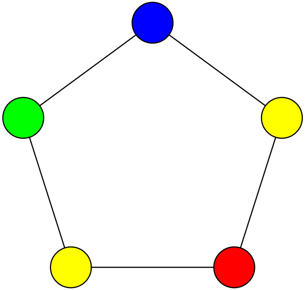
On edge b-colorings

V. Campos[1] C. Lima[1] N. Martins[1] L. Sampaio[1]
M. Santos[1] A. Silva[1]

[1]ParGO - Universidade Federal do Ceará, Brazil

April 2nd, 2013

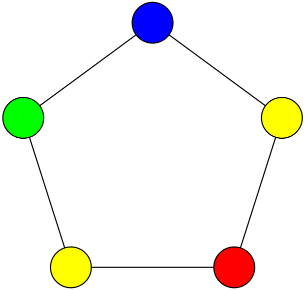
Vertex Coloring



Vertex Coloring

- one color for each vertex
- adjacent vertices get different colors

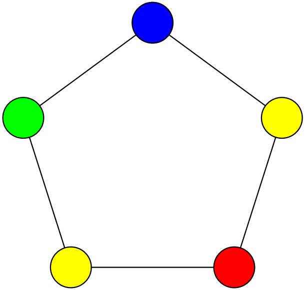
Vertex coloring problem



Definition: chromatic number

- $\chi(G) =$ min number of colors in a vertex coloring of G

Vertex coloring problem



Definition: vertex coloring problem

- $\chi(G) =$ min number of colors in a vertex coloring of G
- Vertex coloring problem: find $\chi(G)$

Some considerations

Dificulties:

- Decision version is NP-complete

R. Karp
Reducibility among combinatorial problems.
Complexity of Computations, 1972.

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

C. Lund e M. Yannakakis
On the hardness of approximating minimization problems.
Journal of the ACM, 1994.

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm
  - b-coloring algorithm

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm
  - b-coloring algorithm
- Not guaranteed to find optimal solutions

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm
  - b-coloring algorithm
- Not guaranteed to find optimal solutions
- So we study the worst case of these algorithms:

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm
  - b-coloring algorithm
- Not guaranteed to find optimal solutions
- So we study the worst case of these algorithms:
  - The worst case of the greedy algorithm is the *Grundy number* ($\Gamma(G)$)

Some considerations

Dificulties:

- Decision version is NP-complete
- Impossible to approximate by a factor of $n^{1-\epsilon}$ unless P=NP

One approach:

- Study polinomial coloring algorithms
  - Greedy algorithm
  - b-coloring algorithm
- Not guaranteed to find optimal solutions
- So we study the worst case of these algorithms:
  - The worst case of the greedy algorithm is the *Grundy number* ($\Gamma(G)$)
  - The worst case of the b-coloring algorithm is the *b-chromatic number* ($b(G)$)

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

F. Havet e L. Sampaio
On the grundy number of a graph.
IPEC, 2010.

J. Kratochvil, Z. Tuza e M. Voigt
On the b-chromatic number of graphs.
LNCS, 2002.

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

One approach

- Work with graph classes

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

One approach

- Work with graph classes
  - graph products

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

One approach

- Work with graph classes
  - graph products
  - $(q, q-4)$ graphs

Other considerations

It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

One approach

- Work with graph classes
  - graph products
  - $(q, q-4)$ graphs
  - cacti

Other considerations

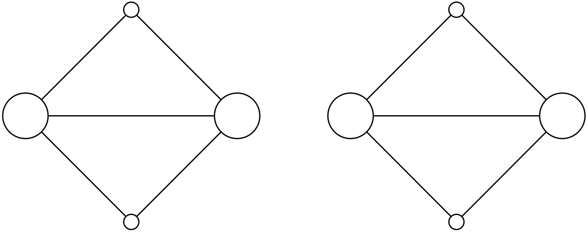
It's hard!

- finding $\Gamma(G)$ or $b(G)$ is NP-hard
  - even if G is bipartite

One approach

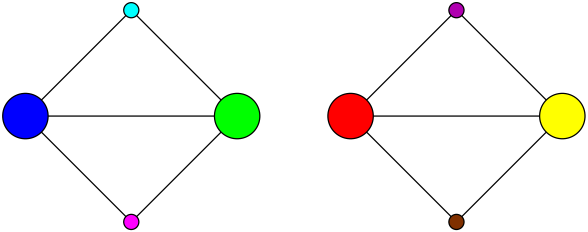
- Work with graph classes
  - graph products
  - $(q, q-4)$ graphs
  - cacti
  - line graphs

b-coloring algorithm



Algorithm:

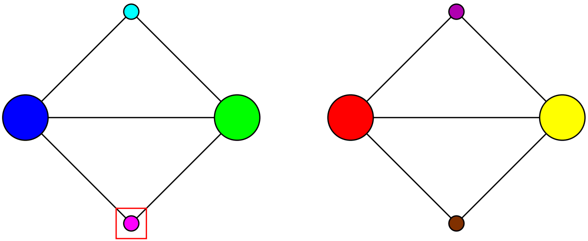
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.

