

DISTRIBUTED RECOLORING

Marthe Bonamy, Paul Ouvrard, Mikaël Rabie, Jukka Suomela, Jara Uitto

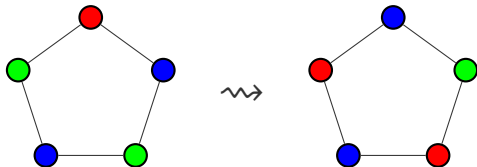
March 28, 2018

ANR DESCARTES meeting

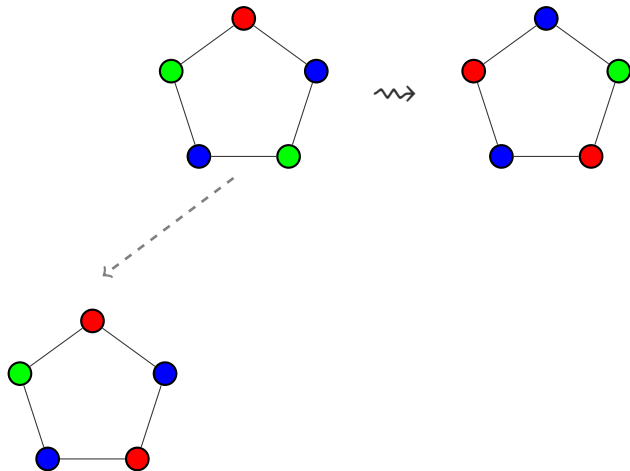


université
de **BORDEAUX**

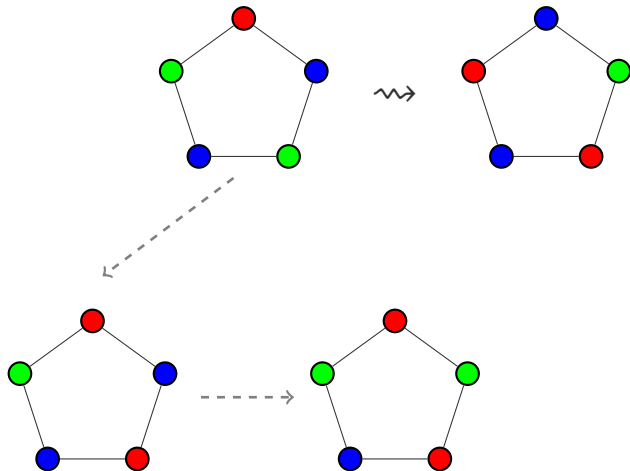
Recoloring



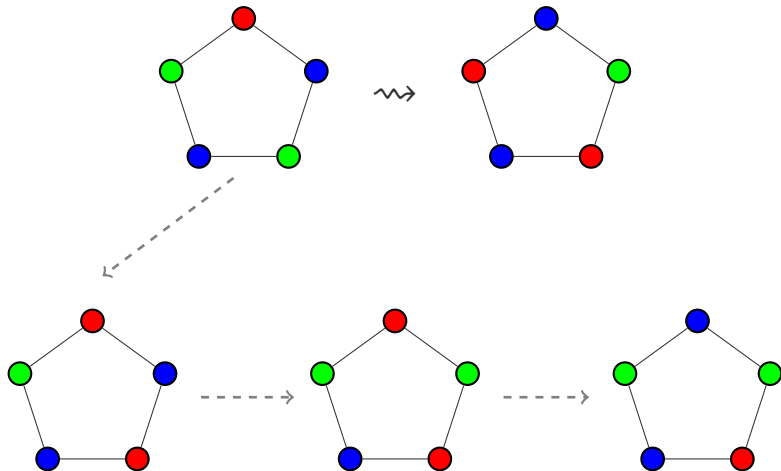
Recoloring



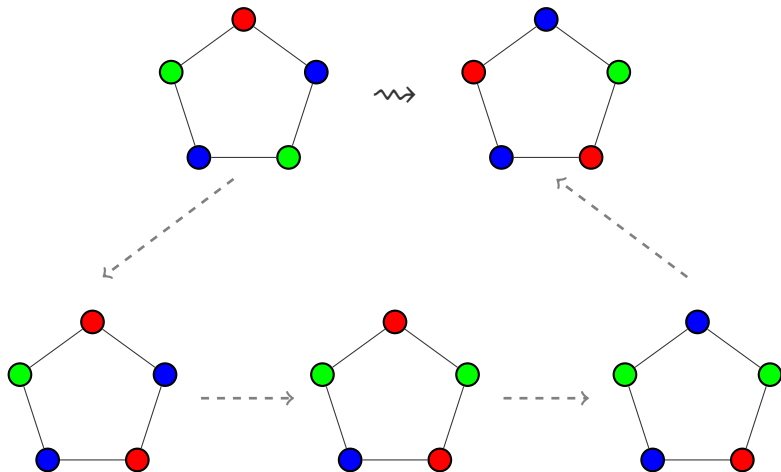
Recoloring



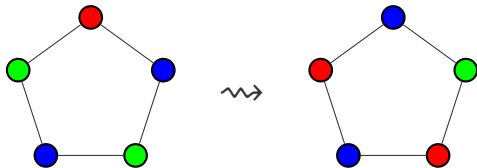
Recoloring



Recoloring



Recoloring

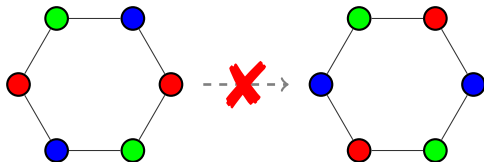


Distributed recoloring

- We use the LOCAL model;
