Algorithmic Graph Theory

Section 7. Vertex coloring

Motivation

Consider the following problem that arises in wireless networking:

Frequency assignment:
Instance: Given is a number of antennas for a mobile communication network. For every pair of antennas, we know whether they are located close enough to cause interference, when operating on the same frequency.
Question: When assigning frequencies to antennas, what is the minimum number of frequencies that is needed to ensure there is no interference?

Modeled as a graph problem:

- Vertices represent antennas
- An edge uv indicates that u and v may interfere.
- A frequency assignment is represented by a *coloring* of the vertices.
- An assignment is *feasible* (i.e. no interference) if adjacent vertices receive different colors.
- *Objective:* minimize the number of colors used.

Instead of minimizing the number of frequencies, one may also ask:
- When k frequencies are assigned uniformly at random to the antennas, what is the probability that there is no interference?

A different application: *scheduling tasks that share resources*:
- The vertices represent tasks that should be assigned to time slots.
- Two vertices are adjacent if the tasks share a common resource (i.e. cannot be carried out simultaneously).
- What is the minimum number of time slots that is required to finish all tasks?

Definitions

- A *vertex coloring* or *coloring* for a graph $G = (V, E)$ is an assignment $\phi : V \to C$ such that for all $uv \in E$, $\phi(u) \neq \phi(v)$.
Elements of C are called *colors*. Usually we take $C = \{1, \dots, k\}$.

- A graph G is called *k-colorable* if there exists a coloring $\phi : V \to C$ with $|C| = k$.

- The *chromatic number* $\chi(G)$ of G is the minimum k such that G is k-colorable.
If $\chi(G) = k$, then G is also called *k-chromatic*.

Observation 7.1

Any k-coloring can be represented by a partition of the vertices into k independent sets, and vice versa.

Examples

Q: What is $\chi(G)$ if G is a tree?

Q: What is $\chi(K_{n,m})$?

Observation 7.2

$\chi(G) \leq 2$ *if and only if* G *is bipartite.*

Q: What is $\chi(C_n)$?

Q: What is $\chi(K_n)$?

- The *clique number* $\omega(G)$ is the largest k such that G contains a K_k subgraph.

Observation 7.3

$\chi(G) \geq \omega(G)$.

Relation to edge coloring

- Let $G = (V, E)$ be a graph. Then its *line graph $L(G)$* has vertex set E, and $e, f \in E$ are adjacent in $L(G)$ if they share an end in G.
H is called a *line graph* if there exists a simple graph G such that $H = L(G)$.

Observation 7.4

For all G: $\chi(L(G)) = \chi'(G)$.

- Hence edge coloring is a special case of vertex coloring.

Proposition 7.5

If H is a line graph, then $\omega(H) \leq \chi(H) \leq \omega(H) + 1$.

Proposition 7.6

Let G be a graph, and let $\chi = \chi(G)$ and $m = |E(G)|$. Then $m \geq \frac{1}{2}\chi(\chi - 1)$.

Greedy coloring

Greedy coloring:
A simple algorithm for coloring $G = (V, E)$ with colors $C = \{1, \dots, k\}$:
- Number (order) the vertices $V = \{v_1, \dots, v_n\}$.
- For $i = 1, \dots, n$: color v_i with the lowest color that is not yet used for its neighbors.

• When using an arbitrary vertex order, or a given vertex order, this algorithm clearly terminates in time $O(m)$.

• The number of colors used depends strongly on the chosen vertex order, but any order will give a color using at most $\Delta(G) + 1$ colors.

Observation 7.7

$\chi(G) \leq \Delta(G) + 1.$

Observation 7.8

For every graph $G = (V, E)$, there exists an order for V such that the greedy coloring algorithm uses exactly $\chi(G)$ colors.

Proof sketch: Let $k = \chi(G)$, and let $\phi : V \rightarrow \{1, \ldots, k\}$ be an optimal coloring.

Order the vertices $v_1, \ldots, v_n$ such that for all $i < j$, $\phi(i) \leq \phi(j)$.

Let $\psi : V \rightarrow \{1, \ldots, k\}$ be a coloring obtained by greedy coloring, using this order. Then by induction over i, it can be shown that $\psi(v_i) \leq \phi(v_i)$, so ψ uses $k = \chi(G)$ colors as well. $\square$

- Observation 7.8 gives a method to determine $\chi(G)$ (try all orders), however it is very inefficient.