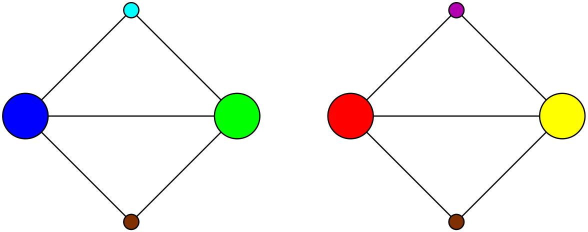
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.

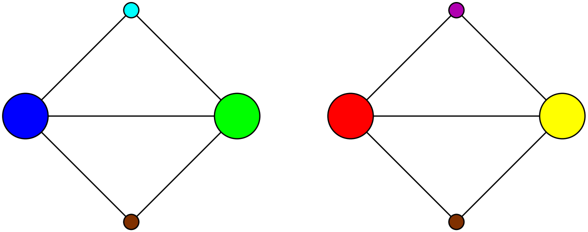
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.

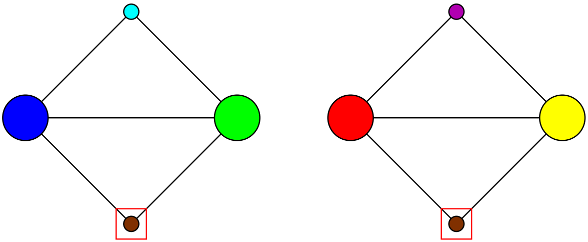
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

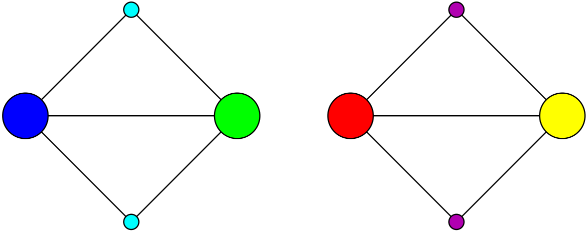
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

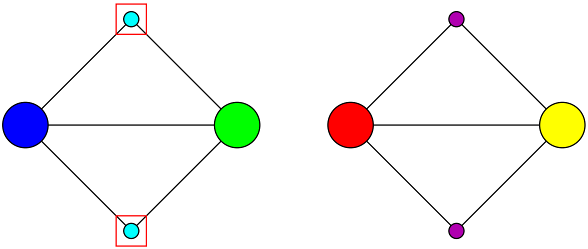
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

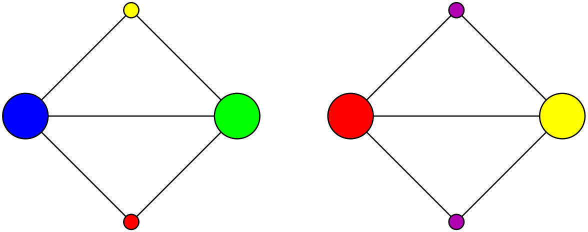
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

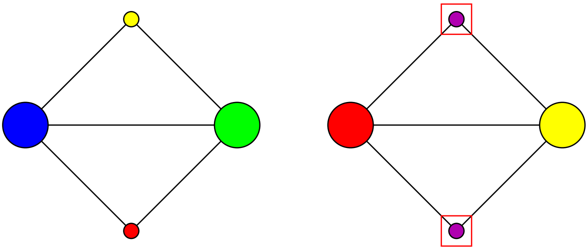
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

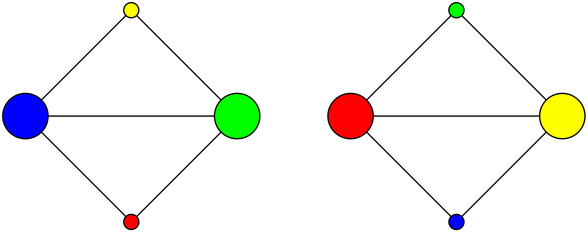
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

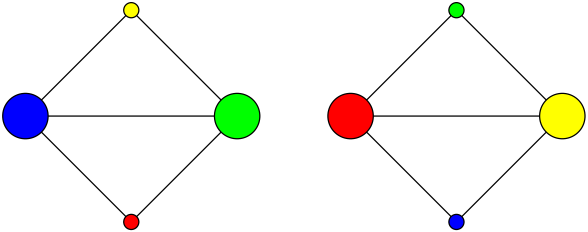
b-coloring algorithm



Algorithm:

1. Give different colors to all vertices of G.
2. Choose a color class we can recolor.
3. Recolor vertices in this color class.
4. Whenever possible, go to step 2.