 - works in synchronous rounds;
 - each node knows only itself;
 - at each round, a node can:
 - send (and receive) a message (unlimited size) to its neighbors;
 - do some local computation.
- at each step, we can recolor a stable set, not only one vertex;

Distributed recoloring

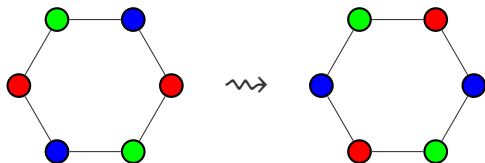
- Recoloring is not always possible:



- \Rightarrow we allow extra colors.

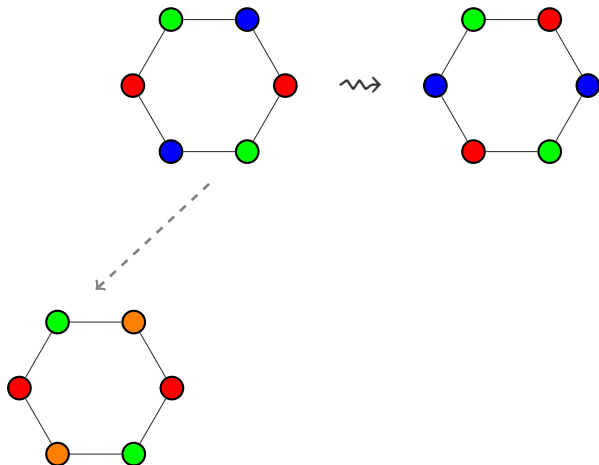
Distributed recoloring with extra colors

- Extra color : ●



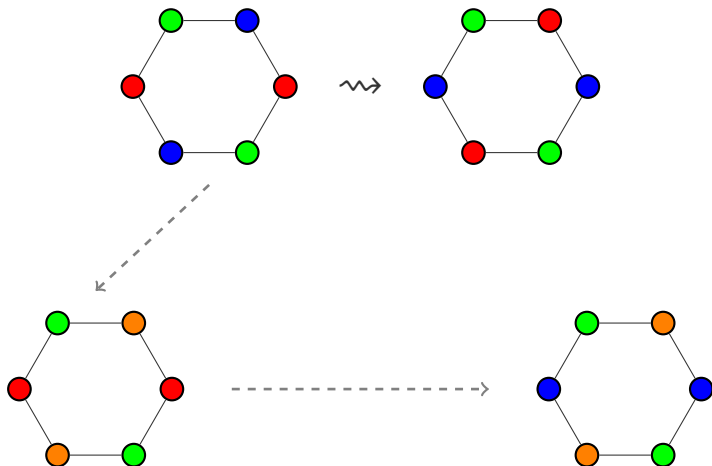
Distributed recoloring with extra colors

