Randomized Algorithms, Lecture 9

Jacob Holm (jaho@di.ku.dk)

May 21th 2019

Good afternoon.

Today's Lecture

Note about Chernoff bounds

The probabilistic method, part II

- Overview
- Oblivious Routing Revisited
- Lovasz Local Lemma
- Method of conditional probabilities

Chernoff bounds

Theorem

Let X be a sum of independent Poisson trials, and let $\mu = \mathbb{E}[X]$.
For any $\bar{\delta} > 0$ and $\bar{\mu} \geq \mu$.

$$\Pr[X > (1+\bar{\delta})\bar{\mu}] < \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}}$$

For any $0 < \bar{\delta} < 1$ and $\bar{\mu} \leq \mu$.

$$\Pr[X < (1-\bar{\delta})\bar{\mu}] < \left(\frac{e^{-\bar{\delta}}}{(1-\bar{\delta})^{(1-\bar{\delta})}}\right)^{\bar{\mu}} < e^{-\frac{\bar{\delta}^2\bar{\mu}}{2}}$$

These versions of the Chernoff bounds are not part of the curriculum, but are quite useful. I believe you have already had one exercise where they would have been useful.
The point is that we often only have an upper or lower bound on the expectation. Thus version of the theorem essentially says that it is valid to use this bound instead of the actual value when estimating the probability.

Chernoff bounds

Proof $X > (1+\bar{\delta})\bar{\mu}$.

Let $\bar{\mu} \geq \mu$, and choose δ such that $(1+\delta)\mu = (1+\bar{\delta})\bar{\mu}$. Note that $\delta \geq \bar{\delta}$

$$\begin{aligned}
\Pr[X > (1+\bar{\delta})\bar{\mu}] &= \Pr[X > (1+\delta)\mu] \\
&< \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu} && \text{(By Chernoff)} \\
&= \left(\frac{e^{1-\frac{1}{1+\delta}}}{1+\delta}\right)^{(1+\delta)\mu} \\
&= \left(\frac{e^{1-\frac{1}{1+\delta}}}{1+\delta}\right)^{(1+\bar{\delta})\bar{\mu}} && ((1+\delta)\mu = (1+\bar{\delta})\bar{\mu}) \\
&\leq \left(\frac{e^{1-\frac{1}{1+\bar{\delta}}}}{1+\bar{\delta}}\right)^{(1+\bar{\delta})\bar{\mu}} && (\bar{\delta} \leq \delta) \\
&= \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \square
\end{aligned}$$

Chernoff bounds

Proof $X < (1-\bar{\delta})\bar{\mu}$.

Let $\bar{\mu} \leq \mu$, and choose δ such that $(1-\delta)\mu = (1-\bar{\delta})\bar{\mu}$. Note that $\delta \geq \bar{\delta}$

$$
\begin{aligned}
\Pr[X < (1-\bar{\delta})\bar{\mu}] &= \Pr[X < (1-\delta)\mu] \\
&< \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu} && \text{(By Chernoff)} \\
&= \left(\frac{e^{1-\frac{1}{1-\delta}}}{1-\delta}\right)^{(1-\delta)\mu} \\
&= \left(\frac{e^{1-\frac{1}{1-\delta}}}{1-\delta}\right)^{(1-\bar{\delta})\bar{\mu}} && ((1-\delta)\mu = (1-\bar{\delta})\bar{\mu}) \\
&\leq \left(\frac{e^{1-\frac{1}{1-\bar{\delta}}}}{1-\bar{\delta}}\right)^{(1-\bar{\delta})\bar{\mu}} && (\bar{\delta} \leq \delta) \\
&= \left(\frac{e^{-\bar{\delta}}}{(1-\bar{\delta})^{(1-\bar{\delta})}}\right)^{\bar{\mu}} < e^{-\frac{\bar{\delta}^2\bar{\mu}}{2}} && \square
\end{aligned}
$$

The probabilistic method: Core ideas

1. Any random variable X takes some value $\leq \mathbb{E}[X]$ and some value $\geq \mathbb{E}[X]$.
2. If a random object taken from a universe U has nonzero probability of satisfying a property P, then there must be an object in U satisfying P.

The probabilistic method: Core ideas

1. Any random variable X takes some value $\leq \mathbb{E}[X]$ and some value $\geq \mathbb{E}[X]$.
2. If a random object taken from a universe U has nonzero probability of satisfying a property P, then there must be an object in U satisfying P.