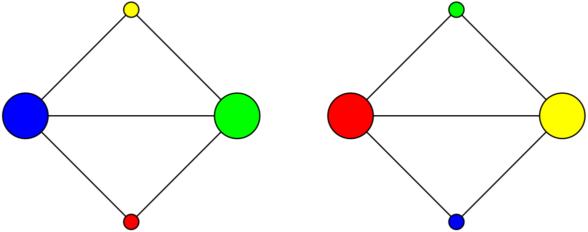
b-coloring



Definitions:

- b-vertex: at least one neighbor of each color other than its own.

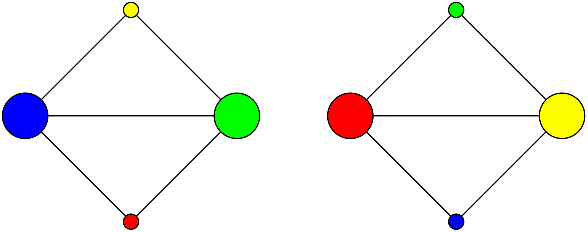
b-coloring



Definitions:

- b-vertex: at least one neighbor of each color other than its own.
- b-coloring: each color class has a b-vertex.

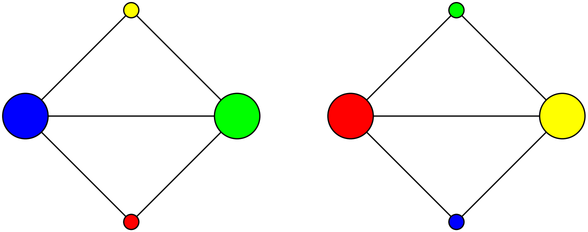
b-coloring



Definitions:

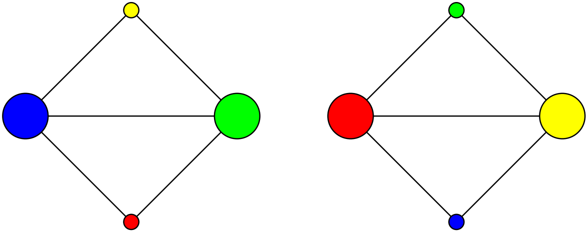
- b-vertex: at least one neighbor of each color other than its own.
- b-coloring: each color class has a b-vertex.
- $b(G) =$ max number of colors in a b-coloring of G.

Upper bound



Has 4 vertices of degree 3.

Upper bound

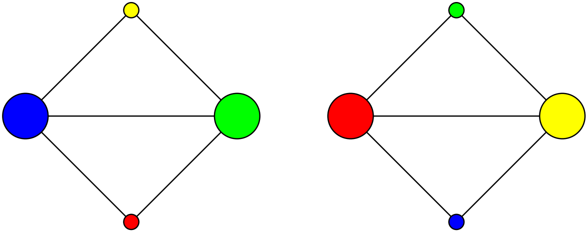


Has 4 vertices of degree 3.

Definition: *m*-degree

$$m(G) = \max\{k \mid G \text{ has } k \text{ vertices with degree } \geq k-1\}$$

Upper bound



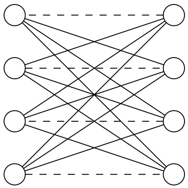
Has 4 vertices of degree 3.

Definition: m-degree

$$m(G) = \max\{k \mid G \text{ has } k \text{ vertices with degree } \geq k-1\}$$

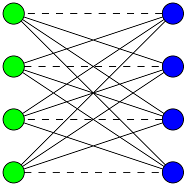
$$\chi(G) \leq b(G) \leq m(G)$$

b-continuity



$K_{4,4}$ minus a perfect matching.

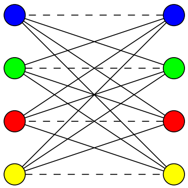
b-continuity



$K_{4,4}$ minus a perfect matching.

- Has b-coloring with 2 colors.

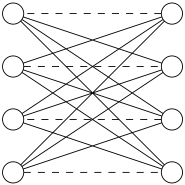
b-continuity



$K_{4,4}$ minus a perfect matching.

