Vertex Coloring & Maximum Independent Sets

Martin Biely

Embedded Computing Systems Group
Vienna University of Technology
European Union

December 1, 2006
Outline

1. Vertex Coloring

2. Maximum Independent Sets
The Coloring Problem

Definition 1 (Vertex Coloring)

A legal *vertex coloring* for a graph is an assignment of a color $\varphi_v$ to each vertex $v$, such that no two adjacent vertices have the same color. I.e., $\varphi_v \neq \varphi_w$ for every $(v, w) \in E$. 
Chromatic Number

Note that, the Ids of vertices form a legal coloring. However, we want as few colors as possible. But: how many are as few as possible?
Note that, the Ids of vertices form a legal coloring. However, we want as few colors as possible. But: how many are as few as possible?

**Definition 2 (Chromatic Number)**

The *chromatic number* $\chi(G)$ of a graph $G$ is the minimum number of colors needed for a legal vertex coloring of the graph.
Typical Chromatic Numbers

- $\chi(G) = 1$, iff $G$ is disconnected ($E = \emptyset$)
- $\chi(G) \geq 3$, iff $G$ has an odd cycle ($G$ is not bipartite)
- $\chi(G) \geq |\text{biggest clique}|$, e.g., trees have $\chi(G) = 2$
- $\chi(G) \leq 4$, if $G$ is planar. (“four color theorem”)
- $\chi(G) \leq \Delta(G) + 1$, ($\Delta$ is the maximum degree)
Reducing the palette

**function** first-free($W, P$)

  let $c$ be the smallest color in $P$ not in use by any vertex in $W$
  return $c$

**function** reduce($m$)

  for round $i = \Delta + 2$ to $m$ do
    if $\varphi_v = i$ then
      $\varphi_v \leftarrow \text{first} - \text{free}(\mathcal{N}_v, P_{\Delta + 1})$
      inform neighbours ($\mathcal{N}_v$) of this choice
Lemma 3

The function reduce produces a legal coloring of the network $G$ with $\Delta + 1$ colors, and $\text{TIME}(\text{reduce}) = m - \Delta + 1$.

Proof.

Consider iteration $i$ of the procedure. Since there are at most $\Delta$ neighbours at most $\Delta$ different colors can be in use by the neighbours already, therefore one of the $\Delta + 1$ colors will be returned by \textit{first-free}. In iteration $i$, only vertices with color $i$ change their color. Since we have a legal coloring before the $i$th iteration, no two neighbours ever change their color in the same round.
Algorithm 6-Color:

$\varphi_v \leftarrow id_v$

repeat

$\ell \leftarrow |\varphi_v|$

if $v$ is root then

$l_v \leftarrow 0$

else

$l_v \leftarrow \min\{i \mid \varphi_v[i] \neq \varphi_{parent(v)}[i]\}$

$\varphi_v \leftarrow \langle l_v; \varphi_v[l_v] \rangle$

inform children of new color

until $|\varphi_v| = \ell$
Algorithm \texttt{6-COLOR} (Proof)

Lemma 4

\textit{In each iteration, \texttt{6-COLOR} produces a legal coloring.}
Lemma 4

In each iteration, 6-\textsc{Color} produces a legal coloring.

Proof.

Consider two adjacent vertices of the tree, $v$ and $w$, and let $v = \text{parent}_w$. If $I_v \neq I_w$, then the new colors will differ in their first component, otherwise, the choice of $I_v$ ensures that $\varphi_v[I_v] \neq \varphi_w[I_w]$.\hfill $\square$
Algorithm 6-\textsc{Color} (Proof)

**Definition 5**

Let $K_i$ denote the number of bits after the $i$th iteration. 
$K_0 = |id| = O(\log n)$.

**Lemma 6**

The following hold:

- $K_{i+1} = \lceil \log K_i \rceil + 1$
- $K_{i+1} < K_i$ as long as $K_i \geq 4$
- $K_i \leq \left\lceil \log^{(i)} K \right\rceil + 2$ for every $i$

**Lemma 7**

The final coloring consists of 6 colors.