Q: Are there examples of graphs G with $\chi(G) = \Delta(G) + 1$?

A: Yes, K_n and C_n with n odd.
It turns out these are the only (connected) examples:

Theorem 7.9 (Brooks, 1941 (+Lovász, 1975))

Let G be a connected graph that is not a complete graph or an odd cycle. Then $\chi(G) \leq \Delta(G)$. Furthermore, such a coloring can be found in polynomial time.

Proof: Omitted. $\square$

Q: Is there a smart way to chose the vertex order for greedy coloring?

A: Yes: informally, color vertices with high degree first. More precisely:

Observation 7.10

If the vertices of a graph G can be numbered $v_1, \ldots, v_n$ such that for every i, $|N(v_i) \cap \{v_1, \ldots, v_{i-1}\}| \leq k$, then $\chi(G) \leq k+1$.

- The *degeneracy* $\deg(G)$ of G is the maximum of $\delta(H)$ over all subgraphs H of G.

Proposition 7.11

For every G, $\chi(G) \leq \deg(G) + 1$.

- For a good heuristic, one may also choose a dynamic order: at any point, color the uncolored vertex that currently has the highest number of different colors in its neighborhood.

Q: Are there also graphs for which greedy coloring may go very wrong?

A: Yes, consider a graph G obtained from $K_{n,n}$ by deleting a perfect matching. Then $\chi(G) = 2$, but greedy coloring using the wrong vertex order may require n colors!

Q: The only graph we saw until now with high chromatic number is K_n, with $\chi(K_n) = n$.
Is it also true that every graph with high chromatic number contains a K_n subgraph?

A: No: there are graphs with arbitrary high $\chi(G)$ that contain no K_3 subgraphs (i.e. triangles).

Theorem 7.12

For every $k \in \mathbb{N}^+$ there exist graphs G with $\chi(G) = k$ without triangles.

Proof: Induction over k. For $k \in \{1, 2\}$, K_k satisfies the conditions.

Now assume G_k is a graph without triangles with $\chi(G_k) = k$, with $V(G_k) = \{v_1, \ldots, v_n\}$ (induction).

Construct G_{k+1} from G_k by adding vertices $u_1, \ldots, u_n$ with $N(u_i) = N_{G_k}(v_i)$ for all i. Next, add a vertex v with $N(v) = \{u_1, \ldots, u_n\}$.

- G_{k+1} contains no triangles. (The u_i vertices form an indep. set.)
- $\chi(G_{k+1}) \leq \chi(G_k) + 1$. (Set $\phi(u_i) := \phi(v_i)$ and $\phi(v) := k + 1$.)
- $\chi(G_{k+1}) \geq k + 1$:

Suppose G_{k+1} admits a k-coloring ϕ. W.l.o.g. assume $\phi(v) = k$.
For every v_i with $\phi(v_i) < k$, set $\psi(v_i) := \phi(v_i)$.
For every v_i with $\phi(v_i) = k$, set $\psi(v_i) := \phi(u_i)$. ($\phi(u_i) < k$.)
This gives a $(k-1)$-coloring ψ of G_k, a contradiction. □

- The graphs constructed in the previous proof contained no triangles, but did contain 4-cycles.

Erdös proved a surprising and strong result, showing that the result can be strengthened to exclude any fixed length cycle:

- The *girth* of a graph is the length of a shortest cycle.

Theorem 7.13 (Erdös, 1959)

For every $k \in \mathbb{N}^+$, there exists a graph G with girth at least k and $\chi(G) \geq k$.

Proof: Omitted. (The proof is non-constructive!) □

Lower and upper bounds for $\chi(G)$ - summary

- Although in some cases we can give good lower and upper bounds for $\chi(G)$, in general these bounds can be very far off.

For instance:

- There exist graphs with arbitrary high $\delta(G)$, but $\chi(G) = 2$: $K_{n,n}$.

- There exist graphs with arbitrary high $\chi(G)$, but which locally look like trees (Erdös).

Algorithms

- In view of the previous results, it is not surprising that determining $\chi(G)$ is hard:

Theorem 7.14

Deciding whether $\chi(G) = 3$ is NP-complete.

Proof: See Theoretische Informatik II. $\square$

- In fact, even approximating $\chi(G)$ is rather hopeless:

Vertex Coloring:
Instance: A graph $G = (V, E)$ with $|V| = n$.
Feasible solution: A (vertex) coloring $\phi : V \to C$ of G.
Objective value: The number of colors $|C|$.
Goal: Minimize $|C|$.