- Has b-coloring with 2 colors.
- Has b-coloring with 4 colors.

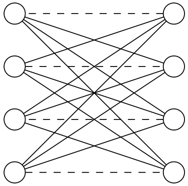
b-continuity



$K_{4,4}$ minus a perfect matching.

- Has b-coloring with 2 colors.
- Has b-coloring with 4 colors.
- Has **no** b-coloring with 3 colors.

b-continuity



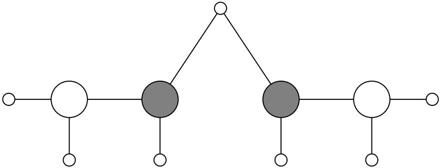
$K_{4,4}$ minus a perfect matching.

- Has b-coloring with 2 colors.
- Has b-coloring with 4 colors.
- Has **no** b-coloring with 3 colors.

b-continuity

G is *b-continuous* if if has a b-coloring with k colors for every $k \in \{\chi(G), ..., b(G)\}$

b-chromatic number of trees



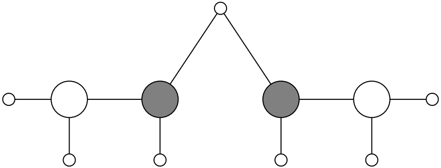
Theorem [IM]

If T is a tree, then

- $b(T) \in \{m(T) - 1, m(T)\}$

R. Irving e D. Manlove
The b-chromatic number of a graph.
Discrete Applied Mathematics 91, 1999.

b-chromatic number of trees



Theorem [IM]

If T is a tree, then

- $b(T) \in \{m(T) - 1, m(T)\}$
- There exists a polinomial time algorithm to decide $b(T)$.

R. Irving e D. Manlove
The b-chromatic number of a graph.
Discrete Applied Mathematics 91, 1999.

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?
- Define χ', b' and m' analogous to vertices but for edges.

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?
- Define χ', b' and m' analogous to vertices but for edges.

Edge b-chromatic problem

- Instance: Graph G
- Question: Is $b'(G) = m'(G)$?

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?
- Define χ', b' and m' analogous to vertices but for edges.

Edge b-chromatic problem

- Instance: Graph G
- Question: Is $b'(G) = m'(G)$?

Theorem [CLMSSS]

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?
- Define χ', b' and m' analogous to vertices but for edges.

Edge b-chromatic problem

- Instance: Graph G
- Question: Is $b'(G) = m'(G)$?

Theorem [CLMSSS]

- Edge b-chromatic problem is NP-complete.

Our results

- What happens if we consider edge b-colorings instead of vertex b-colorings?
- Define χ', b' and m' analogous to vertices but for edges.

Edge b-chromatic problem

- Instance: Graph G
- Question: Is $b'(G) = m'(G)$?

Theorem [CLMSSS]

- Edge b-chromatic problem is NP-complete.
- It remains NP-complete even if G is a comparability graph or C_k-free graph, for $k \geq 4$.

Other questions of interest

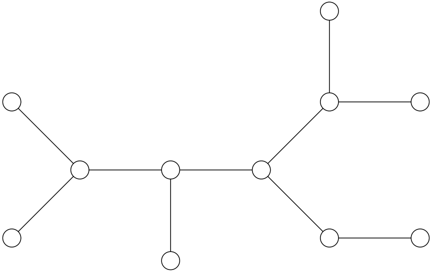
- What about edge b-colorings of trees?

Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.

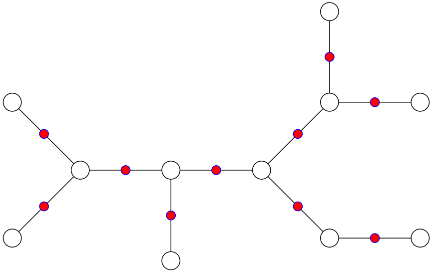
Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



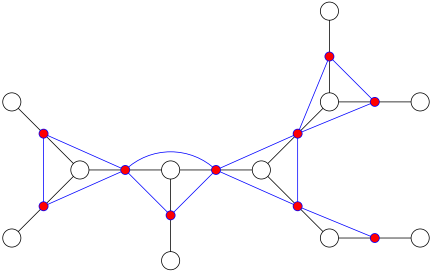
Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



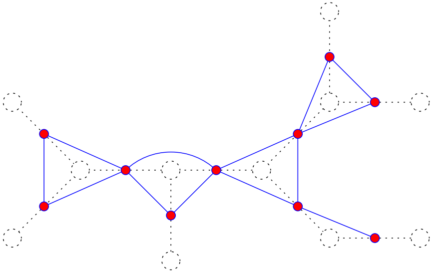
Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