○ Extra color : ●



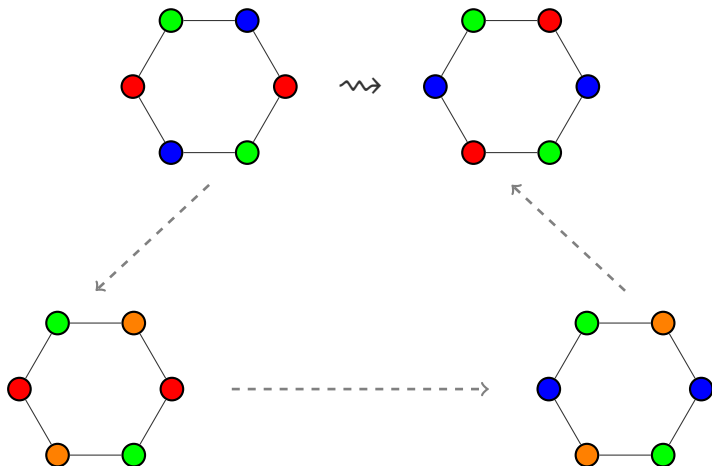
Distributed recoloring with extra colors

○ Extra color : ●



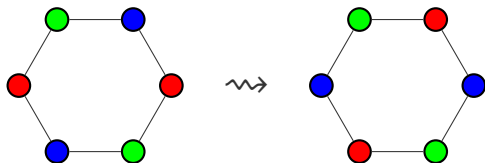
Distributed recoloring with extra colors

○ Extra color : ●



Distributed recoloring with extra colors

○ Extra color : ●



Problem definition

Notation

For a vertex u , we denote by $c_i(u)$ the color of u at step i .

Distributed recoloring

Input : a graph $G = (V, E)$, two k -colorings α and β of G , c extra colors.

Output : for each node $u \in V$ a *recoloring schedule*:

$\langle c_0(u) = \alpha(u), c_1(u), \dots, c_i(u), \dots, c_{\ell(u)-1}(u), c_{\ell(u)}(u) = \beta(u) \rangle$.

Length of the solution (schedule)

$$L = \max_{v \in V} \ell(v)$$

Goal

Minimize L and the number of rounds.

Recoloring trees without extra color

Input : a tree T , α and β two 3-colorings of T .

We want : to recolor T from α to β without extra color.

Theorem [Bonamy, O., Rabie, Suomela, Uitto '18]

For every tree T with radius at most p and for any two 3-colorings α, β of T , we can compute in $O(p)$ rounds how to 3-recolor T from α to β with a schedule of length $O(p)$.

Recoloring trees without extra color

Key idea

- Recolor α and β into a given 2-coloring c' of T ;
- since each recoloring is reversible, we can recolor α into β as follows: $\alpha \rightsquigarrow c' \rightsquigarrow \beta$.

How can we recolor a 3-coloring into c' ?

- Root T on a node r at distance $\leq p$ of any node of T ;
- in $O(\log p)$ rounds, r knows the full tree;
- r computes a schedule of length $O(p)$ and sends it to every node.

Recoloring trees without extra color

Existence of a schedule of length $O(p)$

Identification operation

If u is a leaf and v its grandparent, we recolor u with $c(v)$ and pretend that u and v are a single node i.e. we remove u and reflect any recoloring of v to u (possible since they are not adjacent).

At each step:

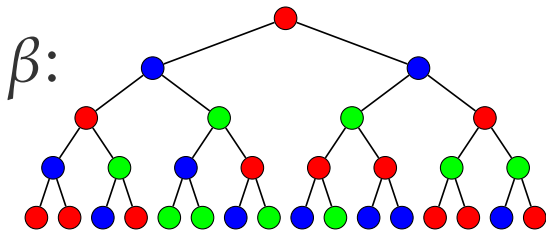
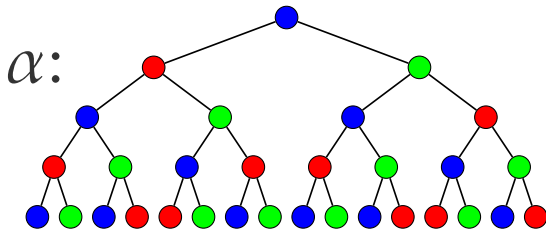
- A : set of leaves with a grandparent;
- for every leaf $u \in A$, identify u with its grandparent.