- An n-approximation algorithm for Vertex Coloring is trivial: color every vertex with a different color.

- Surprisingly, one cannot do significantly better!

Theorem 7.15 (Zuckermann, 2007, based on Feige and Kilian, 1998)

For every $\epsilon > 0$, there is no $n^{1-\epsilon}$-approximation algorithm for Vertex Coloring, *unless P=NP.* *($n = |V(G)|$.)*

Q: If approximating Vertex Coloring seems hopeless, how can this problem be approached?

A1: Restrict the problem to certain graph classes.

A2: Use algorithms that may require exponential time in the worst case, but fast in other cases.

A3: Use heuristics based on the above approaches.

Graph classes

Q: For which graph classes can VERTEX COLORING be solved efficiently, or be approximated well?

- For *bipartite graphs* we can compute $\chi(G)$ efficiently.
- If $\Delta(G) \leq 3$, we can compute $\chi(G)$ efficiently (Theorem 7.9).
- For line graphs $H = L(G)$, there exists a $4/3$-approximation algorithm (assuming G is known) (Proposition 6.25, Observation 7.4).
- If G is planar: We know that $\chi(G) \leq 4$ (Four Color Theorem, see Theoretische Informatik III). A 4-coloring can be found efficiently.

This gives a $4/3$-approximation algorithm. (Why?)

Since even for planar graphs it is NP-complete to decide whether the graph is 3-colorable (Garey, Johnson, Stockmeyer, 1976), the approximation ratio $4/3$ is best possible, unless P=NP.

- Later in Section 7: *chordal graphs*, which are a special case of *perfect graphs*.
- Next year: *graphs of bounded treewidth*.

Algorithmic Graph Theory

Section 7. Vertex coloring

Chromatic polynomials - definition and motivation

Q: Given is a graph G in which the n vertices correspond to antennas, and edges indicate possible interference.
What is the probability that a random assignment of x frequencies to the antennas gives no interference?

A:

$$\frac{\text{Number of } x\text{-colorings of } G}{\text{Number of functions } \phi : V \to \{1, \ldots, x\}} = \frac{P_G(x)}{x^n}.$$

• By $P_G(x)$ we denote the number of x-colorings of the graph G. P_G is called the *chromatic polynomial* of G.

Observation 7.16

If P_G is known, $\chi(G)$ can be computed in polynomial time.

Examples

Examples: compute $P_G(x)$ for:

- $G = \overline{K_n}$. (An edgeless graph on n vertices.)

$P_G(x) = x^n$.

- $G = K_n$.

$P_G(x) = x \cdot (x-1) \cdot \ldots \cdot (x-n+1)$.

- G is a tree on n vertices.

$P_G(x) = x \cdot (x-1)^{n-1}$.

Direct method for computing $P_G(x)$

- A vertex $v \in V(G)$ is called a *simplicial vertex* if $G[N(v)]$ is a complete graph.

Proposition 7.17

If $v \in V(G)$ is a simplicial vertex of G, and $d(v) = k$, then $P_G(x) = (x - k) \cdot P_{G-v}(x)$.

Q: What can we do in case $G = C_4$? (No simplicial vertex.)

- By $G \cdot uv$ we denote the graph obtained from G by identifying the vertices u and v (such that the resulting graph is simple).

Proposition 7.18

Let $G = (V, E)$ be a graph with distinct $u, v \in V$ and $uv \notin E$. Then $P_G(x) = P_{G+uv}(x) + P_{G \cdot uv}(x)$.

Q: For some examples (such as C_6), this requires many steps. Is there an alternative?

Proposition 7.19

Let $G = (V, E)$ be a graph with $uv \in E$. Then $P_G(x) = P_{G-uv}(x) - P_{G \cdot uv}(x)$.

Method for computing $P_G(x)$

- Using the Propositions 7.18 and 7.19, computing $P_G(x)$ can be reduced to computing $P_H(x)$ for a number of simpler graphs H, for which Proposition 7.17 can be applied.

- The number of steps depends on the choices of u and v.

- For certain graphs (e.g. graphs with large girth, but many cycles), this method requires an exponential number of steps in any case.

