Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

Distributed (Δ+1)-Coloring in Linear (in Δ) Time

Nico Eigenmann

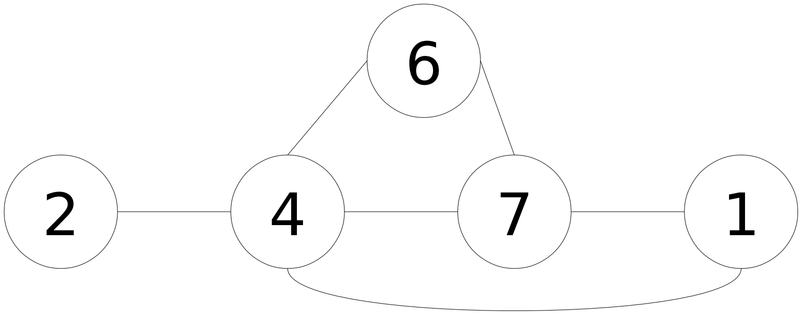
Authors

- Leonid Barenboim
- Michael Elkin

- University of Negev

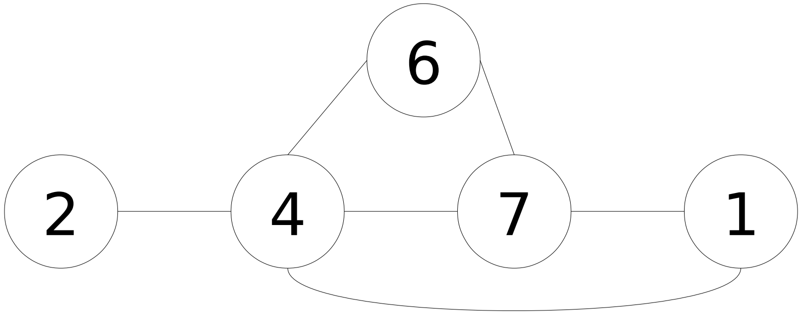
Message Passing Model

- Undirected Graph G(V,E)
- Synchronous
- Reliable Message Transfer
- Unlimited Computing Power
- Unlimited Message Size



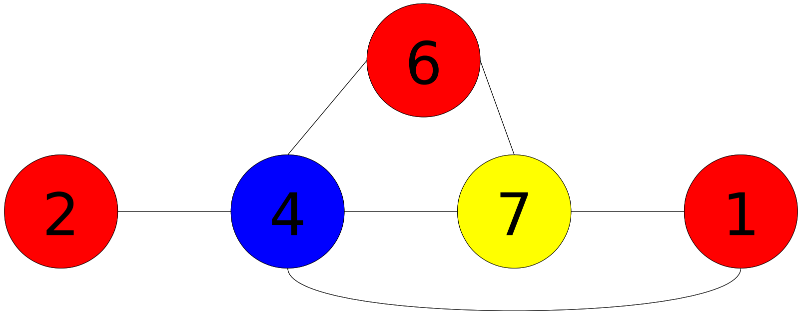
What is a coloring?

A coloring is a function $\varphi:V\rightarrow\mathbb{N}$, that assigns a color to each vertex, such that for all edges $(u,v)\in E$ $\varphi(u)\neq\varphi(v)$



What is a coloring?

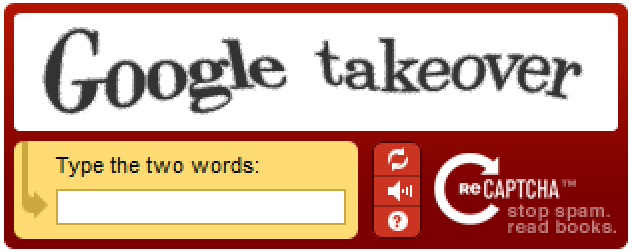
A coloring is a function $\varphi: V \rightarrow \mathbb{N}$, that assigns a color to each vertex, such that for all edges $(u,v) \in E$ $\varphi(u) \neq \varphi(v)$



