Local Computing

• The LOCAL model
• Deterministic \((\Delta + 1)\)-coloring arbitrary graphs with maximum degree \(\Delta\)
LOCAL Model

• Each process is located at a node of a network modeled as an \( n \)-node graph (\( n = \#\text{processes} \))

• Each process has a unique ID in \( \{1, \ldots, n\} \)

• Computation proceeds in synchronous rounds during which every process:

  1. Sends a message to each neighbor

  2. Receives a message from each neighbor

  3. Performs individual computation (same algorithm for all nodes)
Lemma If a problem $P$ can be solved in $t$ rounds in the LOCAL model by an algorithm $A$, then there is a $t$-round algorithm $B$ solving $P$ in which every node proceeds in two phases: (1) Gather all data in the $t$-ball around it; (2) Simulate and compute the solution.
(Δ+1)-coloring

Δ = maximum node degree of the graph

(Δ+1)-coloring = assign colors to nodes such that every pair of adjacent nodes are assigned different colors.

**Lemma** Every graph is (Δ+1)-colorable

**Theorem** (Brooks, 1941)
Every graph G is Δ-colorable, unless G is a complete graph, or an odd cycle.

**Lemma** (Δ+1)-coloring can be sequentially computed by a simple greedy algorithm treating each node individually.
Let $k \geq \Delta + 1$

If there exists a $t$-round $k$-coloring algorithm then there exists a $(\Delta + 1)$-coloring algorithm running in $t + (k - (\Delta + 1))$ rounds.
Coloring graphs of max degree $\Delta$ with $\Delta^{O(\Delta)}$ colors in $O(\log^* n)$ rounds

Every node $u$ maintains an array $c(u) = (c_1(u), \ldots, c_\Delta(u))$ of colors, ordered according to the IDs of its neighbors.

- Initially $c(u) = (\text{ID}(u), \ldots, \text{ID}(u))$
- Repeat
  - performs C&V with each neighbor independently, in parallel.
Correctness

Claim: Proper coloring is preserved after each iteration of C&V, transforming color \( c(u) \) of \( u \) into \( c'(u) \)

- Let \( c'_i(u) = (p, b) \) and \( c'_i(v) = (p', b') \)
- If \( p \neq p' \) then \( c'(u) \neq c'(v) \)
- If \( p = p' \) then, as \( p \) is the first bit-position at which \( c_i(u) \) and \( c_i(v) \) differ, we have \( b \neq b' \), and thus \( c'(u) \neq c'(v) \)
Complexity

- Colors are initially on $\Delta \cdot \lfloor \log_2 n \rfloor$ bits
- Assuming colors on $k$ bits
- After one iteration: colors on $f(k) = \Delta(\lfloor \log_2 k \rfloor + 1)$ bits
- For $k = \alpha \Delta \log \Delta$ with $\alpha$ sufficiently large, we have $f(k) < k$
- Thus, after $O(\log^* n)$ iterations, colors on $O(\Delta \log \Delta)$ bits
- That is, $2^{O(\Delta \log \Delta)} = \Delta^{O(\Delta)}$ colors.
3-Coloring Rooted Trees

- Apply C&V with parent for $O(\log^* n)$ rounds, to 6-color the tree

- For $i = 6$ down to 4 do
  - adopt color of parent
  - recolor nodes colored $i$ with a color in $\{1, 2, 3\}$
1-Factors

• Let $G = (V, E)$ be a graph

• Assume each node $v \in V$ selects one of its incident edges

• Let $F \subseteq E$ be the set of selected edges

**Claim** $F$ is a collection of « pseudo-trees » of the form
Pseudo-Forest Decomposition
A Connected Component

Not a tree, but almost… Hence the names *pseudo-tree* and *pseudo forest*

Remark: *For port 1, one gets « real » trees*
Coloring with $3^\Delta$ colors in $O(\log^* n)$ rounds

- Every node $u$ orders its incident links from 1 to $\deg(u)$ according to the IDs of its neighbors.
- This results in $\Delta$ pseudo-forests $F_1, \ldots, F_\Delta$.
- Color each pseudo tree in each pseudo forest in parallel, in $O(\log^* n)$ rounds.
- Each node gets a color $c(u) = (c_1(u), \ldots, c_\Delta(u))$ where $c_i(u) \in \{1,2,3\}$, hence $3^\Delta$ colors.
\[(\Delta + 1)-\text{Coloring in } O(\Delta^2 + \log^* n) \text{ rounds}\]

- Nodes first compute a \(3^\Delta\)-coloring in \(O(\log^* n)\) rounds

- For each \(u, c(u) = (c_1(u), \ldots, c_\Delta(u))\) with \(c_i(u) \in \{1,2,3\}\)

- Iteratively compute a \((\Delta + 1)\)-coloring \(c'_i\) of \(\bigcup_{j=1}^{i} F_j\) for \(i = 1,\ldots, \Delta\)
  - \(c'_1 = c_1\) is a 3-coloring of \(F_1\)
  - Given \(c'_i\), let us view \((c'_i, c_{i+1})\) as a \(3(\Delta + 1)\)-coloring of \(\bigcup_{j=1}^{i+1} F_j\)
  - The coloring \((c'_i, c_{i+1})\) can be transformed into a \((\Delta + 1)\)
    -coloring \(c'_{i+1}\) of \(\bigcup_{j=1}^{i+1} F_j\) in \(2(\Delta + 1)\) rounds

