Parallel Algorithms

PRAM

- $P$ processors, each with a RAM, local registers
- global memory of $M$ locations
- each processor can in one step do a RAM op or read/write to one global memory location
- synchronous parallel steps
- various conflict resolutions (CREW, EREW, CRCW)
- not realistic, but explores “degree of parallelism”

Randomization in parallel:

- load balancing
- symmetry breaking
- isolating solutions

Classes:

- NC: poly processor, polylog steps
- RNC: with randomization. polylog runtime, monte carlo
- ZNC: las vegas NC
- immune to choice of R/W conflict resolution

Practical observations:

- very little can be done in $o(\log n)$ with poly processors (binary tree of data aggregation usually needed)
- lots can be done in $\Theta(\log n)$
- often concerned about work which is processors times time
- algorithm is “optimal” if work equals best sequential

Basic operations

- and, or
- counting ones
- parallel prefix
Addition

- Prefix sum over “kill”, “propagate”, “carry” operations
- handles $n$-bit numbers in $O(\log n)$ time
- multiplication as $n^2$ additions (better methods exist)

Sorting

Quicksort in parallel:

- $n$ processors
- each takes one item, compares to splitter
- count number of predecessors less than splitter
- determines location of item in split
- total time $O(\log n)$
- combine: $O(\log n)$ per layer with $n$ processors
- problem: $\Omega(\log^2 n)$ time bound
- problem: $n \log^2 n$ work
- tweak (using $\sqrt{n}$ splitters) to get optimal

Perfect Matching

We focus on bipartite; book does general case.
Last time, saw detection algorithm in $\mathcal{RNC}$:

- Tutte matrix
- Symbolic determinant nonzero iff PM
- assign random values in $1, \ldots , 2m$
- Matrix Mul, Determinant in $\mathcal{NC}$

How about finding one?

- If unique, no problem
- Since only one nonzero term, ok to replace each entry by a 1.
- Remove each edge, see if still PM in parallel
- multiplies processors by $m$
• but still NC

Idea:
• make unique minimum weight perfect matching
• find it

Isolating lemma: [MVV]
• Family of distinct sets over \( x_1, \ldots, x_m \)
• assign random weights in \( 1, \ldots, 2m \)
• \( \Pr(\text{unique min-weight set}) \geq 1/2 \)
• Odd: no dependence on number of sets!
• (of course \(< 2^m \))

Proof:
• Fix item \( x_i \)
• \( Y \) is min-value sets containing \( x_i \)
• \( N \) is min-value sets not containing \( x_i \)
• true min-sets are either those in \( Y \) or in \( N \)
• how decide? Value of \( x_i \)
• For \( x_i = -\infty \), min-sets are \( Y \)
• For \( x_i = +\infty \), min-sets are \( N \)
• As increase from \( -\infty \) to \( \infty \), single transition value when both \( X \) and \( Y \) are min-weight
• If only \( Y \) min-weight, then \( x_i \) in every min-set
• If only \( X \) min-weight, then \( x_i \) in no min-set
• If both min-weight, \( x_i \) is ambiguous
• Suppose no \( x_i \) ambiguous. Then min-weight set unique!
• Exactly one value for \( x_i \) makes it ambiguous given remainder
• So \( \Pr(\text{ambiguous}) = 1/2m \)
• So \( \Pr(\text{any ambiguous}) < m/2m = 1/2 \)

Usage:
- Consider tutte matrix $A$
- Assign random value $2^{w_i}$ to $x_i$, with $w_i \in 1, \ldots, 2m$
- Weight of matching is $2\sum w_i$
- Let $W$ be minimum sum
- Unique w/pr $1/2$
- If so, determinant is odd multiple of $2^W$
- Try removing edges one at a time
- Edge in PM iff new determinant$/2^W$ is even.
- Big numbers? No problem: values have poly number of bits

$NC$ algorithm open.
For exact matching, $P$ algorithm open.

**Maximal independent set**

trivial sequential algorithm

- inherently sequential
- from node point of view: each thinks can join MIS if others stay out
- randomization breaks this symmetry

Randomized idea

- each node joins with some probability
- all neighbors excluded
- many nodes join
- few phases needed

Algorithm:

- all degree 0 nodes join
- node $v$ joins with probability $1/2d(v)$
- if edge $(u, v)$ has both ends marked, unmark lower degree vertex
- put all marked nodes in IS
- delete all neighbors
Intuition: $d$-regular graph

- vertex vanishes if it or neighbor gets chosen
- mark with probability $1/2d$
- prob (no neighbor marked) is $(1 - 1/2d)^d$, constant
- so const prob. of neighbor of $v$ marked—destroys $v$
- what about unmarking of $v$’s neighbor?
- prob(unmarking forced) only constant as argued above.
- So just changes constants
- const fraction of nodes vanish: $O(\log n)$ phases
- Implementing a phase trivial in $O(\log n)$.