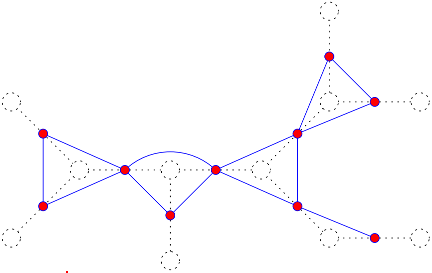
Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



Other questions of interest

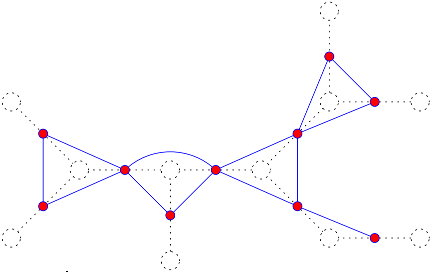
- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



- Special block graphs.

Other questions of interest

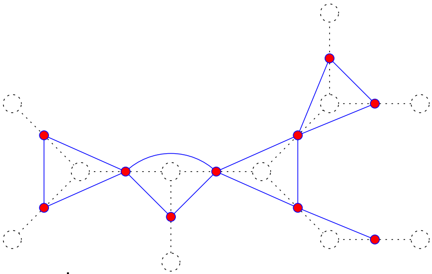
- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



- Special block graphs.
- b-coloring block graphs are hard!

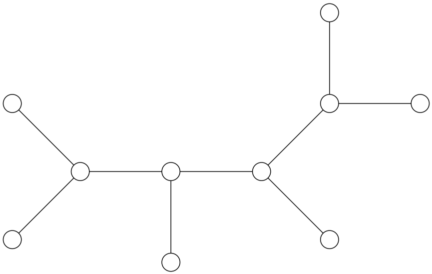
Other questions of interest

- What about edge b-colorings of trees?
- Consider vertex b-colorings of line graphs of trees.



- Special block graphs.
- b-coloring block graphs are hard! (in the sense that it's unknown)

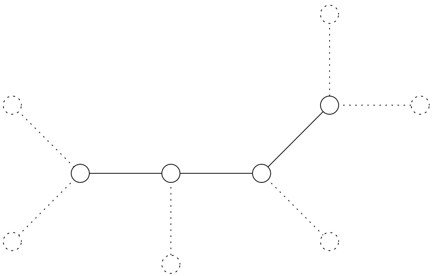
Caterpillar trees



Caterpillar T

Deleting all leaves of T produces a path.

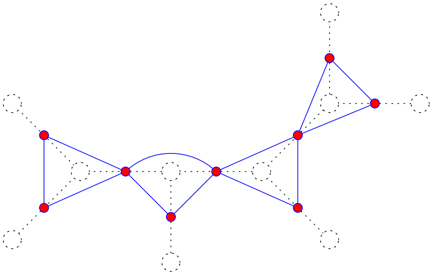
Caterpillar trees



Caterpillar T

Deleting all leaves of T produces a path.

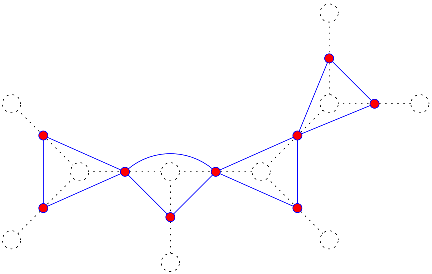
Caterpillar trees



Caterpillar T

Deleting all leaves of T produces a path.

Caterpillar trees

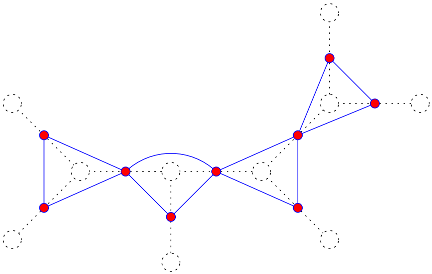


Caterpillar T

Deleting all leaves of T produces a path.

- Line graphs of trees are chordal.

Caterpillar trees

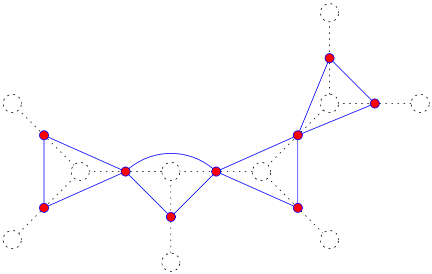


Caterpillar T

Deleting all leaves of T produces a path.

