Approximations for λ-colorings of graphs

Simone Bianco - 1986936
Network Algorithms
Sapienza University of Rome

Introduction

Article: “Approximations for λ-colorings of graphs” by Bodlaender et al. (2004)

Focus: Upper and lower bounds for L(2,1), L(1,1) and L(0,1) labelings in graph classes

Introduction

Article: “Approximations for λ-colorings of graphs” by Bodlaender et al. (2004)

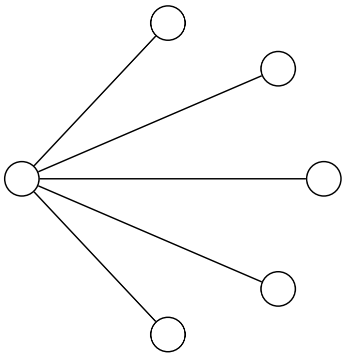
Focus: Upper and lower bounds for L(2,1), L(1,1) and L(0,1) labelings in graph classes

Summary of the presentation:

- Upper bound on graphs with treewidth k
- Upper bound on outerplanar graphs
- Lower bound on split graphs

Recall

General lower bounds for L(p,q)-labeling are given by the tree graph $K_{1,\Delta}$



$\Delta + 1 \leq \lambda_{2,1}$ $\Delta \leq \lambda_{1,1}$ $\Delta - 1 \leq \lambda_{0,1}$

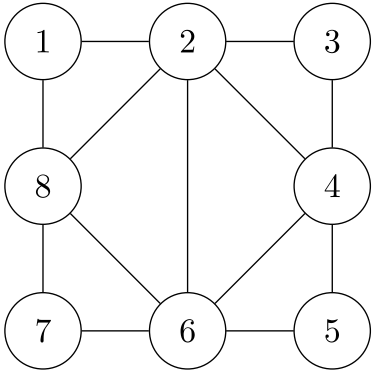
Tree decomposition

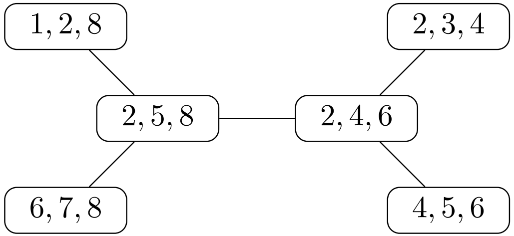
Given a graph G, a **tree decomposition** of G is a tree T whose vertices $X_1, \dots, X_k$ are subsets of V(G) that satisfy the following properties:

1) $X_1, \dots, X_k$ are a cover of V(G)

2) If $v \in X_i \cap X_j$ then each subset X_h in the path from X_i to X_j contains v

3) For each edge (u,v) of G at least one subset X_i contains u and v

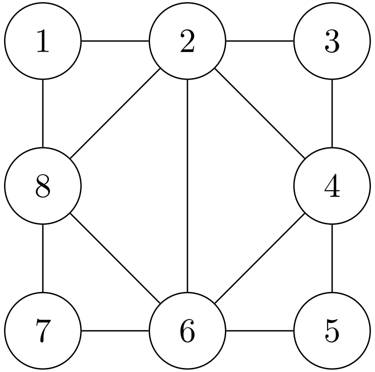


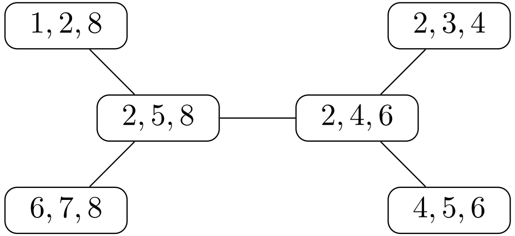


Treewidth

Width of a tree decomposition: size of the largest vertex of T, minus 1

Treewidth of a graph: smallest width of all the tree decompositions





k-Trees

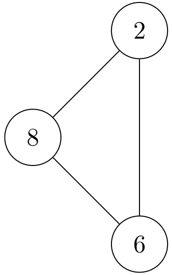
Inductive definition:

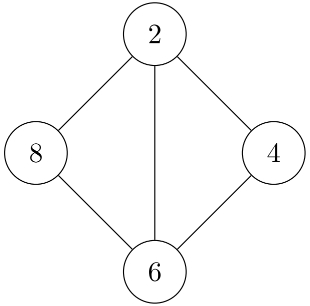
- The complete graph K_{k+1} is a k-tree
- A k-tree with $n > k+1$ nodes can be built from a k-tree G' with n-1 nodes by adding a new node and connecting it to k vertices that form a k-clique in G'

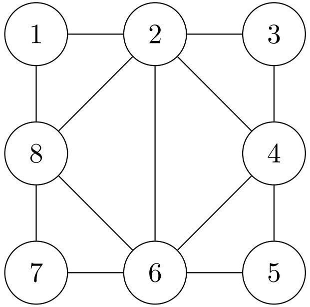
k-Trees

Inductive definition:

- The complete graph K_{k+1} is a k-tree
- A k-tree with n > k+1 nodes can be built from a k-tree G' with n-1 nodes by adding a new node and connecting it to k vertices that form a k-clique in G'

