

Distributed Algorithms 2020

LOCAL model: Unique identifiers

LOCAL model port-numbering model + unique identifiers

Nodes have distinct labels from {1, 2, ..., poly(*n*)}

LOCAL model

- Everything can be solved in diam(G)+1 rounds!
- Universal algorithm: "each node tells its neighbors everything it knows"
 - 1 round: everyone aware of its adjacent nodes and incident edges
 - **T rounds:** everyone aware of all nodes and edges within distance *T*
 - diam(G)+1 rounds: everyone knows G

LOCAL model

• Not so interesting:

"What can be computed?"

• Very interesting:

"What can be computed efficiently?" (efficient ≈ o(diam(G)) rounds)

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	O(log* n)	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta + 1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta + 1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	O(log* n)	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta + 1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta + 1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Greedy color reduction

• If I am a *local maximum*:

- pick the smallest free color that is not used by any of my neighbors
- k+1 colors $\rightarrow k$ colors • provided that $k \ge \Delta+1$



Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	O(log* n)	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta + 1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta + 1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Rotating clocks

- $q = \text{prime}, q > 2\Delta$
- q^2 colors $\rightarrow q$ colors in q rounds
- If no conflicts:
 - $(a, b) \rightarrow (0, b)$
- Otherwise:
 - $(a, b) \rightarrow (a, b+a \mod q)$





Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	O(log* n)	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta + 1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta + 1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

• Old color of node v is a set $S(v) \subseteq \{1, 2, ..., m\}$

{1, 4, 7}

6}

• Promise:

• new color of v is an element of S(v)

•Safe:

 pick an element of S(v) that is not in any S(u) for any neighbor u

• Old color of node v is a set $S(v) \subseteq \{1, 2, ..., m\}$

4, 6}

6}

• Bad:

- sets of neighbors cover all values in my set
- no safe choice left

1-cover-free family

• For up to 1 neighbor these sets are good:

$$S_{1} = \{ 1, 2 \}$$

$$S_{2} = \{ 1, 3 \}$$

$$S_{3} = \{ 1, 4 \}$$

$$S_{4} = \{ 2, 3 \}$$

$$S_{5} = \{ 2, 4 \}$$

$$S_{6} = \{ 3, 4 \}$$

2-cover-free family

• For up to 2 neighbors these sets are good:

$$S_{1} = \{ 1, 2, 3 \}$$

$$S_{2} = \{ 3, 4, 5 \}$$

$$S_{3} = \{ 5, 6, 7 \}$$

$$S_{4} = \{ 1, 4, 7 \}$$

$$S_{5} = \{ 2, 4, 6 \}$$

- •Assume: \mathbf{X} -coloring, maximum degree $\mathbf{\Delta}$
- There is a Δ-cover-free family S₁, S₂, ..., S_x
 all subsets of {1, 2, ..., m}
- Nodes of color c pick set S_c
- There is always a safe choice for any node!
- Color reduction from x to m

- Δ-cover-free family S₁, S₂, ..., S_x
 all subsets of {1, 2, ..., m}
- Good if:
 - Δ large \rightarrow works in high-degree graphs
 - x large → tolerates many input colors
 m small → produces a good output color
 - m small \rightarrow produces a good output coloring
- E.g. x = m is trivial (why?)

• q = prime, **GF(q)** ≈ integers modulo q

- *f* = degree-*d* **polynomial** over GF(*q*)
 - at most d points where f(x) = g(x)
 - *q^{d+1}* possible polynomials

• $S_f = \{ (x, f(x)) | x = 0, 1, ..., q-1 \}$

- base set: all q^2 possible pairs (x, y)
- q^{d+1} possible subsets, each with q elements
- intersection of S_f and S_g has size at most d
- If $q > \Delta d$: a Δ -cover-free family
 - why?



f(*x*), *g*(*x*)

• Construct Δ -cover-free families with suitable parameters

•
$$n \rightarrow \approx \Delta^2 \log^2 n$$

 $\rightarrow \approx \Delta^2 \log^2 \log n$
 $\rightarrow \approx \Delta^2 \log^2 \log \log n$
...
 $\rightarrow \approx \Delta^2 \log^2 \Delta$
 $\rightarrow \approx \Delta^2$

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	O(log* n)	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta + 1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta + 1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms