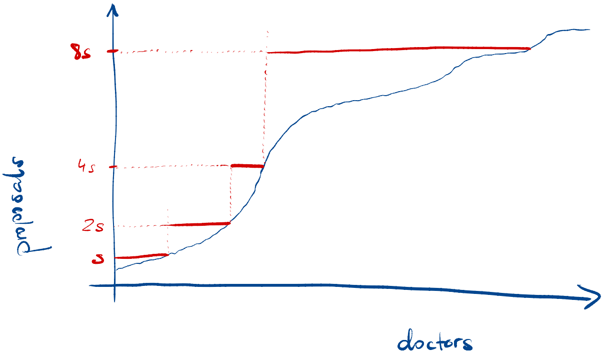
slightly too much to apply the union bound over.

– Solution: consider auxiliary scenarios $(\hat{x}_d)_d$ in which we "round x_d up".

Final Touch

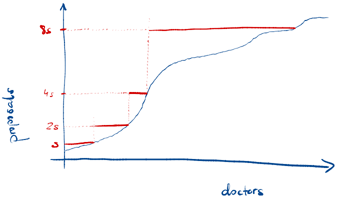


Final Touch



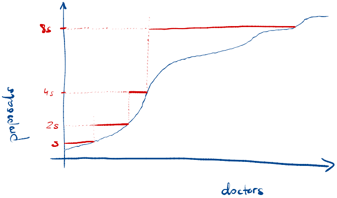
$s = \lceil n/(ac \log n) \rceil$

Final Touch



– If the original scenario $(x_b)_{b \in B}$ makes at least $k \cdot \lfloor c \log n/4 \rfloor = (1 + o(1))n/20$ proposals to a set of $\lfloor c \log n/4 \rfloor$ hospitals, then the auxiliary scenario $(\hat{x}_b)_{b \in B}$ does it too.

Final Touch



– If the original scenario $(x_b)_{b\in B}$ makes at least $k \cdot \lfloor c \log n/4 \rfloor = (1+o(1))n/20$ proposals to a set of $\lfloor c \log n/4 \rfloor$ hospitals, then the auxiliary scenario $(\hat{x}_b)_{b\in B}$ does it too.

– The total number of proposals is comparable:

$$\sum_{b\in B} \hat{x}_b \leq (n+1) \cdot \lceil n/(ac \log n) \rceil + 2 \sum_{b \in B} x_b \leq (1+o(1))\, 3\ell.$$

Final Touch

The advantage is that there are substantially less auxiliary scenarios than the original ones:

$$\begin{aligned}
\sum_{z_1 \geq z_2 \geq \dots} \binom{n}{z_1}\binom{z_1}{z_2}\binom{z_2}{z_3}\cdots &\leq \sum_{z_1 \geq z_2 \geq \dots} \binom{n}{n/2}\binom{n}{n/2}\binom{n/2}{n/4}\binom{n/4}{n/8}\cdots \\
&\leq \sum_{z_1 \geq z_2 \geq \dots} 2^{n+n+n/2+n/4+\dots} \\
&\leq n^{O(\log n)} \cdot 2^{3n} \\
&= \exp(O(\log^2 n)) \cdot 2^{3n},
\end{aligned}$$

where z_i is the number of values of $\hat{x}_b$ that are at least $2^i \lceil n/(ac \log n) \rceil$

Final Touch

– Fix any set of $\lfloor c \log n/4 \rfloor$ hospitals and any auxiliary configuration $(\hat{x}_b)_{b\in B}$ with $\sum_{b\in B} \hat{x}_b \leq (1+o(1))\, 3\ell \leq 4\ell$.

Final Touch

– Fix any set of $\lfloor c \log n/4 \rfloor$ hospitals and any auxiliary configuration $(\hat{x}_b)_{b \in B}$ with $\sum_{b \in B} \hat{x}_b \leq (1 + o(1))\, 3\ell \leq 4\ell$.

– Ignore active doctors that proposed at least $n/2$ times; there are not too many of them so at least $n/40$ proposals made to the selected hospitals have to come from non-active doctors.

Final Touch

– Fix any set of $\lfloor c \log n/4 \rfloor$ hospitals and any auxiliary configuration $(\hat{x}_b)_{b \in B}$ with $\sum_{b \in B} \hat{x}_b \leq (1 + o(1))\, 3\ell \leq 4\ell$.