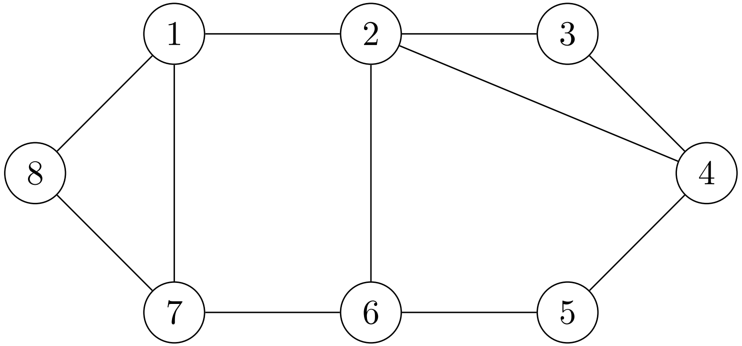


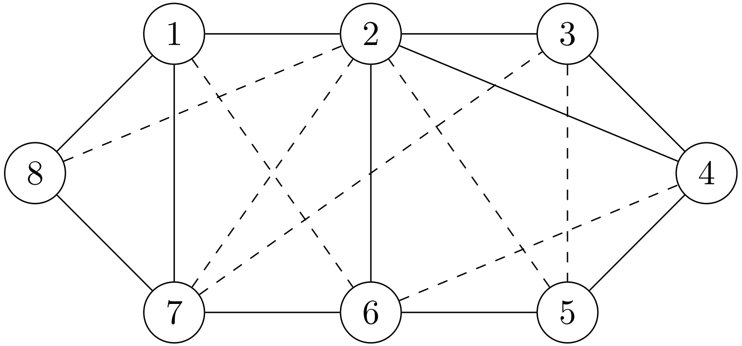




Partial k-tree

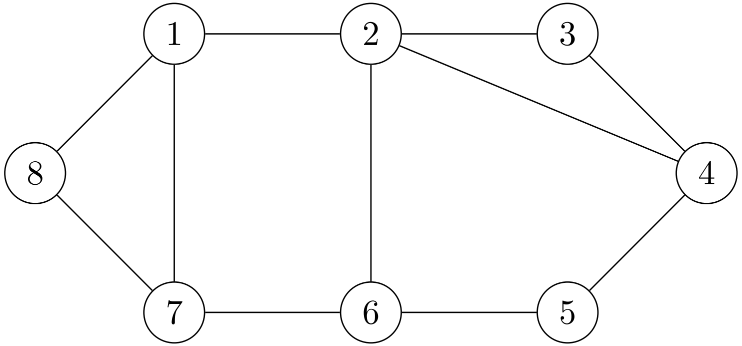
Partial k-tree: any graph that is a subgraph of a k-tree

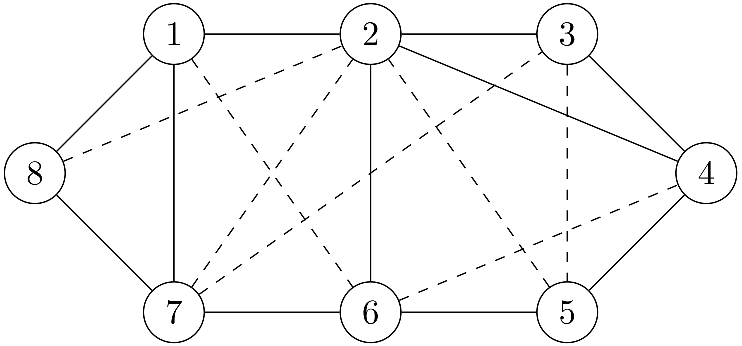




Partial k-tree

Partial k-tree: any graph that is a subgraph of a k-tree





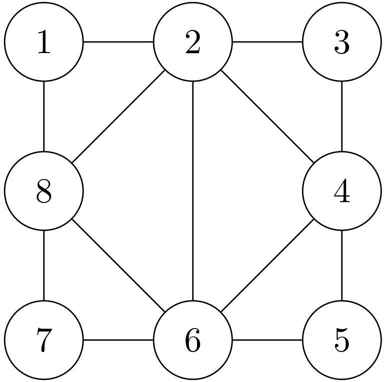
(Thm) G has treewidth ≤ k if and only if G is a partial k-tree

(Cor) G has treewidth k if and only if k is the smallest integer such that G is a partial k-tree

Cordal graphs

K-trees are a special type of **cordal (or triangulated)** graph.

A graph is **chordal** when all cycles of 4+ vertices have a **chord**, i.e. an edge that is not part of the cycle but connects two vertices of the cycle.

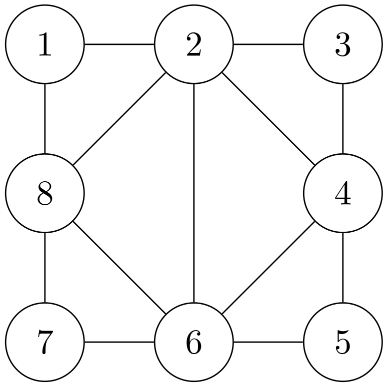


Cordal graphs

K-trees are a special type of **cordal (or triangulated)** graph.

A graph is **chordal** when all cycles of 4+ vertices have a **chord**, i.e. an edge that is not part of the cycle but connects two vertices of the cycle.

Equivalently, a chordal graph can be defined as a graph in which every induced cycle in the graph has **exactly three vertices** (hence the alternative name).



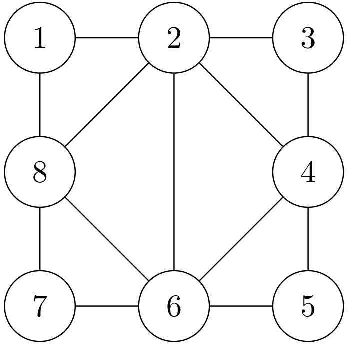
Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

A **perfect elimination sequence** is an ordering of the vertices such that for each node all of its neighbors that occur after it in the sequence form a clique with it.

