Week 3

Andrew Yao (姚期智)

2024-3-12

- For any graph G, let w(G) denote its *clique number*, i.e., the largest size of any clique contained in G. We show that, for large n, a random n-vertex graph G has w(G) close to $2 \log_2 n$.

Theorem For any fixed $0 < \varepsilon < 1$, and large n, a random n-vertex graph G satisfies the condition $(2-\varepsilon) \log_2 n \le w(G) \le (2+\varepsilon) \log_2 n$ with probability $1 - o(1)$.

Proof. *Upper bound:* $Pr\{w(G) > (2+\varepsilon) \log_2 n\} = o(1)$.

Similarly to the argument used in the proof of Erdös Theorem. Let $s = (2+\varepsilon) \log_2 n$, W be the family of vertex subsets of size s.

$$\Pr\{w(G) > s\} \le \Pr\{\cup_{V \in W} (V \textit{ is a clique in } G)\}$$

$$\le \sum_{V \in W} \Pr\{V \textit{ is a clique in } G\}$$

Largest clique in a random graph *(continued)*

$$\Pr\{w(G) > s\} \le \sum_{V \in W} \Pr\{V \textit{ is a clique in } G\}$$

$$= \binom{n}{s} \frac{1}{2^{\binom{s}{2}}} \le n^s \frac{1}{2^{s(s-1)/2}}$$

$$\le \left(\frac{n\sqrt{2}}{2^{(s-1)/2}}\right)^s$$

$$\le 2\left(\frac{\sqrt{2}}{n^{\varepsilon/2}}\right)^{2\log_2 n}$$

$$= n^{-\Omega(\log n)} = o(1)$$

This proves the upper bound.

Largest clique in a random graph *(continued)*

- Lower Bound: Let $m=(2-\varepsilon)\log_2 n$, M= the family of vertex subsets of size m, and let T be the event that $w(G)\geq m$.

Prove: $\Pr\{T\} = 1- o(1)$

Define for each $V \subseteq M$ a random variable

$A_V(G) = 1$ if V is a clique in G, and $A_V(G)=0$ otherwise.

Consider random variable $X= \sum_{V\in W} A_V(G)$.

Note that T is the same as the event $X > 0$, thus

$$\Pr\{T\} = \Pr\{X > 0\}$$

- Our strategy is to use Chebyshev's Inequality to show that

$$\Pr\{X > 0\} = 1- o(1)$$

We'll do it in two steps:

1) $E(X) \rightarrow \infty$ as $n \rightarrow \infty$
2) $\mathrm{Var}(X) = (E(X))^2 \cdot o(1)$

Largest clique in a random graph *(continued)*

- We prove the lower bound Pr{X > 0} = 1− o(1), where $X = \sum_{V \in W} A_V(G)$ in two steps:
  1) $E(X) \to \infty$ as $n \to \infty$
  2) $Var(X) = (E(X))^2 \cdot o(1)$

  It then follows from Chebyshev that

$$\Pr\{X \le 0\} \le \Pr\left\{ |X - E(X)| > \frac{1}{2}E(X) \right\}$$

$$\le \frac{Var(X)}{(\frac{1}{2}E(X))^2} = o(1)$$

- To prove 1), note by Linearity of Expectation,

$$\begin{aligned}
E(X) &= \sum_{V \in M} E(A_V) = \binom{n}{m}\frac{1}{2^{\binom{m}{2}}} \ge \Omega\left(\frac{n^m}{\sqrt{2\pi m}(\frac{m}{e})^m} \cdot \frac{1}{2^{\frac{1}{2}m(m-1)}}\right) \\
&= \Omega\left(\frac{en}{(2\pi m)^{\frac{1}{2m}} \cdot m} \cdot \frac{1}{2^{\frac{1}{2}m}}\right)^m = \Omega\left(\left(\frac{0.01n}{\log_2 n} \cdot \frac{1}{n^{1-\frac{1}{2}\epsilon}}\right)^m\right) \\
&= \Omega\left(\left(\frac{0.01n^{\frac{1}{2}\epsilon}}{\log_2 n}\right)^{\log_2 n}\right) = n^{\Omega(\log n)}.
\end{aligned}$$

