Kishore Kothapalli
Department of Computer Science
Johns Hopkins University
Baltimore, MD USA
Vertex Coloring

Proper coloring

Not a Proper coloring
Coloring a Cycle Graph

- Cycle graph on $n$ nodes
- Distributed algorithm: Every round, every uncolored node chooses a color among \{R,G,B\} independently and uniformly at random. [Luby]
Coloring a Cycle Graph

- Cycle graph on $n$ nodes
- Distributed algorithm: Every round, every uncolored node chooses a color among \{R,G,B\} independently and uniformly at random.
Coloring a Cycle Graph

- Cycle graph on $n$ nodes
- Algorithm: Every round, every uncolored node chooses a color among \{R,G,B\} independently and uniformly at random.
- Known: Every node gets colored in $O(\log n)$ rounds.
Coloring an Oriented Cycle Graph

- Cycle graph on $n$ nodes, with edges oriented.
- Edge $\{u,v\}$ oriented $u \rightarrow v$, then $u$ gets preference over $v$. 
Coloring an Oriented Cycle Graph

- Cycle graph on $n$ nodes, with edges oriented.
- Algorithm: Nodes choose color ind. and u.a.r.
Coloring an Oriented Cycle Graph

- Cycle graph on $n$ nodes, with edges oriented.
- Algorithm: Nodes choose color ind. and u.a.r.
Coloring an Oriented Cycle Graph

- Cycle graph on $n$ nodes, with edges oriented.
- Algorithm: Nodes choose color ind. and u.a.r.
- Use orientation to break symmetry.
- Claim: Can color in $O(\sqrt{\log n})$ rounds, w.h.p. !!
  - Proof coming up...
Coloring an Oriented Cycle Graph

- Consider situation after $r = 4 \sqrt{\log n}$ rounds.
- Claim: Any arc $P$ of length $l = \sqrt{\log n}$ has at least one colored node.
  - $\Pr[\text{a given path } P \text{ has no colored node after } r \text{ rounds}] 
    \leq (1/2)^{lr} \leq 1/n^4$.
  - Number of arcs of length $l \leq 2n$. 
Coloring an **Oriented Cycle Graph**

- **Claim:** Can finish in a further $\sqrt{\log n}$ rounds.
  - Key: Orientation !!
- **Number of rounds** $= O(\sqrt{\log n})$, with high probability.
Questions

1) Can faster algorithms be designed for arbitrary oriented graphs?
2) How many bits have to be exchanged?
Outline

- Introduction
  - Model
  - Related Work
  - Lower bounds
  - Upper bound for constant degree graphs
  - Upper bound for general graphs
  - Experimental results
  - Conclusion
Introduction

- Distributed system
  - Computation done by exchange of messages.
  - Graph $G = (V,E)$ representing the topology.
- Vertex coloring is a fundamental problem.
- Applications to
  - Scheduling
  - Routing
  - Clustering
Introduction

• Distributed vertex coloring

• Number of colors
  - Minimum required is called *chromatic number*, $\chi(G)$.
  - NP-hard to compute $\chi(G)$. [GJ, 1979]
  - Hard to approximate $\chi(G)$ to any reasonable value.

• Try to color with $O(\Delta)$ or $\Delta+1$ colors.
Introduction

• Deterministic algorithms
  - No poly-logarithmic algorithms known.
  - Best known runs in $O(n^{1/\sqrt{\log n}})$ rounds [PS 1996].
  - Cannot be solved deterministically without unique node identification numbers [BFFS, 2003].
Introduction

- Randomized algorithm
  - Decreases the number of rounds required exponentially.
  - Known since more than a decade [Luby 1985].
  - $O(\log n)$ rounds to obtain a $(\Delta+1)$-coloring.
Outline

- Introduction
  - Model
- Related Work
- Lower bounds
- Upper bound for constant degree graphs
- Upper bound for general graphs
- Experimental results
- Conclusion
Model

- Graph $G = (V, E)$ modeling the topology of the distributed system.
- Degree $= \Delta$, $|V| = n$
- Edges have orientation.
  - Bits can flow in both directions.
• *l*-acyclic orientation: Minimum length of directed cycled induced by the orientation $\geq l$. 