The probabilistic method: Core ideas

1. Any random variable X takes some value $\leq \mathbb{E}[X]$ and some value $\geq \mathbb{E}[X]$.
2. If a random object taken from a universe U has nonzero probability of satisfying a property P, then there must be an object in U satisfying P.

Oblivious Routing: Reminder

Definition

In an *oblivious* routing scheme, the route v_i takes to reach $d(i)$ is independent of $d(j)$ for all $i \neq j$.

Theorem

Any deterministic oblivious permutation routing scheme on a network of N nodes of out degree d uses $\Omega(\sqrt{N/d})$ steps in the worst case.

Theorem

Valiant's scheme for oblivious routing on the hypercube with $N = 2^n$ nodes uses Nn random bits and expected $\mathcal{O}(n)$ steps.

Oblivious Routing: Reminder

Definition

In an *oblivious* routing scheme, the route v_i takes to reach $d(i)$ is independent of $d(j)$ for all $i \neq j$.

Theorem

Any deterministic oblivious permutation routing scheme on a network of N nodes of out degree d uses $\Omega(\sqrt{N/d})$ steps in the worst case.

Theorem

Valiant's scheme for oblivious routing on the hypercube with $N = 2^n$ nodes uses Nn random bits and expected $\mathcal{O}(n)$ steps.

Oblivious Routing: Reminder

Definition

In an *oblivious* routing scheme, the route v_i takes to reach $d(i)$ is independent of $d(j)$ for all $i \neq j$.

Theorem

Any deterministic oblivious permutation routing scheme on a network of N nodes of out degree d uses $\Omega(\sqrt{N/d})$ steps in the worst case.

Theorem

Valiant's scheme for oblivious routing on the hypercube with $N = 2^n$ nodes uses Nn random bits and expected $\mathcal{O}(n)$ steps.

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$.
Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Rand. lower bound

Theorem

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes that uses k random bits runs in $\Omega(2^{-k}\sqrt{N/n})$ steps.

Proof.

Any randomized algorithm A can be viewed as a random choice between deterministic algorithms $\{A_1, \ldots, A_R\}$. Since only k bits used, $R \leq 2^k$. Some fixed algorithm A_i is chosen with probability $\geq \frac{1}{R} \geq 2^{-k}$ in *any* run. By the lower bound, there is an instance I_i on which A_i require $t(n) = \Omega(\sqrt{N/n})$ steps. The expected number of steps when running A on I_i is at least $2^{-k}t(n) = \Omega(2^{-k}\sqrt{N/n})$. □

Oblivious Routing: Corollary

Corollary

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes must use $\Omega(n)$ random bits to run in expected $\mathcal{O}(n)$ steps.

Using the probabilistic method, we'll show

Theorem

For every n, there exists a randomized oblivious scheme for permutation routing on a hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Most of these details are irrelevant.

$$\begin{aligned}
2^{-k}\sqrt{\frac{N}{n}} &\leq Cn \\
2^{-2k}\frac{N}{n} &\leq C^2 n^2 \\
2^{-2k} N &\leq C^2 n^3 \\
2^{n-2k} &\leq C^2 n^3 && \text{(Using } N = 2^n\text{)} \\
n - 2k &\leq 3\log_2 n + \log_2 C^2 \\
n - 3\log_2 n - \log_2 C^2 &\leq 2k \\
\frac{n - 3\log_2 n - \log_2 C^2}{2} &\leq k \\
\frac{n - 4\log_2 n}{2} &\leq k && \text{(for } n \geq C^2\text{)} \\
\frac{17 - 4\log_2 17}{34} n &\leq k && \text{(for } n \geq \max\{17, C^2\}\text{)} \\
k &\in \Omega(n)
\end{aligned}$$

Oblivious Routing: Corollary

Corollary

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes must use $\Omega(n)$ random bits to run in expected $\mathcal{O}(n)$ steps.

Using the probabilistic method, we'll show

Theorem

For every n, there exists a randomized oblivious scheme for permutation routing on a hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Oblivious Routing: Corollary

Corollary

Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes must use $\Omega(n)$ random bits to run in expected $\mathcal{O}(n)$ steps.