Prob chosen for IS, given marked, exceeds 1/2

- suppose $w$ marked. only unmarked if higher degree neighbor marked
- higher degree neighbor marked with prob. $\leq 1/2d(w)$
- only $d(w)$ neighbors
- prob. any superior neighbor marked at most 1/2.

For general case, define good vertices

- good: at least $1/3$ neighbors have lower degree
- prob. no neighbor of good marked $\leq (1 - 1/2d(v))^{d(v)/3} \leq e^{-1/6}$.
- So some neighbor marked with prob. $1 - e^{-1/6}$
- Stays marked with prob. $1/2$
- deduce prob. good vertex killed exceeds $(1 - e^{-1/6})/2$
- Problem: perhaps only one good vertex?

Good edges

- any edge with a good neighbor
- has const prob. to vanish
- show half edges good
- deduce $O(\log n)$ iterations.
Proof

- Let $V_B$ be bad vertices; we count edges with both ends in $V_B$.
- direct edges from lower to higher degree $d_i$ is indegree, $d_o$ outdegree
- if $v$ bad, then $d_i(v) \leq d(v)/3$
- deduce

$$\sum_{v \in V_B} d_i(v) \leq \frac{1}{3} \sum_{v \in V_B} d(v) = \frac{1}{3} \sum_{v \in V_B} (d_i(v) + d_o(v))$$

- so $\sum_{v \in V_B} d_i(v) \leq \frac{1}{2} \sum_{v \in V_B} d_o(v)$
- which means indegree can only “catch” half of outdegree; other half must go to good vertices.
- more carefully,
  - $d_o(v) - d_i(v) \geq \frac{1}{3} (d(v)) = \frac{1}{3} (d_o(v) + d_i(v))$.
  - Let $V_G, V_B$ be good, bad vertices
  - degree of bad vertices is

$$2e(V_B, V_B) + e(V_B, V_G) + e(V_G, V_B) = \sum_{v \in V_B} d_o(v) + d_i(v)$$

$$\leq 3 \sum_{v \in V_B} (d_o(v) - d_i(v))$$

$$= 3(e(V_B, V_G) - e(V_G, V_B))$$

$$\leq 3(e(V_B, V_G) + e(V_G, V_B))$$

Deduce $e(V_B, V_B) \leq e(V_B, V_G) + e(V_G, V_B)$. result follows.

Derandomization:

- Analysis focuses on edges,
- so unsurprisingly, pairwise independence sufficient
- not immediately obvious, but again consider $d$-uniform case
- prob vertex marked $1/2d$
- neighbors $1, \ldots, d$ in increasing degree order
- Let $E_i$ be event that $i$ is marked.
- Let $E'_i$ be $E_i$ but no $E_j$ for $j < i$
- $A_i$ event no neighbor of $i$ chosen
Then prob eliminate \( v \) at least

\[
\sum \Pr[E'_i \cap A_i] = \sum \Pr[E'_i] \Pr[A_i \mid E'_i] \\
\geq \sum \Pr[E'_i] \Pr[A_i]
\]

Wait: show \( \Pr[A_i \mid E'_i] \geq \Pr[A_i] \)

\begin{itemize}
  \item true if independent
  \item measure \( \Pr[\neg A_i \mid E'_i] \leq \sum \Pr[E_w \mid E'_i] \) (sum over neighbors \( w \) of \( i \))
  \item measure

\[
\Pr[E_w \mid E'_i] = \frac{\Pr[E_w \cap E']}{\Pr[E'_i]}
= \frac{\Pr[(E_w \cap \neg E_1 \cap \cdots) \cap E_i]}{\Pr[\neg E_1 \cap \cdots \mid E'_i]}
= \frac{\Pr[E_w \cap \neg E_1 \cap \cdots]}{\Pr[\neg E_1 \cap \cdots \mid E_i]}
\leq 1 - \sum_{j \leq i} \Pr[E_j \mid E_i]
= \Theta(\Pr[E_w])
\]

(last step assumes \( d \)-regular so only \( d \) neighbors with odds \( 1/2d \))
\end{itemize}

But expected marked neighbors \( 1/2 \), so by Markov \( \Pr[A_i] > 1/2 \)

so prob eliminate \( v \) exceeds \( \sum \Pr[E'_i] = \Pr[\cup E_i] \)

lower bound as \( \sum \Pr[E_i] - \sum \Pr[E_i \cap E_j] = 1/2 - d(d - 1)/8d^2 > 1/4 \)

so \( 1/2d \) prob. \( v \) marked but no neighbor marked, so \( v \) chosen

Generate pairwise independent with \( O(\log n) \) bits

try all polynomial seeds in parallel

\begin{itemize}
  \item one works
  \item gives deterministic \( NC \) algorithm
\end{itemize}

with care, \( O(m) \) processors and \( O(\log n) \) time (randomized)

LFMIS \( P \)-complete.