Applications

- Scheduling

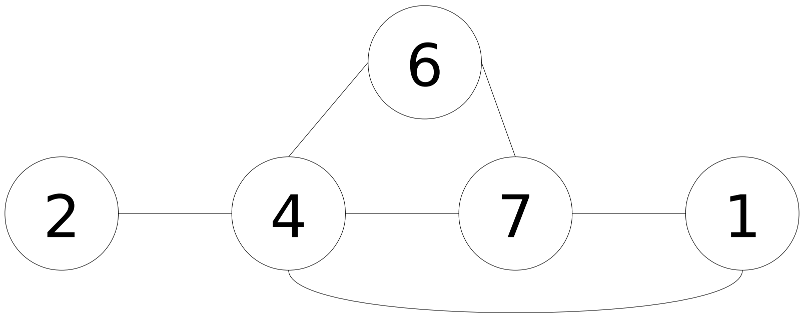
- Pattern Matching



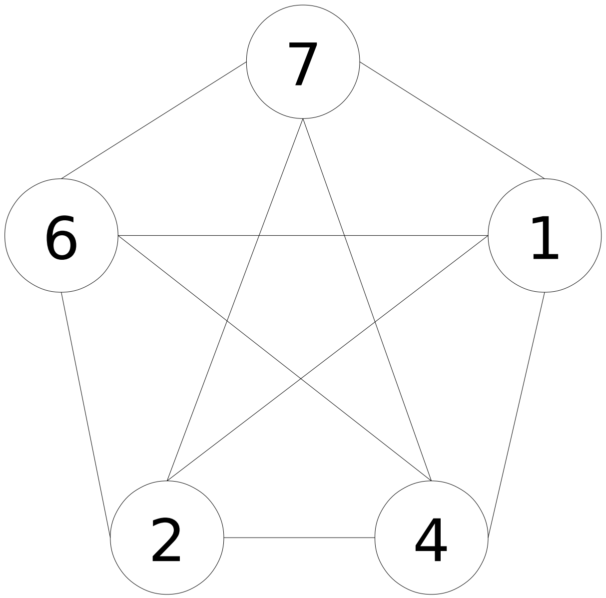
- Radio Frequency Assignment

What is Δ?

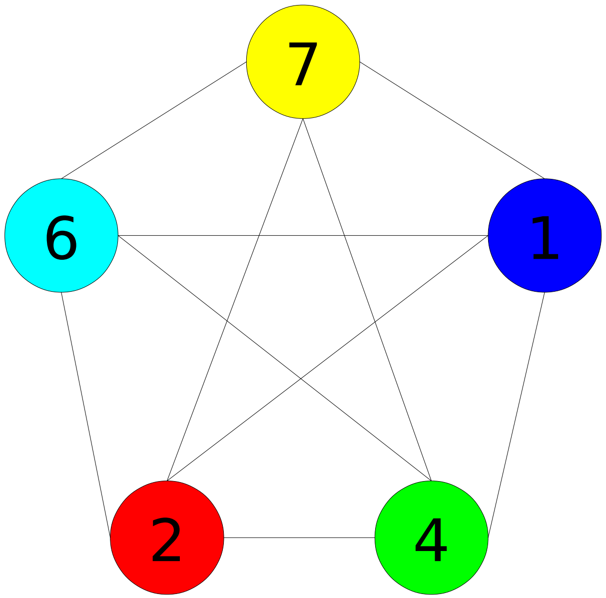
- deg(v): #edges adjacent to v
- $\Delta = \max_{v \in V}\{deg(v)\}$



Why (Δ+1)-coloring?



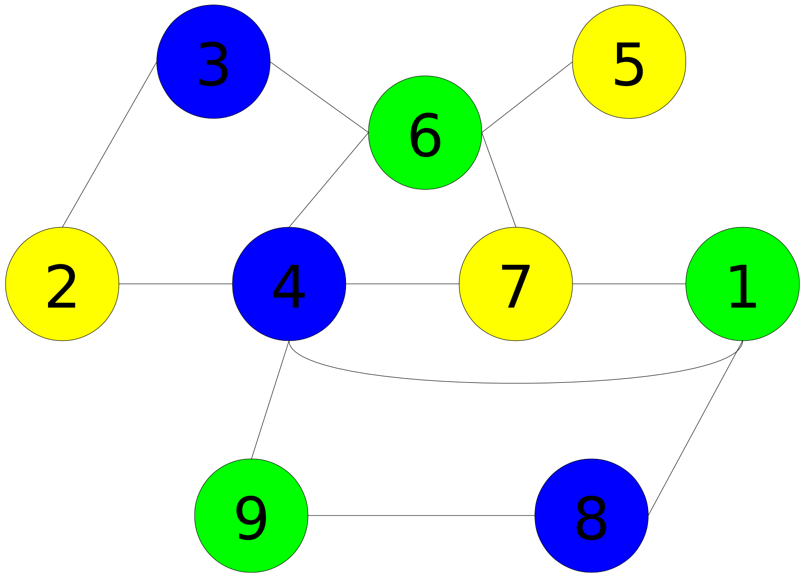
Why (Δ+1)-coloring?



