Distributed Coloring in $\tilde{O}(\sqrt{\log n})$ Bit Rounds

Kishore Kothapalli
John Hopkins University

Christian Scheideler
Technische Universität München

Melih Onus
Arizona State University

Christian Schindelhauer
Heinz Nixdorf Institute and
University of Paderborn

Presented by Ohn Tsarfati
Distributed Algorithms
May 2010

Agenda

- **Problem Definition**
- Applications
- Background and Examples
- Model and Definitions
- Approach of Algorithm
- Lower Bound
- Algorithm
- Analysis – Upper Bounds
- Summary and Conclusions

Problem Definition (1)

- Given a graph G=(V, E)
- Assign “colors” to vertices such that no two adjacent vertices share the same color
- Integers represent colors
- In our case – do it **distributively**

Problem Definition (2)

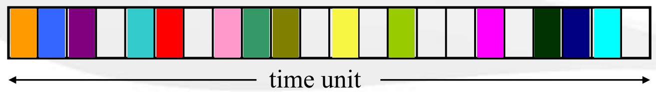
- Challenges:
  - Use as fewer colors as possible (Find the *Chromatic Number*)
  - Minimize runtime complexity
  - Minimize bit complexity (communication rounds)
  - In **distributed algorithms**:
    Color the graph using Δ+1 colors, and minimize the bit complexity

Agenda

- Problem Definition
- **Applications**
- Background and Examples
- Model and Definitions
- Approach of Algorithm
- Lower Bound
- Algorithm
- Analysis – Upper Bounds
- Summary and Conclusions

Applications (1)

- Resource allocation in distributed systems
- Scheduling in wireless networks
  - TDMA example



- Solving Sudoku...
  (81 nodes ; 9 colors ; Δ=8+6+6=20)

Agenda

- Problem Definition
- Applications
- **Background and Examples**
- Model and Definitions
- Approach of Algorithm
- Lower Bound
- Algorithm
- Analysis – Upper Bounds
- Summary and Conclusions

Background and Examples (1)

- How many colors would you use to color a graph with the topology of:
  - a clique?
  - a ring?
  - a star?
- See why it is hard for a distributed algorithm to minimize the colors' number?

Background and Examples (2)

- Cole and Vishkin algorithms:
  - Reach 3-coloring of n-cycle in $O(\log^* n)$ *communication rounds*
- Linial proved coloring an n-cycle is bounded by $\Omega(\log^* n)$ *communication rounds*

Background and Examples (3)

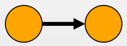
- Fastest known algorithm, by Barenboim and Elkin, finds Δ+1 colors in $O(\Delta)$ + $O(\log^* n)$ rounds for non-cycle graphs
  - Optimality proven by Linial.
- But what if the maximum degree, Δ, is unbounded?
- What if Δ is a function of n?
  - Star topology, for example

Agenda

Model and Definitions (1)

- Definitions:
  - G ≡ (V, E)
  - n ≡ |V|
  - d_u ≡ degree of node u
  - Δ ≡ maximal degree in G
  - ***l-acyclic Orientation*** – Any directed cycle in the graph is at least l vertices long
  - ***With High Probability*** – A probability that is at least 1-(1/n) … probability of (1/n) or less to **fail**

Model and Definitions (2)

- Model:
  - Processors start at the same time
  - Rounds are synchronized
  - Each node knows its own neighbors and degree
  - Each node knows the value of $\log n$
  - Edges are oriented 
  - Graph is $\sqrt{\log n}$-acyclic
  - A ***bit round*** is a synchronized round where each node can send/receive at most 1 bit from each of its neighbors
  - Performance is measured with ***bit complexity*** – the number of ***bit rounds*** the algorithm performs

Agenda

Approach of Algorithm (1)

- Random algorithms suffer from symmetry between neighboring vertices
- Having orientation on the edges is a property that was never studied before the paper, in the context of vertex coloring
- Having orientation helps breaking symmetry

Approach of Algorithm (2)

- This is what random algorithms face in non-oriented graphs:

color choice is illegal – nodes keep random colors until legal
two nodes start with no color

Eventually they succeed, but communication rounds were lost

Approach of Algorithm (3)

- Orientation breaks the symmetry:

Upon illegal choice at random:

Outgoing-edge nodes keep their color

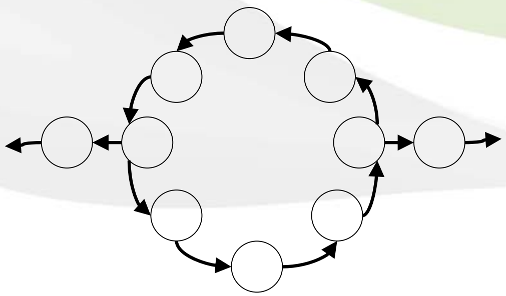
Ingoing-edge nodes change to another color