Using the probabilistic method, we'll show

Theorem

For every n, there exists a randomized oblivious scheme for permutation routing on a hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned} \Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\ &< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\ &\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)} \end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned} \Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\ &< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\ &\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)} \end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} \quad \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} \qquad \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \le \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \le \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\
&\le e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \le \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \le N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \le 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2 \bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.
For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2 \bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.
For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\tfrac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} \quad \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2 \bar{\mu}}{4}} = e^{-\frac{N^2}{4}} \qquad \text{(By Theorem 4.3)}
\end{aligned}$$

$$\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] \leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} && \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} && \text{(By Theorem 4.3)} \\
\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] &\leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1
\end{aligned}$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} \quad \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} \qquad \text{(By Theorem 4.3)} \\
\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] &\leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1
\end{aligned}$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$. For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\frac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} \quad \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2 \bar{\mu}}{4}} = e^{-\frac{N^2}{4}} \qquad \text{(By Theorem 4.3)} \\
\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] &\leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1
\end{aligned}$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.
For any permutation π_i, let $X_{ij} = [B_j \text{ uses } > 14n \text{ steps on } \pi_i]$, and let $X_i = \sum_{j=1}^{t} X_{ij}$. We know $\mathbb{E}[X_{ij}] \leq \frac{1}{N}$ for all i, j, so $\mu = \mathbb{E}[\sum_{j=1}^{t} X_{ij}] \leq \frac{t}{N} = N^2$. Let $\bar{\mu} = N^2$ and $\bar{\delta} = 1$ then

$$\begin{aligned}
\Pr[X_i > 2N^2] &= \Pr[X_i > (1+\bar{\delta})\bar{\mu}] \\
&< \left(\tfrac{e^{\bar{\delta}}}{(1+\bar{\delta})^{(1+\bar{\delta})}}\right)^{\bar{\mu}} \quad \text{(By note at beginning)} \\
&\leq e^{-\frac{\bar{\delta}^2\bar{\mu}}{4}} = e^{-\frac{N^2}{4}} \qquad \text{(By Theorem 4.3)} \\
\Pr[\cup_{i=1}^{N!}\{X_i > 2N^2\}] &\leq \sum_{i=1}^{N!} \Pr[X_i > 2N^2] \leq N! \cdot e^{-\frac{N^2}{4}} < 1
\end{aligned}$$

Thus there exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \le 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned} \mathbb{E}[X] &\le \Pr[X \le 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\ &\le \max_{0 \le q \le \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\ &= \max_{0 \le q \le \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\ &< 15n \end{aligned}$$

□

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1-q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n
\end{aligned}$$

□

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.
There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.
Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.
Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n \qquad\qquad \square
\end{aligned}$$

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n
\end{aligned}$$

□

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n \qquad \square
\end{aligned}$$

We need to take the max here, because we don't have the exact probability.

We don't need sup because the range for q is compact and the function we are taking max over is continuous.

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n \qquad \square
\end{aligned}$$

Oblivious Routing: Algorithm proof

Valiant's scheme can be seen as picking a deterministic algorithm uniformly at random from $\mathcal{A} = \{A_i\}_{(i \in [N^N])}$. Let $t = N^3$, and for $j \in [t]$ pick $B_j \in \mathcal{A}$ independently, uniformly at random, with replacement, and let $\mathcal{B} = \{B_j\}_{(j \in [t])}$.

There exists such a $\mathcal{B}$ where $X_i \leq 2N^2$ for all π_i.

Thus, for any given permutation π, if we pick $B \in \mathcal{B}$ uniformly at random, with probability at least $1 - \frac{2N^2}{N^3} = 1 - \frac{2}{N}$, B uses at most $14n$ steps on π.

Let X be the number of steps used by B on π. We conclude

$$\begin{aligned}
\mathbb{E}[X] &\leq \Pr[X \leq 14n] \cdot 14n + \Pr[X > 14n] \cdot 2(N - 1 + n) \\
&\leq \max_{0 \leq q \leq \frac{2}{N}} (1 - q) \cdot 14n + q \cdot 2(N - 1 + n) \\
&= \max_{0 \leq q \leq \frac{2}{N}} 14n + q \cdot 2(N - 1 - 6n) \\
&< 15n
\end{aligned}$$

□

If $n \leq 4$ then $N - 1 - 6n < 0$ so the maximum is obtained for $q = 0$, and is $14n < 15n$.

