Algorithmic Methods 13/12/2010

Lecture Notes 9: Symmetry breaking in distributed networks

Professor: Yossi Azar *Scribe:Ilan Cohen*

1 Introduction

Given a distributed computer ring network C_n with n computers, each computer (processor) has a unique ID, we want that each processor choose a color, so that, there is no two adjacent processors with the same color.
Notice: If n is odd number the number of colors needed is 3.

Distributed Computing

In each step each processor can send and receive messages from it's neighbors.

Simple solution

We give one of the processors a token, starting with color 0, this processor choose the color of the token and pass the token to it's right neighbor, with the opposite color, the last processor choose the color 2, this solution would take $\theta(n)$. We will show a solution that take $\log^*(n) + O(1)$ where $\log^*(n) = min\{i | \log^{(i)} n \leq 1\}$. Note that $\log^{(0)} n = n$ and $\log^{(i)} n = \log \log^{(i-1)} n$ (i.e., applying log iteratively i times).

2 Coloring in distributed computers

The algorithm works in steps.
In step k each processor has a color between $[0, \ldots, l_k - 1]$.
Init: C(x) = ID(x).
At the beginning of step k+1 there is a legal coloring with l_k colors at the the end of step, there is a legal coloring with l_{k+1} colors.
Step k: each processor X looks at the color of its right neighbor Y. Consider the binary representation of $C_k(X)$ and $C_k(Y)$.
$C_k(X) = (x_r, \ldots, x_0)$
$C_k(Y) = (y_r, \ldots, y_0)$
let i be the minimal index that this color are different $i = min\{i | x_i \neq y_i\}$
The new color or X would be $C_{k+1}(X) = (i, x_i) = 2 \cdot i + x_i$.
The number of colors descends from r to $\lceil \log_2 r \rceil + 1$.

Claim: if $C_k(X) \neq C_k(Y)$ then $C_{k+1}(X) \neq C_{k+1}(Y)$

- Case 1: $i_x \neq i_y$ then $(i_x, x_{i_x}) \neq (i_y, y_{i_y})$.
- Case 2: $i_x = i_y$, by definition $x_{i_x} \neq y_{i_y}$ hence $(i_x, x_{i_x}) \neq (i_y, y_{i_y})$.

Let r_k be the number of bits in step k then $r_{k+1} = \lceil \log_2 r_k \rceil + 1$.
If $r_k \geq 4$ then r_k descends.
If $r_k = 3$ then in the next step there would be 6 colors.
Claim: If $\log^{k-1} n \geq 3$ then $r_k \leq \lceil \log^k n \rceil + 2$.
By induction: base $r_0 = n + 1$ because ID is n bits.
Inductive Step:
$r_k = \lceil \log_2(r_{k-1}) + 1 \rceil \leq \lceil \log_2(\lceil \log^{(k-1)}(n) \rceil + 2) \rceil + 1$
$\leq \lceil \log_2(\log^{(k-1)}(n) + 3) \rceil + 1 \leq \lceil \log_2(2 \cdot \log^{(k-1)}(n)) \rceil + 1 \leq \lceil \log^{(k)}(n) \rceil + 2$

From 6 colors to 3 colors in 3 steps.

Each processor has one of the color (0...5), the algorithm works in 3 steps for $3 \leq i \leq 5$.
Step i: each processor with color i, chooses a color between 0,1,2 that is different from its neighbors.
Note that if a processor chooses a color, then the color of its neighbors do not change in this iteration, and the processor can always choose a color different from its neighbors.

3 Lower bound for distributed ring coloring

Theorem: Any local algorithm for coloring with constant number of colors requires at least $\frac{1}{2} \log^* n - 1$ steps.

Assume we got algorithm that colors the ring after t steps.
After t steps, the information that each processor has $(x_{i-t}, \ldots, x_i, \ldots, x_{i+t})$, the length of the vector is $2t + 1$ and all the x_i are different.
Any algorithm is a function f from these vectors to the colors 0,1,2.
Consider two vertices:
$v_1 = (x_{i-t}, \ldots, x_i, \ldots, x_{i+t})$
$v_2 = (x_{i-t+1}, \ldots, x_{i+1}, \ldots, x_{i+t+1})$
We call v_2 a legal 1-shift of v_1 if v_2 is a shift by 1 of v_1 while all x_i are different and $x_{i-t} \neq x_{i+t+1}$.
Observation: if v_2 is a legal 1-shift of v_1, then $f(v_1) \neq f(v_2)$
Proof: v_1 and v_2 might be adjacent on the ring so they must be colored differently.

Let us build a graph $G^t(V, E)$ its vertices would be $2t + 1$ length vectors with different integer number (even between 0 to n).
Two vertices are connected if one is a legal 1-shift of the other.
The f function yields a legal coloring of the graph G^t with 3 colors.
Let $\chi(G)$ be the coloring number of G.
We prove that if $t < \frac{1}{2} \log^* n - 1$ then $\chi(G^t) > 3$. We will actually show that it is true even for a subgraph of G^t. Specifically, there is $H \subseteq G^t$ such that $\chi(H) > 3$.

Line Graph

Let $G = (V, E)$ be a directed graph, we will define $LG(G)$ as follows

- The edges of G are the vertices of LG.
- Two vertices in LG $e_i = (v_{i_1}, v_{i_2}), e_j = (v_{j_1}, v_{j_2})$, are connected if $v_{j_1} = v_{i_2}$

Claim: for our G $\chi(G) \leq 2^{\chi(LG)}$
Prove: assume we have a color of LGthat use k colors, we will show coloring of LG with 2^k colors. The coloring would be:
$C(v) = (b_0, b_1, \ldots, b_{k-1}), b_i = 1$ if there exists edge (v, X) with color i.
The number of colors is 2^k, now we will prove that it is a legal coloring.
Assume $(X, Y) \in E$ in LG the color of (X, Y) is i, that mean in the coordinate i of $C(X)$ is 1, assume that also in the i coordinate of $C(Y)$ is 1, that mean that exist Z, so that $(Y, Z) \in E$ and its color is also i, $((X, Y), (Y, Z)) \in E(LG)$ and $(X, Y), (Y, Z)$ has the same color, contradiction that the coloring of LG is legal.

Lets define H_0 graph with n vertices indexed from 0 to $n - 1$, and exist edge (i,j) if $i < j$, $\chi(H_0) = n$ (full graph).
$H_1 = LG(H_0)$ - vertices (i, j) $i < j$, there exist edge between (i,j) and (j,k) if $i < j < k$. By the claim $\chi(H_1) \geq \log_2(n)$.
$H_2 = LG(H_1)$ - vertices (i, j, k), exist edge between (i,j,k) to (j,k,l) if $i < j < k < l$. $\chi(H_2) \geq \log_2(\log_2(n))$.
...
In general the vertices of H_k are $(i_0, i_1, \ldots, i_k)$ while for $i_0 < i_1 < \ldots < i_k$ there exist an edge between $(i_0, i_1, \ldots, i_k)$ to $(i_1, i_1, \ldots, i_{k+1})$ if $i_i < i_1 < \ldots < i_k < i_{k+1}$.

By a simple induction we have $\chi(H_k) \geq \log^{(k)}(n)$. Note that on H_{2t+1} subgraph of G^t (we just add a requirement of sequence monotonicity). If $2t + 1 < \log^* n$ then :
$\log^{(2t+1)}(n) \leq \chi(H_{2t+1}) \leq \chi(G^t) \leq C$ (the constant number that the algorithm color G).
This implies that $t \geq \frac{1}{2} \log^* n - 1$.