236610
Distributed Graph Algorithms

Spring 2018
Keren Censor-Hillel
Lower Bound for Set Disjointness

• **Theorem**: The communication complexity of *set disjointness* is $k+1$

• **Proof**:
  • Any protocol $\pi$ that solves set disjointness has $CC(\pi)=k+1$:
  • Will be proven by **Claims 1** and **2** next.
Rectangles

• Given two subsets $X, Y$ of $2^k$, the set $X \times Y$ is called a rectangle

• Illustration

• Given a function $f : 2^k \times 2^k \rightarrow \{0, 1\}$, a rectangle $X \times Y$ is f-monochromatic if $f(x, y) = f(x', y')$ for all $x, x' \in X$ and $y, y' \in Y$. 
Monochromatic Rectangles

• **Claim 1**: A protocol $\pi$ with $CC(\pi) = b$ divides the input domain $2^k \times 2^k$ to at most $2^b$ disjoint monochromatic rectangles.

• **Proof**: by induction on the depth $i$ in the protocol tree, with the property of corresponding to disjoint rectangles.

• **Base case $i=0$**: the root corresponds to the set of all inputs.
Monochromatic Rectangles

• **Induction hypothesis**: after \( i-1 \) bits of the protocol \( \pi \), the input domain \( 2^k \times 2^k \) is divided to **disjoint rectangles**
Monochromatic Rectangles

• **Induction step**: Consider a string of depth $i$, and let $XxY$ be the corresponding rectangle.

• Assume that Bob sends the next bit.
Monochromatic Rectangles

• Assume that Bob sends the next bit.
  – $Y'$ in $Y$: the inputs for which the next bit is 0
  – $Y''$ in $Y$: the inputs for which the next bit is 1

• $XxY'$ and $XxY''$ are two disjoint rectangles corresponding to $s$

• The leaves are disjoint rectangles, and they must be monochromatic. For $CC(\pi)=b$, there are at most $2^b$ leaves.
Rectangles in Set Disjointness

• **Claim 2**: A protocol $\pi$ that solves Set Disjointness divides the input domain $2^k \times 2^k$ to at least $2^k + 1$ disjoint monochromatic rectangles

• **Proof**: Consider the $2^k$ pairs of the form $(u, \bar{u})$. The output for all of them is 0.
Rectangles in Set Disjointness

• **Claim 2**: A protocol $\pi$ that solves Set Disjointness divides the input domain $2^k \times 2^k$ to at least $2^k + 1$ disjoint monochromatic rectangles

• **Proof**: No two such pairs $(u, \bar{u})$ and $(u', \bar{u}')$ can be in the same monochromatic rectangle because then the output for $(u, \bar{u}')$ and $(u', \bar{u})$ is also 0, but it has to be 1 for one of them.
Rectangles in Set Disjointness

• **Claim 2**: A protocol \( \pi \) that solves Set Disjointness divides the input domain \( 2^k \times 2^k \) to at least \( 2^k + 1 \) disjoint monochromatic rectangles.

• **Proof**: This gives \( 2^k \) disjoint monochromatic rectangles.

• There is at least one more additional rectangle for outputs 1.
Lower Bound for Set Disjointness

• **Theorem**: The communication complexity of *set disjointness* is $k+1$

Any protocol $\pi$ that solves set disjointness has $\text{CC}(\pi)=k+1$:

• **Proof**:  
  • By **Claims 1** and **2**, $\pi$ needs at least $\log(2^k+1)$ bits, which is at least $k+1$ bits.
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 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 c-coloring then \( G \) is 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. \(b \leftarrow \text{ID}_v[\log n - |x|]\)
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 \(U_x = \{v | \text{Id}_v = yx\}\). 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 $U_x = \{v | \text{Id}_v = yx\}$. 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: $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log \log n})})$ rounds

- Lower bound: $\Omega(\log^* n)$