- The coloring \(c'_\Delta\) is a \((\Delta + 1)\)-coloring of \(\bigcup_{j=1}^{\Delta} F_j = G\), obtained in
\[(\Delta - 1)(2(\Delta + 1)) = O(\Delta^2)\] rounds.
(Δ + 1)-Coloring in $O(Δ + \log^* n)$ rounds

Four phases:

1. $3^Δ$-coloring in $O(\log^* n)$ rounds (cf. previous slides)

2. Reducing to $O(Δ^3)$-coloring in 1 round

3. Reducing number of colors to $O(Δ^2)$ in 1 round

4. Reducing number of colors to $Δ + 1$ in $O(Δ)$ rounds
Phase 2: From $3^\Delta$ to $O(\Delta^3)$ colors in a single round

Lemma [Erdös, Frankl, Füredi, 1985]
For any $k > \Delta \geq 2$, there exists a family $\mathcal{F}$ of $k$ subsets of $\{1, \ldots, 5\lceil \Delta^2 \log k \rceil \}$ such that, for any $\Delta + 1$ sets $F_0, \ldots, F_\Delta$ in $\mathcal{F}$, we have $F_0 \not\subseteq \cup_{i=1}^\Delta F_i$

Algorithm:
- Range of colors $[1, k]$ with $k = 3^\Delta$
- Node $u$ with color $c(u) \in \{1, \ldots, k\}$ picks set $F_{c(u)} \in \mathcal{F}$
- By the lemma, $\exists x \not\in \cup_{v \in N(u)} F_{c(v)}$
- Node $u$ updates its color $c(u)$ to $x$ i.e. $c(u) \leftarrow x$
- Reduction of #colors: $3^\Delta \rightarrow O(\Delta^2 \log(3^\Delta)) = O(\Delta^3)$
Polynomials on Finite Fields

- For a prime integer $q$, let $\mathbb{F}_q = \mathbb{Z}/q\mathbb{Z}$ i.e., $\mathbb{F}_q$ is the set $\{0,\ldots, q - 1\}$ with arithmetic modulo $q$
- $\mathbb{F}_q$ is a finite field
- A polynomial of degree $d$ on $\mathbb{F}_q$ is of the form
  \[ a_0 + a_1X + \ldots + a_dX^d \]

**Lemma** A polynomial of degree $d$ on $\mathbb{F}_q$ has at most $d$ roots

**Corollary** Two polynomials of degree $d$ on $\mathbb{F}_q$ may coincide on at most $d$ values.
Phase 3: From \( O(\Delta^3) \) to \( O(\Delta^2) \)
colors in a single round

- Say colors in \([1, \alpha \Delta^3]\) for some \( \alpha > 0 \)
- Let \( q = O(\Delta) \) prime with \( 3\Delta < q \) and \( q^4 \geq \alpha \Delta^3 \)
- There are \( q^4 \) polynomials of degree 3 in \( \mathbb{F}_q \)
- Node \( u \) with color \( c(u) = i \in [1, \alpha \Delta^3] \) picks set
  \[
  S_{c(u)} = S_i = \{(x, p_i(x)) : x \in \mathbb{F}_q\} \subseteq \mathbb{F}_q \times \mathbb{F}_q
  \]
- For every \( i \neq j \) we have \( |S_i \cap S_j| \leq 3 \)
- Thus \( |S_{c(u)} \cup \bigcup_{v \in N(u)} S_{c(v)}| \geq |S_{c(u)}| - 3\Delta > 0 \)
- Node \( u \) updates its colors by picking one element in
  \( S_{c(u)} \cup \bigcup_{v \in N(u)} S_{c(v)} \)
Phase 4: From $O(\Delta^2)$ to $\Delta + 1$ colors in $O(\Delta)$ rounds

- Say colors in $[1, \beta \Delta^2]$ for some $\beta > 0$
- Let $q = O(\Delta)$ prime with $6\Delta < q$ and $q^4 \geq \beta \Delta^2$
- Node $u$ with color $c(u) = i \in [1, \beta \Delta^2]$ picks sequence
  \[
  \sigma_{c(u)} = \sigma_i = (p_i(0), p_i(1), \ldots, p_i(q - 1))
  \]
  For $x = 0$ to $q - 1$ do
  - if uncolored then propose color $p_i(x)$
  - if no conflicts, then adopt color $p_i(x)$ and terminate
- At most 3 conflicting iterations for each non-terminated neighbor and at most 3 conflicting iterations for each terminated neighbor
- Reduce #colors from $q$ to $\Delta + 1$ in $q - (\Delta + 1) = O(\Delta)$ rounds
State of the Art and Open Problems

Best known algorithm performs $(\Delta + 1)$-coloring in

- $O(\log^* n + \sqrt{\Delta \log \Delta})$ rounds
- $O(\log n \cdot \log^2 \Delta) \leq O(\log^3 n)$ rounds

Can we improve this complexity?

Is there a distributed algorithm running in $O(\log^* n)$ rounds in LOCAL that properly colors every graph of maximum degree $\Delta$ with $o(\Delta^2)$ colors?
End Lecture 2