Recoloring trees without extra color

- At the end of this process, T consists only of the root r and its children;
- select (arbitrarily) a children and identify the others with it:
- T is only an edge e : if necessary, recolor e into c' (possible since we have 3 colors and only two vertices).

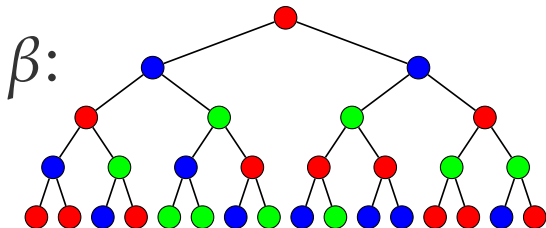
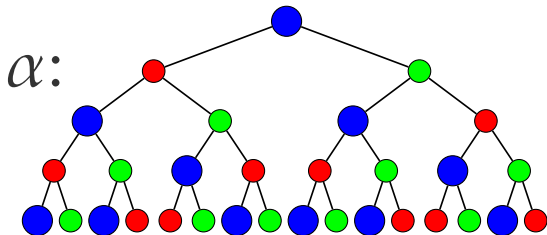
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



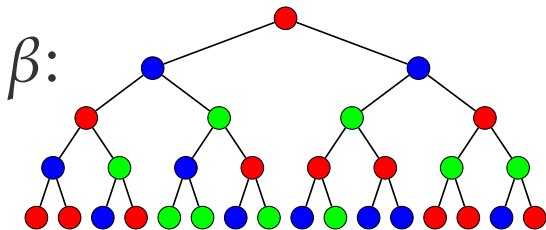
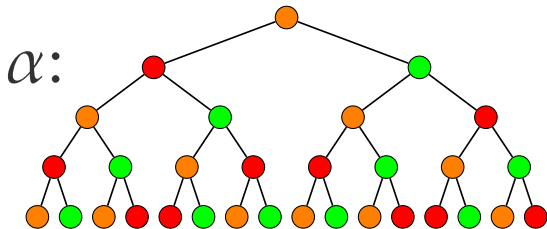
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



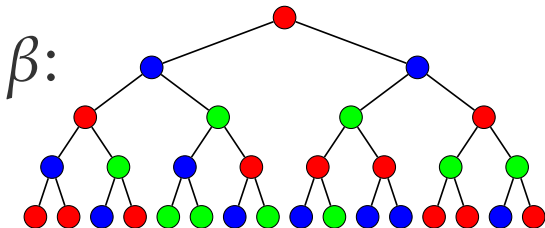
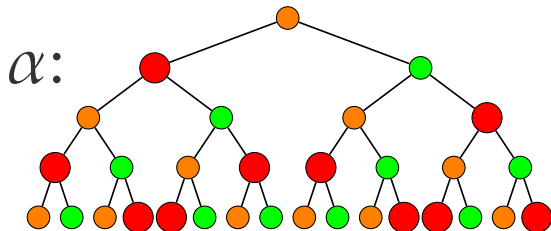
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



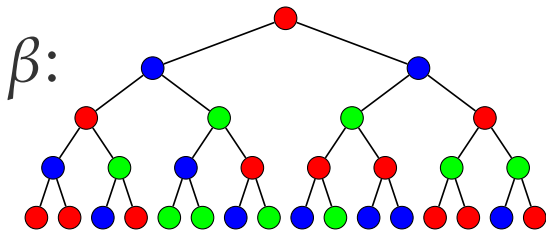
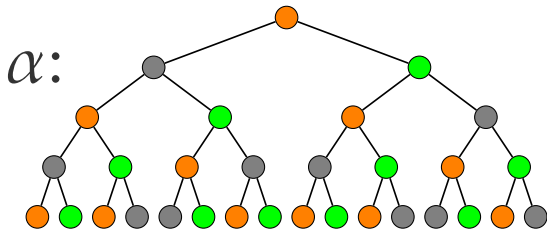
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