$l = 4$
Model

- Nodes do not need unique labels, know only $n$ and $\Delta$
- Synchronized rounds
- Local computation is not counted.
Model

- Every bit round, every node can send (receive) at most 1 bit to (resp. from) its neighbors.
  - Bit complexity is the maximum number of bit rounds required.
- One round of the algorithm contains several bit rounds.
Model

• How easy is it to provide orientation?
• Three scenarios:
  1) Dynamic networks.
  2) Know a reference, such as distance to destination.
  3) Nodes are labeled distinctly.
Outline

• Introduction
• Model

→ Related Work
• Lower bounds
• Upper bound for constant degree graphs
• Upper bound for general graphs
• Experimental results
• Conclusion
Related Work - Coloring

• Luby [Luby, 1985] gave an $O(\log n)$ round algorithm to $\Delta+1$ color a graph.

• Special cases:
  - Cycle: $O(\log^* n)$ rounds [CV 1986, GPS 1987], shown to be optimal [Linial, 1992]
  - Rooted Trees: 6-coloring in $O(\log^* n)$ rounds [GPS, 1987]

• Unlimited local computation
  - $\Delta$ coloring in $O(\log^*(n/\Delta))$ rounds, [De Marco and Pelc, 2001]

• Many more...
Related Work – Oriented Graphs

- Sense of direction
  - Equivalent notion to that of orientation on edges.
- Leader election with $O(n)$ messages [Singh 87].
- Spanning tree, depth first traversal [Flocchini, Mans, Santoro 97].
Outline

- Introduction
- Model
- Related Work
  - Lower bounds
  - Upper bound for constant degree graphs
  - Upper bound for general graphs
  - Experimental results
- Conclusion
Lower bound -1

- (Unoriented) cycle graph on $n$ nodes
- *Any* finite number of colors, *any* Las Vegas algorithm.
- Need $\Omega(\log n)$ bit rounds, with high probability, to arrive at a proper coloring.
• Cycle graph on \( n \) nodes, \textit{oriented} in the same direction.

• \textit{Any} finite number of colors, \textit{any} Las Vegas algorithm.

• Need \( \Omega(\sqrt{\log n}) \) bit rounds, with high probability, to arrive at a proper coloring.
Outline

- Introduction
- Model
- Related Work
- Lower bounds
  - Upper bound for constant degree graphs
- Upper bound for general graphs
- Experimental results
- Conclusion
Upper bound for constant degree oriented graphs

- \( G = (V,E), \Delta = O(1) \)
- \( G \) is provided with a \( \sqrt{\log n} \)-acyclic orientation.
- Algorithm:
  - Uncolored nodes choose color independently and u.a.r. among \( 2\Delta \) colors
  - Conflicts resolved using orientation.
- Two phase analysis
Upper Bound – constant degree graphs

- Phase I
- Claim: After $r = 4\sqrt{\log n}$ rounds, every oriented path $P$ of length $l = \sqrt{\log n}$ has at least one colored node, w.h.p.
- Can be shown using union bound.
  - Number of paths of length $l \leq n\Delta^l$
Upper bound – constant degree graphs

- Situation at the end of Phase I
  - Connected components of uncolored nodes
  - Diameter of any such component < $\sqrt{\log n}$
- Phase II: Same algorithm
Upper bound – constant degree graphs

- Consider the following layering process
  - Layer 0 = \{u: no uncolored node \( u \) with \( u \rightarrow v \}\}
  - Remove nodes in layer 0.
  - Continue until no nodes are left.
Upper bound – constant degree graphs

• Claim: The layering process terminates in less than $\sqrt{\log n}$ rounds

• Phase II requires less than $\sqrt{\log n}$ rounds.
  - Each round, at least one node gets colored.
Upper bound – constant degree graphs

• Claim: $\text{label } (v) < \sqrt{\log n}$ for any node.
• Phase II requires less than $\sqrt{\log n}$ rounds.
  - Each round, at least one node gets colored.
Upper bound – constant degree graphs

- Claim: label \((v) < \sqrt{\log n}\) for any node.
- Phase II requires less than \(\sqrt{\log n}\) rounds.
  - Each round, at least one node gets colored.
- Nodes do not have to explicitly compute the labels.
Constant degree oriented graphs