- Line graphs of trees are chordal.
- Chordal graphs are perfect.

Caterpillar trees

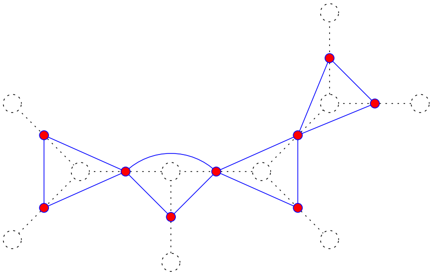


Caterpillar T

Deleting all leaves of T produces a path.

- Line graphs of trees are chordal.
- Chordal graphs are perfect.
- $\chi = \omega =$ size of largest clique.

Caterpillar trees

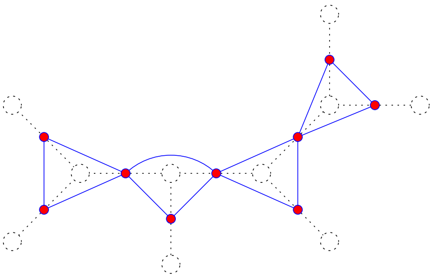


Caterpillar T

Deleting all leaves of T produces a path.

- Line graphs of trees are chordal.
- Chordal graphs are perfect.
- $\chi = \omega =$ size of largest clique. (polynomial)

Caterpillar trees



Caterpillar T

Deleting all leaves of T produces a path.

- Line graphs of trees are chordal.
- Chordal graphs are perfect.
- $\chi = \omega =$ size of largest clique.
- From now on, let G be a line graph of a caterpillar.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
    - then $\chi(G) = b(G) = m(G)$.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
  - then $\chi(G) = b(G) = m(G)$.

Theorem [CLMSSS]

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
  - then $\chi(G) = b(G) = m(G)$.

Theorem [CLMSSS]

- If $\chi(G) < m(G)$ and $k \in \{\chi(G), \ldots, m(G) - 1\}$,

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
  - then $\chi(G) = b(G) = m(G)$.

Theorem [CLMSSS]

- If $\chi(G) < m(G)$ and $k \in \{\chi(G), \ldots, m(G) - 1\}$,
  - there exists a b-coloring of G with k colors.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
  - then $\chi(G) = b(G) = m(G)$.

Theorem [CLMSSS]

- If $\chi(G) < m(G)$ and $k \in \{\chi(G), \ldots, m(G) - 1\}$,
  - there exists a b-coloring of G with k colors.

Corollary [CLMSSS]

G is b-continuous.

Some results on line graphs of caterpillars

- $\chi(G)$ is polynomial for G.
- $m(G)$ is polynomial.
- if $\chi(G) = m(G)$,
    - then $\chi(G) = b(G) = m(G)$.

Theorem [CLMSSS]

- If $\chi(G) < m(G)$ and $k \in \{\chi(G), \ldots, m(G) - 1\}$,
    - there exists a b-coloring of G with k colors.

Corollary [CLMSSS]

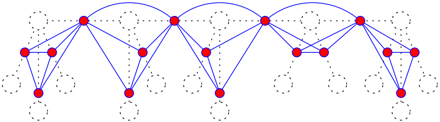
G is b-continuous.

Corollary [CLMSSS]

$b(G) \in \{m(G) - 1, m(G)\}$.

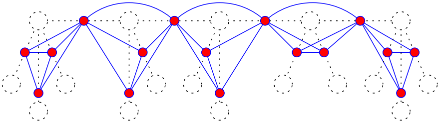
Deciding the value of $b(G)$

- Suppose $\chi(G) < m(G)$.



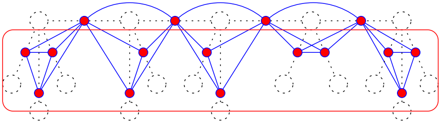
Deciding the value of $b(G)$

- Suppose $\chi(G) < m(G)$.
- Consider a coloring of G with $m(G)$ colors.



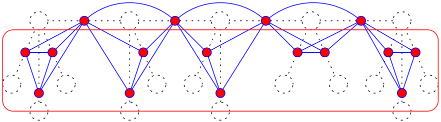
Deciding the value of $b(G)$

- Suppose $\chi(G) < m(G)$.
- Consider a coloring of G with $m(G)$ colors.
- These vertices have at most $\omega(G) - 1$ neighbours.



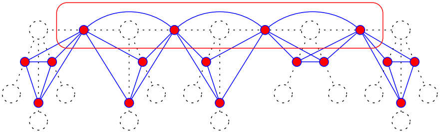
Deciding the value of $b(G)$

- Suppose $\chi(G) < m(G)$.
- Consider a coloring of G with $m(G)$ colors.
- These vertices have at most $\omega(G) - 1$ neighbours.
  - so they cannot be b-vertices.