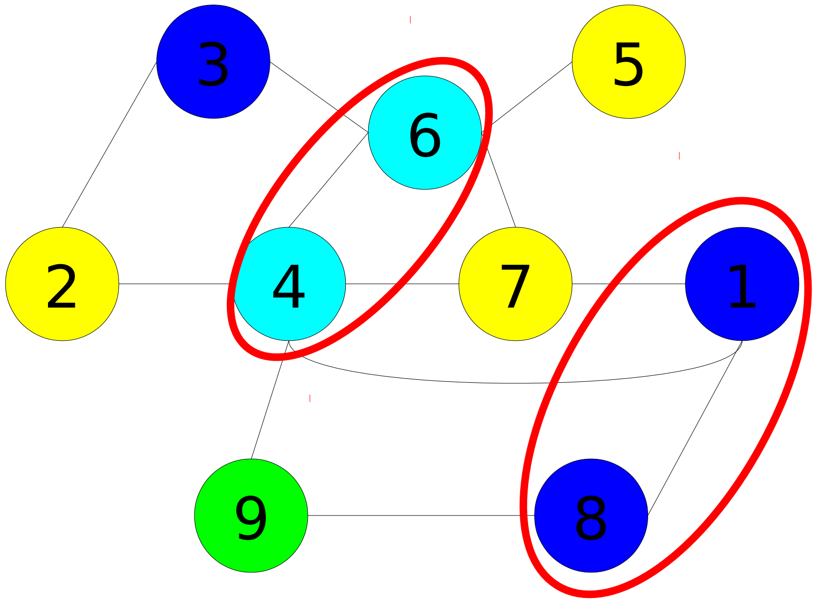
log*

- $\log^i(n) = \log(\log^{i-1}(n))$
- $\log^*(n) = \min\{i \mid \log^i(n) < 2\}$

m-defective Coloring



m-defective Coloring



Previous Work

- Kuhn, Wattenhofer
  deterministic: $O(\Delta \cdot \log \Delta + \log^* n)$
  randomized: $O(\Delta \cdot \log \log n)$

- New deterministic
  $O(\Delta) + \frac{1}{2} \log^* n$

Algorithms used in the paper

- Szegedy Vishwanatan-algorithm
  - Input: Graph G
  - Output: valid $O(\Delta^2)$-coloring
  - Running Time: ½log*n +O(1)
- Kuhn Wattenhofer-iteration
  - Input: valid m-coloring
  - Output: valid $(\Delta+1)$-coloring
  - Running Time: $O(\Delta \log(m/\Delta))$

Idea of the algorithm

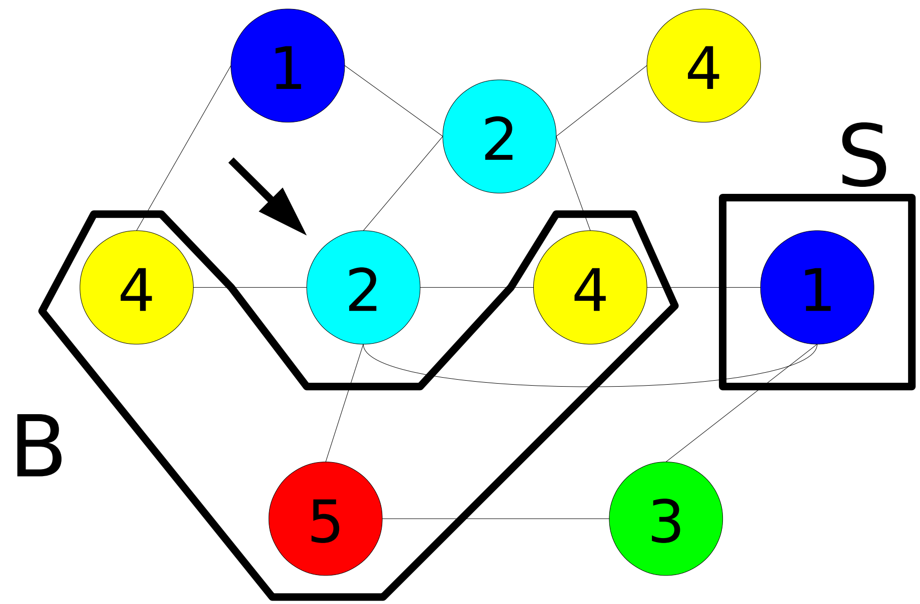
- Divide Graph into subgraphs
  - Procedure Defective Color
- Color each subgraph
- Merge colorings of subgraphs