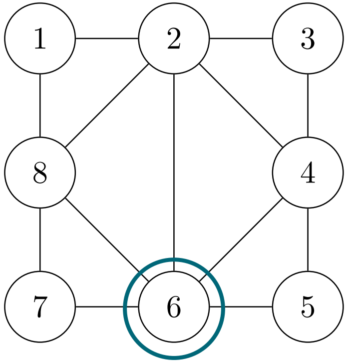


(1,3,5,7,4,2,8,6)

Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

A **perfect elimination sequence** is an ordering of the vertices such that for each node all of its neighbors that occur after it in the sequence form a clique with it.

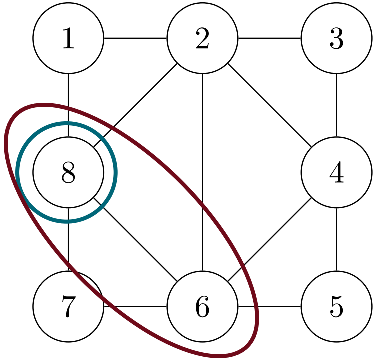


(1,3,5,7,4,2,8,6)

Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

A **perfect elimination sequence** is an ordering of the vertices such that for each node all of its neighbors that occur after it in the sequence form a clique with it.

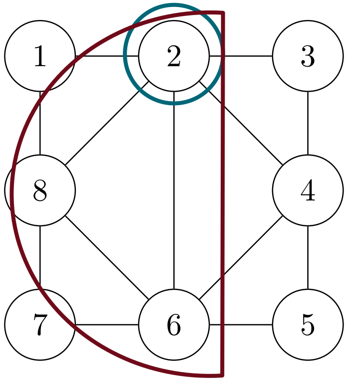


(1,3,5,7,4,2,8,6)

Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

A **perfect elimination sequence** is an ordering of the vertices such that for each node all of its neighbors that occur after it in the sequence form a clique with it.

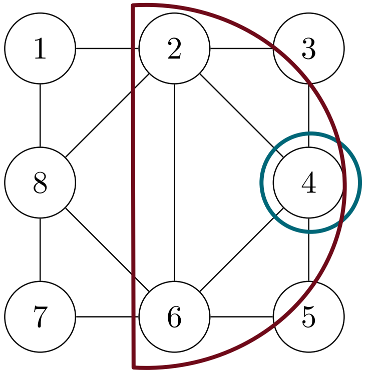


(1,3,5,7,4,2,8,6)

Perfect elimination sequence

(Thm) A graph is chordal if and only if it has a perfect elimination sequence

A **perfect elimination sequence** is an ordering of the vertices such that for each node all of its neighbors that occur after it in the sequence form a clique with it.



(1,3,5,7,4,2,8,6)

Algorithm for graphs with treewidth k

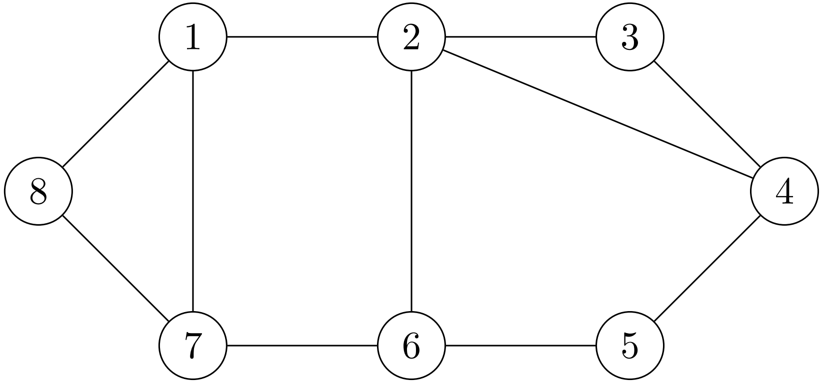
Given (G, λ, p, q) in input:

1) Build a k-tree H that contains G

2) Construct a perfect elimination sequence $v_1, \ldots, v_n$ on H

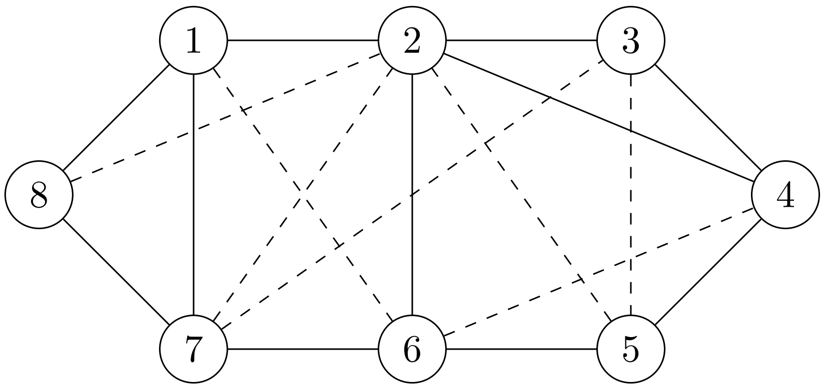
3) For $i = n, \ldots, 1$:
   Color v_i using the smallest color in $\{0, \ldots, \lambda\}$ that satisfies the L(p,q)-constraints in G