**Proof.**

By previous lemma, in the final iteration, since $K_{i} \leq K_{i-1} + 6$, 2
Algorithm 6-\textcolor{red}{COLOR} (Proof)

Lemma 6

\textit{The following hold:}

- $K_{i+1} = \lceil \log K_i \rceil + 1$
- $K_{i+1} < K_i$ as long as $K_i \geq 4$
- $K_i \leq \left\lceil \log^{(i)} K \right\rceil + 2$ for every $i$

Lemma 7

\textit{The final coloring consists of 6 colors.}

Proof.

By previous lemma, in the final iteration $i$ we have $K_i = K_{i-1} \leq 3$. Therefore, there are 3 choices for the bit $I_v$ and 2 possible values for the bit. I.e., a total of 6 colors.
From 6 to 3 Colored Trees

**Algorithm Six2Three:**

```plaintext
for \( x \in \{4, 5, 6\} \) do
    use parent’s color (root just chooses any new one)
    Each vertex with \( \varphi_v = x \): \( \varphi \leftarrow \text{first free}(N_v, \mathcal{P}_3) \)
```

**Lemma 8**

*Each iteration removes one color.*
Complexity of 3-coloring a Tree

How many rounds does $\text{6-COLOR}$ need?
- $K_{i+1} = \lceil \log K_i \rceil + 1$
- $K_i \leq \log^{(i)} K + 2$ for every $i$

**Definition 9**

We define $\log^{(*)} n = \min\{i \mid \log^{(i)} n \leq 2\}$.

(E.g., $\log^{(*)} 65535 = 3$, since $\log \log \log 65535 \approx 1.999$)

**Theorem 10**

Algorithms $\text{6-COLOR}$ and $\text{Six2Three}$ color any tree in time $O(\log^{(*)} n)$. 
Let $W_{s,n} = \{(x_1, \ldots, x_s) \mid 1 \leq x_i \leq n, x_i \neq x_j\}$.

In $t$ rounds any vertex $v$ can learn the topology of its $t$ neighbourhood $\Gamma_t(v)$, as $W_{2t+1,n}$.

Define $B_{s,n} = (W_{s,n}, E_{s,n})$, where $(w_1, w_2) \in E_{s,n}$ when $w_1$ and $w_2$ could be views of neighbouring processes.

**Algorithm** $\Pi_t$:

Collect topology for $t$ rounds
Let $\zeta(v) \in W_{2t+1,n}$ be the information gathered

$\varphi_v \leftarrow \tilde{\varphi}_\Pi(\zeta(v))$, where

$\varphi_v \leftarrow \tilde{\varphi}_\Pi : W_{s,n} \rightarrow \{1, \ldots, c_{\max}\}$ is the coloring function of $\Pi_t$
Lemma 11

If $\Pi_t$ produces a legal coloring for any $n$-vertex ring, then $\varphi_v \leftarrow \tilde{\varphi}_\Pi$ defines a legal coloring for $B_{2t+1,n}$.

Proof.

Assume otherwise, then for two neighbouring vertices $\zeta$ and $\zeta'$ of $B_{2t+1,n}$ we have $\varphi_v \leftarrow \tilde{\varphi}_\Pi(\zeta) = \varphi_v \leftarrow \tilde{\varphi}_\Pi(\zeta')$. But then $\Pi_t$ will yield a coloring for a ring where 2 neighbouring vertices have views $\zeta$ and $\zeta'$ with the same color.

Corollary 12

If the $n$-vertex ring can be colored in $t$ rounds using $c_{\text{max}}$ colors, then $\chi(B_{2t+1,n}) \leq c_{\text{max}}$. 


Corollary 12

If the $n$-vertex ring can be colored in $t$ rounds using $c_{\text{max}}$ colors, then $\chi(B_{2t+1,n}) \leq c_{\text{max}}$.

Theorem 13

Any deterministic distributed algorithm for coloring the $(2n)$-vertex ring with two colors requires at least $n - 1$ rounds.

