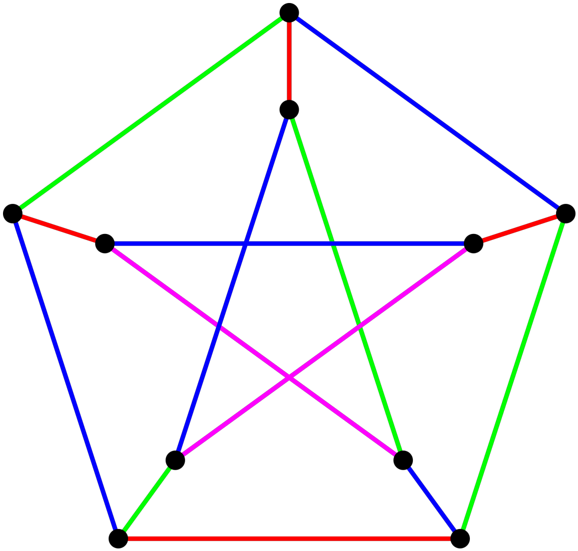
Online Max-Edge-Coloring of Paths and Trees

Lene M. Favrholdt and Jesper W. Mikkelsen

Department of Mathematics and Computer Science
University of Southern Denmark

TOLA, July 7, 2014

Edge Coloring



An edge coloring of the Petersen graph using 4 colors.

Minimum Edge Coloring

Classical edge coloring:

- Color the edges of a graph using as few colors as possible.

Minimum Edge Coloring

Classical edge coloring:

- Color the edges of a graph using as few colors as possible.

Vizing's Theorem

Let G be a simple graph of maximum degree $\Delta(G)$. The minimum number of colors needed to color all edges of G is either $\Delta(G)$ or $\Delta(G) + 1$.

Dual Edge Coloring

There is a *dual* version known as Edge-k-Coloring:

Dual Edge Coloring

There is a *dual* version known as Edge-k-Coloring:

- A fixed number, k, of colors is available.
- The goal is to color as many edges as possible.

Dual Edge Coloring

There is a *dual* version known as Edge-k-Coloring:

- A fixed number, k, of colors is available.
- The goal is to color as many edges as possible.

We label the k colors $1, 2, \ldots, k$.

Dual Edge Coloring

There is a *dual* version known as Edge-k-Coloring:

- A fixed number, k, of colors is available.
- The goal is to color as many edges as possible.

We label the k colors $1, 2, \ldots, k$.
For $k = 1$, this is the maximum matching problem.

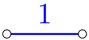
Online Edge Coloring

Online Edge-k-Coloring

- Edges arrive one by one.
- Must immediately color a newly arrived edge with one of the k colors or reject the edge.
- The decision is *irrevocable*.

Example for $k = 2$

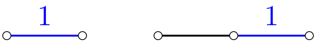
Example for $k = 2$



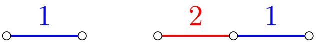
Example for $k = 2$

Example for $k = 2$

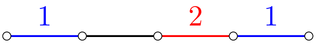
Example for $k = 2$



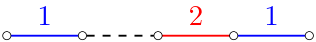
Example for $k = 2$



Example for $k = 2$



Example for $k = 2$



Online Edge Coloring

Competitive analysis [Sleator, Tarjan '85], [Karlin et al. '88]

An algorithm A is c-*competitive* if

$$\mathsf{A}(\sigma) \geq c \cdot \mathsf{OPT}(\sigma) - b$$

for all sequence of edges σ.
For a randomized algorithm, replace $\mathsf{A}(\sigma)$ with $E[\mathsf{A}(\sigma)]$.
The competitive ratio $C = \sup\{c : \mathsf{A} \text{ is } c\text{-competitive}\}$.

Online Edge Coloring

Competitive analysis [Sleator, Tarjan '85], [Karlin et al. '88]

An algorithm A is c-*competitive* if

$$\mathsf{A}(\sigma) \geq c \cdot \mathsf{OPT}(\sigma) - b$$

for all sequence of edges σ.
For a randomized algorithm, replace $\mathsf{A}(\sigma)$ with $E[\mathsf{A}(\sigma)]$.
The competitive ratio $C = \sup\{c : \mathsf{A} \text{ is } c\text{-competitive}\}$.

Previous results

(Favrholdt, Nielsen '03)

- Negative results:

No deterministic algorithm has a competitive ratio better than $\frac{1}{2}$.
No randomized algorithm has a competitive ratio better than $\frac{4}{7}$.

Previous results

(Favrholdt, Nielsen '03)

- Negative results:

No deterministic algorithm has a competitive ratio better than $\frac{1}{2}$.
No randomized algorithm has a competitive ratio better than $\frac{4}{7}$.

- Positive results:

The competitive ratio of a *fair* algorithm is at least $2\sqrt{3} - 3 \approx 0.46$
An algorithm is *fair* if it never rejects an edge unless forced to do so.

What?

- In order to obtain a more fine-grained analysis, we study Edge-k-Coloring on some basic graph classes:
- For paths, we give an optimal (randomized) algorithm.
- For trees, we show that a natural algorithm called First-Fit is optimal among deterministic algorithms.
- For trees and "tree-like" graphs, we show that any fair algorithm for online Edge-k-Coloring performs well if k (the number of colors) is sufficiently large.

Why?

Why paths and trees?

Why?

Why paths and trees?

- Natural building blocks for studying more complicated graph classes.

Why?

Why paths and trees?

- Natural building blocks for studying more complicated graph classes.
- All previous (negative) results for Edge-k-Coloring holds when the input graph is a bipartite graph.

Algorithms

Recall that the competitive ratio of a fair algorithm is at least $2\sqrt{3} - 3 \approx 0.46$ and at most $\frac{1}{2}$ (Favrholdt, Nielsen '03).
The following fair and deterministic algorithms have been studied:

- First-Fit uses the lowest available color when coloring an edge. It can be viewed as the natural greedy strategy.

Algorithms

Recall that the competitive ratio of a fair algorithm is at least $2\sqrt{3} - 3 \approx 0.46$ and at most $\frac{1}{2}$ (Favrholdt, Nielsen '03).
The following fair and deterministic algorithms have been studied:

- First-Fit uses the lowest available color when coloring an edge. It can be viewed as the natural greedy strategy.
- Next-Fit remembers the last used color c_{last}. When coloring an edge, it uses the first available color in the ordered sequence $\langle c_{\text{last}} + 1, \ldots, k, 1, \ldots, c_{\text{last}} \rangle$.

Algorithms

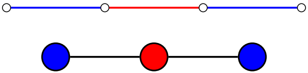
Recall that the competitive ratio of a fair algorithm is at least $2\sqrt{3} - 3 \approx 0.46$ and at most $\frac{1}{2}$ (Favrholdt, Nielsen '03).
The following fair and deterministic algorithms have been studied:

- First-Fit uses the lowest available color when coloring an edge. It can be viewed as the natural greedy strategy.
- Next-Fit remembers the last used color c_{last}. When coloring an edge, it uses the first available color in the ordered sequence $\langle c_{\text{last}} + 1, \ldots, k, 1, \ldots, c_{\text{last}} \rangle$.

Next-Fit is shown to have a competitive ratio of exactly $2\sqrt{3} - 3$.
The competitive ratio of First-Fit is shown to be at most 0.48.

Relationship to vertex coloring

- Edge coloring a graph G is equivalent to vertex coloring the line graph of G.
- This also holds in an online setting.
- In particular, online Edge-k-Coloring on paths is exactly the same as online *dual* vertex coloring on paths.



Edge-2-Coloring on Paths

- Next-Fit has a competitive ratio of $\frac{1}{2}$ on paths.

Edge-2-Coloring on Paths

- Next-Fit has a competitive ratio of $\frac{1}{2}$ on paths.

Edge-2-Coloring on Paths

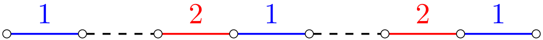
- Next-Fit has a competitive ratio of $\frac{1}{2}$ on paths.
- First-Fit has a competitive ratio of $\frac{2}{3}$ on paths.

Edge-2-Coloring on Paths

- Next-Fit has a competitive ratio of $\frac{1}{2}$ on paths.
- First-Fit has a competitive ratio of $\frac{2}{3}$ on paths.

Edge-2-Coloring on Paths

- Next-Fit has a competitive ratio of $\frac{1}{2}$ on paths.
- First-Fit has a competitive ratio of $\frac{2}{3}$ on paths.



No deterministic algorithm can do better than $\frac{2}{3}$.

Edge-2-Coloring on Paths

- Can a randomized algorithm do better than $\frac{2}{3}$?

Edge-2-Coloring on Paths

- Can a randomized algorithm do better than $\frac{2}{3}$?
- Yes! There is a randomized algorithm with a competitive ratio of $\frac{4}{5}$.

Rand$_p$

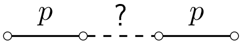
Let $\frac{1}{2} \leq p \leq 1$. Define Rand$_p$ as follows:

- For isolated edges, use the color 1 with probability p and the color 2 with probability $1 - p$. Non-isolated edges are colored if possible.

Rand$_p$

Let $\frac{1}{2} \leq p \leq 1$. Define Rand$_p$ as follows:

- For isolated edges, use the color 1 with probability p and the color 2 with probability $1 - p$. Non-isolated edges are colored if possible.
- Two types of rejections:

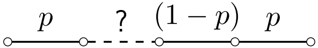


Dashed edge is colored with probability $p^2 + (1 - p)^2$.

Rand_p

Let $\frac{1}{2} \leq p \leq 1$. Define Rand_p as follows:

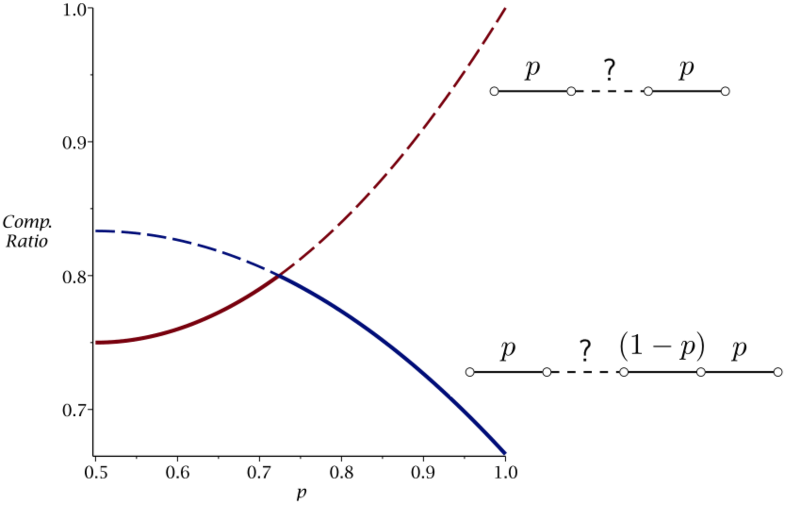
- For isolated edges, use the color 1 with probability p and the color 2 with probability $1-p$. Non-isolated edges are colored if possible.
- Two types of rejections:



Dashed edge is colored with probability $p(1-p)+(1-p)p$.

Rand$_p$

Choose the parameter p so that we balance the two situations:
$p = \frac{\varphi}{\sqrt{5}} \approx 0.72$ gives a competitive ratio of $\frac{4}{5}$.



Randomization

- Can a randomized algorithm do better than $\frac{4}{5}$?

Randomization

- Can a randomized algorithm do better than $\frac{4}{5}$?
- No. We prove this using Yao's minimax principle.

Edge-k-Coloring on Trees

- Suppose that the input graph is a tree.

Edge-k-Coloring on Trees

- Suppose that the input graph is a tree.
- For $k \geq 2$, we show that:
  - The competitive ratio of any fair algorithm is at least $\frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.

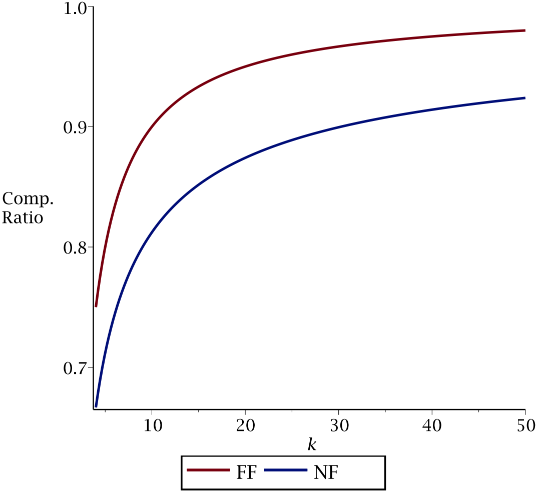
Edge-k-Coloring on Trees

- Suppose that the input graph is a tree.
- For $k \geq 2$, we show that:
  - The competitive ratio of any fair algorithm is at least $\frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.
  - The competitive ratio of First-Fit is exactly $\frac{k-1}{k}$.

Edge-k-Coloring on Trees

- Suppose that the input graph is a tree.
- For $k \geq 2$, we show that:
  - The competitive ratio of any fair algorithm is at least $\frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.
  - The competitive ratio of First-Fit is exactly $\frac{k-1}{k}$.
  - First-Fit is *optimal* among deterministic or fair algorithms.

First-Fit vs Next-Fit on Trees



Charging technique for proving positive results

Three types of edges for a deterministic algorithm A:

- Double colored: Colored by both A and OPT.
- Single colored: Colored only by A.
- Rejected: Colored only by OPT.

Charging technique for proving positive results

Three types of edges for a deterministic algorithm A:

- Double colored: Colored by both A and OPT.
- Single colored: Colored only by A.
- Rejected: Colored only by OPT.

We want to prove that A is C-competitive. Suppose that A earns a dollar whenever it colors an edge. We need to show that A can buy all of the edges colored by OPT, paying at least C for each.

Charging technique for proving positive results

Three types of edges for a deterministic algorithm A:

- Double colored: Colored by both A and OPT.
- Single colored: Colored only by A.
- Rejected: Colored only by OPT.

We want to prove that A is C-competitive. Suppose that A earns a dollar whenever it colors an edge. We need to show that A can buy all of the edges colored by OPT, paying at least C for each.
Double colored edges will pay for themselves and therefore have a surplus of $1 - C$.
Single colored edges will have a surplus of 1.

Fair Algorithm on Trees

Any fair algorithm F has a competitive ratio of at least $C = \frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.

Fair Algorithm on Trees

Any fair algorithm F has a competitive ratio of at least $C = \frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.

- Double colored edges have a surplus of $1 - C = \frac{1}{2\sqrt{k}-1}$.
- Single colored edges have a surplus of 1.
- Rejected edges need to receive a value of at least C from the colored edges.

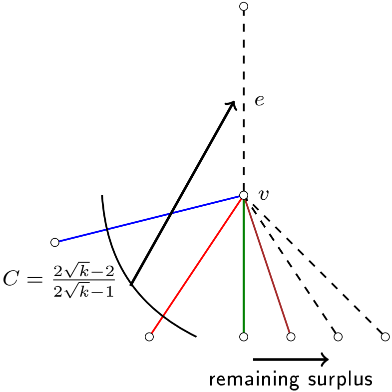
Fair Algorithm on Trees

Any fair algorithm F has a competitive ratio of at least $C = \frac{2\sqrt{k}-2}{2\sqrt{k}-1}$.

- Double colored edges have a surplus of $1 - C = \frac{1}{2\sqrt{k}-1}$.
- Single colored edges have a surplus of 1.
- Rejected edges need to receive a value of at least C from the colored edges.

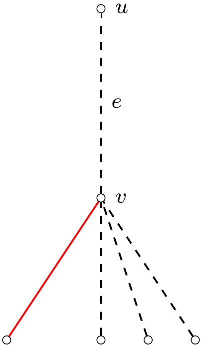
Fair Algorithm on Trees

Strategy for redistributing the surplus:



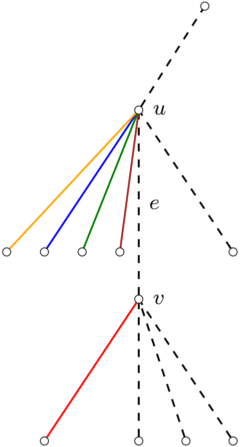
Fair Algorithm on Trees

What if a rejected edge e has only a few colored child edges?



