Counting Problems: Network Reliability

Διοίκηση & Οικονομική Τηλ/κών Δικτύων, ΕΚΠΑ

Χ. Κοντέλλης, ΜΟΠ084

The Problem

- Network Reliability
  - Given a connected, undirected graph, with failure probability specified for each edge, compute the probability that the graph becomes disconnected
- Applications
  - Network Design, Systems Reliability

Background

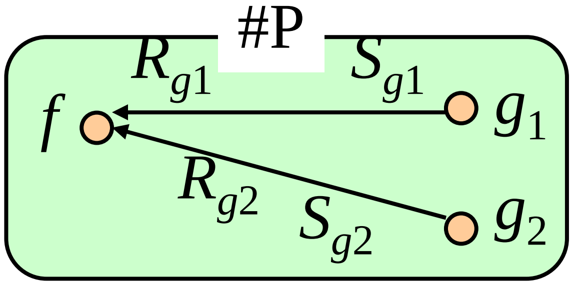
- Counting Problems: Counting # of solutions
  - Markov chain Monte Carlo method
  - Combinatorial Algorithms
- #P, #P-complete (‘Sharp’P, ‘Sharp’P-complete)
- Fully Polynomial Randomized Approximation Scheme (FPRAS)
- Counting DNF solutions

Definitions (1/3): of #P

- #P denotes a class of counting problems.
- We use the following notations for the definition.
  - L: a language in NP
    - all instances satisfying constraints of an NP problem
    - $L_{3SAT}=\{(x_1 \vee x_1 \vee x_1), (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3), \dots\}$
  - M: associated verifier for L
    - $M((x_1 \vee x_1 \vee x_2),((x_1, x_2)=(1, 1)))$:
  - p: polynomial bounding the length of M's Yes certificates (y).
    - $p_{3SAT}(|x|) \leqq c_1 n \leqq c_2 |x|$ (x: instance, n: # variables, c_1, c_2: constants).
  - $f(x)$: the number of strings y s.t. $|y|<p(|x|)$ and $M(x,y)$ accepts.
- Such $f(x)$ constitutes the class #P.

Definitions (2/3): of #P-complete

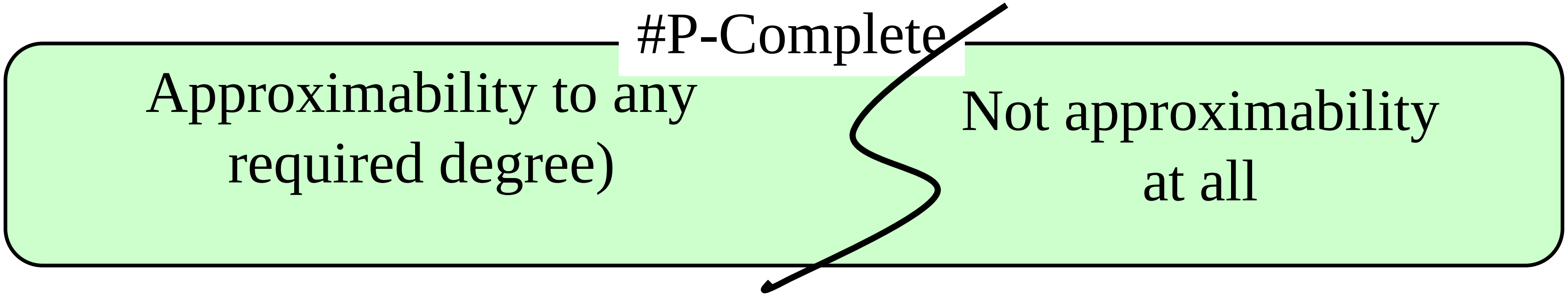
- #P-complete intuitively means one of the most intractable counting problems in NP.
- *f* is #P-complete if
  - *f* is in #P.



  - For any *g* in #P, *g* is reducible to *f* as follows:
    - There are a transducer *R* and a function *S* that are polynomial time computable.
      - $R(x)\in L_f \Leftrightarrow x\in L_g$.
      - $g(x)=S(x, f(R(x)))$.

Definitions (3/3): of FPRAS

- The solution counting versions of almost all known NP-complete problems are #P-complete.
- #P-complete problems admit only two (2) possibilities.



#P-Complete

Approximability to any required degree)

Not approximability at all

  - An algorithm *A* is an FPRAS
    - if, for any instance *x*,
    - *A* runs in poly. time in |*x*| and 1/ε, and

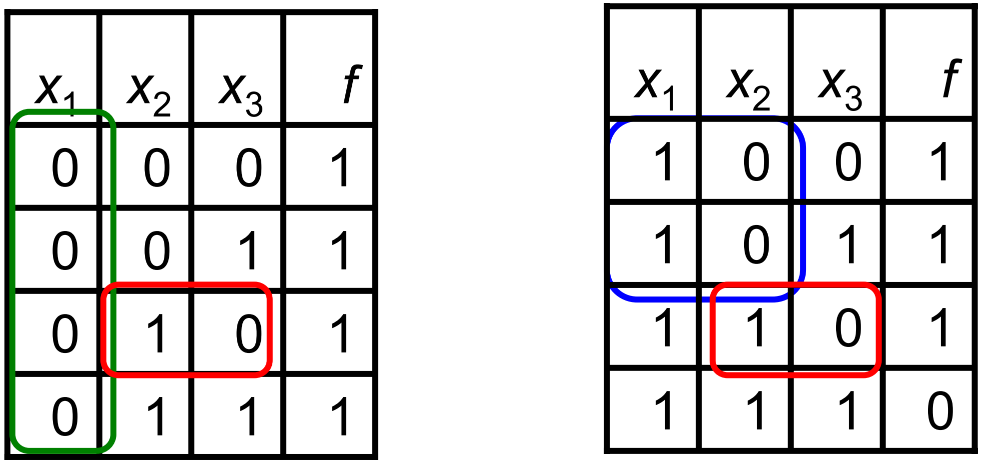
$$\Pr[|A(x) - f(x)| \leq \varepsilon\ f(x)] \geq \frac{3}{4}.$$

Issues in this chapter

- Definitions for counting # solutions
  - #P, #P-complete, fully polynomial randomized approximation scheme (FPRAS).
- Counting DNF solutions
- Network reliability

Counting DNF solutions

- Input:
  - a formula f in disjunctive normal form (DNF) on n Boolean variables.
    - E.g., $f_{EX}=\underline{(x_1\wedge\neg x_2)}\vee\underline{(x_2\wedge\neg x_3)}\vee\underline{(\neg x_1)}$.
- Output:
  - The number of satisfying truth assignments of f.
    - Let #f be the number (#f_{EX} is 7).



x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

x_1	x_2	x_3	f
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Efficiently approximate #f

- The main idea
    - Estimating #f by sampling a random variable X.
        - X must be an unbiased estimator, i.e., $\mathbf{E}[X]$=#f.
        - The standard deviation of X must be within a polynomial factor of $\mathbf{E}[X]$.
- A straightforward FPRAS
    - Sample X a poly. number of times (in n and $1/\varepsilon$).
    - Output the mean.