QED

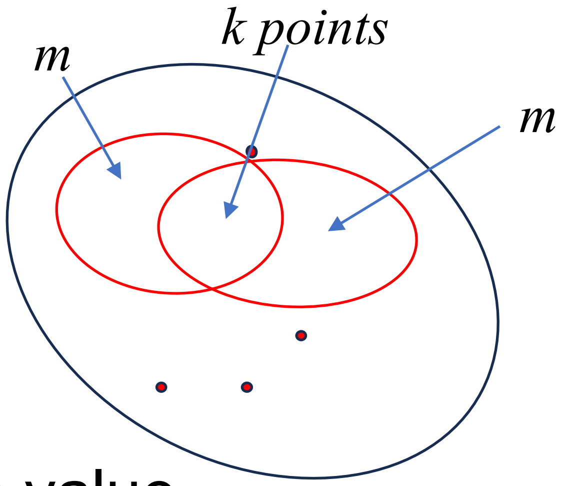
We now prove 2): $Var(X) = (E(X))^2 \cdot o(1)$

- $Var(X) = (E(\sum_{V\in W} A_V)^2) - (E(X))2$

  $\le (E(\sum_V \sum_{V'} A_V A_{V'}) - \sum_V \sum_{|V\cap V'|\le 1} E(A_V)E(A_{V'})$

  $= E(\sum_V A_V + \sum_V \sum_{|V\cap V'|\le 1} A_V A_{V'} + \sum_V \sum_{|V\cap V'|>1} A_V A_{V'}) - \sum_V \sum_{|V\cap V'|\le 1} E(A_V)E(AV_{'})$

Note that $E(A_V A_{V'}) = E(A_V)E(A_{V'})$ if $|V \cap V'| \le 1$, thus



- $Var(X) \le E(X) + \sum_{2\le k\le m} \sum_V \sum_{|V\cap V'|=k} E(A_{V,} A_{V'})$

  $= E(X) + \sum_{2\le k\le m} \sum_V \sum_{|V\cap V'|=k} Pr\{A_V=1, A_{V'}=1\}$

  $= E(X) + \sum_{2\le k\le m} \sum_V \sum_{|V\cap V'|=k} Pr\{A_V=1\} Pr\{A_V=1|A_{V'}=1\}$

- By symmetry, all $\Pr\{A_V=1| A_{V'})=1\}$ with $|V \cap V'| = k$ have the same value

$$
\begin{aligned}
Var(X) &\le E(X) + \sum_{2\le k\le m} \sum_V \Pr\{A_V = 1\} \cdot \binom{m}{k}\binom{n-m}{m-k}\frac{1}{2^{\binom{m}{2}-\binom{k}{2}}} \\
&= E(X) + E(X) \cdot \sum_{2\le k\le m} \binom{m}{k}\binom{n-m}{m-k}\frac{1}{2^{\binom{m}{2}-\binom{k}{2}}}
\end{aligned}
$$

Lemma $\displaystyle\sum_{2\le k\le m}\binom{m}{k}\binom{n-m}{m-k}\frac{1}{2^{\binom{m}{2}-\binom{k}{2}}}\le\frac{m^5}{n-m+1}E(X).$

Proof. (Homework)

- By this Lemma and the fast E(X) → ∞, we have for large n,

$$\mathrm{Var}(X) \le E(X) + \frac{64(\log_2 n)5}{n} E(X)^2$$
$$\le \frac{128(\log_2 n)5}{n} E(X)^2$$

- By Chebyshev's Inequality,

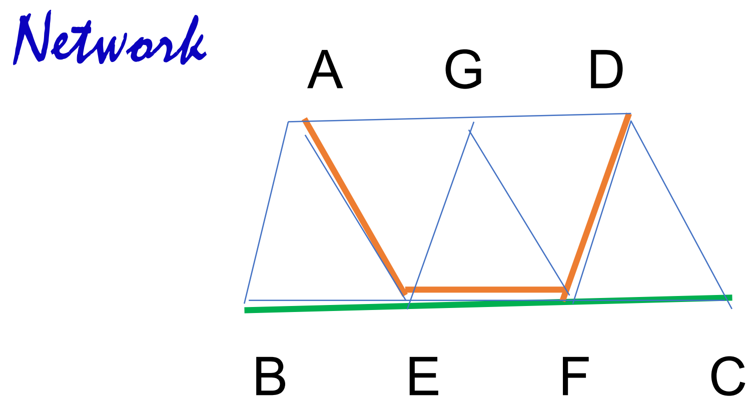
$$\Pr\{|X-E(X)| \ge ½\, E(X)\} \le \frac{Var(X)}{(½\, E(X))2} = O(\frac{(\log_2 n)5}{n})$$