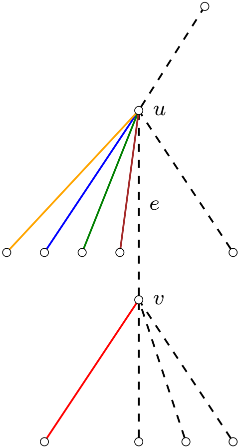
Fair Algorithm on Trees

What if a rejected edge e has only a few colored child edges?



Fair Algorithm on Trees

What if a rejected edge e has only a few colored child edges?



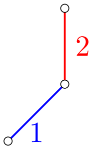
Worst-case: Roughly $\sqrt{k}$ colored child edges.

First-Fit on Trees

First-Fit has a competitive ratio of at least $\frac{k-1}{k}$ on trees. Use the same strategy as before with the following addition:

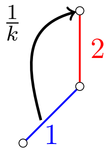
First-Fit on Trees

First-Fit has a competitive ratio of at least $\frac{k-1}{k}$ on trees. Use the same strategy as before with the following addition:



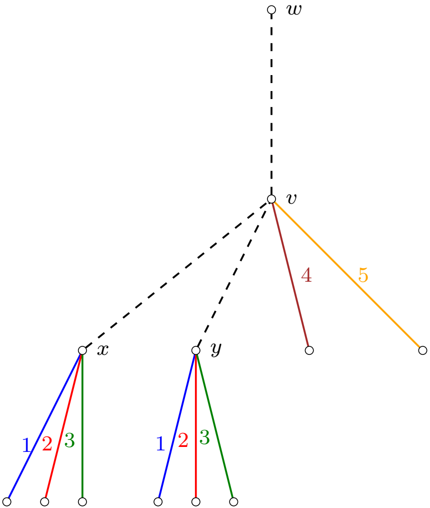
First-Fit on Trees

First-Fit has a competitive ratio of at least $\frac{k-1}{k}$ on trees. Use the same strategy as before with the following addition:



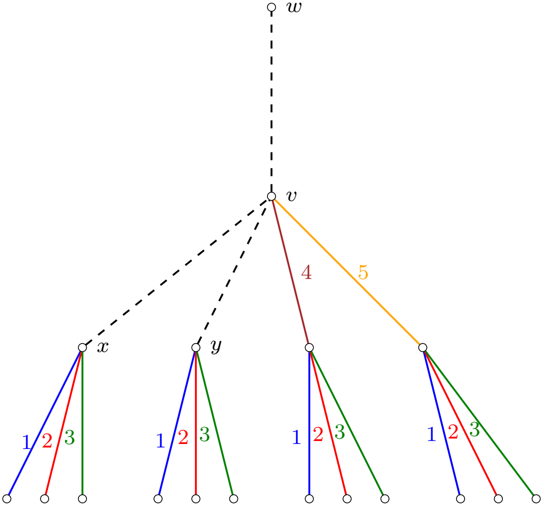
First-Fit on Trees

Example: $k = 5$, only double colored.



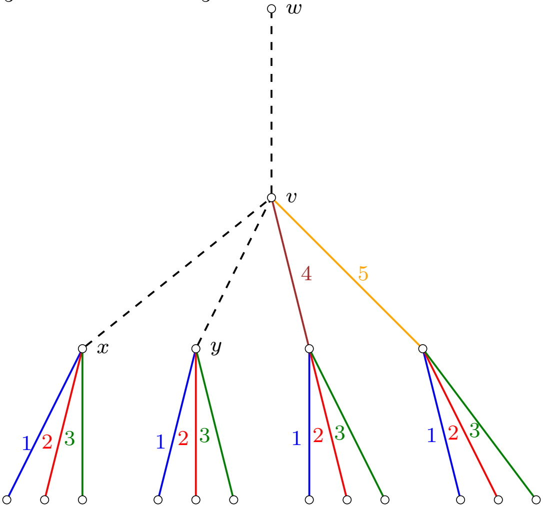
First-Fit on Trees

Example: $k = 5$, only double colored.



First-Fit on Trees

Example: $k = 5$, only double colored.
$m(v) = \frac{8}{5}$.
v transfers $\frac{4}{5}$ to (w, v) and $\frac{2}{5}$ to each of (v, x) and (v, y).



First-Fit on Trees

First-Fit has a competitive ratio of at least $\frac{k-1}{k}$ on trees.