An unbiased estimator for #f

- Y and $Y(\tau)$ are defined as follows:
  - $Y(\tau)$: 2^n (τ satisfies f), 0 (otherwise).
  - $\Pr(Y)$: uniform distribution on all 2^n truth assignments.

$$
\begin{aligned}
E[Y(\tau)] &= \sum_{\tau} \left(\Pr(\tau)Y(\tau)\right) \\
&= \sum_{\tau:\tau \text{ satisfies } f} \frac{1}{2^n}2^n + \sum_{\tau:\tau \text{ does not satisfy } f} \frac{1}{2^n}0 \\
&= \sum_{\tau:\tau \text{ satisfies } f} 1 = \# f.
\end{aligned}
$$

- **E**[$Y(\tau)$] is then an unbiased estimator

x_1	x_2	x_3	f	Y
0	0	0	1	8
0	0	1	1	8
0	1	0	1	8
0	1	1	1	8

x_1	x_2	x_3	f	Y
1	0	0	1	8
1	0	1	1	8
1	1	0	1	8
1	1	1	0	0

Y is not efficient

$$\sigma^2[Y(\tau)] = \sum_{\tau} \Pr(\tau)\left(Y(\tau) - E[Y(\tau)]\right)^2$$

$$= \sum_{\tau:\tau \text{ satisfies } f} \frac{1}{2^n}\left(2^n - \# f\right)^2 + \sum_{\tau:\tau \text{ does not satisfy } f} \frac{1}{2^n}\left(0 - \# f\right)^2$$

$$= \sum_{\tau:\tau \text{ satisfies } f} \frac{1}{2^n}\left(2^{2n} - 2^{n+1}\# f + (\# f)^2\right) + \sum_{\tau:\tau \text{ does not satisfy } f} \frac{1}{2^n}\left(\# f\right)^2$$

$$= \frac{\# f}{2^n}\left(2^{2n} - 2^{n+1}\# f + (\# f)^2\right) + \frac{2^n - \# f}{2^n}\left(\# f\right)^2$$

$$= 2^n \# f - (\# f)^2.$$

Not bounded by a polynomial of *n*.
Not useful for constructing an FPRAS.

Constructing a new random variable

- X: a random variable with $X(\tau)>0$ only if τ satisfies f.
- S_i: a set of truth assignments that satisfy clause C_i.
  - $|S_i|=2^{n-ri}$ where r_i is the number of literals in clause C_i.
  - $\#f=|\cup S_i|$.
  - $c(\tau)$: # clauses that τ satisfies.
  - M: multiset union of the sets S_i.
    - $|M|=\Sigma|S_i|=\Sigma 2^{n-ri}$ is easy to compute.
- $X(\tau)$: $|M|/c(\tau)$.

Constructing a new random variable

- Example:
  - $f_{EX}=\underline{(x_1 \wedge \neg x_2)} \vee \underline{(x_2 \wedge \neg x_3)} \vee \underline{(\neg x_1)}$.
  - S_i: a set of truth assignments that satisfy clause C_i.
    - $S_1=\{(1,0,0), (1,0,1)\}$, $S_2=\{(0,1,0), (1,1,0)\}$, $S_3=\{(0,0,0), (0,0,1), (0,1,0), (0,1,1)\}$, $|S_1|=2^{3-2}=2$, $|S_2|=2^{3-2}=2$,$|S_3|=2^{3-1}=4$.
    - $\#f=|\cup S_i|$, $c(\tau)$: # clauses that τ satisfies.
    - M: multiset union of the sets S_i.
      - $M_{EX}=\langle$ (1,0,0), (1,0,1), $\underline{(0,1,0),}$ (1,1,0), (0,0,0), (0,0,1), $\underline{(0,1,0),}$ (0,1,1)$\rangle$.
    - $X(\tau)=|M|/c(\tau)$.

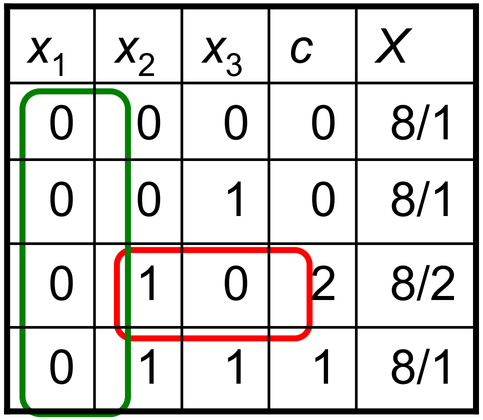
x_1	x_2	x_3	c	X
0	0	0	0	8/1
0	0	1	0	8/1
0	1	0	2	8/2
0	1	1	1	8/1

x_1	x_2	x_3	c	X
1	0	0	1	8/1
1	0	1	1	8/1
1	1	0	1	8/1
1	1	1	0	0

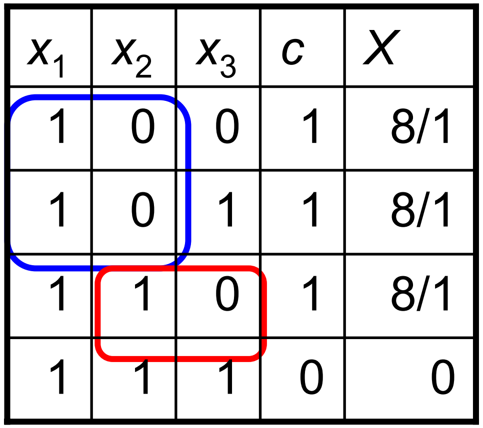
An FPRAS

- Example
  - $f_{EX}=(x_1 \wedge \neg x_2) \vee (x_2 \wedge \neg x_3) \vee (\neg x_1)$.
- for i=1 to k
  - Pick one clause C_j from f with prob. $|S_j|/|M|$.
    - C_1 with prob. 2/8.
  - Pick a truth assignment τ_i satisfying C_i at random.
    - τ_i= (1,0,1).
  - Find $c(\tau_i)$ and $X(\tau_i)= |M|/c(\tau_i)$.
    - $c(\tau_i)=1$, $X(\tau_i)=8/1=8$.
- end-for
- output $X_k=(X(\tau_1)+\ldots+X(\tau_k))/k$
  - $X_k=(8+4+8+8)/4=7$.

From lemma 28.2, τ is picked with prob. $c(\tau)/|M|$.

