Colorful Vertex Recoloring of Bipartite Graphs

Boaz Patt-Shamir
Tel Aviv U.

Adi Rosén
CNRS & U. Paris Cité

William Umboh
U. Melbourne

Sometimes, living in the same place is hard

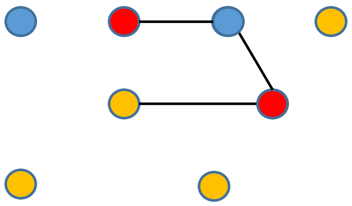
Also in computer systems.

Jobs (VMs) may need to be separated due to…

- Conflicting resource requirements
- Security concerns
- Performance considerations

- “Anti-affinity” rules

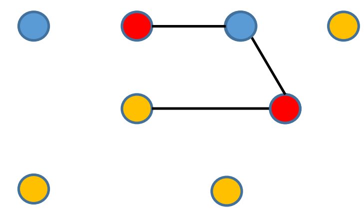
Abstract Model: Online Recoloring



- Start with colored vertices
  - Vertex = job, color = location
- In each step, a new edge arrives
  - Edge = conflict
- The algorithm must maintain legal vertex coloring at all times
  - If the new edge is monochromatic, the algorithm must recolor (at least) one of its endpoints
  - i.e., migrate at least one of the conflicting jobs

Cost = total number of recolorings

Measure



- Aim: minimize **competitive ratio**

$$CR = \max\left\{\frac{\mathrm{cost}\big(\mathrm{ALG}(\sigma)\big)}{\mathrm{cost}\big(\mathrm{OPT}(\sigma)\big)} : \mathrm{input}\ \sigma\right\}$$

- Note: $\mathrm{cost}\big(\mathrm{OPT}(\sigma)\big) \leq n$ for all σ
  - Offline solution recolors each vertex at most once

Previous work

- Online coloring (no recoloring)
  - LST’89, HS’94: $\Theta\left(\frac{n}{\log n}\right)$ competitiveness
- Coloring with recourse (forget competitiveness)
  - Dynamic algorithms: absolute cost (#steps), regardless of optimum cost [BCKLRRV’19, BDFPZ’20]
  - Special graphs: Bipartite, bounded-degree, bounded arboricity, interval graphs [KNNP’19, BDFPZ’20, HNW20]
  - General graphs, using an (NP-hard) oracle for coloring: tradeoff between # colors and # recolorings [BCKLRRV’19,SW’19]

Previous work [AMPT'22]

- The competitive ratio of recoloring a bipartite graph with 2 colors is $\Theta(\log n)$.
- The deterministic competitive ratio of $(\Delta + 1)$-recoloring is $\Theta(\Delta)$, and $\Theta(\log \Delta)$ for randomized algorithms.

---

* Δ is the largest degree in the graph

Our Model

Allow the online algorithm to use more colors than the offline adversary (resource augmentation)

- Hence “colorful”

Additional colors may be more expensive (**weighted** resource augmentation)

A natural extension for the motivating scenario

Our Results

Focus on bipartite graphs (2 colors for the adversary)

1. Uniform cost: given c colors, c.r = $O\left(\frac{\log n}{c}\right)$
2. New Δ colors at cost D: c.r. = $O(\log D + \beta^2)$
   - $\beta \leq c \leq \log n$, where β is the **bond** of the graph

This talk

Lower Bounds:

1. Competitiveness is $\Omega(\min\{\log n, D\})$ if additional colors cost D, even if randomization is allowed.
2. Even for a collection of paths, competitiveness is $\Omega(\log D)$, if additional colors cost D

The **Bond** β

- A measure of "tree-ness"

Definition: the maximal number of edges that can be removed so that the number of connected components increases by 1