First-Fit on Trees

First-Fit has a competitive ratio of at least $\frac{k-1}{k}$ on trees.

Step 1 Consider in turn all edges $e = (v, u) \in E_{\mathsf{c}}$. Let c be the color assigned to e by First-Fit and let $e' = (w, v)$ be the parent edge of e.

- **Step 1.1** If $e' \in E_{\mathsf{d}}$ and e' has been colored with a color $c' > c$, then e transfers a value of $\frac{1}{k}$ to w.
- **Step 1.2** Any surplus remaining at e is transferred to v.

For each vertex v, let $m(v)$ denote the value transferred to v in step 1.

Step 2 Consider in turn all vertices $v \in V$.

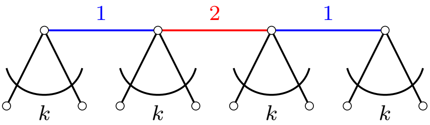
- **Step 2.1** If v has a parent edge e' and $e' \in E_{\mathsf{r}}$, then v transfers a value of $\min\left\{m(v), \frac{k-1}{k}\right\}$ to e'.
- **Step 2.2** Any value remaining at v is distributed equally among the child edges of v belonging to E_{r}.

Negative result for Edge-k-Coloring of Trees

First-Fit has a competitive ratio of at most $\frac{k-1}{k}$.

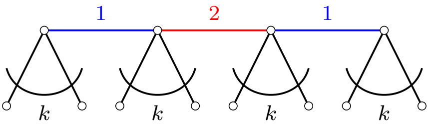
Negative result for Edge-k-Coloring of Trees

First-Fit has a competitive ratio of at most $\frac{k-1}{k}$.



Negative result for Edge-k-Coloring of Trees

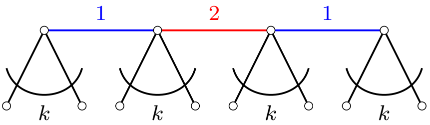
First-Fit has a competitive ratio of at most $\frac{k-1}{k}$.



First-Fit colors $N(k-2)+N = N(k-1)$ and OPT colors Nk edges.

Negative result for Edge-k-Coloring of Trees

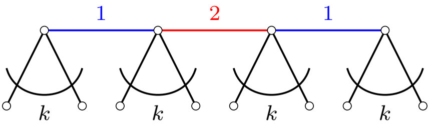
First-Fit has a competitive ratio of at most $\frac{k-1}{k}$.



First-Fit colors $N(k-2)+N = N(k-1)$ and OPT colors Nk edges.
A similar construction shows that no fair or deterministic algorithm can do better than $\frac{k-1}{k}$.

Negative result for Edge-k-Coloring of Trees

First-Fit has a competitive ratio of at most $\frac{k-1}{k}$.



First-Fit colors $N(k-2)+N = N(k-1)$ and OPT colors Nk edges.

A similar construction shows that no fair or deterministic algorithm can do better than $\frac{k-1}{k}$.

Furthermore, one can show that the competitive ratio of Next-Fit is no better than $\frac{2\sqrt{k}-2}{2\sqrt{k}-1}$ when k is a square number.

Randomization on Trees

- First-Fit is optimal on trees among fair or deterministic algorithms with a competitive ratio of $\frac{k-1}{k}$.
- Can a randomized algorithm do better than $\frac{k-1}{k}$?

Randomization on Trees

- First-Fit is optimal on trees among fair or deterministic algorithms with a competitive ratio of $\frac{k-1}{k}$.
- Can a randomized algorithm do better than $\frac{k-1}{k}$?
- Maybe, but not better than $\frac{k}{k+1}$.

If it looks like a tree...

- There exists several measures of how “tree-like” a graph is.

If it looks like a tree...

- There exists several measures of how “tree-like” a graph is.
- Treewidth, arboricity, degeneracy, pseudoarboricity etc.

If it looks like a tree...

- There exists several measures of how “tree-like” a graph is.
- Treewidth, arboricity, degeneracy, pseudoarboricity etc.
- The *pseudoarboricity (PA)* of G is the minimum t such that the edges of G can be oriented to form a digraph where each vertex has outdegree at most t.

If it looks like a tree...