Approach of Algorithm (4)

- So simple!
- Apparently, not so trivial

Approach of Algorithm (5)

- Unfortunately, symmetry still exists in loops



- We may have a race of colors in the **same** round
- We may suffer from the race of colors after **some** rounds
- Note that this problem also radiates **outside** of the loop

Approach of Algorithm (6)

- Reminder:
  - The model demands are of a $\sqrt{\log n}$-*acyclic* graph
- We will prove that after $O(\sqrt{\log n})$ rounds, we have oriented paths of uncolored nodes of length less than $\sqrt{\log n}$, w.h.p
- **(hence, without loops. Why?)**
  - With this property we can color at least one vertex from each path, round by round.
  - This guarantees completing the coloring in at most $O(\sqrt{\log n})$ additional rounds

Agenda

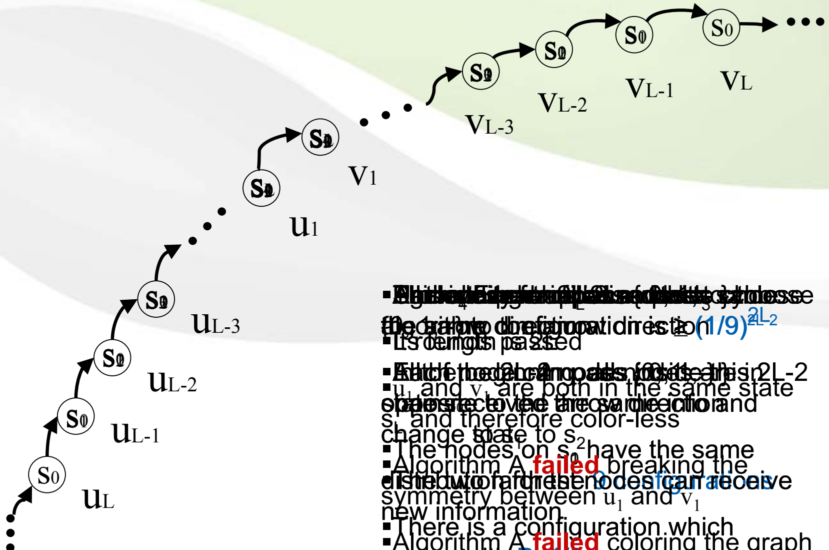
Lower Bound (1)

- Theorem 2.2
  - *For every Las Vegas algorithm A there is an infinite family of* ***oriented*** *graphs G, s.t A has a bit complexity of at least* $\Omega(\sqrt{\log n})$ *on G, w.h.p, to compute proper vertex coloring*

Lower Bound (2) - Proof

- Theorem 2.2, **in simple words:**
  - No random algorithm can perform vertex coloring on an oriented graph in less than $\Omega(\sqrt{\log n})$ bit rounds
- Proof:
  - Counter example
  - We will see a graph that every algorithm fails to color, with a probability larger than $1/n$
  - This graph is a cycle of n nodes

Lower Bound (3) - Proof



[illegible]

- u_1 and v_1 are both in the same state s_1 and therefore color-less
- Algorithm A **failed** breaking the symmetry between u_1 and v_1
- There is a configuration which occurs with $P \geq 1/9$
- Algorithm A **failed** coloring the graph

Lower Bound (4) - Proof

- Algorithm did not stop after L rounds of 1-bit exchange
- Each round of this scenario occurs with $P \geq (1/9)^{2(L-i+1)}$
- For L rounds the probability to **fail** is

$$\left(\frac{1}{9}\right)^{\sum_{i=1}^{\ell} 2(\ell-i+1)} \geq \left(\frac{1}{9}\right)^{\ell^2/2}$$

Lower Bound (5) - Proof

- When choosing:

$$L = \sqrt{2\log_9(n / 2\log^2 n)}$$

this results in $P_{fail} \geq (2\log^2 n)/n$

- This P is for a chain of 2L nodes
- We can have as many as $n/2L$ chains like this in the cycle
- The probability for all cycles to avoid this scenario is **at most**:

$$[\ 1-(2\log^2 n) / n\ ]^{n/2L} \leq 1/n$$

  - (put L and work it out)

Lower Bound (6) - Proof

- Therefore, we need $\Omega(L) = \Omega(\sqrt{\log n})$ bit rounds to succeed coloring w.h.p

■

Agenda

- Problem Definition
- Applications
- Background and Examples
- Model and Definitions
- Lower Bounds
- Approach of Algorithm
- **Algorithm**
- Analysis – Upper Bounds
- Summary and Conclusions

Algorithm (1)