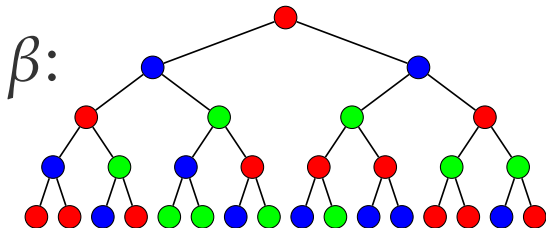
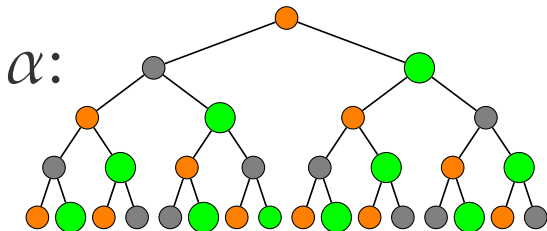
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



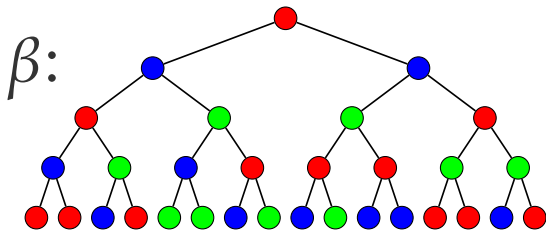
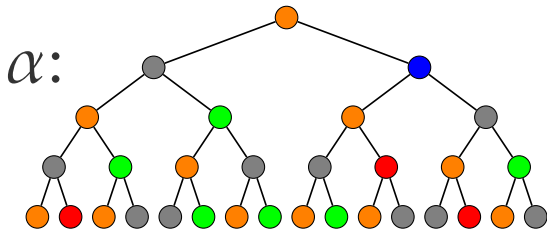
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



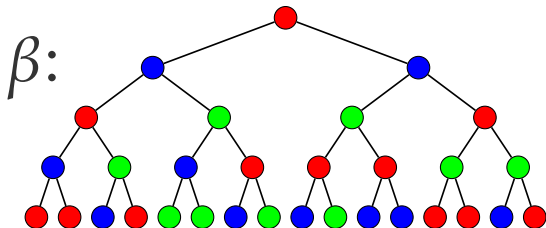
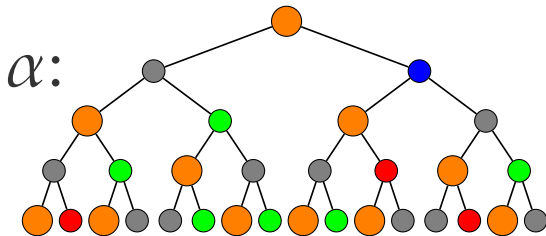
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



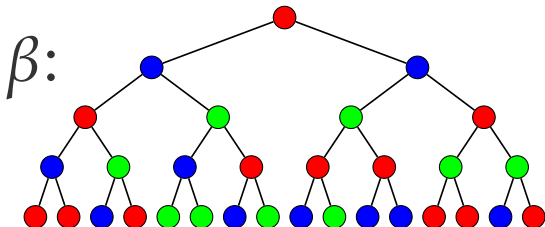
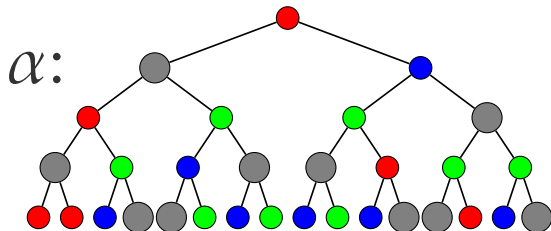
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