Deciding the value of $b(G)$

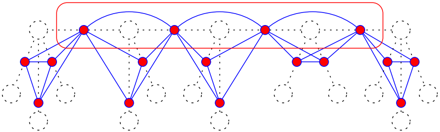
- Suppose $\chi(G) < m(G)$.
- Consider a coloring of G with $m(G)$ colors.
- These vertices have at most $\omega(G) - 1$ neighbours.
  - so they cannot be b-vertices.
- Only possible b-vertices are in the *central path*.



An interesting problem

Central path coloring problem

- Instance: Graph G and subset $W = \{w_1, \ldots, w_k\}$ of vertices in the central path
- Question: Is there a b-coloring of G with $m(G)$ colors such that the vertices of W are b-vertices of different colors?



Using the Central path coloring problem

Central path coloring problem

- Instance: Graph G and subset $W = \{w_1, \ldots, w_k\}$ of vertices in the central path
- Question: Is there a b-coloring of G with $m(G)$ colors such that the vertices of W are b-vertices of different colors?

- We can decide if G has a b-coloring with $m(G)$ colors by solving the Central path coloring problem $\binom{n}{m(G)}$ times.

Using the Central path coloring problem

Central path coloring problem

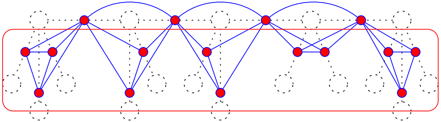
- Instance: Graph G and subset $W = \{w_1, \ldots, w_k\}$ of vertices in the central path
- Question: Is there a b-coloring of G with $m(G)$ colors such that the vertices of W are b-vertices of different colors?

- We can decide if G has a b-coloring with $m(G)$ colors by solving the Central path coloring problem $\binom{n}{m(G)}$ times.

Theorem [CLMSSS]

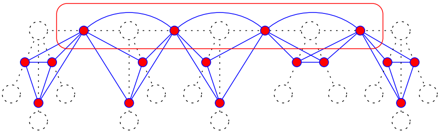
We can decide if G has a b-coloring with $m(G)$ colors by solving the Central path coloring problem n times.

Solving the CENTRAL PATH COLORING PROBLEM



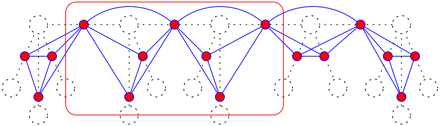
- Give colors to cliques.

Solving the CENTRAL PATH COLORING PROBLEM



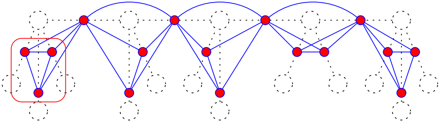
- Give colors to cliques.
- Give colors to central vertices.

Solving the CENTRAL PATH COLORING PROBLEM



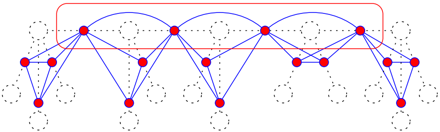
- Give colors to cliques.
- Give colors to central vertices.
- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

Solving the Central path coloring problem



- Give colors to cliques.
- Give colors to central vertices.
- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.
- Clique C gets at most $|C|$ colors.

Solving the Central path coloring problem



- Give colors to cliques.
- Give colors to central vertices.
- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.
- Clique C gets at most $|C|$ colors.
- Central vertices get one color.

Solving the Central path coloring problem

- Give colors to cliques.
- Give colors to central vertices.
- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.
- Clique C gets at most $|C|$ colors.
- Central vertices get one color.

Solving the CENTRAL PATH COLORING PROBLEM

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

- Clique C gets at most $|C|$ colors.

- Central vertices get one color.

Solving the CENTRAL PATH COLORING PROBLEM

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

$$y_{v,i} = \begin{cases} 1, & \text{if } v \text{ is colored } i \text{ (or } v = w_i\text{)} \\ 0, & \text{othewise} \end{cases}$$

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.
- Clique C gets at most $|C|$ colors.
- Central vertices get one color.

Solving the Central path coloring problem

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

$$y_{v,i} = \begin{cases} 1, & \text{if } v \text{ is colored } i \text{ (or } v = w_i) \\ 0, & \text{othewise} \end{cases}$$

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

$$x_{C_l(w_i),j} + x_{C_r(w_i),j} + y_{v_l(w_i),j} + y_{v_r(w_i),j} \geq 1$$

- Clique C gets at most $|C|$ colors.