- Thus $\Pr\{X > 0\} \le \Pr\{|X-E(X)| \ge ½\, E(X)\} = O(\frac{(\log_2 n)5}{n}) = o(1)$. QED

This proves the lower bound, and the theorem on random graph clique size.

A Network Routing Problem

Assume:



- message M_A goes A → D via A→E→F→D
  message M_B goes B → C via B→E→F→C
- both M_A , M_B start at t=0, each link takes 1 time unit,
  each link's transport capacity = I message

At t=1, both M_A and M_B arrive at E to use link EF, hence one of them must wait in a queue and gets routed at t=2 through EF.

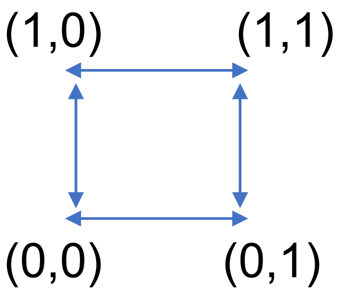
Question: *How to design a routing algorithm to avoid congestion and long delays?*

Hypercube Network

An n-Hypercube has $N=2^n$ nodes $V=\{0,1\}^n$, and directed edges $E=\{d_H(i, i')=1\}$.

Note $|E|= n\cdot N$

n=2



$|E| = 2 \cdot 2^n = 8$ edges

Hypercube Network (continued)

A routing task is specified by a $\sigma \epsilon S_n$. Starting at t=0, each node $i\ \epsilon V$ has a message M_i to be routed to destination $\sigma(i)$. The goal is to get all messages successfully delivered within a reasonably short time.

Bit-Fixing Algorithm (BSA):

$$
\begin{aligned}
i &= 0\underline{0}11010\\
&\ 011\underline{1}010\\
&\ 01100\underline{1}0\\
\sigma(i) &= 0110001
\end{aligned}
$$

- Path($i,\sigma(i)$) is of length = $d_H(i,\sigma(i))$
- Each node along path decreases distance d_H to destination $\sigma(i)$ by 1
- Call such paths *geodesics*

Fact This routing algorithm has exponential delay in the worst case.

pf. Only need to exhibit one bad σ. Let n=odd and define σ such that $\sigma(u0v)=v1u$ where $|u|=|v|= (n-1)/2$. The corresponding path looks like: u0v ….v0v, v1v ….v1u

In particular, the path from $i = u0^{(n+1)/2}$ to $\sigma(i) = 0^{(n-1)/2}1u$ must contain the (middle) link $e=(0^n,\ 0^{(n-1)/2}\, 1\, 0^{(n-1)/2})$. Hence at least $2^{(n-1)/2}$ messages need to be routed through e, causing a time delay of $2^{(n-1)/2}$ for some message M_i

QED

- Actually, this kind of worst-case exponential delay also happens to many other deterministic routing algorithms. How can we avoid it?

Randomized BSA (Valiant 1981)

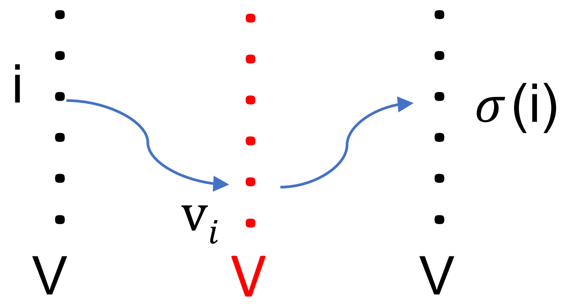
Let σ be the permutation specifying the routing task. Let $V=\{0,1\}^n$, and recall $N=2^n$.

Phase 1. For each node $i \in V$, generate a random $v_i \in V$.

Use BFA to route message M_i from node i to v_i .

Phase 2. At time $t=6n$, for each $i \in V$,

use BFA to route message M_i from node v_i to $\sigma(i)$.