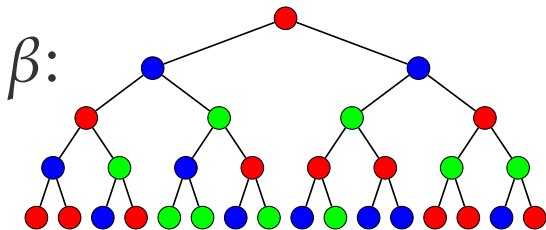
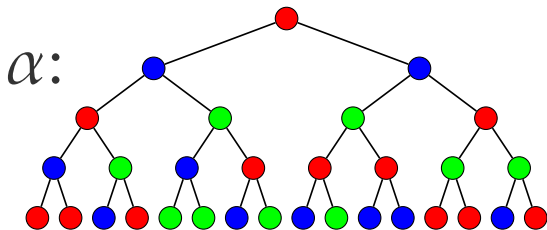
Recoloring trees with 2 extra colors

Two extra colors: ●, ●



Recoloring trees with 2 extra colors

Two extra colors: ●, ●



Recoloring trees with one extra color

Recap

We have seen:

- $3+0$: in $O(n)$ rounds and schedule $O(n)$;
- $3+2$: no communication needed and constant schedule;
- ... but what about $3+1$ extra color?

Recoloring trees with one extra color

Recap

We have seen:

- $3+0$: in $O(n)$ rounds and schedule $O(n)$;
- $3+2$: no communication needed and constant schedule;
- ... but what about $3+1$ extra color?

Theorem [Bonamy, O., Rabie, Suomela, Uitto]

For any $k \in \mathbb{N}$, for every tree T on n nodes, for any two k -colorings α, β of T , we can compute in $O(\log n)$ rounds how to recolor T from α to β with 1 extra color and a schedule of length $O(1)$.

Key lemma:

Tree decomposition

There is an $O(\log n)$ -round algorithm that finds an MIS in a tree, such that every component induced by non-MIS nodes is of size one or two.

light h -labeling

Assign to each node a label between 1 and h such that for any $1 \leq i \leq h$ we have:

- Any node labeled i has at most two neighbors with label $\geq i$, at most one of which with label $\geq i + 1$.
- No two adjacent nodes labeled i both have a neighbor with label $\geq i + 1$.

Light $(2 \log n)$ -labeling for trees

There is an $O(\log n)$ -round algorithm that finds a light $(2 \log n)$ -labeling of a tree.

Proof (sketch):

- Adaptation of the *rake and compress* method by Reif and Miller;
- at step i , remove all nodes of degree 1 and all nodes of degree 2 that belong to a chain of at least three nodes of degree 2 and assign them label i .
- at every step, we remove at least a sixth of the nodes;
- the algorithm ends after $2 \log n$ steps.

Computing the MIS

- Compute a 3-coloring of T (can be done in $O(\log n)$ rounds);
- compute a light $O(\log n)$ labeling;
- Let $S = \emptyset$
- consider labels decreasingly:
 - if a node u labeled i has a neighbor of label $\geq i + 1$ not in S , add u to S ;
 - complete S with the 3-coloring.
- by construction, S is a MIS.

Computing the MIS

All the nodes of a connected component of $T \setminus S$ have the same label (due to the first operation).

What about the size of these connected components ?

- C : connected component of size ≥ 3 ;
- uvw : a path in C
- $\Rightarrow u, v$ and w are labeled i
- u has no other neighbor of label $\geq i$
- \Rightarrow contradiction.

Recoloring algorithm

- Compute the MIS S ;
- recolor S with the extra color;
- remove S and recolor each connected component from α to β (without the extra color);
- recolor each node in S to its final color.

Conclusion

For trees:

- $3+0$: $O(n)$ rounds, schedule $O(n)$;
- $3+1$: $O(\log n)$ rounds, schedule $O(1)$;
- $3+2$ no communication, schedule $O(1)$.

Some results for:

- subcubic graphs;
- toroidal grids;
- list coloring (on trees and subcubic graphs);

Conclusion

For trees:

- $3+0$: $O(n)$ rounds, schedule $O(n)$;
- $3+1$: $O(\log n)$ rounds, schedule $O(1)$;
- $3+2$ no communication, schedule $O(1)$.

Some results for:

- subcubic graphs;
- toroidal grids;
- list coloring (on trees and subcubic graphs);

Thank you!