Example of (2,1)-labeling



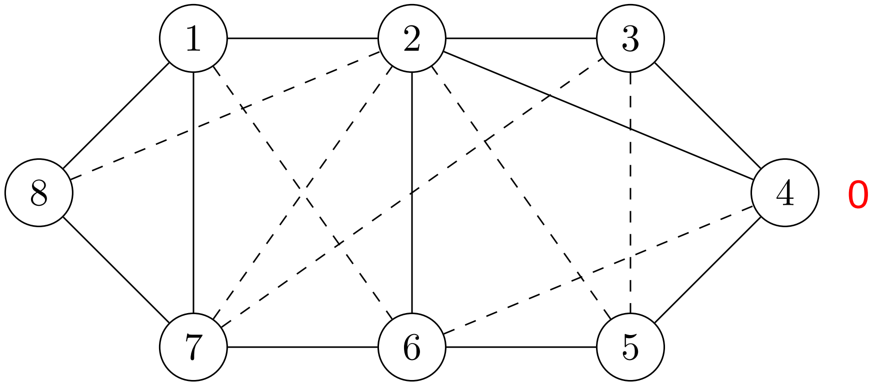
Treewidth: 3

Example of (2,1)-labeling



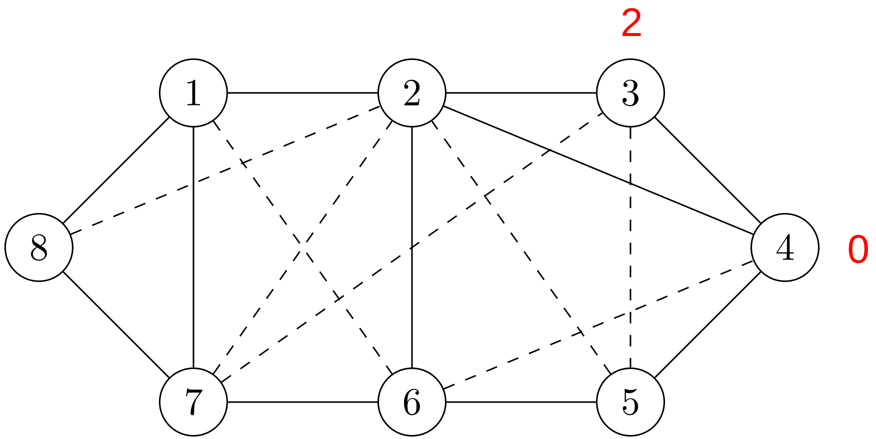
Sequence: (8,1,7,2,6,5,3,4)

Example of (2,1)-labeling



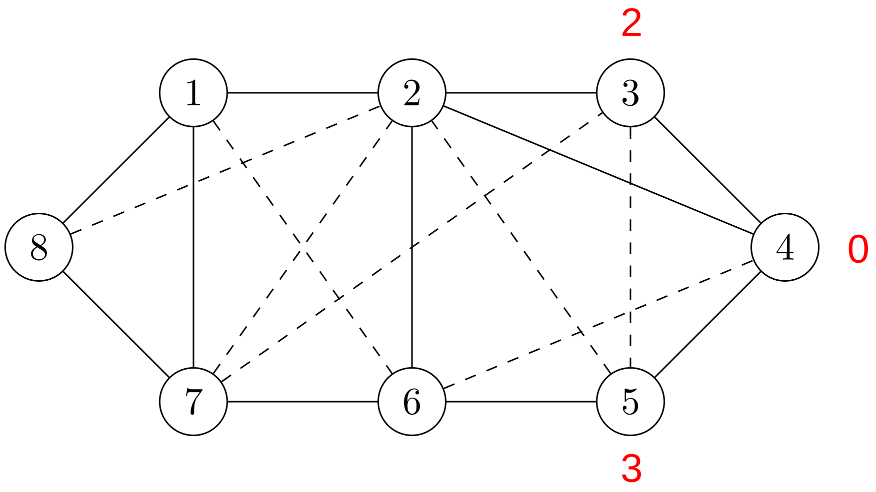
Sequence: (8,1,7,2,6,5,3,4) **Forbidden Colors: ---**

Example of (2,1)-labeling



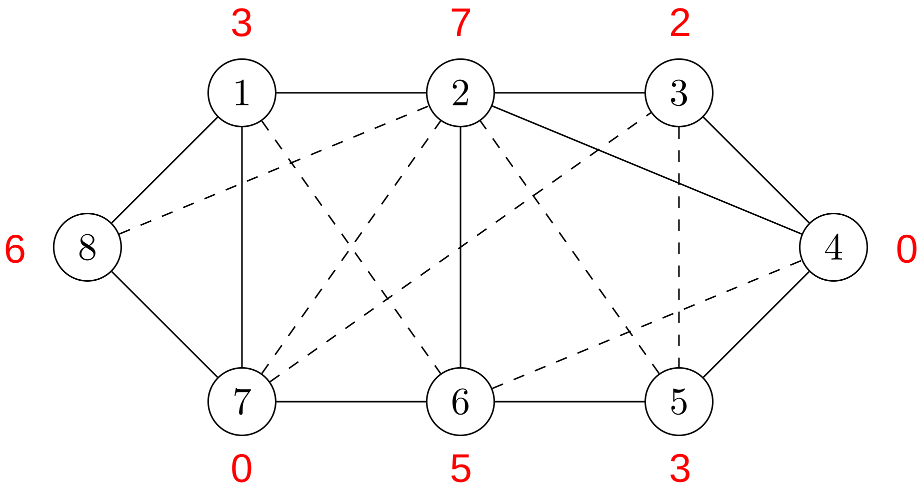
Sequence: (8,1,7,2,6,5,3,4) **Forbidden Colors: {0,1}**

Example of (2,1)-labeling



Sequence: (8,1,7,2,6,5,3,4) **Forbidden Colors: {0,1,2}**

Example of (2,1)-labeling



Sequence: (8,1,7,2,6,5,3,4) **λ ≥ 7 in order to work**

Algorithm for graphs with treewidth k

(Thm) Given any graph G of treewidth k, the previous algorithm finds:

- An $L(2, 1)$-labeling using the set $\{0, \dots, k\Delta + 2k\}$.
- An $L(1, 1)$-labeling using the set $\{0, \dots, k\Delta\}$.
- An $L(0, 1)$-labeling using the set $\{0, \dots, k\Delta - k\}$.