- This is a randomized algorithm, whose randomness comes from the choice of the intermediate node $v_i \in V$ for each $i \in V$. Let U be the set of all possible mappings $V \rightarrow V$, clearly $|U|=N^N$. Thus the probability space is $P=(U, p)$, where $p=1/|U|$. The delivery time for M_i is a random variable in P. For any routing task σ, let B_σ be the event that, for all $i \in V$, message M_i reaches destination $\sigma(i)$ by time 12n.

Theorem 1. For any σ, $\Pr\{B_\sigma\} > 1 - (2^{-3n})$.

- *Event B_σ*: all message M_i reach destination $\sigma(i)$ by time 12n.

Theorem 1 For any σ, $\Pr\{B_\sigma\} > 1 - (2^{-3n})$.

- It suffices to prove, for each of Phase 1 and 2, the probability for any M_i not to reach destination in time 6n is $O(2^{-3n})$. We'll prove it for Phase 1. (The proof of Phase 2 is similar and left as exercise.)
- In Phase 1, let T_i be the arrival time for message M_i to reach its intermediate node v_i.

Theorem 1' $\Pr\{\exists\, i \in V \text{ with } T_i > 6n\} = O(2^{-3n})$.

Note that Theorem 1' can be reduced to proving the following:

Main Lemma Fix any $i \in V$ and $u \in V$. Then $\Pr\{T_i > 6n \mid v_i = u\} = O(2^{-4n})$.

The Main Lemma together with distributive law implies, for any $i \in V$,

$$\begin{aligned}\Pr\{T_i > 6n\} &= \sum_{u \in V} \Pr\{v_i = u\} \cdot \Pr\{T_i > 6n \mid v_i = u\} \\ &= O(2^{-4n}) \sum_{u \in V} \Pr\{v_i = u\} \\ &= O(2^{-4n})\end{aligned}$$

By union bound, $\Pr\{\exists\, i \in V \text{ with } T_i > 6n\} \le |V|\, O(2^{-4n}) = O(2^{-3n})$, which is Theorem 1'

Main Lemma For any fixed $i \epsilon V$ and $u \epsilon V$, $\Pr\{ T_i > 6n \mid v_i = u \} = O(2^{-4n})$.

pf. Consider the random variable $S = \{ j \mid j \neq i, \text{Path}(j, v_j) \cap \text{Path}(i, v_i = u) \neq \emptyset \}$ (i.e., the two paths share at least 1 edge).

- Key Insight $T_i \leq d_H(i, v_i) + |S|$. (Prove this in homework)

- Note that $d_H(i, v_i) \leq n$ is the absolutely minimum time needed to traverse the hypercube from i to v_i. The above inequality says that the additional 'delay' in delivering message M_i is no greater than the number of messages M_j intersecting the path taken by M_i .This key insight amazingly transforms the analysis of an algorithm into the analysis of a 'static' combinatorial quantity |S|.
- Given the Key Insight, to prove the Main Lemma, we will show that

Proposition $\Pr\{|S| > 5n\} = O(2^{-4n})$.

pf. We first generate for i a new independent random variable v_i'. Define

$$S' = \begin{cases} S \cup \{i\} & \text{if } \text{Path}(i, v_{i'}) \cap \text{Path}(i, u) \neq \emptyset \\ S & \text{otherwise,} \end{cases}$$

For all $k \epsilon V$, let $X_k = 1$ if $k \epsilon S'$ and 0 otherwise, then $|S'| = \sum_{k \epsilon V} X_k$

Note that $|S'| = \sum_{k \epsilon V} X_k$ is a sum of independent Boolean variables.

Since $S \subseteq S'$, we only need to prove

Proposition' $\Pr\{|S'|>5n\} = O(2^{-4n})$.

pf. We will need the following Lemma

Lemma A $E(|S'|) \le n/2$.

Using Lemma A we can obtain $\Pr\{|S'|-E(S')> 4n\} \le 2^{-4n}$, because by Corollary 2 to Chernoff's bound: $\Pr\{ Z - E(Z) > c \} \le 2^{-c}$ if $c > 6E(Z)$.

Using Lemma A again, $\Pr\{|S'| > 5n\} \le 2^{-4n}$, proving the Proposition.