Procedure Refine

- Input
  - m-defective c-coloring
  - Parameter p, $1 \leq p \leq \Delta$
- Output
  - $(m+\lfloor \Delta/p \rfloor)$-defective p^2-coloring
- Running Time
  - $O(c)$

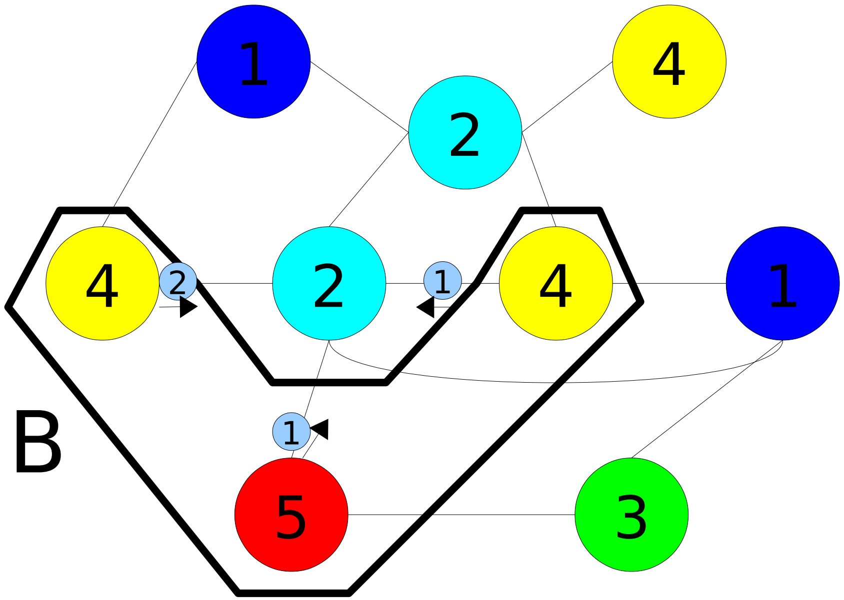
Procedure Refine

- S(v): all neighbors of v with "smaller" color
- B(v): all neighbors of v with "bigger" color



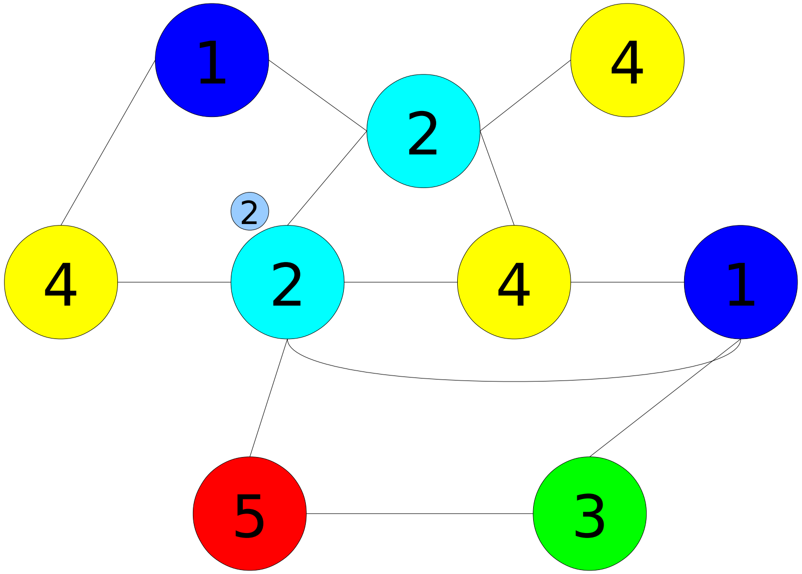
Procedure Refine

- Each vertex:
  - If v has no vertices in B(v) choose number b∊{1..p} at random and send it to all neighbors
  - Else wait until received b from all neighbors in B