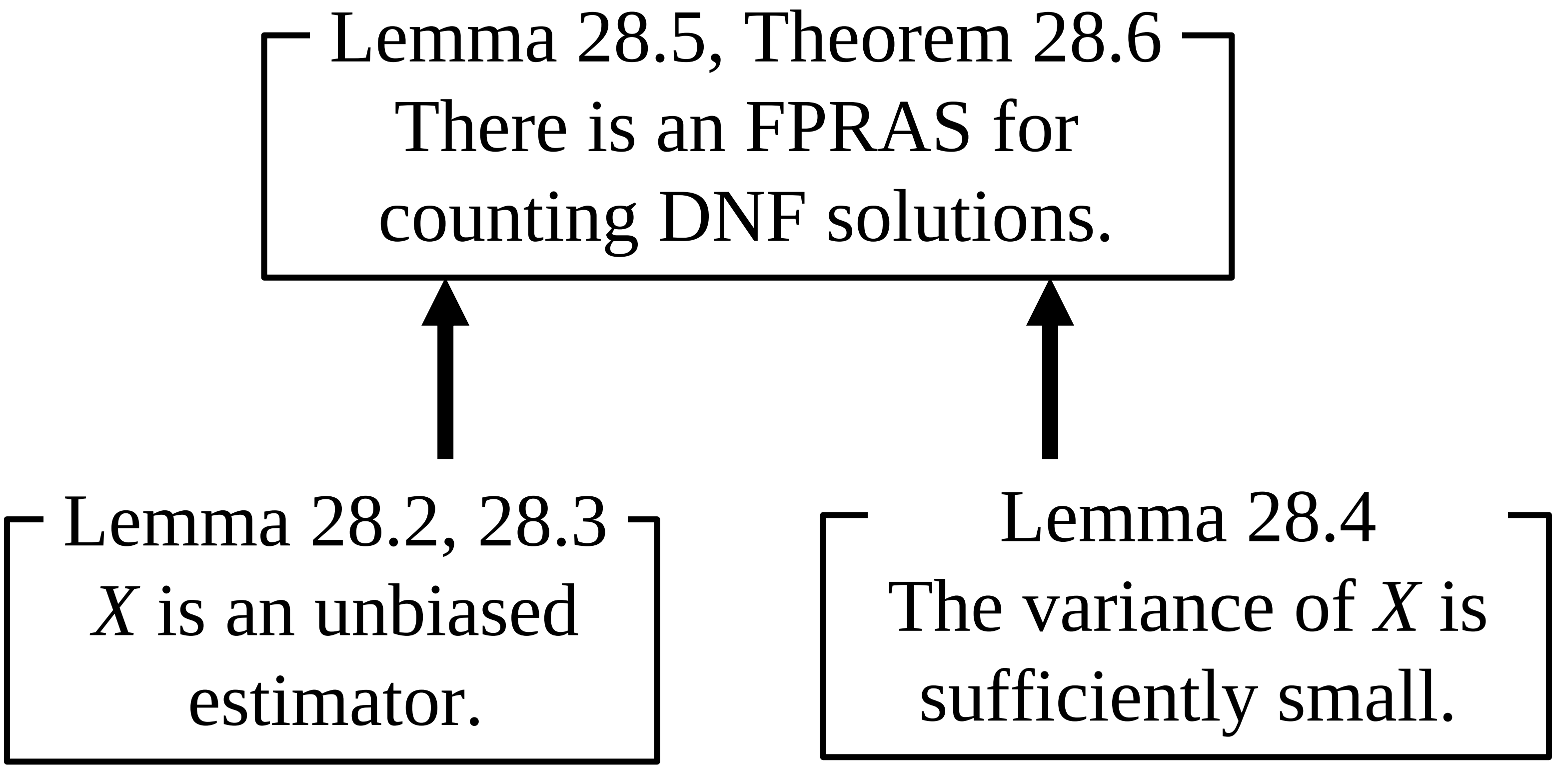


x_1	x_2	x_3	c	X
0	0	0	0	8/1
0	0	1	0	8/1
0	1	0	2	8/2
0	1	1	1	8/1



x_1	x_2	x_3	c	X
1	0	0	1	8/1
1	0	1	1	8/1
1	1	0	1	8/1
1	1	1	0	0

Overview



Lemma 28.5, Theorem 28.6
There is an FPRAS for counting DNF solutions.

Lemma 28.2, 28.3
X is an unbiased estimator.

Lemma 28.4
The variance of X is sufficiently small.

Lemma 28.2

- Random variable *X* can be efficiently sampled.
  - Sampling *X* is done with picking a random element from the multiset *M*.
    - 1. pick a clause so that the probability of picking clause C_i is $|S_i|/|M|$.
    - 2. among the truth assignments satisfying the picked clause, pick one at random.
  - The probability with which truth assignment τ is picked is

$$\sum_{i:\tau \text{ satisfies } C_i} \frac{|S_i|}{|M|}\frac{1}{|S_i|} = \frac{c(\tau)}{|M|}.$$

Lemma 28.3

- *X* is an unbiased estimator for #f.

$$E[X] = \sum_{\tau} \Pr[\tau \text{ is picked}] X(\tau)$$

$$= \sum_{\tau:\tau \text{ satisfies } f} \frac{c(\tau)}{|M|} \frac{|M|}{c(\tau)} = \# f.$$

Lemma 28.4

- α=|*M*|/*m*.
- If *m* denotes the number of clauses in *f*, then

$$\frac{\sigma(X)}{E[X]} \le m-1.$$

---

the average number of truth assignments satisfying one clause

$$\alpha = \frac{|M|}{m} = \frac{\sum_{i=1}^{m} |S_i|}{m} \le E[X] = \# f = |\cup_{i=1}^{m} S_i|.$$

the number of truth assignments satisfying at least one clause

---

clauses satisfied with τ

$$1 \le c(\tau) \le m. \Rightarrow \frac{|M|}{m} \le X(\tau) \le \frac{|M|}{1}. \Rightarrow |X(\tau) - E[X]| \le (m-1)\alpha.$$

#clauses in *f*.

$$X(\tau) = \frac{|M|}{c(\tau)}.$$

$$\alpha \le X(\tau) \le m\alpha.$$

$$\sigma(X) \le (m-1)\alpha$$

$$\le (m-1)E[X].$$

Lemma 28.5 and Theorem 28.6

- Let k=4(m - 1)^2/ε^2. For any ε>0,

$$\Pr[|X_k - \#f| \le \varepsilon \# f] \ge \frac{3}{4}.$$

Chebyshev's inequality $\Pr[|X - E[X]| \ge a] \le \left(\frac{\sigma(X)}{a}\right)^2.$

$$\Pr[|X_k - E[X_k]| \ge \varepsilon E[X_k]] \le \left(\frac{\sigma(X_k)}{\varepsilon E[X_k]}\right)^2 = \left(\frac{\sigma(X)}{\varepsilon\sqrt{k}\,E[X]}\right)^2$$

$$= \left(\frac{1}{\varepsilon}\frac{1}{\sqrt{k}}\frac{\sigma(X)}{E[X]}\right)^2 = \left(\frac{1}{\varepsilon}\frac{\varepsilon}{2(m-1)}(m-1)\right)^2 = \frac{1}{4}.$$