- We now prove Lemma A. First introduce a random variable Y_e for each edge in the network: $Y_e \equiv$ # of nodes $j \epsilon V$ in the network such that Path(j, v_j) contains e.

  Fact. $E(Y_e) = 1/2$ for each edge e. (Homework)
- To prove Lemma A, we write Path(i, v_i)=$e_1 e_2 \cdots e_\ell$ with $\ell = d_H(i, v_i) \le n$.

  Note $|S'| \le \sum_{1 \le k \le \ell} Y_{e_k}$ since every node of S' is counted at least once on the RHS.

From $|S'| \leq \sum_{1 \leq k \leq \ell} Y_{e_k}$ it follows that

$$E(|S'|) \leq E\left(\sum_{1 \leq k \leq \ell} Y_{e_k}\right) = \sum_{1 \leq k \leq \ell} E(Y_{e_k})$$

$$= \ell \cdot \frac{1}{2}$$

$$\leq n/2.$$

This completes the proof of Lemma A, and hence the Main Lemma, and Theorem 1.

QED

Comment:

The above analysis shows that randomization sometimes leads to simple and more efficient algorithms than the standard algorithms.

We'll later discuss a result that a wide class of deterministic routing algorithms must have exponential congestion and hence delay time, just like the Bit-Fixing Alg.

This course:

✓ Topic 1: Probability Theory

Models and Tools:

- Probability space
- Event

  Union bound

- Random variable X

  Expectation E(X), Variance VAR(X)

  Linearity of expectation

Independent random variables X_i ; sum and product

- Conditional Probability

  Conditional Expectation

  Chain rules for conditional probability

  Law of total Probability

  Law of total Expectation

Topic 1: Probability Theory

Models and Tools: (continued)

- Tail Estimates

  Markov's Inequality, Chebyshev's Inequality, Chernoff's Inequality

Application Examples:

- Erdos Theorem on Random numbers (union bound)
- Number of cycles in random permutations
- Analysis of Greedy Clique Algorithm in random graph (union bound)
- Max clique size in random graphs (union bound, Chebyshev)
- Second moment method to prove $Pr\{X>0\} = 1- o(1)$
- Randomized Routing (linearity of expectation, total probability, key insight, Chernoff)

Some Classical Open Problems:

1. Lower bounds to Ramsey numbers
2. Can we find clique in random graphs of size $c\log_2 n$ for $c > 1$?

This course:

- ✓ Topic 1: Probability Theory

  Topic 2: *Graph Theory / Combinatorics*

  -- Counting problems

  -- Complexity questions

- Introduce a widely-used technique called "generating functions".

  Let $\langle a_k \rangle = a_0\, a_1\, a_2\, \ldots$ be an infinite sequence of complex numbers.

  Its generating function is defined as $A(x) = \sum_{k \geq 0} a_k x^k$. This conceptually provides an innovative alternative way to view the sequence $\langle a_k \rangle$. The rich set of tools available in the mature fields of real/complex analysis often makes it possible to obtain explicit information on $\langle a_k \rangle$.

- We begin with a simple example. Let X be a random variable with range $N = \{0,1,2, \ldots \}$ and $p_k = \Pr\{X=k\}$ for $k \in N$. Assume that the generating function $A(x) = \sum_{k \geq 0} p_k x^k$ is convergent (and hence analytic) in a neighborhood of $x=0$.

- $A(x) = \sum_{k\geq 0} p_k x^k$ where $p_k = \Pr\{X=k\}$ for $k \in N$.

Theorem 1. $E(X)=A'(1)$ and $Var(X)=A''(1) + A'(1)-A'(1)^2$

Pf. $A'(x) = \sum_{k\geq 0} k\, p_k\, x^{k-1}$

$A''(x) = \sum_{k\geq 0} k(k-1)\, p_k\, x^{k-2}$

It follows that $A'(1) = \sum_{k\geq 0} k\, p_k = E(X)$

and $A''(1) = \sum_{k\geq 0} k^2 p_k - \sum_{k\geq 0} k\, p_k = E(X^2) - E(X)$

Thus, $Var(X)=E(X^2) - (E(X))^2$.