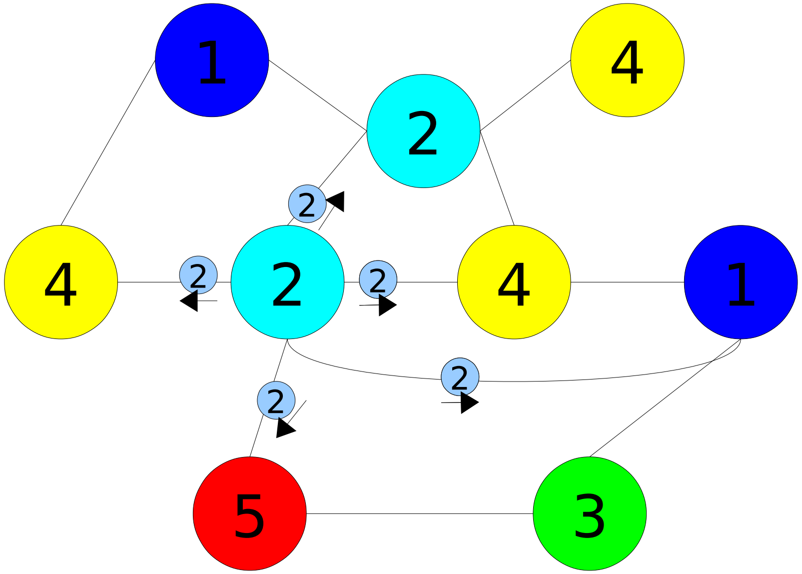
Procedure Refine

- Each vertex:
  - If received all numbers b from neighbors in B
  - Choose number b∊{1..p}, which is has least occurrence in all of the received b



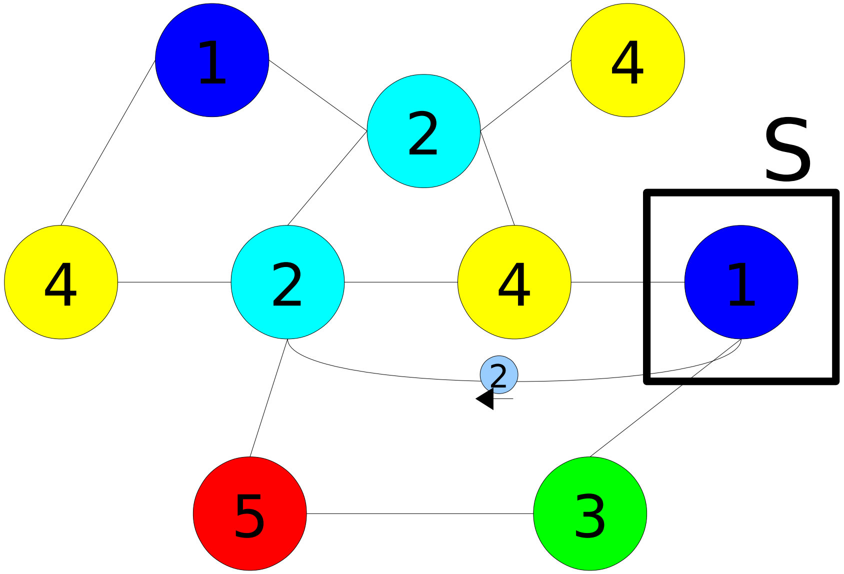
Procedure Refine

- Each vertex:
  - If received all numbers b from neighbors in B
  - Choose number b∊{1..p}, which is has least occurrence in all of the received b
  - Send b to all neighbors



