236358
Distributed Graph Algorithms

Spring 2017
Class by Keren Censor-Hillel
Local/Global Problems

• We need $\Omega(D)$ rounds for BFS, MST

• Do we need $\Omega(D)$ rounds for all interesting problems?

• No:
  – BFS, MST are global problems
  – some problems are local problems
c-Coloring

• A function \( \varphi : V \rightarrow \{1, \ldots, c\} \) is a \textit{c-coloring} if for every \( u, v \in V \) such that \( \{u, v\} \in E \) it holds that \( \varphi(u) \neq \varphi(v) \)

• If \( G \) has a \textit{c-coloring} then \( G \) is \textit{c-colorable}
Chromatic Number

• The **chromatic number** $\chi(G)$ of $G$ is the smallest $c$ for which $G$ is $c$-colorable
  – Finding $\chi(G)$ or a $\chi(G)$-coloring is NP-hard

• Every graph has a $(\Delta+1)$-coloring
  – $\Delta$ is the maximal degree in the graph

• **Proof:** The greedy sequential algorithm
Distributed Coloring

• LOCAL model

• Each node $v$ outputs a color $\varphi(v)$ such that $\varphi$ is a $c$-coloring

• Greedy can be simulated in $n$ rounds in LOCAL
Color Reduction

- Given a $c$-coloring $\varphi$, obtain a $(\Delta+1)$-coloring

1. for $i=c,\ldots,\Delta+2$ do
2. if $\varphi(v)=i$ then
3. $\varphi(v) \leftarrow \min\{x \mid \varphi(u) \neq x \text{ for all } u \in N(v)\}$
4. send $\varphi(v)$ to all neighbors
5. return $\varphi(v)$
Color Reduction

• **Correctness:**
  - For each $v$, $\varphi(v) \leq \Delta + 1$ because at most $\Delta$ colors are used by neighbors.
  
  - The coloring is valid, by induction on the round number (starts valid and remains valid at the end of the round).

• **Round complexity:** $c-\Delta-1$
  - This is $O(n-\Delta)$ if we start with IDs as colors
Recursive $(\Delta+1)$-coloring

**Recurse**($x$):
1. if $|x| = \log n$ then
2. \( \varphi(v) \leftarrow 1 \)
3. return $\varphi(v)$
4. $b \leftarrow ID_v[\log n - |x|]$
5. $\varphi(v) \leftarrow \text{Recurse}(bx)$
6. if $b=1$
7. for $i=1,\ldots,\Delta+1$
8. if $\varphi(v)=i$
9. \( \varphi(v) \leftarrow \min\{x \mid \varphi(u) \neq x \text{ for all } u \in N(v)\} \)
10. send $\varphi(v)$ to all neighbors
11. return $\varphi(v)$
Recursive \((\Delta+1)\)-coloring

- **Illustration**

- **Correctness**: Gives a \((\Delta+1)\)-coloring when executed with \(x=\varepsilon\). Let \(U_x = \{v \mid \text{Id}_v = bx\}\). By induction, at the end of each recursion level, \(U_x\) has a \((\Delta+1)\)-coloring.

- **Base case**: When \(|x| = \log n\), each node \(v\) has \(\varphi(v) = 1\), and \(U_x = \{v\}\).

- **Induction hypothesis**: Assume the claim holds for \(|x| = i\), that is, for all \(0x'\) and \(1x'\) such that \(|x'| = i-1\).
Recursive $(\Delta+1)$-coloring

- **Induction step**: By the induction hypothesis, all nodes in $U_{0x'}$ have a valid coloring. The nodes in $U_{1x'}$ adjust their colors accordingly, and since this is done according to their colors, no two neighbors in $U_{1x'}$ adjust at the same round. This gives a valid $(\Delta+1)$-coloring for $U_{x'}$. 
Recursive $(\Delta+1)$-coloring

- **Correctness**: Gives a $(\Delta+1)$-coloring when executed with $x=\varepsilon$. Let $U_x = \{v \mid Id_v = bx\}$. By induction, at the end of each recursion level, $U_x$ has a $(\Delta+1)$-coloring.