$$\because E[X_k] = E[X],$$

$$\sigma(X_k) = \sigma(X)/\sqrt{k}.$$

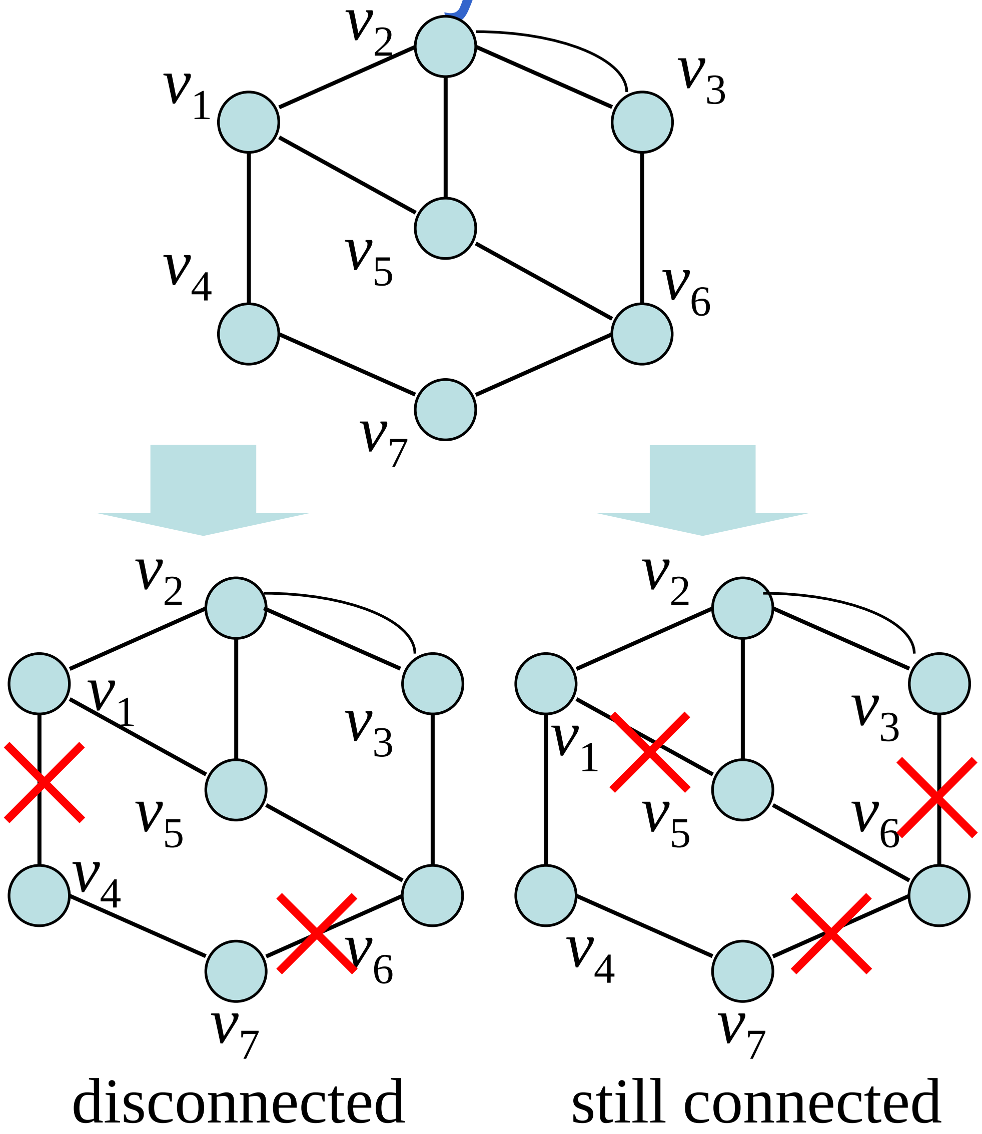
There is an FPRAS for the problem of counting DNF solutions.

Issues in this chapter

- Definitions for counting # solutions
  - #P, #P-complete, fully polynomial randomized approximation scheme (FPRAS).
- Counting DNF solutions
- Network reliability

Network reliability

- Input:
  - a connected undirected graph *G*=(*V*, *E*), with failure prob. for each edge *e*.
    - Parallel edges between two nodes are allowed.
- Output:
  - The prob. that the graph becomes disconnected.
    - Denote the prob. by FAIL(*p*).



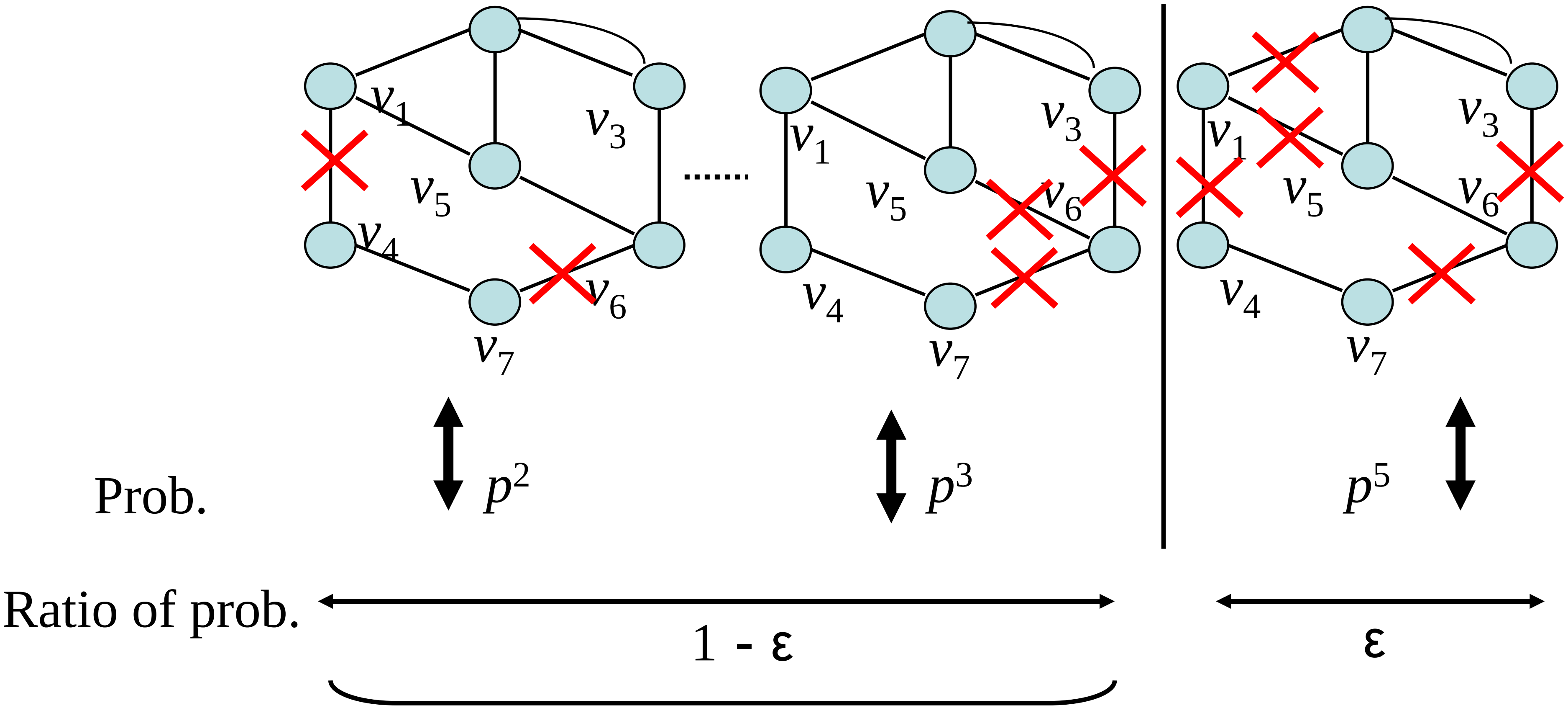
Tractability of FAIL(p)