- Algorithm Color-Random (C_u)
- While u is not colored, do: (each iteration is one round)
  1. Node u randomly selects color c_u from $\{1..C_u\}$
  2. Node u communicates its choice c_u to all of its uncolored, out-edge neighbors
  3. If none of u's in-neighbors chose c_u as their color, then u chooses colored with c_u.
     Otherwise, u remains uncolored
  4. If u is colored during step 3, it informs all of its uncolored neighbors about its color choice
     (once colored, u never changes its choice)
  5. Node u removes colors from C_u according to its neighbors' final color choices **(in and out neighbors)**

Algorithm (2)

- Algorithm Color (C_u)

Phase I

1. Set $C_u := c_1\Delta$ for a constant $c_1 \geq 3$
2. While $\delta_u \geq c_2 \log n$ for a constant c_2 do:
   3. Use Algorithm **Color-Random** (C_u)

Phase II

4. Set $C_u := 2c_2 \log n$
5. Use Algorithm **Color-Random** (C_u)

•δ_u is the number of uncolored neighbors of node u

•$C_u = 3\Delta$ but it can be reduced to $\Delta+1$ using 'advanced techniques'

Agenda

- Problem Definition
- Applications
- Background and Examples
- Model and Definitions
- Lower Bounds
- Approach of Algorithm
- Algorithm
- **Analysis – Upper Bounds**
- Summary and Conclusions

Analysis – Upper Bounds (1)

- **Analysis for Phase I**

• During round i, the probability for node u to remain uncolored is:

the neighbors of u chose colors

u chose 1 color from palette

$$P_u(i) \leq \sum_{j=1}^{d_u(i)} \frac{1}{C_u(i)} \leq \frac{\hat{d}(i)}{\alpha\Delta}$$

This is the probability that u chooses same color as one of its neighbors

$\hat{d}(i)$ denotes $\max_u d_u(i)$

$\alpha\Delta$ is just C_u

Analysis – Upper Bounds (2)

- **Analysis for Phase I – continued**
  - The expected number of uncolored neighbors of u, after i rounds, is:

After i rounds

$$E[d_u(i+1)] = \sum_{v \in N_u(i)} P_v(i) \leq \hat{d}(i)^2/\alpha\Delta$$

All neighbors of u, bounded by $\hat{d}(i)$

Probability of each neighbor to be colorless in round i

Analysis – Upper Bounds (3)

- **Analysis for Phase I – continued**
  - We reach the following recurrence relation:

$$\hat{d}(i+1) \leq \frac{\hat{d}^2(i)}{\alpha\Delta} + \sqrt{\gamma \hat{d}(i) \log n} \;\; *$$

  - Where * is exceeded with a probability of less than $1/n^2$
  - Solving the recurrence and finding $d(i) \leq c_2 \log n$ results in $i = O(\log\log n)$ rounds

Analysis – Upper Bounds (4)

- **Analysis for Phase I – Summary**
  - $O(\log\log n)$ rounds, with high probability
  - Communication consists of 3Δ colors – $O(\log\Delta)$ bits.
- **Bit complexity of Phase I is $O(\log\Delta \log\log n)$**

Analysis – Upper Bounds (5)

- **Analysis for Phase II (a)**
  - Lemma 4.2:
    - *After $O(\sqrt{\log n})$ additional rounds, every path of length $L = \sqrt{\log n}$ will have at least one colored node, w.h.p*
  - After $O(\sqrt{\log n})$ additional rounds, the graph will have no uncolored cycles!

Analysis – Upper Bounds (6)

▪ Analysis for Phase II (a) - continued

- Proof of Lemma 4.2:
  - » For simplicity of proof we set $C_u = 2\Delta$
  - » Probability for node u with δ_u uncolored neighbors to remain uncolored at the end of a round:

$$P = \delta_u / C_u \leq \delta_u / [\, 2\Delta - (d_u - \delta_u)\,] \leq 1/2$$

Choose same color as uncolored neighbors

Analysis – Upper Bounds (7)

▪ Analysis for Phase II (a) - continued

- Proof of Lemma 4.2 - continued:
  - » Probability for all nodes in L-nodes path to stay uncolored after r rounds is:

$$P \leq (1/2)^{Lr}$$

Analysis – Upper Bounds (8)

▪ Analysis for Phase II (a) - continued

- Proof of Lemma 4.2 - continued:
  - » The number of such paths in the graph is bounded by $n(c_2 \log n)^L$

n nodes. Theoretically, a path with length L can start from each one of them

Reminder:
By definition, Phase I ends when all nodes have at most $c_2 \log n$ uncolored neighbors

Analysis – Upper Bounds (9)

