Routing

Second main application of Chernoff: analysis of load balancing.

- Already saw balls in bins example
- synchronous message passing
- bidirectional links, one message per step
- queues on links
- **permutation** routing
- **oblivious algorithms** only consider self packet.
- **Theorem** Any deterministic oblivious permutation routing requires $\Omega(\sqrt{N/d})$ steps on an N node degree d machine.
  - reason: some edge has lots of paths through it.
  - **homework:** special case
- Hypercube.
  - N nodes, $n = \log_2 N$ dimensions
  - Nn directed edges
  - bit representation
  - natural routing: bit fixing (left to right)
  - paths of length n—lower bound on routing time
  - Nn edges for N length n paths suggest no congestion bound
  - but deterministic bound $\Omega(\sqrt{N/n})$
- Randomized routing algorithm:
  - $O(n) = O(\log N)$ randomized
  - how? load balance paths.
- First idea: random destination (not permutation!), bit correction
  - Average case, but a good start.

  - $T(e_i)$ = number of paths using e_i
  - by symmetry, all $E[T(e_i)]$ equal
  - expected path length $n/2$
  - LOE: expected total path length $Nn/2$
  - nN edges in hypercube
  - Deduce $E[T(e_i)] = 1/2$
  - Chernoff: every edge gets $\leq 3n$ (prob $1 - 1/N$)

- Naive usage:
  - n phases, one per bit
  - $3n$ time per phase
  - $O(n^2)$ total

- Worst case destinations
  - Idea [Valiant-Brebner] From intermediate destination, route back!
  - routes **any** permutation in $O(n^2)$ expected time.
  - what's going in with $\sqrt{N/n}$ lower bound?
  - Adversary doesn't know our routing so cannot plan worst permutation

- What if don't wait for next phase?
  - FIFO queuing
  - total time is length plus **delay**
  - Expected delay $\leq E[\sum T(e_l)] = n/2$.
  - Chernoff bound? no. dependence of $T(e_i)$.

- High prob. bound:
  - consider paths sharing i's fixed route $(e_0, \ldots, e_k)$
  - Suppose S packets intersect route (use at least one of e_r)
  - claim delay $\leq |S|$

- Suppose true, and let $H_{ij} = 1$ if j hits i's (fixed) route.

$$\begin{aligned} E[\text{delay}] &\leq E[\sum H_{ij}] \\ &\leq E[\sum T(e_l)] \\ &\leq n/2 \end{aligned}$$

- Now Chernoff **does** apply (H_{ij} independent for fixed i-route).
- $|S| = O(n)$ w.p. $1 - 2^{-5n}$, so $O(n)$ delay for all 2^n paths.

- Lag argument
  - Exercise: once packets separate, don't rejoin
  - Route for i is $\rho_i = (e_1, \ldots, e_k)$
  - charge each delay to a departure of a packet from ρ_i.
  - Packet waiting to follow e_j at time t has: **Lag** $t - j$
  - Delay of i is lag crossing e_k
  - When i delay rises to $l + 1$, some packet from S has lag l (since crosses e_j instead of i).
  - Consider last time t' where a lag-l packet exists on path
    - some lag-l packet w crosses $e_{j'}$ at t' (others increase to lag-$(l + 1)$)
    - w leaves at this point (if not, then l at $e_{j'+1}$ next time)
    - charge one delay to w.

Summary:

- 2 key roles for chernoff
- sampling
- load balancing
- "high probability" results at $\log n$ means.

The Probabilistic Method—Value of Random Answers

Idea: to show an object with certain properties exists

- generate a random object
- prove it has properties with nonzero probability
- often, "certain properties" means "good solution to our problem"

Max-Cut:

- Define
- NP-complete
- Approximation algorithms
- factor 2
- "expected performance," so doesn't really fit our RP/ZPP framework
- but does show such a cut **exists**

Set balancing.

- minimize max bias.
- $4\sqrt{n \ln n}$.
- Spencer—10 lectures on the probabilistic method

Expanders

Existence vs. constriction

- Of course, many probabilistic method constructions yield constructive algorithms
- In maxcut, just try till succeed
- Other times, are only existential proofs, or very bad algorithms
- But motivate search for good algorithm

Definition: (n, d, α, c) OR-concentrator

- bipartite $2n$ vertices
- degree at most d in L
- expansion c on sets $< \alpha n$.