- Tractable if FAIL(p) is not small.
  - "Small" means at least inverse polynomial.
  - FAIL(p) can be estimated by sampling.
    - We will explain it later (in the proof of Theorem 28.11).
- Intractable if FAIL(p) is small.
  - Sampling approaches do not work.
    - Many samplings are required for the estimation.
  - In the following, we assume that FAIL(p) $\leqq n^{-4}$.
- Pr(cut $(C,\overline{C})$ gets disconnected)=p^c.
  - where capacity c is the number of edges crossing the cut.
  - p^c decreases exponentially with capacity (# edges, c).

Ideas of the algorithm

- For any ε>0, we will show that only polynomially many "small'' cuts (in *n* and 1/ε) are responsible for 1 - ε fraction of the total failure probability FAIL(*p*). Moreover, these cuts, say $E_1,\ldots,E_k$, $E_i \subseteq E$, can be enumerated in polynomial time.
- We refrain to compute the probability that one of the above cuts fails; because of correlations, this is non trivial, instead:
- We will construct a polynomial sized DNF formula *f* whose probability of being satisfied is precisely the probability that at least one of these cuts fails.

Illustration of the idea (1/2)

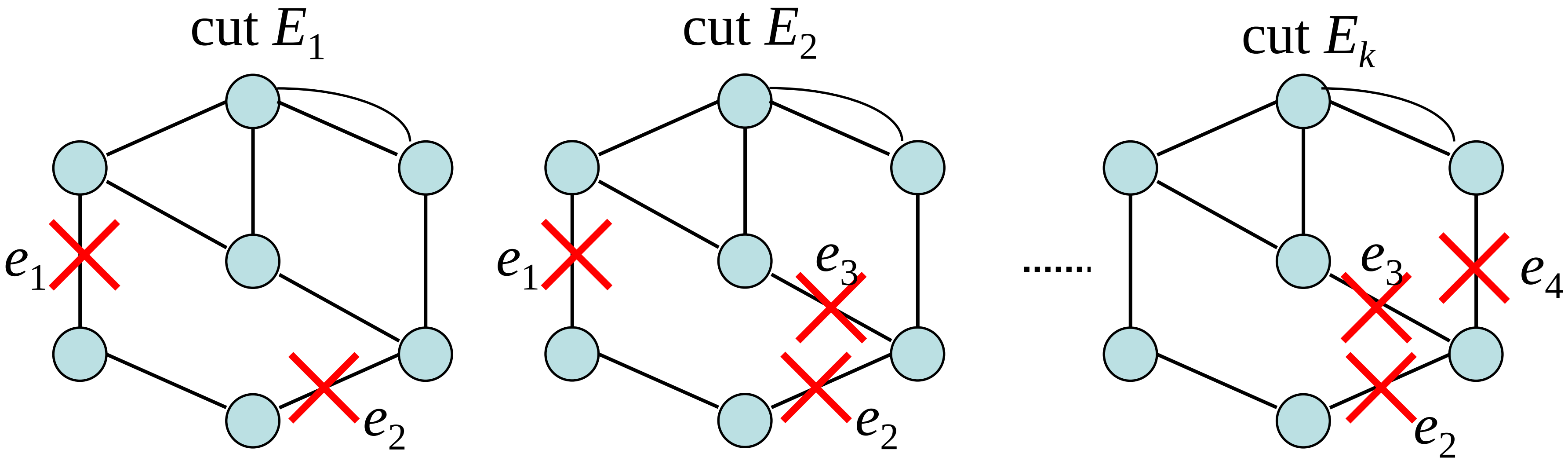


Prob. p^2 p^3 p^5

Ratio of prob. 1 - ε ε

Enumerable in polynomial time
(Exercise 28.11-13)

Illustration of the idea (2/2)



One-to-one correspondence ↕

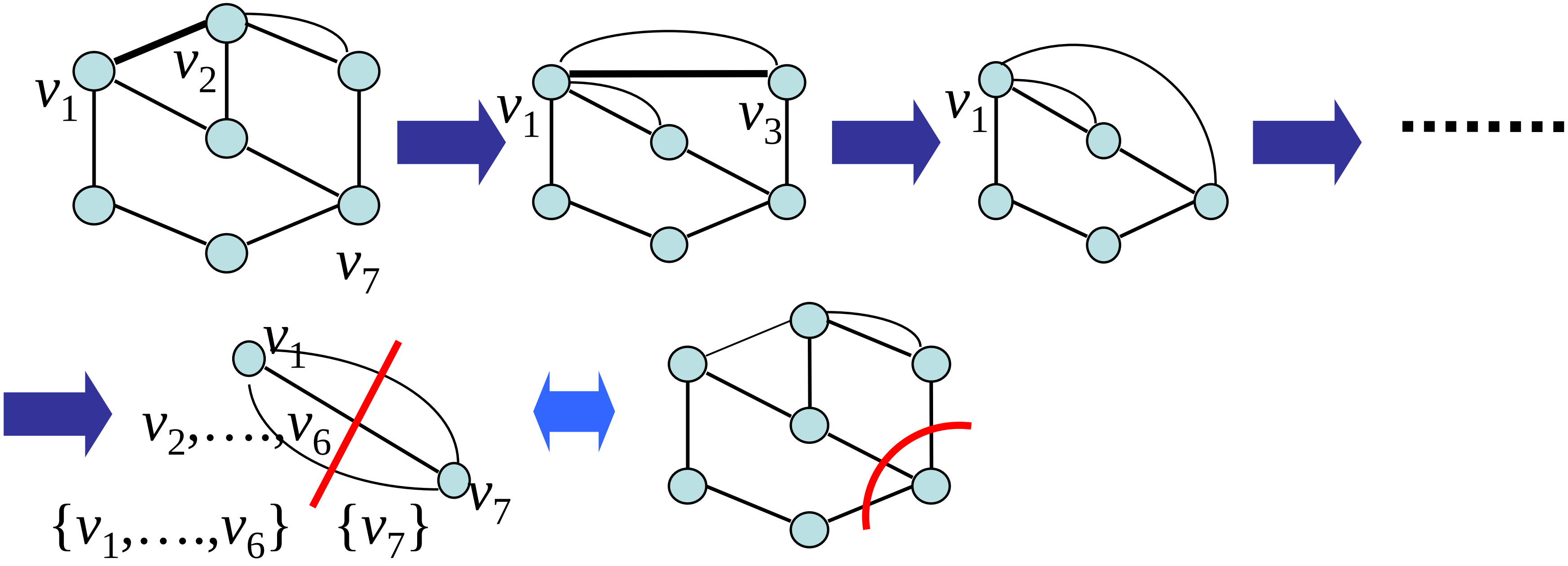
$$D_1 = x_{e_1} \wedge x_{e_2}, \qquad D_2 = x_{e_1} \wedge x_{e_2} \wedge x_{e_3}, \qquad D_k = x_{e_2} \wedge x_{e_3} \wedge x_{e_4},$$