(Cor) Given any graph G of treewidth k it holds that:

$$\lambda_{2,1} \leq k\Delta + 2k \qquad \lambda_{1,1} \leq k\Delta \qquad \lambda_{0,1} \leq k\Delta - k$$

Algorithm for graphs with treewidth k

Dim. For each v_i there are 3 types of already-colored nodes that forbid colors to v_i:

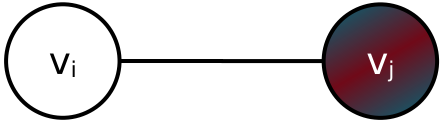
1) α vertices at distance 1 from v_i in G
2) β vertices at distance 2 from v_i in G that have a common neighbor with v_i in G that has not yet been colored
3) γ vertices at distance 2 from v_i in G that have a common neighbor with v_i in G that has already been colored

Algorithm for graphs with treewidth k

Dim. (cont.)

Let v_j be a node with $i < j$:

1) If v_j is a type 1 node than it is one of such at most k clique-neighbors

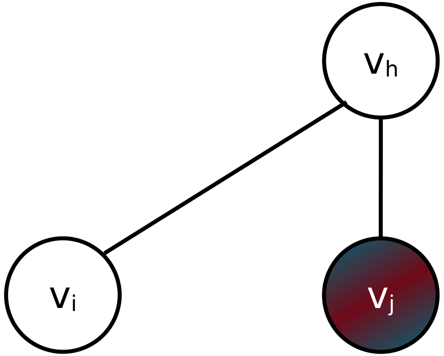


Algorithm for graphs with treewidth k

Dim. (cont.)

2) If v_j is a type 2 node and v_h is the common neighbor that has not yet been colored

$\Longrightarrow h < i < j$ and $v_i \sim v_h \sim v_j$



Algorithm for graphs with treewidth k

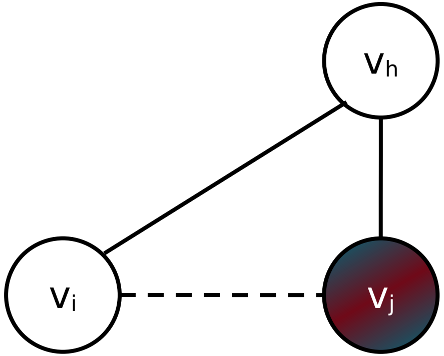
Dim. (cont.)

2) If v_j is a type 2 node and v_h is the common neighbor that has not yet been colored

   $\Longrightarrow h < i < j$ and $v_i \sim v_h \sim v_j$

   $\Longrightarrow v_i, v_j$ are in v_h's at most k clique-neighbors

   $\Longrightarrow v_j$ is one of v_i's at most k clique-neighbors

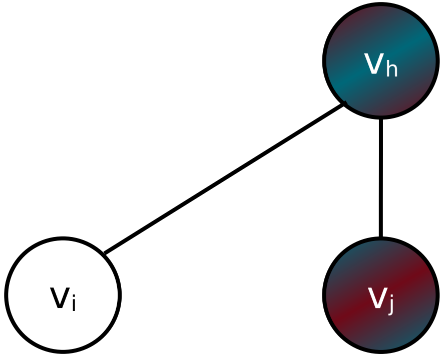


Algorithm for graphs with treewidth k

Dim. (cont.)

3) If v_j is a type 3 node and v_h is the common neighbor that has not yet been colored

$\Longrightarrow$ v_i, v_h are adjacent in H



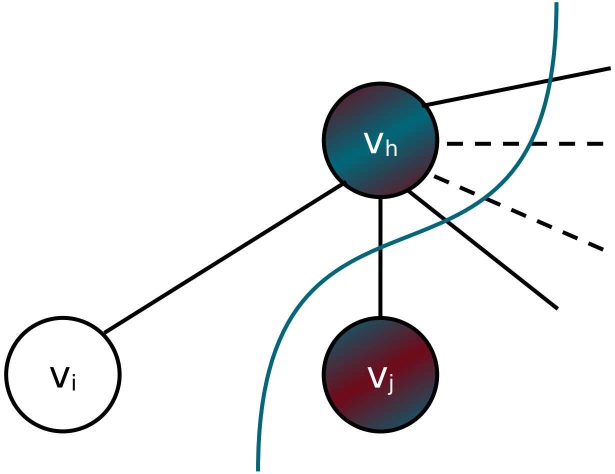
Algorithm for graphs with treewidth k

Dim. (cont.)

3) If v_j is a type 3 node and v_h is the common neighbor that has not yet been colored

$\Longrightarrow$ v_i, v_h are adjacent in H

$\Longrightarrow$ $i < h$ hence v_h may have at most $\Delta - 1$ already colored neighbors



Algorithm for graphs with treewidth k

Dim. (cont.) Each case has at least one of v_i's at most k clique-neighbors $\implies \alpha + \beta + \gamma \leq k$

Algorithm for graphs with treewidth k

Dim. (cont.) Each case has at least one of v_i's at most k clique-neighbors $\Longrightarrow \alpha + \beta + \gamma \leq k$

Let $x_{p,q}$ denote the number of colors needed to color v_i for $L(p,q)$.