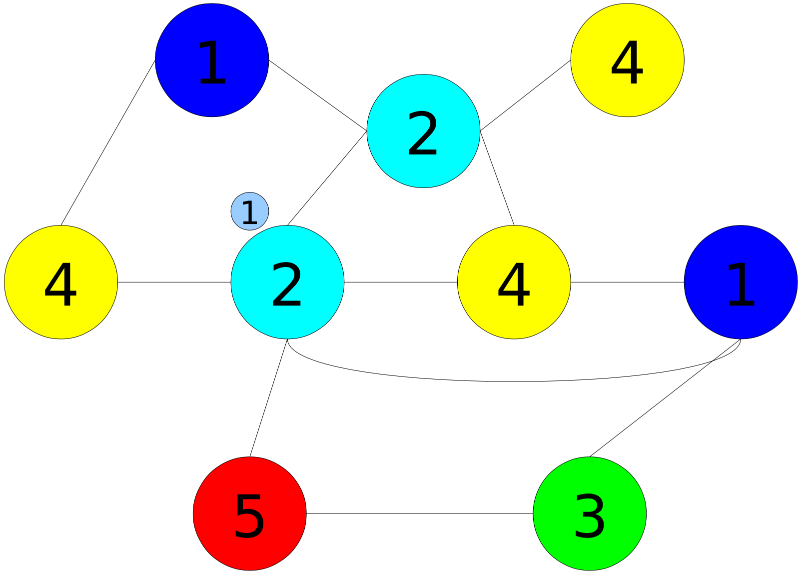
Procedure Refine

- Each vertex:
  - If v has no vertices in S(v) choose number s∊{1..p} at random and send it to all neighbors
  - Else wait until received s from all neighbors in S



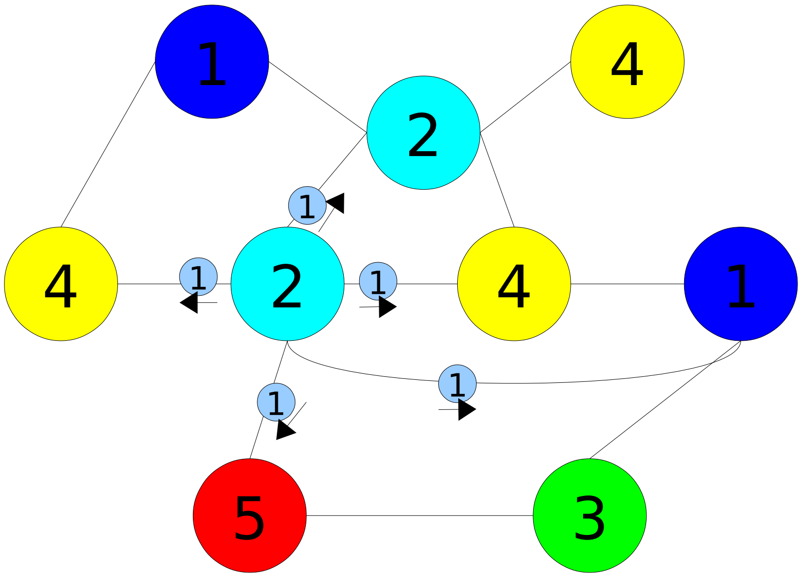
Procedure Refine

- Each vertex:
  - If received all numbers s from neighbors in S
  - Choose number s϶{1..p}, which is has least occurrence in all of the received s



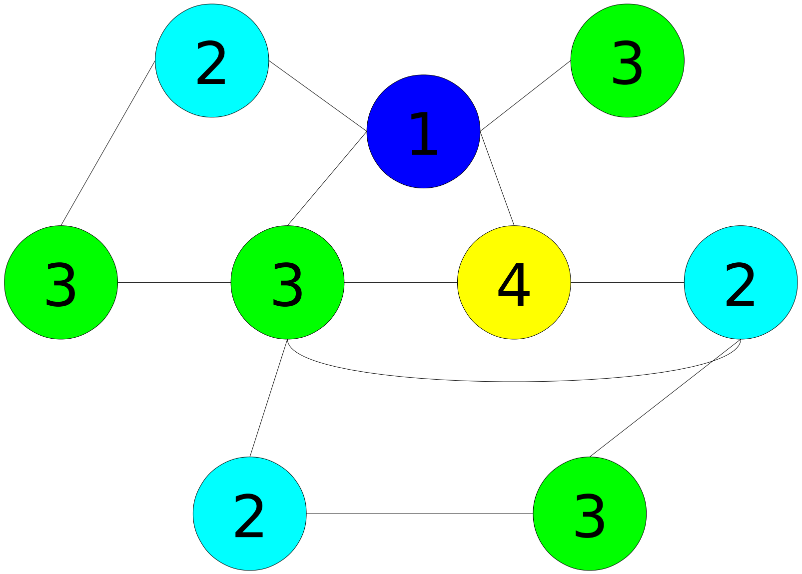
Procedure Refine

- Each vertex:
  - If received all numbers s from neighbors in S
  - Chose number s∈{1..p}, which is has least occurrence in all of the received s
  - Send s to all neighbors