x_{ei} is true with probability p_{ei}.

$$f = D_1 \vee \cdots \vee D_k.$$

Lemma 28.8 (1/3)

- The number of minimum cuts in $G=(V,E)$ is bounded by $n(n-1)/2$.
  - Contractions of an edge.



Cut $(\{v_1,\ldots,v_6\}, \{v_7\})$ *survives*.

Lemma 28.8 (2/3)

- Let *M* be the number of minimum cuts in *G*.
  - *M* is bounded by *n*(*n*-1)/2 if

$$\Pr[(C,\overline{C})\,\text{survives}] \geq \frac{2}{n(n-1)}.$$

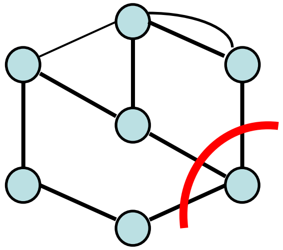
$$1 = \sum_{(C,\overline{C})} \Pr[(C,\overline{C})\,\text{survives}].$$

$$1 \geq \sum_{\substack{(C,\overline{C}):s.t.(C,\overline{C})\,\text{is} \\ \text{a minimum cut}}} \Pr[(C,\overline{C})\,\text{survives}] \geq \frac{2M}{n(n-1)}.$$

$$\frac{n(n-1)}{2} \geq M.$$

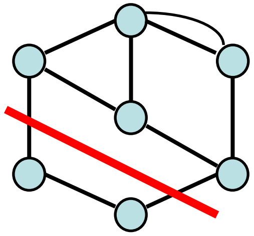
Lemma 28.8 (3/3)

- *H*: a graph at the beginning of contraction process.
  - Contractions never decrease the capacity of the minimum cut.
    - The degree of each node in *H* is at least *c*.
    - *m* is the number of nodes in *H*.
    - Hence, *H* must have at least *cm*/2 edges.
  - The minimum cut survives with the probability (1-*c*/#edges).



$$\left(1-\frac{c}{\#edges}\right)\ge\left(1-\frac{c}{cm/2}\right)=\left(1-\frac{2}{m}\right).$$

$$\Pr[(C,\overline{C})\,\text{survives}]\ge\left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{3}\right)=\frac{2}{n(n-1)}.$$



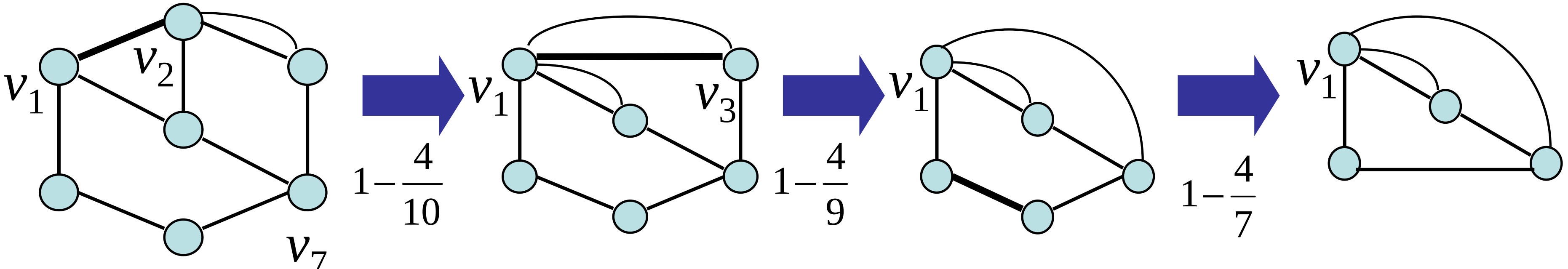
Lemma 28.9 (1/3)

- For any $\alpha \geqq 1$, the number of α-min cuts in G is at most $n^{2\alpha}$.
  - A cut is an α-min cut if its capacity is at most αc.
  - We assume α is a half-integer. Let $k=2\alpha$.
    - *(for arbitrary α can be proved by applying same ideas to generalized binomial coefficients - left as an exercise)*
  - Consider the two-phase process.
    - 1. Contract edges at random until there remain k nodes in the graph.
    - 2. Pick up a cut from all $2^{k-1} - 1$ at random.

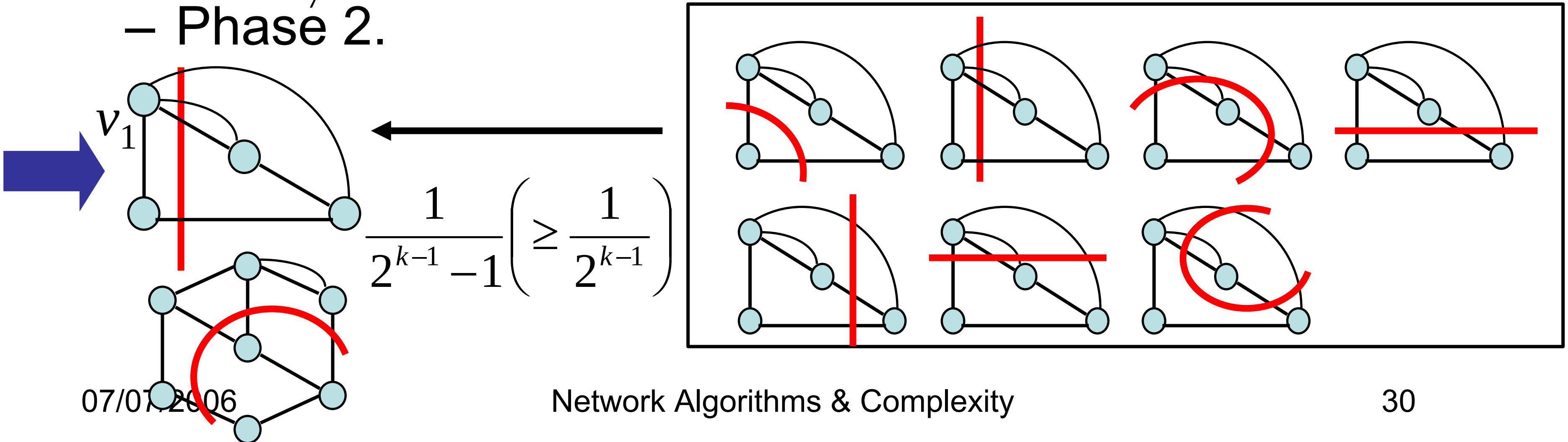
Lemma 28.9 (2/3)

- Example
  - k=4.
  - Phase 1.

$$\Pr[(C,\overline{C})\text{ survives}] \ge \left(1-\frac{k}{n}\right)\left(1-\frac{k}{n-1}\right)\cdots\left(1-\frac{k}{k+1}\right)=\frac{k(k-1)\cdots 1}{n(n-1)\cdots(n-k+1)}.$$



  - Phase 2.