- **Round Complexity**: $O(\Delta \log n)$ rounds, since there are $\log n$ iterations of $O(\Delta)$ rounds each.
Distributed \((\Delta+1)\)-coloring

- Distributed algorithms for \((\Delta+1)\)-coloring:
  - Greedy: \(O(n)\) rounds
  - Color reduction: \(O(n-\Delta)\) rounds
  - Recursive: \(O(\Delta \log n)\) rounds

- Best known complexities are:
  - Deterministic: \(\tilde{O}(\sqrt{\Delta \log^*n})\) rounds
  - Randomized: \(O(\sqrt{\log \Delta}) \cdot 2^{O(\sqrt{\log \log n})}\) rounds

- Lower bound: \(\Omega(\log^*n)\)
A ring is 3-colorable
  – A ring of even length is 2-colorable
  – 2-coloring a ring of even length requires $\Omega(n)$ rounds

What about 3-coloring a ring?
  – We saw $(\Delta+1)$-coloring algorithms for general graphs
  – Can do much better: $O(\log^* n)$ rounds
Coloring a ring

• We find a 6-coloring algorithm for the ring in $O(\log^* n)$ rounds

• Then use the color reduction algorithm from $c=6$ colors to $(\Delta+1)=3$ colors
  – An additional $O(1)$ rounds

• We assume the ring is oriented
  – Assumption can be removed
6-Coloring a ring

1. $\varphi(v) \leftarrow \text{ID}_v$
2. For $O(\log^* n)$ rounds
3. $i \leftarrow \min\{ k \mid \varphi(v)[k] \neq \varphi(\text{parent}(v))[k] \}$
4. $\varphi(v) \leftarrow <i, \varphi(v)[i]>$
5. send $\varphi(v)$ to all neighbors
6. return $\varphi(v)$
6-Coloring a ring

- **Illustration**

- **Claim 1**: At the end of each iteration, $\varphi$ is a valid coloring

- **Claim 2**: After $O(\log^* n)$ rounds, $\varphi$ is a 6-coloring
6-Coloring a ring

• **Claim 1**: At the end of each iteration, $\varphi$ is a valid coloring

• **Proof**: Let $v, w$ be neighbors, $w = \text{parent}(v)$. By induction, $\varphi(v) \neq \varphi(w)$ at the end of each iteration.

• **Base case**: IDs are unique.
6-Coloring a ring

• **Claim 1:** At the end of each iteration, $\varphi$ is a valid coloring

• **Induction step:** Let $i_v, i_w$ be the indices chosen in Line 3. If $i_v \neq i_w$ then $\varphi(v) \neq \varphi(w)$ after Line 4.

• Otherwise, let $i = i_v = i_w$. Since at the beginning of the iteration $\varphi$ is valid, $\varphi(v)[i] \neq \varphi(w)[i]$, and hence $\varphi(v) \neq \varphi(w)$ after Line 4.
6-Coloring a ring

• **Claim 2**: After $O(\log^* n)$ rounds, $\varphi$ is a 6-coloring

• **Proof**: Each iteration that starts with a coloring of $b$ bits (a $2^b$-coloring) ends with a coloring of $\log b + 1$ bits (index and its value). After $O(\log^* n)$ iterations we get a coloring with 3 bits (an 8-coloring).

• But actually this is a 6-coloring: the first two bits represent an index, so 11 cannot occur.
3-coloring a ring

• Can be extended to $(\Delta+1)$-coloring:
  – In each iteration the new color is a sequence of $\Delta$ pairs $<\text{index}, \text{value}>$
  – A $b$-bit coloring becomes a $\Delta(\log b + 1)$-bit coloring
  – $O(\Delta)$-bit coloring after $O(\log^* n)$ rounds
  – Color reduction to $\Delta+1$ colors in a number of rounds that is exponential in $\Delta$
    • Constant number of rounds if $\Delta$ is constant

• Can be extended to rooted oriented trees
Lower bound for \((\Delta+1)\)-coloring

- **Theorem:** Any algorithm \(A\) for 3-coloring a ring requires \(\Omega(\log^* n)\) rounds