- There exists several measures of how "tree-like" a graph is.
- Treewidth, arboricity, degeneracy, pseudoarboricity etc.
- The *pseudoarboricity (PA)* of G is the minimum t such that the edges of G can be oriented to form a digraph where each vertex has outdegree at most t.
- Trees have PA $= 1$. Planar graphs have PA at most 3.

If it looks like a tree...

- There exists several measures of how "tree-like" a graph is.
- Treewidth, arboricity, degeneracy, pseudoarboricity etc.
- The *pseudoarboricity (PA)* of G is the minimum t such that the edges of G can be oriented to form a digraph where each vertex has outdegree at most t.
- Trees have PA $= 1$. Planar graphs have PA at most 3.
- Graphs of bounded degree, treewidth, degeneracy or genus has bounded PA.

Parameterized Competitive Ratio

- We *parameterize* the competitive ratio by the PA of the input graph.

Parameterized Competitive Ratio

- We *parameterize* the competitive ratio by the PA of the input graph.

Theorem

Suppose that the input graph is k-colorable and has PA at most t. If $t \leq \frac{1}{4}k$, then the competitive ratio of any fair algorithm is at least

$$\frac{2\sqrt{k/t} - 2}{2\sqrt{k/t} - 1}.$$

Parameterized Competitive Ratio

- We *parameterize* the competitive ratio by the PA of the input graph.

Theorem

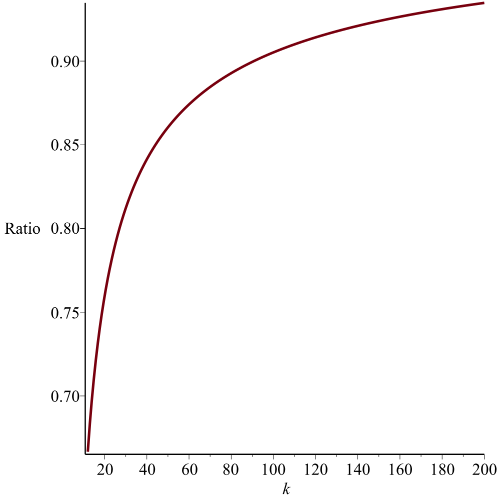
Suppose that the input graph is k-colorable and has PA at most t. If $t \leq \frac{1}{4}k$, then the competitive ratio of any fair algorithm is at least

$$\frac{2\sqrt{k/t}-2}{2\sqrt{k/t}-1}.$$

The competitive ratio on k-colorable graph is also known as the competitive ratio on *accommodating sequences* [Boyar, Larsen, Nielsen '98].

Parameterized Competitive Ratio

A lower bound for any fair algorithm on planar graphs (PA ≤ 3).



Conclusion

- Rand_p is optimal on paths and better than any deterministic algorithm.

Conclusion

- Rand_p is optimal on paths and better than any deterministic algorithm.
- First-Fit is optimal among deterministic algorithms on paths and trees.

Conclusion

- Rand_p is optimal on paths and better than any deterministic algorithm.
- First-Fit is optimal among deterministic algorithms on paths and trees.
- On tree-like graphs, any fair algorithm for online Edge-k-Coloring performs well if it has a sufficiently large number of colors.

Open Problems

- Find the optimal online algorithm for Edge-k-Coloring in general and on other graph classes.

Open Problems

- Find the optimal online algorithm for Edge-k-Coloring in general and on other graph classes.

Is it possible to achieve a competitive ratio better than $2\sqrt{3} - 3$ for Edge-k-Coloring?

Open Problems

- Find the optimal online algorithm for Edge-k-Coloring in general and on other graph classes.

Is it possible to achieve a competitive ratio better than $2\sqrt{3} - 3$ for Edge-k-Coloring?
Does First-Fit have a competitive ratio better than $2\sqrt{3} - 3$ for Edge-k-Coloring?

Open Problems

- Find the optimal online algorithm for Edge-k-Coloring in general and on other graph classes.

Is it possible to achieve a competitive ratio better than $2\sqrt{3}-3$ for Edge-k-Coloring?
Does First-Fit have a competitive ratio better than $2\sqrt{3}-3$ for Edge-k-Coloring?On bipartite graphs?

THANK YOU