• Summary
  - Lower bound of $\Omega(\sqrt{\log n})$ on bit complexity.
  - Our algorithm achieves an upper bound of $O(\sqrt{\log n})$.
  - Nodes need to know only their local degree.
  - With known techniques, can reduce number of colors to $\Delta+1$.
  - Simple algorithm.
  - Can arrive at local coloring: node $u$ with degree $d_u$ gets a color between 1 and $d_u+1$. 
Outline

- Introduction
- Model
- Related Work
- Lower bounds
- Upper bound for constant degree graphs
- Upper bound for general graphs
- Experimental results
- Conclusion
Upper bound - Arbitrary graphs

• $G = (V,E)$, degree $= \Delta$.
• $G$ is provided with a $\sqrt{\log n}$-acyclic orientation
• Problems:
  1) High degree graphs: Too many paths
  2) Low degree also poses a problem
• Have to extend our algorithm and analysis.
• Goal: Arrive at a $(1+\varepsilon)\Delta$-coloring with $O(\sqrt{\log n})$ bits for any constant $\varepsilon > 0$. 
Arbitrary Graphs

• 3-phase algorithm
  - Phase I: Reduce degree to $c \cdot \log n$
  - Phase II: Break into connected components of uncolored nodes with diameter less than $\sqrt{\log n}$
  - Phase III: Finish using orientation.

• Bit rounds:
  - Phase I: $O(\log \Delta (\log \log n))$
  - Phase II: $\tilde{O}(\sqrt{\log n})$
  - Phase III: $O(\sqrt{\log n})$
Arbitrary Graphs

• Phase I
• Algorithm:
  - Uncolored nodes choose a color between 1 and $c'\Delta$.
  - Conflicts resolved using orientation.
• Using ideas from:
  - A particular balls-and-bins game.
Arbitrary Graphs

• Phase II
• Reduced palette.
• Algorithm:
  - Uncolored nodes choose a color between 1 and $\min\{2\Delta, 2c \log n\}$
  - Conflicts resolved using orientation.
• Goal: Reduce degree of any node to $\sqrt{\log n}$
• Key Ideas:
  - Orientation
  - Union bound
Arbitrary Graphs

- End of Phase II
  - Graph broken into connected components of uncolored nodes.
  - Diameter of any connected component less than $\sqrt{\log n}$
Arbitrary Graphs

• Phase III: Key idea
  - Use a layering process as in Phase II of constant degree oriented graphs.
  - Phase III takes less than $\sqrt{\log n}$ rounds.
Few Improvements

- Can reduce complexity of Phase I to $O(\log \Delta)$.
- Can reduce complexity of Phase II and Phase III to $O(\sqrt{\log n \log \log n})$. 
Arbitrary oriented graphs

- Summary:
  - Need $\sqrt{\log n}$-acyclicity of the orientation
  - Bit complexity = $O(\log \Delta) + \tilde{O}(\sqrt{\log n})$
  - For graphs with $\Delta > 2\sqrt{\log n}$, best possible in general.
  - If every node has degree $> \log n$, then can also arrive at a local coloring.
  - With known techniques, can reduce the number of colors to $(1+\varepsilon)\Delta$ for any constant $\varepsilon > 0$. 
Outline

- Introduction
- Model
- Related Work
- Lower bounds
- Upper bound for constant degree graphs
- Upper bound for general graphs
  → Experimental Results
- Conclusions
Experimental results

- Methodology
  - ANSI C implementation
  - SHA-1 hash function used as random number generator
  - Multiple runs and average value.

- Input: Oriented cycle graph
Experimental results

- Cycle output picture

![Graph showing cycle output picture](image)
Conclusions

• Orientation helps in symmetry breaking.
• Tight results.
• Further work:
  - any "good" orientations?
  - any graph parameter?
  - In preparation: O(1)-coloring trees and planar graphs.
Acknowledgements

- Joint work with:
  - Christian Scheideler, Johns Hopkins University
    (presently at Technische Universität München, Germany).
  - Melih Onus, Arizona State University
  - Christian Schindelhauer, University of Paderborn, Germany.
- Preliminary results are to appear as "Distributed Coloring in $\tilde{O}(\sqrt{\log n})$ bit rounds", in Proc. of the IEEE Intl. Parallel and Distributed Processing Symposium (IPDPS), 2006.
Thank You.