▪ Analysis for Phase II (a) - continued

- Proof of Lemma 4.2 - continued:
  - » Take $L = \sqrt{\log n}$, $r = 4\sqrt{\log n}$ rounds
  - » The probability to fail in any of the $n(c_2 \log n)^L$ paths of length L, for r consecutive rounds, is:

$$P \leq n(c_2 \log n)^L (1/2)^{Lr} \leq 1/n$$

Low probability for the scenario to occur in any of the potential paths

Analysis – Upper Bounds (10)

▪Analysis for Phase II (a) - Summary

- r = $4\sqrt{\log n}$ rounds
- Each round, each node transfers one of $c_2\log n$ possible colors
- Bit complexity of this sub-phase:

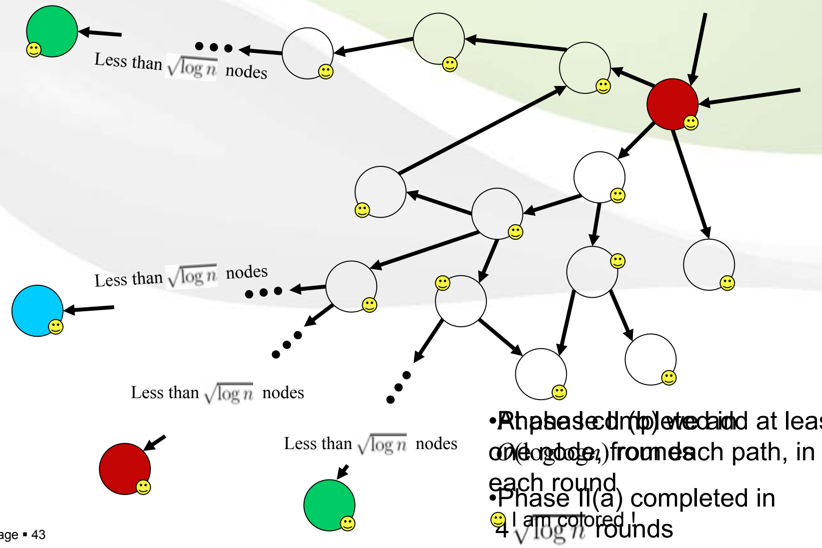
$$O(\sqrt{\log n}\log\log n)$$

Analysis – Upper Bounds (11)

▪Analysis for Phase II (b) - continued

- The rest of phase II ends after at most $\sqrt{\log n}$ additional rounds!
- Proof:
  - »Uncolored directed paths are not longer than $\sqrt{\log n}$ nodes (phase II(a))
  - »At each round, at least one node of each path gets colored (Orientation!) ■

Analysis – Upper Bounds (12)



•At each round we add at least one node, from each path, in each round
•Phase II(b) completed in $O(\log\log n)$ rounds
•Phase II(a) completed in $4\sqrt{\log n}$ rounds

Analysis – Upper Bounds (13)

▪Algorithm Color(C_u) analysis summary

- Bit complexity of phase I: $O(\log\Delta \log\log n)$
- Bit complexity of phase II(a): $O(\sqrt{\log n} \log\log n)$
- Bit complexity of phase II(b): $O(\sqrt{\log n} \log\log n)$
- **Overall bit complexity:** $O(\log\Delta \log\log n + \sqrt{\log n} \log\log n)$

Analysis – Upper Bounds (14)

- The paper offers further improvements:
  - Phase I can be reduced to $O(\log\Delta)$
    - » Done by decreasing the color palette whenever $\delta_u(i)$ decreases below $\sqrt{\delta_u(i-1)}$
  - Phase II is not changed
  - If $\Delta \geq \log n$ then Phase II bit complexity is $O(\sqrt{\log n \log\log n})$
    - » This is done by different approach in calculation
    - » Splitting Phase II to 3 sub-phases:

      (a) $\delta_u \leq O(\sqrt{\log n}\log\log n)$ after $O(\sqrt{\log n}/\log\log n)$, w.h.p

      (b) After $\sqrt{\log n/\log\log n}$ rounds, every path of length $\sqrt{\log n/\log\log n}$ has at least one colored node, w.h.p

      (c) Finish in $\sqrt{\log n/\log\log n}$ more rounds
  - Overall bit complexity of $O(\log\Delta + \sqrt{\log n \log\log n})$

Agenda

Summary

- We learned about the lower bound of Vertex Coloring in oriented graphs (non-oriented graphs lower bound proof is almost identical)
- We learned about how to break symmetry using orientation of the edges
- We analyzed a random algorithm that uses edges' orientation to effectively color graphs with $\Delta = f(n)$

Conclusions

- There is room to investigate how to achieve directed, l-acyclic graphs, since the results are great
  - Possible with UIDs
  - Possible with "which node was the first to discover the link"
  - Possible by coin toss
- The paper introduces a new approach (orientation) to an old question
  - Leaves room for further investigation of algorithms