If $n \geq 5$, then the maximum is obtained for $q = 2/N$, and is $14n + 4\frac{N-1-6n}{N} < 14n + 4 < 14n + n = 15n$

Oblivious Routing: Summary

We have seen that, for every n, there exists a randomized oblivious scheme for permutation routing on the hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Why is this not an efficient algorithm?

Oblivious Routing: Summary

We have seen that, for every n, there exists a randomized oblivious scheme for permutation routing on the hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Why is this not an efficient algorithm?

Oblivious Routing: Summary

We have seen that, for every n, there exists a randomized oblivious scheme for permutation routing on the hypercube with $N = 2^n$ nodes that use $3n$ random bits and expected at most $15n$ steps.

Why is this not an efficient algorithm? Because we don't know how to construct $\mathcal{B}$ efficiently, and we need a new one for each n (non-uniform).

Lovasz Local Lemma: Definitions

Recall that for *independent* (bad) events $\mathcal{E}_1, \ldots, \mathcal{E}_n$, we have $\Pr[\cap_{i=1}^n \overline{\mathcal{E}}_i] = \prod_{i=1}^n (1 - \Pr[\mathcal{E}_i])$.

We will generalize this slightly, to the case where there is *some* dependencies.

An event $\mathcal{E}_i$ is *mutually independent* of a set S of events, if $\Pr[\mathcal{E}_i \mid \cap_{\mathcal{E}_j \in T} \mathcal{E}_j] = \Pr[\mathcal{E}_i]$ for all $T \subseteq S$.

Let V be a set of events. A *dependency graph* for V is any digraph $G = (V, E)$ where each $\mathcal{E}_i \in V$ is mutually independent of $\{\mathcal{E}_j \in V \setminus \{\mathcal{E}_i\} \mid (\mathcal{E}_i, \mathcal{E}_j) \notin E\}$.

Lovasz Local Lemma: Definitions

Recall that for *independent* (bad) events $\mathcal{E}_1, \dots, \mathcal{E}_n$, we have $\Pr[\cap_{i=1}^{n} \overline{\mathcal{E}}_i] = \prod_{i=1}^{n}(1 - \Pr[\mathcal{E}_i])$.

We will generalize this slightly, to the case where there is *some* dependencies.

An event $\mathcal{E}_i$ is *mutually independent* of a set S of events, if $\Pr[\mathcal{E}_i \mid \cap_{\mathcal{E}_j \in T} \mathcal{E}_j] = \Pr[\mathcal{E}_i]$ for all $T \subseteq S$.

Let V be a set of events. A *dependency graph* for V is any digraph $G = (V, E)$ where each $\mathcal{E}_i \in V$ is mutually independent of $\{\mathcal{E}_j \in V \setminus \{\mathcal{E}_i\} \mid (\mathcal{E}_i, \mathcal{E}_j) \notin E\}$.

Lovasz Local Lemma: Definitions

Recall that for *independent* (bad) events $\mathcal{E}_1, \ldots, \mathcal{E}_n$, we have $\Pr[\cap_{i=1}^n \overline{\mathcal{E}}_i] = \prod_{i=1}^n (1 - \Pr[\mathcal{E}_i])$.

We will generalize this slightly, to the case where there is *some* dependencies.

An event $\mathcal{E}_i$ is *mutually independent* of a set S of events, if $\Pr[\mathcal{E}_i \mid \cap_{\mathcal{E}_j \in T} \mathcal{E}_j] = \Pr[\mathcal{E}_i]$ for all $T \subseteq S$.

Let V be a set of events. A *dependency graph* for V is any digraph $G = (V, E)$ where each $\mathcal{E}_i \in V$ is mutually independent of $\{\mathcal{E}_j \in V \setminus \{\mathcal{E}_i\} \mid (\mathcal{E}_i, \mathcal{E}_j) \notin E\}$.

Note that this holds for all $T \subseteq S$ if and only if it holds for all for all $T \subseteq S \cup \{\overline{\mathcal{E}}_j \mid \mathcal{E}_j \in S\}$.
In other words, an event and its complement are equivalent in terms of dependency.