Proof.

By corollary, $\chi(B_{2t+1,2n}) \leq 2$. However, we can find a cycle of odd length in $B_{2t+1,2n}$, for $t \leq n - 2$ But we know $\chi(G) \geq 3$, iff $G$ has an odd cycle. Contradiction.
Back to 3-coloring rings...
Lemma 14

\[ \chi(B_{2t+1,n}) \geq \log^{(2t)} n. \]

We define a new Graph \( \tilde{B}_{s,n} = (\tilde{E}_{s,n}, \tilde{E}_{s,n}) \), s.t.
\[ \tilde{W}_{s,n} = \{(x_1, \ldots, x_s) \mid 1 \leq x_1 < \cdots < x_s \leq n \} \text{ and } (\zeta, \zeta') \in \tilde{E}_{s,n} \text{ if } \zeta = (x_1, x_2, \ldots, x_s) \text{ and } \zeta' = (x_2, \ldots, x_s, b), \text{ with } x_s < b \leq n. \]
Since the undirected version of \( \tilde{B}_{s,n} \) is a subgraph of \( B_{s,n} \) we have:

Lemma 15

\[ \chi(B_{s,n}) \geq \chi(\tilde{B}_{s,n}) \]

To show Lemma 14 we have to bound \( \chi(\tilde{B}_{s,n}) \) from below.
Definition 16 (Line Graph)

Given a directed graph $H = (U, F)$, the line graph of $H$ denoted $DL(H)$, is a directed graph whose vertices are the arcs of $F$ and in which there is an arc from $e$ to $e'$ ($e, e' \in F$) iff in $H$, $e'$ starts at the vertex in which $e$ ends.
Lemma 17

- The undirected version of $\tilde{B}_{1,n}$ is a complete graph of size $n$.
- $\tilde{B}_{s+1,n} = \mathcal{DL}(\tilde{B}_{s,n})$.

Proof of second claim.

We establish an isomorphism between the $\tilde{B}_{s+1,n}$ and $\mathcal{DL}(\tilde{B}_{s,n})$. Consider vertex $e$ of $\mathcal{DL}(\tilde{B}_{s,n})$, then $e$ is an arc in $\tilde{B}_{s,n}$ connecting $(x_1, x_2, \ldots, x_s)$ and $(x_2, \ldots, x_s, y)$. Then we can map $e$ to $(x_1, x_2, \ldots, x_s, y)$ in $\tilde{B}_{s+1,n}$.

Consider an edge $f$ of $\mathcal{DL}(\tilde{B}_{s,n})$, then its start and endpoints, can be mapped to two edges connecting $(x_1, x_2, \ldots, x_s)$, $(x_2, x_3, \ldots, x_s, y)$ and $(x_3, \ldots, x_s, y, z)$. Therefore it can be mapped to the edge in $\tilde{B}_{s+1,n}$ connecting $(x_1, x_2, \ldots, x_s, y)$ and $(x_2, \ldots, x_s, y, z)$.
Lemma 18

For every directed graph $H$, $\chi(\mathcal{DL}(H)) \geq \log(\chi(H))$

Proof.

Let $k = \chi(\mathcal{DL}(H))$, then there is a $k$-coloring $\hat{\phi}$ of $\mathcal{DL}(H)$ and $\hat{\phi}$ can be seen as a edgecoloring of $H$ where if $e$ ends where $e'$ starts, $\hat{\phi}(e) \neq \hat{\phi}(e')$. Clearly we can create a $2^k$-coloring for $H$:

$$\tilde{\phi}(v) = \{\hat{\phi}(e) \mid e \text{ ends in } v\}.$$ 

It is legal: Consider $v, w \in H$, with $e = (v, w) \in H$, clealy $\tilde{\phi}(v)$ contains $\hat{\phi}(e)$, but $\tilde{\phi}(w)$ does not. Otherwise there must be an $e'$ ending in $w$ with that color, which is a contradiction to the assumption that it $\hat{\phi}$ is a legal edge-coloring.