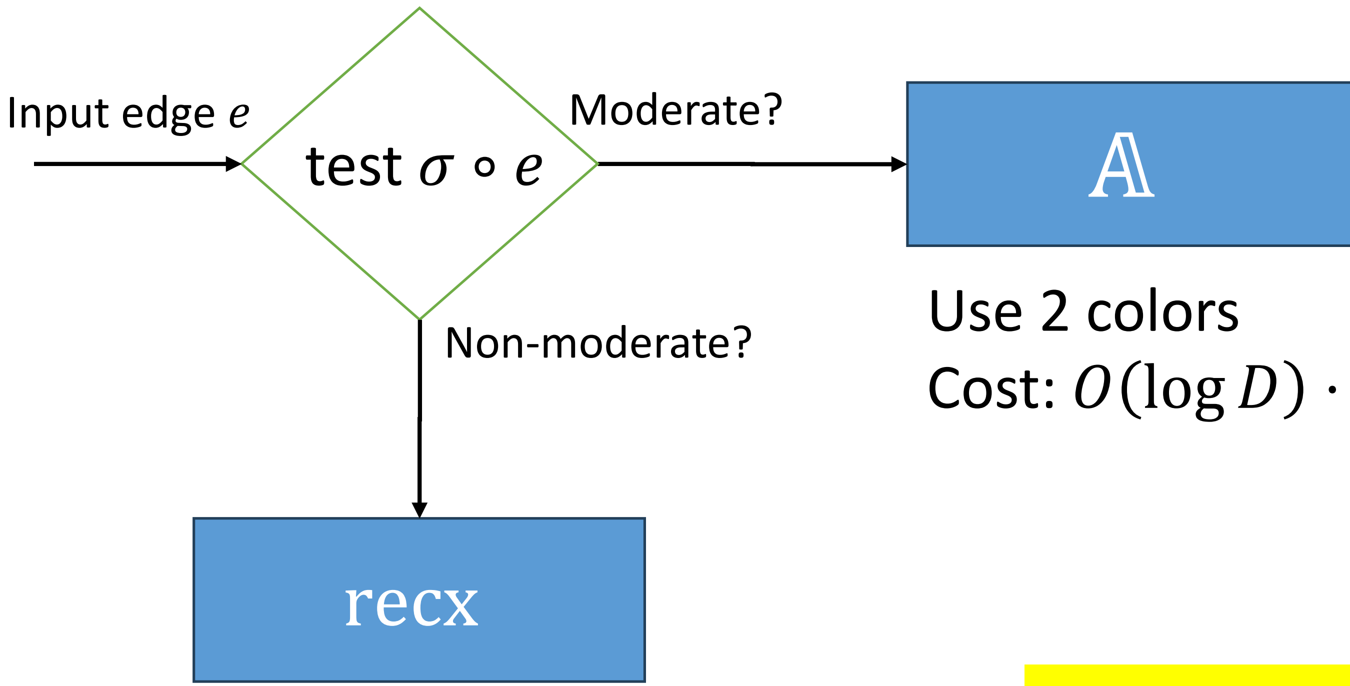
  - Not min cut, not max cut
  - The size of the largest minim**al** cut

- $\beta = 1$ if and only if the graph is a forest

Algorithm $\mathbb{B}$: General Idea

- Recall properties of Algorithm $\mathbb{A}$ [AMPT'22]:
  - Number of vertices recolored $\leq 3\ \mathrm{OPT}$
  - A vertex recolored $\Longrightarrow$ component size grows by a constant factor (roughly)
- We show: If input is "D-moderate" then $\mathbb{A}$ is $O(\log D)$-competitive
- Edge sequence is D-moderate if no edge connects two components of size more than D with a majority of recolored nodes

Algorithm **B**: General Idea (cont.)



Use 2 colors
Cost: $O(\log D) \cdot \mathrm{OPT}$

Use special colors
Cost: $O(\beta^2) \cdot \mathrm{OPT}$

σ is the sequence of edges previously fed to $\mathbb{A}$

Algorithm **B**: Some Details

- Simulation of $\mathbb{A}$ is unaware of the special colors
- To prove that $\mathbb{A}$ is $O(D)$-competitive on D-moderate sequences we use a non-trivial charging scheme
- **`recx`** uses only special colors
  - $\Delta + 1$ is easy, can reduce to Δ
- The number of edges sent to **`recx`** is $O(\beta \cdot \mathrm{OPT}/D)$
  - Because each such edge connects 2 large components with many vertices recolored by $\mathbb{A}$

Another idea: Algorithm C

- Assign a level to each node. Initially, all 1
- Maintain a simulation of $\mathbb{A}_i$ for each level i
  - Each instance uses a distinct pair of colors
- Edges with both endpoints at level i are sent to $\mathbb{A}_i$
  - Edges that connect different levels are okay!
- If new edge makes input to $\mathbb{A}_i$ not X-moderate, increase level of one of its endpoints to $i+1$
  - If input to $\mathbb{A}_{i+1}$ moderate, recolor neighbors of newcomer
  - Otherwise, increase level to $i+2$ and repeat.

Algorithm **C**: Some Details

If new edge makes input to $\mathbb{A}_i$ not X-moderate, increase level of one of its endpoints to $i+1$

If input to $\mathbb{A}_{i+1}$ moderate, recolor neighbors of newcomer

Otherwise, increase level to $i+2$ and repeat.

- X-moderate for $X = \max\{2^{1/\epsilon}, \beta^2\}$
- We show that:
  - Each $\mathbb{A}_i$ is $\log X$-competitive, i.e., $\left(\log \beta + \frac{1}{\epsilon}\right)$-comp.
  - $OPT_i = O\left(\frac{\beta^2}{X} OPT_{i-1}\right)$
- Hence the number of levels is $O(\epsilon^{-1} \log n)$

Conclusion

- We have extended recoloring of bipartite graphs to the resource augmentation setting
- We have introduced and analyzed the weighted resource augmentation model
- We have shown that the concept of bond is useful

Thanks!