Lovasz Local Lemma: Definitions

Recall that for *independent* (bad) events $\mathcal{E}_1, \ldots, \mathcal{E}_n$, we have $\Pr[\cap_{i=1}^n \overline{\mathcal{E}}_i] = \prod_{i=1}^n (1 - \Pr[\mathcal{E}_i])$.

We will generalize this slightly, to the case where there is *some* dependencies.

An event $\mathcal{E}_i$ is *mutually independent* of a set S of events, if $\Pr[\mathcal{E}_i \mid \cap_{\mathcal{E}_j \in T} \mathcal{E}_j] = \Pr[\mathcal{E}_i]$ for all $T \subseteq S$.

Let V be a set of events. A *dependency graph* for V is any digraph $G = (V, E)$ where each $\mathcal{E}_i \in V$ is mutually independent of $\{\mathcal{E}_j \in V \setminus \{\mathcal{E}_i\} \mid (\mathcal{E}_i, \mathcal{E}_j) \notin E\}$.

So each event *may* depend on (some subset of) its neighbors, but don't have to.

Lovasz Local Lemma: Statement

Lemma (Lovasz Local Lemma)

Let $G = (V, E)$ *be a dependency graph for events* $\mathcal{E}_1, \ldots, \mathcal{E}_n$ *in a probability space. Suppose there exists* $x_i \in [0, 1]$ *for* $1 \leq i \leq n$ *such that*

$$\Pr[\mathcal{E}_i] \leq x_i \prod_{(i,j)\in E} (1 - x_j)$$

Then

$$\Pr\left[\cap_{i=1}^{n} \overline{\mathcal{E}}_i\right] \geq \prod_{i=1}^{n} (1 - x_i)$$

The proof of this is *not* part of the curriculum.

Lovasz Local Lemma: Corollary

Corollary (Symmetric Lovasz Local Lemma)

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events in a probability space, with $\Pr[\mathcal{E}_i] \leq p$ for all i. If each event is mutually independent of all other events except for at most d, and if $ep(d+1) \leq 1$, then $\Pr\left[\cap_{i=1}^{n} \overline{\mathcal{E}}_i\right] > 0$

Proving this will be part of Assignment #6.

Lovasz Local Lemma: Corollary

Corollary (Symmetric Lovasz Local Lemma)

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ *be events in a probability space, with* $\Pr[\mathcal{E}_i] \leq p$ *for all* i*. If each event is mutually independent of all other events except for at most* d*, and if* $ep(d+1) \leq 1$*, then* $\Pr\left[\cap_{i=1}^{n} \overline{\mathcal{E}}_i\right] > 0$

Proving this will be part of Assignment #6.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

The book uses $2^{\frac{k}{50}}$ as the upper bound on the number of clauses, because the subsequent algorithm in the book needs such a bound.
That algorithm is obsolete, since newer versions of the Algorithmic Lovasz Local Lemma work with the simpler assumptions here.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: *k*-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

It can only share each of the k variables that it actually contains, and each of these are present in at most $\frac{2^k}{ke} - 1$ *other* clauses.

Lovasz Local Lemma: k-SAT

A k-SAT instance is a set of logical clauses, each containing exactly k literals.

Suppose each variable in a k-SAT instance appears in at most $\frac{2^k}{ke}$ of the m clauses. Consider a random truth assignment (like in MAX-SAT-SIMPLE) where each variable is set to TRUE independently with probability $\frac{1}{2}$.

Let $\mathcal{E}_i$ be the event that clause i is unsatisfied. Then $\Pr[\mathcal{E}_i] = p = 2^{-k}$. This event is independent of all other $\mathcal{E}_j$, except those where clause i and j share a variable.

Each clause shares variables with at most $d = k(\frac{2^k}{ke} - 1) \leq \frac{2^k}{e} - 1$ other clauses. Since $ep(d+1) \leq e(2^{-k})((\frac{2^k}{e} - 1) + 1) = 1$, the corollary tells us that there is a satisfying assignment.

Lovasz Local Lemma: Algorithmic Version

This is not part of the curriculum, but good to know!

Moser and Tardos, JACM 2010:

Theorem (Algorithmic Lovasz Local Lemma)

If the events $\mathcal{E}_1, \ldots, \mathcal{E}_n$ in the Lovasz Local Lemma are determined by a finite set $\mathcal{P}$ of independent random variables, and $x_i < 1$ for all i, then there is a Las Vegas style randomized algorithm running in expected polynomial time that finds an assignment to all variables in $\mathcal{P}$ such that $\cap_{i=1}^{n} \overline{\mathcal{E}}_i$.