Lemma 28.9 (3/3)

$$\Pr[(C,\overline{C})\text{ survives through the two phases}]$$

$$\geq \frac{k(k-1)\cdots 1}{n(n-1)\cdots(n-k+1)}\frac{1}{2^{k-1}}$$

$$=\frac{k}{2n}\frac{k-1}{2(n-1)}\cdots\frac{2}{2(n-k+2)}\frac{1}{n-k+1}$$

$$\geq \frac{1}{n^{k}}=\frac{1}{n^{2\alpha}}.$$

FAIL(p): Analysis

- In case that FAIL(p) $\leqq n^{-4}$.
- The failure probability of a minimum cut is $p^c \leqq$ FAIL(p) $\leqq n^{-4}$.
- *Let* $p^c = n^{-(2+\delta)}$, $\delta \geq 2$
- From lemma 28.9, for any $\alpha \geqq 1$, the total failure probability of all cuts of capacity αc is at most $p^{c\alpha} n^{2\alpha} = n^{-\alpha\delta}$.

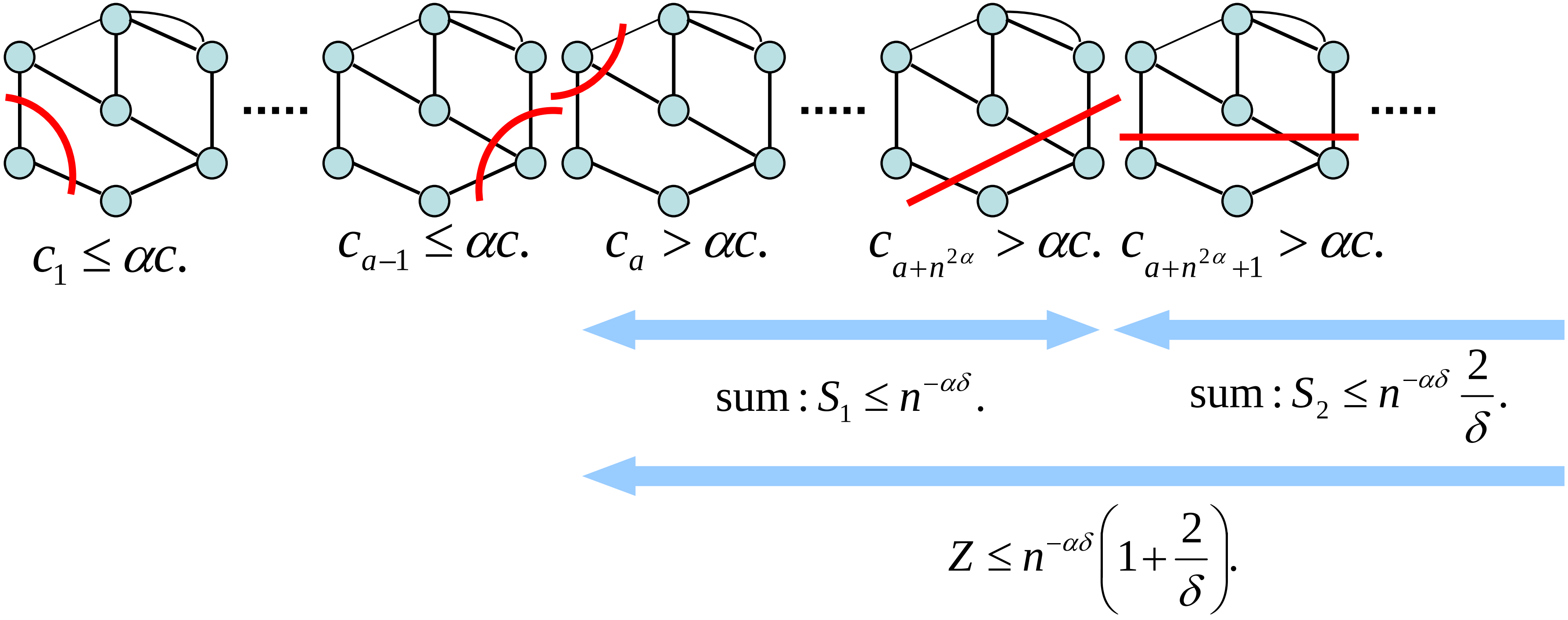
Lemma 28.10 (1/3)

- For any α,
  - $Z = \Pr[\text{some cut of capacity} > \alpha c \text{ fails}] \leq n^{-\alpha\delta}\left(1+\frac{2}{\delta}\right).$
  - For bounding the total failure prob. of "large" capacity cuts.
  - Number all cuts in *G* by increasing capacity.
    - c_k: the capacity of the *k*-th cut in this numbering.
    - p_k: the failure probability of the *k*-th cut.
    - *a*: the number of the first cut of capacity greater than α*c*.
  - It suffices to show that

$$Z = \sum_{k \geq a} p_k = \sum_{k=a}^{a+n^{2\alpha}} p_k + \sum_{k > a+n^{2\alpha}} p_k \leq n^{-\alpha\delta}\left(1+\frac{2}{\delta}\right) \quad \left(\sum_{k=a}^{a+n^{2\alpha}} p_k \leq n^{-\alpha\delta},\ \sum_{k > a+n^{2\alpha}} p_k \leq n^{-\alpha\delta}\frac{2}{\delta}\right).$$

Lemma 28.10 (2/3)

- Illustration of the idea of lemma 28.10
  - Number all cuts in *G* by increasing capacity.