- Central vertices get one color.

Solving the Central path coloring problem

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

$$y_{v,i} = \begin{cases} 1, & \text{if } v \text{ is colored } i \text{ (or } v = w_i) \\ 0, & \text{othewise} \end{cases}$$

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

$$x_{C_l(w_i),j} + x_{C_r(w_i),j} + y_{v_l(w_i),j} + y_{v_r(w_i),j} \geq 1$$

- Clique C gets at most $|C|$ colors.

$$x_{C,1} + \cdots + x_{C,m(G)} \leq |C|$$

- Central vertices get one color.

Solving the Central path coloring problem

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

$$y_{v,i} = \begin{cases} 1, & \text{if } v \text{ is colored } i \text{ (or } v = w_i) \\ 0, & \text{othewise} \end{cases}$$

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

$$x_{C_l(w_i),j} + x_{C_r(w_i),j} + y_{v_l(w_i),j} + y_{v_r(w_i),j} \geq 1$$

- Clique C gets at most $|C|$ colors.

$$x_{C,1} + \cdots + x_{C,m(G)} \leq |C|$$

- Central vertices get one color.

$$y_{v,1} + \cdots + y_{v,m(G)} = 1$$

Solving the Central path coloring problem

- Give colors to cliques.

$$x_{C,i} = \begin{cases} 1, & \text{if color } i \text{ is in } C \\ 0, & \text{othewise} \end{cases}$$

- Give colors to central vertices.

$$y_{v,i} = \begin{cases} 1, & \text{if } v \text{ is colored } i \text{ (or } v = w_i) \\ 0, & \text{othewise} \end{cases}$$

- For every $w_i \in W$ and $j \neq i$, w_i has a neighbor colored j.

$$x_{C_l(w_i),j} + x_{C_r(w_i),j} + y_{v_l(w_i),j} + y_{v_r(w_i),j} \geq 1$$

- Clique C gets at most $|C|$ colors.

$$x_{C,1} + \cdots + x_{C,m(G)} \leq |C|$$

- Central vertices get one color.

$$y_{v,1} + \cdots + y_{v,m(G)} = 1$$

Let this be the Central path polytope.

Results on the Central path polytope

Theorem [CLMSSS]

If the Central path polytope has an integer solution, then the Central path coloring problem is true.

Results on the Central path polytope

Theorem [CLMSSS]

If the Central path polytope has an integer solution, then the Central path coloring problem is true.

Theorem [CLMSSS]

The Central path polytope matrix is totally unimodular.

Results on the Central path polytope

Theorem [CLMSSS]

If the Central path polytope has an integer solution, then the Central path coloring problem is true.

Theorem [CLMSSS]

The Central path polytope matrix is totally unimodular.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide if the Central path polytope has an integer solution.

Results on the Central path polytope

Theorem [CLMSSS]

If the Central path polytope has an integer solution, then the Central path coloring problem is true.

Theorem [CLMSSS]

The Central path polytope matrix is totally unimodular.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide if the Central path polytope has an integer solution.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide the Central path coloring problem.

Results on the CENTRAL PATH POLYTOPE

Theorem [CLMSSS]

If the CENTRAL PATH POLYTOPE has an integer solution, then the CENTRAL PATH COLORING PROBLEM is true.

Theorem [CLMSSS]

The CENTRAL PATH POLYTOPE matrix is totally unimodular.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide if the CENTRAL PATH POLYTOPE has an integer solution.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide the CENTRAL PATH COLORING PROBLEM.

Corollary [CLMSSS]

There is a polynomial time algorithm to decide if G has a b-coloring with $m(G)$ colors.

Open problems

- Obtain a combinatorial algorithm to solve the CENTRAL PATH COLORING PROBLEM.

Open problems

- Obtain a combinatorial algorithm to solve the CENTRAL PATH COLORING PROBLEM.
  - Use the primal-dual method?

Open problems

- Obtain a combinatorial algorithm to solve the Central path coloring problem.
  - Use the primal-dual method?
- Generalize combinatorial algorithm for other classes of trees.

Open problems

- Obtain a combinatorial algorithm to solve the CENTRAL PATH COLORING PROBLEM.
  - Use the primal-dual method?
- Generalize combinatorial algorithm for other classes of trees.
- ???

Open problems

- Obtain a combinatorial algorithm to solve the CENTRAL PATH COLORING PROBLEM.
  - Use the primal-dual method?
- Generalize combinatorial algorithm for other classes of trees.
- ???
- Profit!

The End

Thank You!!