Applications:

- switching/routing
- ECCs

claim: $(n, 18, 1/3, 2)$-concentrator

- Construct by sampling d random neighbors with replacement
  - E_s: Specific size s set has $< cs$ neighbors.
  - fix S of size s. T of size $< cs$.
  - prob. S goes to T at most $(cs/n)^{ds}$
  - $\binom{n}{cs}$ sets T
  - $\binom{n}{s}$ sets S
  -

$$\begin{aligned}
\Pr[] &\leq \binom{n}{s}\binom{n}{cs}(cs/n)^{ds} \\
&\leq (en/s)^s (en/cs)^{cs} (cs/n)^{ds} \\
&= [(s/n)^{d-c-1} e^{c+1} c^{d-c}]^s \\
&\leq [(1/3)^{d-c-1} e^{c+1} c^{d-c}]^s \\
&\leq [(c/3)^d (3e)^{c+1}]^s
\end{aligned}$$

  - Take $c = 2, d = 18$, get $[(2/3)^{18}(3e)^3]^{<} 2^{-s}$
  - sum over s, get < 1

Existence proof

- No known construction this good.
- NP-hard to verify
- but some constructions almost this good
- recent progress via zig-zag product

Wiring

Sometimes, it's hard to get hands on a good probability distribution of random answers.

- Problem formulation
  - $\sqrt{n} \times \sqrt{n}$ *gate array*
  - Manhattan wiring
  - boundaries between gates
  - fixed width boundary means limit on number of crossing wires
  - optimization vs. feasibility: minimize max crossing number
  - focus on single-bend wiring. two choices for route.
  - Generalizes if you know about max-flow
- Linear Programs, integer linear programs
  - Black box
  - Good to know, since great solvers exist in practice
  - Solution techniques in other courses
- IP formulation
  - x_{i0} means x_i starts horizontal, x_{i1} vertical
  - $T_{b0} = \{i \mid \text{net } i \text{ through } b \text{ if } x_{i0}\}$
  - T_{b1}
  - IP

$$\begin{aligned} \min \quad & w \\ x_{i0} + x_{i1} &= 1 \\ \sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} &\leq w \end{aligned}$$

- Solution $\hat{x}_{i0}$, $\hat{x}_{i1}$, value $\hat{w}$.
- rounding is Poisson vars, mean $\hat{w}$.

- $\Pr[\geq (1+\delta)\hat{w}] \leq e^{-\delta^2 \hat{w}/4}$
- need $2n$ boundaries, so aim for prob. bound $1/2n^2$.
- solve, $\delta = \sqrt{(4 \ln 2n^2)/\hat{w}}$.
- So absolute error $\sqrt{8\hat{w} \ln n}$
  - Good ($o(1)$-error) if $\hat{w} \gg 8 \ln n$
  - Bad ($O(\ln n)$ error) is $\hat{w} = 2$
  - General rule: randomized rounding good if target logarithmic, not if constant

MAX SAT

Define.

- literals
- clauses
- NP-complete

random set

- achieve $1 - 2^{-k}$
- very nice for large k, but only $1/2$ for $k = 1$

LP

$$\begin{aligned} \max \quad & \sum z_j \\ \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_1) \geq z_j \end{aligned}$$

Analysis

- $\beta_k = 1 - (1 - 1/k)^k$. values $1, 3/4, .704, \ldots$
- Lemma: k-literal clause sat w/pr at least $\beta_k \hat{z}_j$.
- proof:

- assume all positive literals.
- prob $1-\prod(1-y_i)$
-
- maximize when all $y_i = \hat{z}_j/k$.
- Show $1-(1-\hat{z}/k)^k \geq \beta_k \hat{z}_k$.
- check at $z = 0, 1$

- Result: $(1-1/e)$ approximation (convergence of $(1-1/k)^k$)
- much better for small k: i.e. 1-approx for $k = 1$

LP good for small clauses, random for large.

- Better: try both methods.
- n_1, n_2 number in both methods
- Show $(n_1+n_2)/2 \geq (3/4)\sum \hat{z}_j$
- $n_1 \geq \sum_{C_j \in S^k}(1-2^{-k})\hat{z}_j$
- $n_2 \geq \sum \beta_k \hat{z}_j$
- $n_1+n_2 \geq \sum(1-2^{-k}+\beta_k)\hat{z}_j \geq \sum \frac{3}{2}\hat{z}_j$