Procedure Refine

- Each vertex:
  - Final color: (b-1)·p+s



Procedure Refine

- Input
  - m-defective c-coloring
  - Parameter p, $1 \le p \le \Delta$
- Output
  - $(m+\lfloor \Delta/p \rfloor)$-defective p^2-coloring
- Running Time
  - O(c)

Adriana Lima

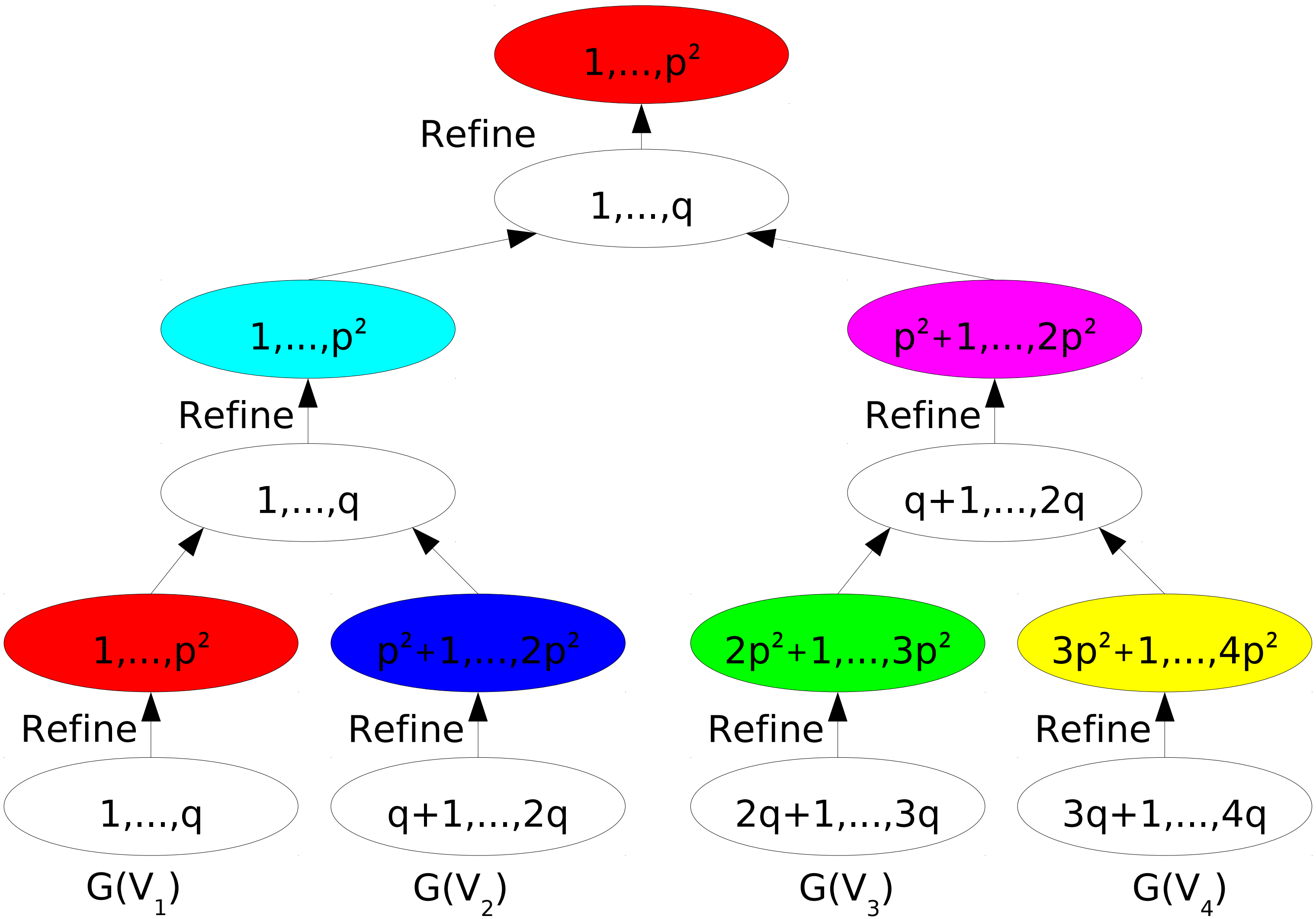
Procedure Defective Color

- Input
  - Graph G
  - Parameter p, $1 \leq p \leq \Delta$
  - Parameter q, $p^2 < q$
- Output
  - $O(\log\Delta/\log(q/p^2) \cdot (\Delta/p))$ defective p^2-coloring of G
- Running Time
  - $O(\log^* n + \log\Delta/\log(q/p^2) \cdot q)$