It follows that $\chi(H) \leq 2^k$. \qed
By induction on $s$ we can now show that

**Lemma 19**

$$\chi(\tilde{B}_s,n) \geq \log^{(s-1)} n.$$  

And finally, Lemma 14, i.e., $\chi(B_{2t+1,n}) \geq \log^{(2t)} n$. From this and our corollary we get: if an algorithm collects information for $t$ rounds, it can color with $\chi(B_{2t+1,n}) \geq \log^{(2t)} n$ colors. Hence, $2t \geq \log^{(*)} n - 1$ for a 3-coloring algorithm. I.o.w.:

**Theorem 20**

Any deterministic distributed algorithm for coloring the $n$-vertex ring with three colors requires at least $\frac{1}{2}(\log^{(*)} n - 1)$ rounds.
Outline

1. Vertex Coloring

2. Maximum Independent Sets
The MIS Problem

Definition 21

An independent set is a set of vertices $U \subseteq V$, no two of which are adjacent. An independent set is maximal of no vertex can be added to it without violating its independence. The MIS problem is the problem of computing a maximal independent set for a given graph.

Definition 22

The LexMIS problem is the problem of finding the lexicographically first maximal independent sets.

(Both of these problems are in PTIME)
... and an NP-hard variant

**Definition 23**

The maximal-size MIS problem is the problem of finding the biggest maximal independent set.
**Algorithm** \textsc{Greedy-MIS}

**Variables:**

\[ U \leftarrow V, \quad M \leftarrow \emptyset \]

\textbf{while} \( U \neq \emptyset \) \textbf{do}

\hspace{1em} pick any \( v \in U \).

\hspace{1em} \( U \leftarrow U \setminus (\{v\} \cup \mathcal{N}_v) \)

\hspace{1em} \( M \leftarrow M \cup \{v\} \)
Distributed greedy MIS construction

MIS is solvable by straightforward DFS.
LexMIS can be solved by the following algorithm...
Distributed greedy MIS construction

**Algorithm** MIS-Rank

**Variables:**

\[ b \in \{0, 1, -1\} \leftarrow -1 \]

**initially:**

- **invoke** `join()`

**when** receiving a message \((\text{DECIDED}, 1)\) from \(w\) **do**

\[ b \leftarrow 0 \]

**send** \((\text{DECIDED}, 0)\) to all neighbours

**when** receiving a message \((\text{DECIDED}, 0)\) from \(w\) **do**

- **invoke** `join()`

**function** `join()`

- **if** received \((\text{DECIDED}, 0)\) from all \(w\) with \(id(w) > \text{myid}\) **then**

\[ b \leftarrow 1 \]

**send** \((\text{DECIDED}, 1)\) to all neighbours.
Reducing coloring to MIS

Given a coloring with \( m \) colors...

**Algorithm** \textsc{Color2MIS}(m)

\[
\text{for } \text{round } i = 0 \text{ to } m \text{ do} \\
\quad \text{if } \varphi_v = i \text{ then} \\
\quad \quad \text{if none of } v \text{'s neighbours has joined the MIS then} \\
\quad \quad \quad b \leftarrow 1 \text{ (join MIS)} \\
\quad \quad \quad \text{inform neighbours} \\
\quad \text{else} \\
\quad \quad b \leftarrow 0
\]
Lemma 24

Algorithm \textbf{COLOR2MIS} constructs an MIS from a \emph{m}-colored graph \emph{in} \emph{m} rounds.

Proof.

The \emph{m} rounds claim is obvious.
To see that the set is independent: No two adjacent vertices can be “activated” in the same round, since we have a legal coloring.
To see that the set is maximal: Assume that there is a vertex with color \(i\) that could be added to the set, then none of its neighbours has joined the set before round \(i\), which would have caused this vertex to join. Therefore no such process can exist.
Lemma 24

Algorithm \textsc{Color2MIS} constructs an MIS from a \textit{m}-colored graph in \textit{m} rounds.

Proof.