Algorithm for graphs with treewidth k

Dim. (cont.) Each case has at least one of v_i's at most k clique-neighbors $\Longrightarrow \alpha + \beta + \gamma \leq k$

Let $x_{p,q}$ denote the number of colors needed to color v_i for L(p,q). Then:

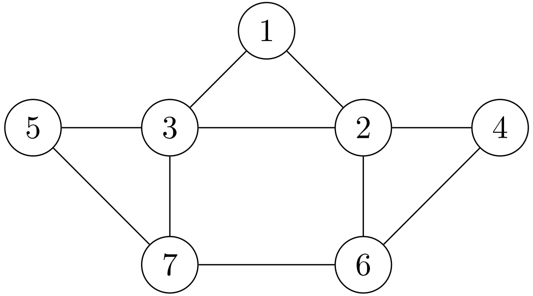
$X_{2,1} \leq 1 + 3\alpha + \beta + \gamma(\Delta\text{-}1) \leq 1 + k\Delta + 2k \quad \Longrightarrow \{0, \ldots, k\Delta + 2k\}$ suffices for L(2,1)

$X_{1,1} \leq 1 + \alpha + \beta + \gamma(\Delta\text{-}1) \quad \leq 1 + k\Delta \quad \Longrightarrow \{0, \ldots, k\Delta\}$ suffices for L(1,1)

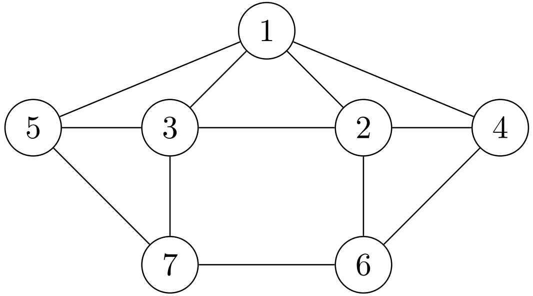
$X_{0,1} \leq 1 + \beta + \gamma(\Delta\text{-}1) \quad \leq 1 + k\Delta - k \quad \Longrightarrow \{0, \ldots, k\Delta - k\}$ suffices for L(0,1)

Outerplanar graphs

Outerplanar: the graph has a planar embedding where all the vertices on the exterior face.



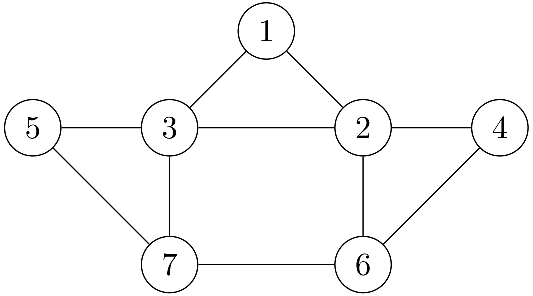
Outerplanar



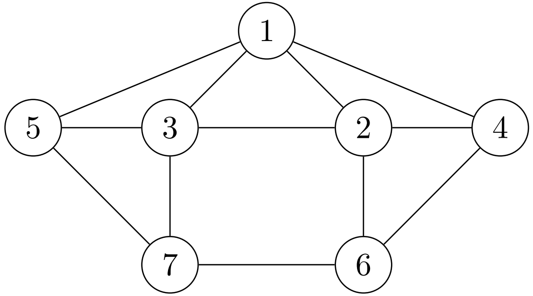
Planar

Outerplanar graphs

Outerplanar: the graph has a planar embedding where all the vertices on the exterior face.



Outerplanar



Planar

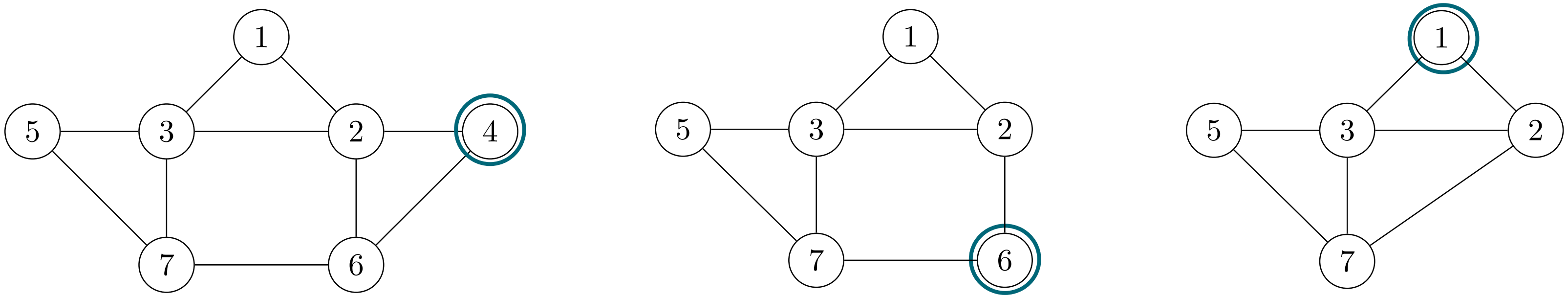
(Lem) Any outerplanar graph has either a node with degree at most 1 or a node with degree at most 2 who has a neighbor of degree at most 4

Algorithm for outerplanar graphs

Given (G, λ, p, q) in input:

1) Let $H = G$, let $S = (v_1, \ldots, v_n)$
2) For $i = 1, \ldots, n$:
    1) Let u be a vertex s.t. $\deg(u) \leq 1$ or $[\deg(u) \leq 2$ and $u \sim v$ where $\deg(v) \leq 4]$
    a) If u has two neighbors x,y not adjacent in H then add the edge (x,y) in H
    b) Set $v_i = u$ and remove u from H
3) For $i = n, \ldots, 1$:
   Color v_i using the smallest color in $\{0, \ldots, \lambda\}$ that satisfies the $L(p,q)$-constraints in G

Example of Sequence Construction



Final Sequence: (3,5,7,2,6,1,4)

Algorithm for outerplanar graphs

(Thm) Given any outerplanar graph G, the previous algorithm finds:

- An $L(2, 1)$-labeling using the set $\{0, \ldots, \Delta + 8\}$.
- An $L(1, 1)$-labeling using the set $\{0, \ldots, \Delta + 4\}$.
- An $L(0, 1)$-labeling using the set $\{0, \ldots, \Delta + 2\}$.

(Cor) Given any outerplanar graph G it holds that:

$\lambda_{2,1} \leq \Delta + 8$ $\lambda_{1,1} \leq \Delta + 4$ $\lambda_{0,1} \leq \Delta + 2$

Algorithm for outerplanar graphs

(Dim) In each iteration, H is always outerplanar $\implies$ The lemma always works

Algorithm for outerplanar graphs

(Dim) In each iteration, H is always outerplanar $\implies$ The lemma always works

Assume the sequence always picks a node v_i with degree ≤ 2 with a neighbor u of degree ≤ 4,

Algorithm for outerplanar graphs

(Dim) In each iteration, H is always outerplanar $\Longrightarrow$ The lemma always works

Assume the sequence always picks a node v_i with degree ≤ 2 with a neighbor u of degree ≤ 4,

We have at most $3 + \Delta - 1$ already-colored neighbors at distance 2 from v_i

Algorithm for outerplanar graphs

(Dim) In each iteration, H is always outerplanar $\Longrightarrow$ The lemma always works

Assume the sequence always picks a node v_i with degree ≤ 2 with a neighbor u of degree ≤ 4,

We have at most $3 + \Delta - 1$ already-colored neighbors at distance 2 from v_i

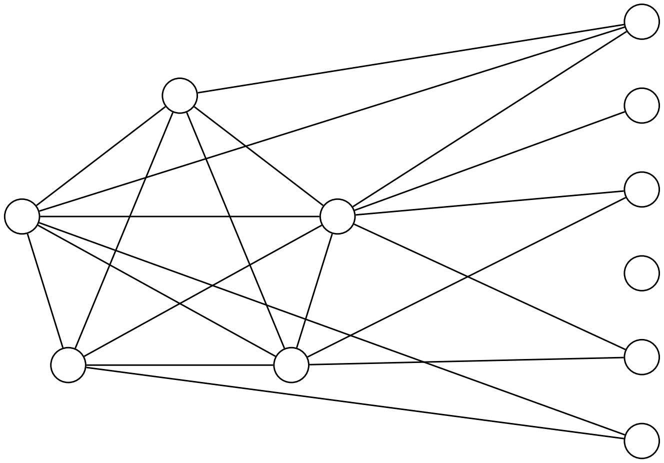
$X_{2,1} \leq 1 + 6 + (3 + \Delta - 1) = 1 + \Delta + 8$ $\Longrightarrow$ $\{0, \ldots, \Delta + 8\}$ suffices for L(2,1)

$X_{1,1} \leq 1 + 2 + (3 + \Delta - 1) = 1 + \Delta + 4$ $\Longrightarrow$ $\{0, \ldots, \Delta + 4\}$ suffices for L(1,1)

$X_{0,1} \leq 1 + (3 + \Delta - 1) = 1 + \Delta + 2$ $\Longrightarrow$ $\{0, \ldots, \Delta + 2\}$ suffices for L(0,1)

Split graphs

A split graph is a graph whose node set can be partitioned into two sets K and S such that K is a clique and S is an independent set in G.



Upper bounds for split graphs

(Thm) Given any graph G of treewidth k, there is an algorithm that finds:

- An L(2, 1)-labeling using the set $\{0, ..., \frac{1}{2}\Delta^{1.5} + 2\Delta\}$.
- An L(1, 1)-labeling using the set $\{0, ..., \frac{1}{2}\Delta^{1.5} + \Delta\}$.
- An L(0, 1)-labeling using the set $\{0, ..., \frac{1}{2}\Delta^{1.5}\}$.

(Cor) Given any split graph G it holds that:

$$\lambda_{2,1} \leq \frac{1}{2}\Delta^{1.5} + 2\Delta \qquad \lambda_{1,1} \leq \frac{1}{2}\Delta^{1.5} + \Delta \qquad \lambda_{0,1} \leq \frac{1}{2}\Delta^{1.5}$$

Lower bounds for split graphs

(Thm) For any $\Delta > 0$, there is a split graph with maximum degree Δ such that:

$$\lambda_{2,1} \geq \lambda_{1,1} \geq \lambda_{0,1} \geq \frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5}$$

Note: a lower value may suffice for some split graphs

Authors' Conjecture: for all split graphs we have that $\lambda_{2,1}$, $\lambda_{1,1}$, $\lambda_{0,1} = \Omega(\Delta^{1.5})$

Lower bounds for split graphs

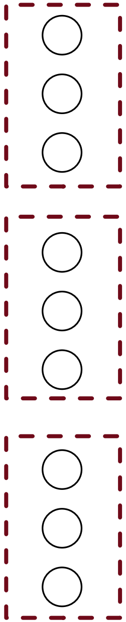
Dim.