Procedure Defective Color

- Compute initial $O(\Delta^2)$-coloring
  #colors $c=d\cdot\Delta^2$

Procedure Defective Color



Procedure Defective Color

- # Iterations:
  - $\log d \cdot \Delta^2 / \log(q/p^2)$
- Procedure Refine
  - Running time $O(q)$
  - Δ/p-defective

Procedure Defective Color

- Input
  - Graph G
  - Parameter p, $1 \le p \le \Delta$
  - Parameter q, $p^2 < q$
- Output
  - $O(\log\Delta/\log(q/p^2) \cdot (\Delta/p))$ defective p^2-coloring of G
- Running Time
  - $O(\log^* n + \log\Delta/\log(q/p^2) \cdot q)$

The first algorithm

- Run Defective Color
  - $p = \log\Delta$
  - $q = \Delta^{\varepsilon}$
- $O(\Delta/\log\Delta)$-defective $(\log\Delta)^2$-coloring
- Create subgraphs Vj for each color $j \in \{1..\lfloor(\log\Delta)^2\rfloor\}$
- $\Delta j = O(\Delta/\log\Delta)$
- Run KW-algorithm on each subgraph with $O(\Delta/\log\Delta)$-colors
- Valid $O((\log\Delta)^2 \cdot \Delta/\log\Delta)) = O(\Delta \cdot \log\Delta)$-coloring
- Run KW-iteration

The first algorithm (Runtime)

- Defective Color: $O(\Delta^{\varepsilon})+½ \log^* n$
- KW-algorithm: $O(\Delta+\log^* n)$
- KW-iteration: $O(\Delta \cdot \log \log \Delta)$

- Total:

  $O(\Delta \cdot \log \log \Delta + \log^* n)$

Recursive Algorithm

- Assume that algorithm A_k computes $(\Delta+1)$-coloring

  Running time: $O(\Delta \log^{(k)}\Delta) + k/2 \log^* n$
- Algorithm A_{k+1}
  - Defective Color
    - $p=\log^k\Delta$
    - $q=\Delta^\varepsilon$
  - Run A_k on all subgraphs
  - Run KW-iteration

Recursive Algorithm (Runtime)

- Defective Color: $O(\Delta^{\varepsilon}) + ½ \log^* n$
- A_k-algorithm: $O(\Delta) + k/2 \log^* n$
- KW-iteration: $O(\Delta \log^{(k+1)} \Delta)$

- Total:

  $O(\Delta \log^{(k+1)} \Delta) + (k+1)/2 \log^* n$

Recursive Algorithm

- A_k-algorithm:

  $O(\Delta \log^{(k)}\Delta + \log^* n)$

- $A_{\log^*\Delta}$-algorithm:

  $O(\Delta + \log^*\Delta \cdot \log^* n)$

Final improvements

- Each iteration calls Defective Color
- Defective Color invokes SV-algorithm
- SV needs ½log*n time

Final algorithm

- Final version:
  $O(\Delta) + ½ \log^* n$

- Trade off version
  $O(\Delta^* t)$-coloring in $O(\Delta / t) + ½ \log^* n$ time

Conclusion

- Improved (Δ+1)-coloring algorithm
- Using defective coloring

- Further Research
  - Improve p-defective q-coloring

Q & A

Theorem by Erdös, Frankl, Füredi

For every positive integer A, there exists a collection T of $\Theta(A^3)$ subsets of $\{1,2,..,cc \cdot A^2\}$ such that for every A+1 subsets

$T_0, T_1, \ldots, T_A \in T$, $$T_0 \not\subseteq \bigcup_{i=1}^{A} (T_i)$$

Procedure Defective Color

- Compute initial coloring φ, #colors $c=d \cdot \Delta^2$
- while $c>p^2$ for each vertex v
  - $j = \min\{\lceil \varphi(v)/q \rceil, \lfloor c/q \rfloor\}$
  - Vertex v joins set V_j
  - $\tau_j(v)$= Invoke Refine on $G(V_j)$
  - $\varphi(v) = \tau_j(v)+(j-1) \cdot p^2$
  - $c = \lfloor c/q \rfloor \cdot p^2$
- end while
- Return φ