The \textit{m} rounds claim is obvious.
To see that the set is independent: No two adjacent vertices can be “activated” in the same round, since we have a legal coloring.
To see that the set is maximal: Assume that there is a vertex with color \textit{i} that could be added to the set, then none of its neighbours has joined the set before round \textit{i}, which would have caused this vertex to join. Therefore no such process can exist.

Corollary 25

Given a coloring algorithm that colors every graph \textit{G} with \textit{f(}\textit{G})-colors in \textit{T(}\textit{G}) rounds, it is possible to construct an MIS for \textit{G} in time \textit{T(}\textit{G}) + \textit{f(}\textit{G}) rounds.

Corollary 26

There exists a deterministic distributed MIS algorithm for trees and for bounded-degree graphs with time complexity \(O(\log^2 n)\).
MIS lower bound on rounds

Theorem 27

Any deterministic distributed MIS algorithm for the n-vertex ring requires at least \( \frac{1}{2}(\log^{(*)} n - 1) - 1 \) rounds.

Proof.

It suffices to show, that we can 3-color a ring given an MIS with one additional round.

Assuming each vertex which neighbours are clockwise and counterclockwise. We can color a tree by requiring every vertex to send a message InMIS in clockwise direction. Every vertex in the MIS uses color 1, those receiving an InMIS message use color 2 and those not receiving such a message use color 3. Without this assumption we can still color in one round, if vertices have ids. If vertices are anonymous 3-coloring is impossible.
Small dominating sets

Definition 28

A set $M$ of vertices in a graph $G = (V, E)$ is said to be dominating if every vertex in $V \setminus M$ has a neighbour in $M$, or iow, if $\text{dist}(v, M) \leq 1$ for every $v \in V$.

Clearly $V$ dominates $G$, therefore we are looking for small dominating sets, i.e., a set $M$ s.t.:

- $M$ dominates $G$ and
- $|M| \leq n/2$
Algorithm \textsc{Small-Dom-Set}

\[ L \leftarrow \min_{l \in \text{leaves}} \text{dist}(v, l) \]

\textbf{if} \ \ l > 2 \ \ \textbf{then} \\
- participate in MIS algorithm \\
- all processes in MIS join dominating set

\textbf{else if} \ \ l = 1 \ \ \textbf{then} \\
- join dominating set
Correctness of **Small-Dom-Set**

Let $L_0$, $L_1$, and $L_2$ denote the vertices at distance 0, 1, and 2 resp. from a leaf. Let $M$ be the outcome of **Small-Dom-Set** on the tree $T$ and $Q$ the outcome of the MIS algorithm on the subtree.

**Lemma 29**

*Then for every vertex $v \notin M$ there exists an adjacent vertex $v' \in M$.*

**Proof.**

Now partition $V$ into $L_0$, $L_1$, $L_2$ and $R$. Clearly the $Q$ dominates $R$. And the vertices in $L_0$ and $L_2$ are dominated by those in $L_1$. \(\square\)
Lemma 30

|M| \leq \frac{n}{2}

Proof.

By construction $M = Q \cup L_1$. Obviously $|L_1| \leq |L_0|$ and therefore

$$|L_1| \leq \frac{|L_0 \cup L_1|}{2}.$$ 

Moreover, each $v \in Q$ has at least one child in $R \cup L_2$ which is not in $Q$. Therefore, $|R \cup L_2| \geq 2 \cdot |Q|$.

It now follows,

$$|M| = |Q| + |L_1| \leq \frac{|R \cup L_2|}{2} + \frac{|L_0 \cup L_1|}{2} = \frac{n}{2}.$$
Since we only need to determine the displance from the leaves for up to 2, vertices can determine their (potential) membership in $L_{0,1,2}$ in a constant number of rounds. Therefore,

**Corollary 31**

There exists a deterministic distributed algorithm for constructing a small dominating set on trees with time complexity $O(\log^*(n))$. 
Proposition 32 (Final proposition)

37 slides suffice.

Proof.

Left as an exercise to the audience.