$c_1 \le \alpha c.$ $c_{a-1} \le \alpha c.$ $c_a > \alpha c.$ $c_{a+n^{2\alpha}} > \alpha c.$ $c_{a+n^{2\alpha}+1} > \alpha c.$

$\mathrm{sum}: S_1 \le n^{-\alpha\delta}.$

$\mathrm{sum}: S_2 \le n^{-\alpha\delta} \frac{2}{\delta}.$

$$Z \le n^{-\alpha\delta}\left(1 + \frac{2}{\delta}\right).$$

Lemma 28.10 (3/3)

- For c_k ($a \leqq k \leqq a+n^{2\alpha}$),
  - $c_k > \alpha c \rightarrow p_k < p^{\alpha c} = n^{-\alpha(2+\delta)}$.

$$\sum_{k=a}^{a+n^{2\alpha}} p_k \leq \sum_{k=a}^{a+n^{2\alpha}} n^{-\alpha(2+\delta)} = n^{2\alpha} n^{-\alpha(2+\delta)} = n^{-\alpha\delta}.$$

- For c_k ($k \geqq a+n^{2\alpha}$),
  - at most $n^{2\alpha}$ cuts with the capacity less than αc exist.
      - from lemma 28.9.
    - Then, for any β, $c_{n^{2\beta}} \geqq \beta c$.
    - Replacing $n^{2\beta}$by k, we obtain $\beta = \log k/(2 \log n)$, and
    - Therefore,

$$p_k \leq (p^c)^{\frac{\ln k}{2\ln n}} = k^{-(1+\delta/2)}.$$

$$\sum_{k>a+n^{2\alpha}} p_k \leq \sum_{k>n^{2\alpha}} p_k \leq \int_{n^{2\alpha}}^{\infty} k^{-(1+\delta/2)} dk = \frac{1}{1+\delta/2} n^{-\alpha\delta} \leq \frac{2}{\delta} n^{-\alpha\delta}.$$

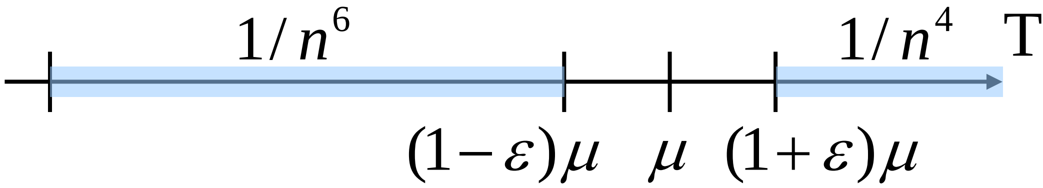
Theorem 28.11 (1/4)

- There is an FPRAS for estimating network reliability.
  - In case that FAIL(*p*)>n^{-4}.
  - The network is connected/disconnected: binomial distribution.
    - Sampling and Chernoff bound are used to estimate FAIL(*p*)=μ.

$$\Pr[X > (1-\varepsilon)\mu] \le e^{-k\mu\varepsilon^2/2}, \Pr[X > (1+\varepsilon)\mu] \le e^{-k\mu\varepsilon^2/3}.$$

$$k = 12\log n/(\varepsilon^2\mu) < 12n^4\log n/\varepsilon^2,$$

$$\Pr[X > (1-\varepsilon)\mu] \le e^{-12\log n/2} = n^{-6}, \Pr[X > (1+\varepsilon)\mu] \le e^{-12\log n/3} = n^{-4}.$$



The light blue areas are less than 1/4 if *n*>2.

Theorem 28.11 (2/4)

- In case that FAIL(p)≦n^{-4}.
- α must be determined for enumerating graphs with high probabilities such that

$$\underbrace{\Pr[\text{some cut of capacity} > \alpha c \text{ fails}] \le n^{-\alpha\delta}\left(1+\frac{2}{\delta}\right)}_{\text{By lemma 28.10}} \underline{\le \varepsilon\, \mathrm{FAIL}(p) \le \varepsilon n^{-(2+\delta)}}.$$

This inequality is given in the textbook, but this seems to contradict with $n^{-(2+\delta)} = \underset{\substack{\uparrow \\ \text{one failure}}}{p^c} \le \underset{\substack{\uparrow \\ \text{total failure}}}{\mathrm{FAIL}(p)}$ in p. 300.

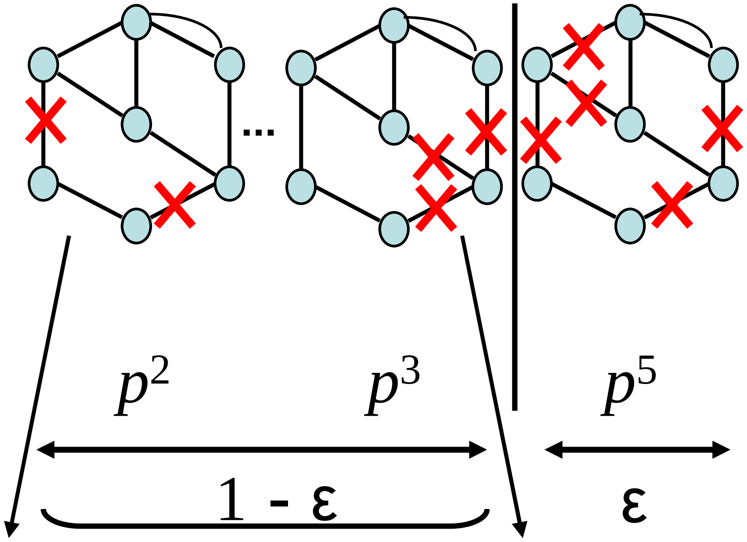
$$n^{-\alpha\delta} \le \varepsilon n^{-(2+\delta)}$$

?

$$\alpha \le 1 + \frac{2}{\delta} - \frac{\log \varepsilon / 2}{\delta \log n} \le 2 - \frac{\log \varepsilon / 2}{2 \log n}.$$

Theorem 28.11 (3/4)

- By lemma 28.9, $c_{n^{2\alpha}} > \alpha c.$
- Pr[one of the first $n^{2\alpha}$ fails] ≧(1 - ε)FAIL(p).
- The first $n^{2\alpha} = O(n^4/\varepsilon)$ cuts are enumerable in polynomial time (Exercise 28.11-13).



$$D_1 = x_{e_1} \wedge x_{e_2}, \quad D_{n^{2\alpha}} = x_{e_1} \wedge x_{e_2} \wedge x_{e_3}. \longrightarrow \quad f = D_1 \vee \cdots \vee D_{n^{2\alpha}}.$$

Theorem 28.11 (4/4)

- To reduce the case of arbitrary edge failure probabilities, parallel edges are used.

failure
probability
p_e

parallel
$-(\ln p_e)/\theta$ edges
failure
probability θ

all edges are disconnected
with prob. $(1-\theta)^{-(\ln p_e)/\theta}$

$$\lim_{\theta\to 0}(1-\theta)^{-(\ln p_e)/\theta} = e^{\ln p_e} = p_e.$$

Open Issues

- Probability that s-t fails
- Probability that s-t remains connected
- Probability the graph remains connected

References

- Vazirani – Approximation Algorithms (ch28)
- Karge - Using randomized sparsification to approximate minimum cuts
- San Diego University, Theory of Parallel Algorithms, Chernoff Bounds
- Kumar, Randomized min cut
- Vempala, minimum cuts

Thank you