- **Definition:** A \((c,k)\)-coloring function is a function
  \[\varphi : X_k \rightarrow [c]\]
  where \(X_k\) is the set of all vectors of \(k\) different IDs, such that
  \[\varphi(x_0,\ldots,x_{k-1}) \neq \varphi(x_1,\ldots,x_k)\]
  for every \(x_0 < x_1 < \ldots < x_k\)
Lower bound for \((\Delta+1)\)-coloring

• **Claim 1**: If \(A\) finishes in \(t\) rounds then \(A\) is a \((3,2t+1)\)-coloring function

• **Proof**: The output of a node \(v\) is determined by its \(t\)-neighborhood (lemma from first class). Hence, we can view \(A\) as a function

\[
A : X_{2t+1} \rightarrow \{0,1,2\}
\]

such that

\[
A(x_0,\ldots,x_{2t}) \neq A(x_1,\ldots,x_{2t+1})
\]

for every \(x_0,\ldots,x_{2t+1}\) that are all different.

• In particular, this holds for every \(x_0 < x_1 < \ldots < x_{2t+1}\)
Lower bound for \((\Delta+1)\)-coloring

• **Claim 2:** If there is a \((c,1)\)-coloring function \(f\) then \(c \geq n\).

• **Proof:** By definition, \(f(x_0) \neq f(x_1)\) for every \(0 \leq x_0 < x_1 < n\), so \(f\) is 1:1, and hence \(c \geq n\).
Lower bound for \((\Delta+1)\)-coloring

• In other words: without communication, we must use \(n\) colors.

• **Main goal**: show that reducing the number of colors requires communication
Lower bound for \((\Delta+1)\)-coloring

• **Claim 3**: If there is a \((c,k)\)-coloring function \(f\) then there is a \((2^c, k-1)\)-coloring function \(g\).

• **Proof of Theorem**: By **Claim 1**, \(A\) is a \((3,2t+1)\)-coloring function. Applying **Claim 3** for \(2t\) times gives that there is a \((T,1)\)-coloring function, where \(T=2^2\ldots^2^3\) (\(t\) times). By **Claim 2** we have \(T\geq n\), and hence \(t\geq \Omega(\log^*n)\).
Lower bound for $(\Delta+1)$-coloring

• **Claim 3**: If there is a $(c,k)$-coloring function $f$ then there is a $(2^c, k-1)$-coloring function $g$.

• **Proof**: Consider a bijection $h:2^c \rightarrow [2^c]$, mapping subsets of $[c]$ into $[2^c]$ (think characteristic vector).

• Define $g: X_{k-1} \rightarrow [2^c]$ by
\[
g(x_0,\ldots,x_{k-2})=h( \{ f(x_0,\ldots,x_{k-1}) \mid x_{k-2} < x_{k-1} < n\} )
\]
Lower bound for $(\Delta+1)$-coloring

- Define $g: X_{k-1} \to [2^c]$ by
  $$g(x_0,\ldots,x_{k-2})=h( \{ f(x_0,\ldots,x_{k-1}) \mid x_{k-2}<x_{k-1}<n \} )$$

- That is, a vector of length $k-1$ is assigned by $g$ a color that is given by $h$ to the set of colors assigned by $f$ to all possible extensions of length $k-2$ of the vector
Lower bound for (Δ+1)-coloring

• We need to show that $g(x_0,\ldots,x_{k-2}) \neq g(x_1,\ldots,x_{k-1})$ for all $x_0 < x_1 \ldots < x_{k-2} < x_{k-1} < n$

• We have $f(x_0,\ldots,x_{k-2},x_{k-1}) \neq f(x_1,\ldots,x_k)$ for all $x_0 < x_1 \ldots < x_{k-1} < x_k < n$ because $f$ is a $(c,k)$-coloring

• So $f(x_0,\ldots,x_{k-2},x_{k-1})$ which belongs to the set that defines $g(x_0,\ldots,x_{k-2})$, does not belong to the set that defines $g(x_1,\ldots,x_{k-1})$, giving $g(x_0,\ldots,x_{k-2}) \neq g(x_1,\ldots,x_{k-1})$. 
Lower bound for \((\Delta+1)\)-coloring

- **Theorem**: Any algorithm \(A\) for 3-coloring a ring requires \(\Omega(\log^* n)\) rounds

- Remarkably, we do not have stronger lower bounds for \((\Delta+1)\)-coloring

- Next: additional local problems