The expected number of "steps" in the algorithm is $\mathcal{O}\left(\sum_{i=1}^{n} \frac{x_i}{1-x_i}\right)$, and each step is assumed to take polynomial time.

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n \times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \le \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \le \sum_i \Pr[\mathcal{E}_i] \le \tfrac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] \ge 1 - \tfrac{2}{n}$$

We will derandomize this using the *method of conditional probabilities*.

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n \times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \leq \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \leq \sum_i \Pr[\mathcal{E}_i] \leq \tfrac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \leq 4\sqrt{n \ln n}\right] \geq 1 - \tfrac{2}{n}$$

We will derandomize this using the *method of conditional probabilities.*

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n\times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \le \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \le \sum_i \Pr[\mathcal{E}_i] \le \tfrac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] \ge 1 - \tfrac{2}{n}$$

We will derandomize this using the *method of conditional probabilities*.

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n \times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \le \frac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \le \sum_i \Pr[\mathcal{E}_i] \le \frac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] \ge 1 - \frac{2}{n}$$

We will derandomize this using the *method of conditional probabilities.*

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n\times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \le \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \le \sum_i \Pr[\mathcal{E}_i] \le \tfrac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] \ge 1 - \tfrac{2}{n}$$

We will derandomize this using the *method of conditional probabilities.*

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n \times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\Pr[\mathcal{E}_i] \le \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] \le \sum_i \Pr[\mathcal{E}_i] \le \tfrac{2}{n}$$

$$\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] \ge 1 - \tfrac{2}{n}$$

We will derandomize this using the *method of conditional probabilities*.

Method of conditional probabilities

Consider the set-balancing problem: Given $\mathbf{A} \in \{0,1\}^{n\times n}$, find column vector $\mathbf{b} \in \{-1,1\}^n$ minimizing $\|\mathbf{Ab}\|_\infty = \max_i |(\mathbf{Ab})_i|$.

If for each i we pick $\mathbf{b}_i \in \{-1,1\}$ independently and uniformly at random, and let $\mathcal{E}_i$ denote the event that $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$, then (See Example 4.5)

$$\begin{aligned}
\Pr[\mathcal{E}_i] \le \tfrac{2}{n^2} \implies \Pr[\cup_i \mathcal{E}_i] &\le \sum_i \Pr[\mathcal{E}_i] \le \tfrac{2}{n} \\
\implies \Pr\left[\max_i |(\mathbf{Ab})_i| \le 4\sqrt{n \ln n}\right] &\ge 1 - \tfrac{2}{n}
\end{aligned}$$

We will derandomize this using the *method of conditional probabilities*.

Method of conditional probabilities

We can view a run of the algorithm as traversing a path from the root r to a leaf in a *computation tree*. Here, this is a complete binary tree of height n, where each edge is labelled either -1 or 1. In the ith step, a child at depth i is chosen uniformly at random, and $\mathbf{b}_i$ is set to the value at the traversed edge.

Method of conditional probabilities

We can view a run of the algorithm as traversing a path from the root r to a leaf in a *computation tree*. Here, this is a complete binary tree of height n, where each edge is labelled either -1 or 1. In the ith step, a child at depth i is chosen uniformly at random, and $\mathbf{b}_i$ is set to the value at the traversed edge.

Method of conditional probabilities

We can view a run of the algorithm as traversing a path from the root r to a leaf in a *computation tree*. Here, this is a complete binary tree of height n, where each edge is labelled either -1 or 1. In the ith step, a child at depth i is chosen uniformly at random, and $\mathbf{b}_i$ is set to the value at the traversed edge.

Method of conditional probabilities

We can view a run of the algorithm as traversing a path from the root r to a leaf in a *computation tree*. Here, this is a complete binary tree of height n, where each edge is labelled either -1 or 1. In the ith step, a child at depth i is chosen uniformly at random, and $\mathbf{b}_i$ is set to the value at the traversed edge.