$=A''(1) + A'(1)-A'(1)^2$

QED

- For instance, let X be the number of 1's in a throw of n independent coin tosses with bias $0<b<1$. Then $p_k = \binom{n}{k} b^k(1-b)^{n-k}$, and $A(x) = \sum_{k\geq 0} p_k x^k = (bx+(1-b))^n$.

It follows that $A'(1) = n(b \cdot 1+(1-b))^{n-1} b = bn$, and

$A''(1) = n(n-1)(b \cdot 1+(1-b))^{n-2} b^2 = n(n-1)b^2$.

By Theorem 1, we have $E(X)=A'(1)= bn$

and $Var(X)=A''(1) + A'(1)-A'(1)^2 = n(n-1)b^2 + bn - (bn)^2$

$= b(1-b)n$, as expected.

We now turn to a more sophisticated usage of generating functions.

- When the explicit form of $<a_k>$ is unknown, it may be easier to obtain its generating function first in some familiar form, and then obtain exact or approximate expressions for the elements in the sequence $<a_k>$. Specifically, we will discuss the solution of recurrence relations via generating functions.

First we need to introduce two basic operations on generating functions:

- Let A(x), B(x) be the generating function of sequences $<a_k>$, $<b_k>$ respectively.

1. $A(x)+B(x)$ is the generating function of the sequence $<c_k>$ where $c_k = a_k + b_k$
2. $A(x) \cdot B(x)$ is the generating function of the sequence $<d_k>$ where

$$d_k = \sum_{0 \le j \le k} a_j b_{k-j} \ .$$

** The sequence $<d_k>$ so defined is often called the convolution of $<a_k>$ and $<b_k>$

Example 1. Fibonacci Numbers

Consider a sequence $<a_k>$ defined by the recurrence relation

$a_0 = 1$, $a_1 = 1$

$a_n = a_{n-1} + a_{n-2}$

At first sight, it is not clear there is a familiar expression for a_n. However, as we'll see, the recurrence relation almost immediately reveals the generating function $A(x)$!

- Indeed, from the recurrence relation we have

$$\begin{aligned} A(x) &= a_0 + a_1x + \textstyle\sum_{n\geq 2} a_n x^n \\ &= 1 + x + \textstyle\sum_{n\geq 2} (a_{n-1} + a_{n-2}) x^n \\ &= 1 + x + x\textstyle\sum_{n\geq 2} a_{n-1} x^{n-1} + x^2 \textstyle\sum_{n\geq 2} a_{n-2} x^{n-2} \\ &= 1 + x + x(A(x) - 1) + x^2A(x) \end{aligned}$$

- Thus $(1 - x - x^2)A(x) = 1$, and $A(x) = \dfrac{1}{1 - x - x^2}$

It is now easy to obtain an explicit form of a_n th by partial fraction as follows.

Example 1. Fibonacci Numbers (continued)

We solve $A(x) = \dfrac{1}{1 - x - x^2}$ as follows:

$$A(x) = \frac{1}{(1-\frac{1}{2}x)^2 - \frac{5}{4}x^2}$$

$$= \frac{1}{(1-\frac{1}{2}x-\frac{\sqrt{5}}{2}x)(1-\frac{1}{2}x+\frac{\sqrt{5}}{2}x)}$$

$$= \frac{1}{(1-\frac{1+\sqrt{5}}{2}x)(1-\frac{1-\sqrt{5}}{2}x)}$$

This is of the form $A(x) = \dfrac{1}{(1-\alpha x)(1-\beta x)}$

$$= \frac{\alpha}{\alpha-\beta}\frac{1}{1-\alpha x} - \frac{\beta}{\alpha-\beta}\frac{1}{1-\beta x}$$

$$= \frac{\alpha}{\alpha-\beta}\sum_{n\geq 0}\alpha^n x^n - \frac{\beta}{\alpha-\beta}\sum_{n\geq 0}\beta^n xn$$

We have thus derived an exact formula for a_n (known as the n-th Fibonacci number):

$$a_n = \frac{\alpha}{\alpha-\beta}(\alpha^{n+1} - \alpha^{n+1}) \text{ for } n\geq 0, \text{ where } \alpha = \frac{1+\sqrt{5}}{2} \approx 1.7 \quad \beta = \frac{1-\sqrt{5}}{2} \approx -0.6$$

End