– Ignore active doctors that proposed at least $n/2$ times; there are not too many of them so at least $n/40$ proposals made to the selected hospitals have to come from non-active doctors.

– Expose proposals from non-active doctors, one by one; proposal is made to one of the selected hospitals with probability at most $c \log n/(2n)$.

Final Touch

– Fix any set of $\lfloor c \log n/4 \rfloor$ hospitals and any auxiliary configuration $(\hat{x}_b)_{b \in B}$ with $\sum_{b \in B} \hat{x}_b \le (1 + o(1))\, 3\ell \le 4\ell$.

– Ignore active doctors that proposed at least $n/2$ times; there are not too many of them so at least $n/40$ proposals made to the selected hospitals have to come from non-active doctors.

– Expose proposals from non-active doctors, one by one; proposal is made to one of the selected hospitals with probability at most $c \log n/(2n)$.

– The number of proposals that are made to the selected hospitals can be stochastically upper bounded by the binomial random variable $X \in \text{Bin}(4\ell, c \log n/(2n))$ with $\mathbb{E}X = 2\ell c \log n/n = (1 + o(1))\, 2n/a$.

Final Touch

– Use some "fancy" Chernoff's bound:

$$\mathbb{P}(X \geq \mathbb{E}X + t) \quad \leq \quad \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{t}{\mathbb{E}X}\right)\right) \quad \leq \quad \exp\left(-\frac{t^2}{2(\mathbb{E}X + t/3)}\right)$$

$$\mathbb{P}(X \leq \mathbb{E}X - t) \quad \leq \quad \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{-t}{\mathbb{E}X}\right)\right) \quad \leq \quad \exp\left(-\frac{t^2}{2\mathbb{E}X}\right),$$

where $\varphi(x) = (1 + x)\log(1 + x) - x$, $x > -1$.

Final Touch

– Use some "fancy" Chernoff's bound:

$$\mathbb{P}(X \geq \mathbb{E}X + t) \leq \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{t}{\mathbb{E}X}\right)\right) \leq \exp\left(-\frac{t^2}{2(\mathbb{E}X + t/3)}\right)$$

$$\mathbb{P}(X \leq \mathbb{E}X - t) \leq \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{-t}{\mathbb{E}X}\right)\right) \leq \exp\left(-\frac{t^2}{2\mathbb{E}X}\right),$$

where $\varphi(x) = (1+x)\log(1+x) - x, x > -1$.

– Conclude that the scenario makes at least $n/40$ proposals to the selected hospitals with probability at most e^{3n}.

Final Touch

– Use some "fancy" Chernoff's bound:

$$\mathbb{P}(X \geq \mathbb{E}X + t) \quad \leq \quad \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{t}{\mathbb{E}X}\right)\right) \quad \leq \quad \exp\left(-\frac{t^2}{2(\mathbb{E}X + t/3)}\right)$$

$$\mathbb{P}(X \leq \mathbb{E}X - t) \quad \leq \quad \exp\left(-\mathbb{E}X \cdot \varphi\left(\frac{-t}{\mathbb{E}X}\right)\right) \quad \leq \quad \exp\left(-\frac{t^2}{2\mathbb{E}X}\right),$$

where $\varphi(x) = (1 + x)\log(1 + x) - x, x > -1$.

– Conclude that the scenario makes at least $n/40$ proposals to the selected hospitals with probability at most e^{3n}.

– We are done by the union bound over all auxiliary scenarios and sets of $\lfloor c \log n/4 \rfloor$ hospitals.

Final Touch

– It takes at least $\ell = \lfloor n^2/(ac \log n) \rfloor$ proposals to converge to a stable matching.

Final Touch

– It takes at least $\ell = \lfloor n^2/(ac \log n) \rfloor$ proposals to converge to a stable matching.

– Doctors propose (on average) at least $\ell/(n+1) = \Omega(n/\log n)$ times.

Final Touch

– It takes at least $\ell = \lfloor n^2/(ac \log n) \rfloor$ proposals to converge to a stable matching.

– Doctors propose (on average) at least $\ell/(n+1) = \Omega(n/\log n)$ times.

– Similar argument shows that there are at most $O(\log n)$ proposals to a given hospital (on average).

THE
END