Take an independent set S with $k=\sqrt{\frac{2}{3}\Delta}$ groups of $\frac{1}{3}\Delta$ nodes

$\Longrightarrow$ $\frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5}$ total nodes in S

$\Longrightarrow$ $\frac{k(k-1)}{2} \leq \frac{1}{3}\Delta$ distinct pairs of groups

$\Delta = 10$

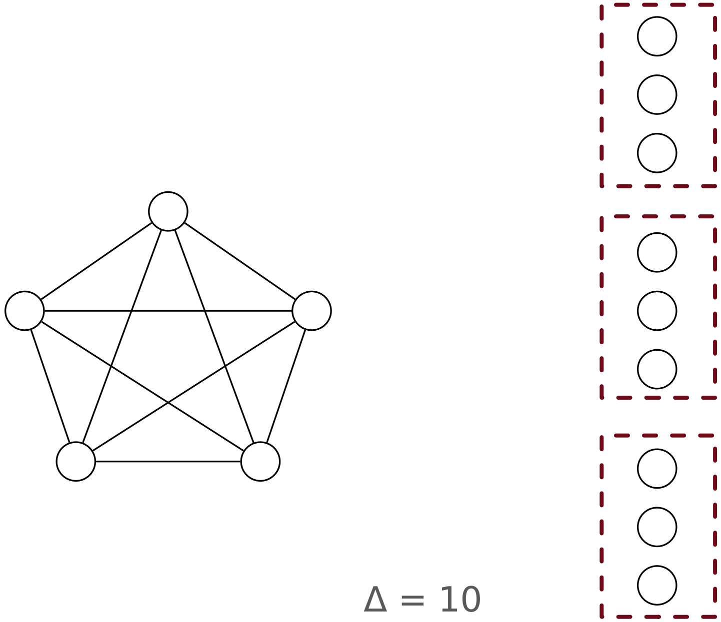


Lower bounds for split graphs

Dim.

Add a clique with $\frac{1}{3}\Delta+1$ nodes

Connect each pair with an unique node



Lower bounds for split graphs

Dim.

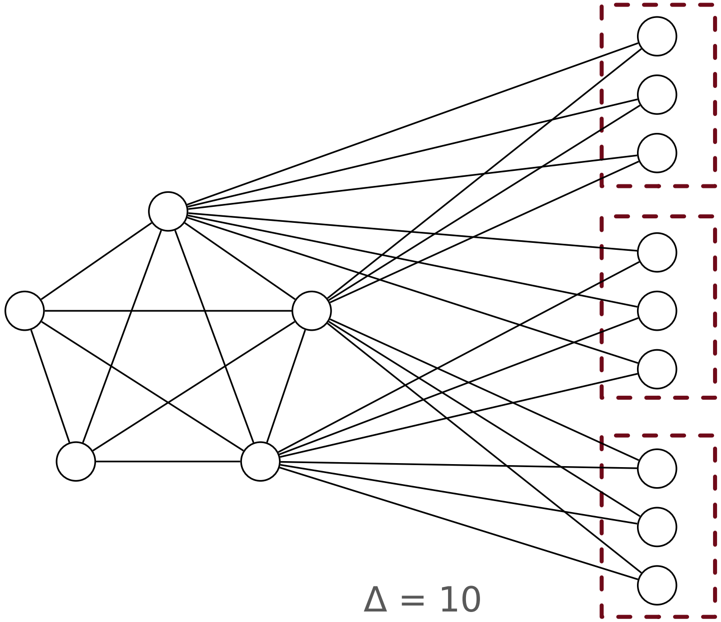
Add a clique with $\frac{1}{3}\Delta+1$ nodes

Connect each pair with an unique node

$\implies$ Each node of the clique has degree Δ

$\implies$ Each pair of nodes in S has distance 2

$\implies$ We need $\frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5}$ colors for S



Recap

Graphs with Treewidth k:

$\Delta + 1 \leq \lambda_{2,1} \leq k\Delta + 2k$ $\Delta \leq \lambda_{1,1} \leq k\Delta$ $\Delta - 1 \leq \lambda_{0,1} \leq k\Delta\text{-}k$

Outerplanar graphs:

$\Delta + 1 \leq \lambda_{2,1} \leq \Delta + 8$ $\Delta \leq \lambda_{1,1} \leq \Delta + 4$ $\Delta - 1 \leq \lambda_{0,1} \leq \Delta + 2$

Split graphs:

$\frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5} \leq \lambda_{2,1} \leq \frac{1}{2}\Delta^{1.5} + 2\Delta$ $\frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5} \leq \lambda_{1,1} \leq \frac{1}{2}\Delta^{1.5} + \Delta$ $\frac{1}{3}\sqrt{\frac{2}{3}}\Delta^{1.5} \leq \lambda_{0,1} \leq \frac{1}{2}\Delta^{1.5}$

Main references

- Hans L. Bodlaender, Ton Kloks, Richard B. Tan, et al. “Approximations for Lambda-Colorings of Graphs”. In: The Computer Journal (2004)

- Jerrold R. Griggs and Roger K. Yeh. “Labelling Graphs with a Condition at Distance 2”. In: SIAM Journal on Discrete Mathematics (1992)

- Frederic Havet, Bruce Reed, and Jean-Sebastien Sereni. “L(2,1)-labelling of graphs”. In: ACM-SIAM symposium on Discrete algorithms (2008)

- H. P. Patil. “On the structure of k-trees”. In: Journal of Combinatorics, Information and System Sciences, (1986)

Thank you for the attention!