Method of conditional probabilities

We can view a run of the algorithm as traversing a path from the root r to a leaf in a *computation tree*. Here, this is a complete binary tree of height n, where each edge is labelled either -1 or 1. In the ith step, a child at depth i is chosen uniformly at random, and $\mathbf{b}_i$ is set to the value at the traversed edge.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

$$\begin{aligned}
P(a) &= \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}] \\
&= \Pr[\cup_i \mathcal{E}_i \cap c \text{ reached} \mid a \text{ reached}] \\
&\quad + \Pr[\cup_i \mathcal{E}_i \cap d \text{ reached} \mid a \text{ reached}] \\
&= \Pr[\cup_i \mathcal{E}_i \mid c \text{ reached}] \cdot \Pr[c \text{ reached} \mid a \text{ reached}] \\
&\quad + \Pr[\cup_i \mathcal{E}_i \mid d \text{ reached}] \cdot \Pr[d \text{ reached} \mid a \text{ reached}] \\
&= P(c) \cdot \tfrac{1}{2} + P(d) \cdot \tfrac{1}{2} \\
&= \frac{P(c) + P(d)}{2}
\end{aligned}$$

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done.

Method of conditional probabilities

For a node a, let $P(a) = \Pr[\cup_i \mathcal{E}_i \mid a \text{ reached}]$. Note $P(r) \leq \frac{2}{n} < 1$ for $n > 2$.

Let c, d be children of a, then $P(a) = \frac{P(c)+P(d)}{2}$, so $\min\{P(c), P(d)\} \leq P(a)$.

For each leaf ℓ, $P(\ell) \in \{0, 1\}$. In particular, $P(\ell) < 1 \implies P(\ell) = 0$.

If we can efficiently select the child minimizing $P(\cdot)$ we are done. For some problems we can! But not for this one.

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d

$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$

3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d

$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$

3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d

$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$

3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d

$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$

3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

For each i,

$$\Pr[\mathcal{E}_i \mid a \text{ reached}] = \frac{\Pr[\mathcal{E}_i \mid c \text{ reached}] + \Pr[\mathcal{E}_i \mid d \text{ reached}]}{2}$$

Thus

$$\begin{aligned}
\widehat{P}(a) &= \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}] \\
&= \sum_{i=1}^{n} \frac{\Pr[\mathcal{E}_i \mid c \text{ reached}] + \Pr[\mathcal{E}_i \mid d \text{ reached}]}{2} \\
&= \frac{\widehat{P}(c) + \widehat{P}(d)}{2} \\
&\geq \min\left\{\widehat{P}(c), \widehat{P}(d)\right\}
\end{aligned}$$

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d

$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$

3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

Assume a has depth $\ell \in [n]$, and that all unfixed bits in $\mathbf{b}$ are set to 0.
For each i, we have $\Pr[\mathcal{E}_i \mid a \text{ reached}] = 2^{-s} \sum_{j=t}^{s} \binom{s}{j}$, where $s \leq n - \ell$ is the number of bit positions selected by $\mathbf{A}_i$ that has not yet been fixed in $\mathbf{b}$, and $t = 4\sqrt{n \ln n} + 1 - |(\mathbf{Ab})_i|$ is the minimum number of these positions that need to be set correctly to make $|(\mathbf{Ab})_i| > 4\sqrt{n \ln n}$.
This can clearly be computed in polynomial time.

Method of conditional probabilities

Define $\widehat{P}(a) := \sum_{i=1}^{n} \Pr[\mathcal{E}_i \mid a \text{ reached}]$.

Then $P(a) \leq \widehat{P}(a)$ (union bound), and

1. $\widehat{P}(r) < 1$
2. For any node a with children c, d
$$\min\left\{\widehat{P}(c), \widehat{P}(d)\right\} \leq \widehat{P}(a)$$
3. For any node a, we can compute $\widehat{P}(a)$ in time polynomial in n.

This gives us a deterministic polynomial-time approximation algorithm.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.

Summary

- We have seen a useful extension to the Chernoff Bounds.
- We saw a lower bound for randomized oblivious routing, showing a tradeoff between the amount of randomness used and the best possible expected number of steps. We also saw a non-uniform matching upper bound.
- We then saw Lovasz Local Lemma in its normal and symmetric form, and used it to show that certain restricted versions of k-SAT always have a satisfying assignment, and I briefly mentioned that there is an algorithmic version of Lovasz Local lemma.
- Finally, we saw how the method of conditional probabilities could be used to derandomize the set-balancing problem.
- Next time: Algebraic techniques.