Theorem 7.20

Let G be a simple graph on n vertices with m edges and k components. Then $P_G(x)$ is a polynomial with alternating signs, and

$$P_G(x) = x^n - mx^{n-1} + \ldots \pm cx^k.$$

Proof: By induction over m, using the fact that $P_G(x) = P_{G-uv}(x) - P_{G\cdot uv}(x)$ (Proposition 7.19).

For $G = \overline{K_n}$, it holds that $P_G(x) = x^n$, so the stated properties hold.

- $P_G(x)$ is a polynomial: holds since $P_{G-uv}(x)$ and $P_{G\cdot uv}(x)$ are polynomials (induction).

- The signs of $P_G(x)$ alternate: holds since $G \cdot uv$ has one vertex fewer than $G - uv$, so they have degree $n-1$ and n respectively (induction), and their signs alternate (induction).

Proof of Theorem 7.20, continued:

- The highest degree term is x^n (with coefficient 1): Because of the number of vertices of $G - uv$ and $G \cdot uv$, the highest degree term of $P_G(x)$ is the same as the one for $G - uv$, which is x^n (induction).

- The second term has coefficient $-m$: The coefficient of x^{n-1} in $P_{G-uv}(x)$ is $-(m-1)$ (induction), and in $P_{G \cdot uv}(x)$ it is 1 (induction).

- The lowest degree of a term is k: $G - uv$ has at least k components, so its lowest degree is at least k (induction). $G \cdot uv$ has exactly k components, so its lowest degree is k. The signs match, so an x^k term remains. $\square$

Algorithmic Graph Theory

Section 7. Vertex coloring

7.1 Basic definitions and bounds

7.2 Chromatic polynomials

7.3 Chordal graphs and perfect graphs

Chordal graphs

- A graph is *chordal* if it has no induced cycles of length at least 4.

- If C is a cycle (subgraph) of G, and $u, v \in V(C)$ and $uv \in E(G) \setminus E(C)$, then the edge uv is called a *chord* of C.

- So in chordal graphs, all cycles of length at least 4 have a chord.

Reasons to study chordal graphs:

- They are an example of the important class of *perfect graphs*.
- VERTEX COLORING can be done efficiently on them.
- In many applications of VERTEX COLORING, the graphs are actually chordal. For instance:

Room assignment:
Instance: A number of meetings that should take place in a conference center is given. For every meeting, the start and end times are given.
Feasible solution: An assignment of rooms to the meetings, such that meetings that overlap in time are in different rooms.
Objective: Minimize the number of required rooms.

- This is VERTEX COLORING restricted to chordal graphs.

(Interval graphs, in fact.)

Lemma 7.21

Let X be a minimal vertex cut of a chordal graph G. Then X is a clique.

Proof: Let C^1 and C^2 be the vertex sets of different components of $G - X$. Let $G_1 = G[C^1 \cup X]$ and $G_2 = G[C^2 \cup X]$.

Suppose X contains nonadjacent vertices x and y.

By minimality of X, both x and y are adjacent to both C^1 and C^2, so we can find two shortest (x, y)-paths P^1 and P^2 in $G_1 - (X \setminus \{x, y\})$ and $G_2 - (X \setminus \{x, y\})$ respectively.

P^1 and P^2 form a chordless cycle together, since:

- they are shortest paths,
- they have no internal vertices in X,
- X separates C^1 from C^2, and
- x and y are nonadjacent.

This contradicts that G is chordal, so X is a clique of G. □

• The reason that VERTEX COLORING can be solved efficiently on chordal graphs is that *every chordal graph G contains a simplicial vertex v*, and clearly $G - v$ is again chordal. Then Proposition 7.17 can be applied.

Q: But how to prove this fact?

A: We need to prove a slightly stronger statement:

Lemma 7.22

A connected chordal graph is either a complete graph, or it contains two nonadjacent simplicial vertices.

Proof of Lemma 7.22: By induction over $n = |V(G)|$.

If $n = 1$ or $n = 2$, then G is a complete graph.

Suppose now that $n \geq 3$. If G is not complete, we can select nonadjacent vertices s and t, and a minimal (s, t)-separating vertex cut $X \subseteq V(G)$. By Lemma 7.21, X is a clique.

Let C_s and C_t be the vertex sets of the components of $G - X$ that contain s and t, respectively. Let $G_s = G[C_s \cup X]$ and $G_t = G[C_t \cup X]$.

Both G_s and G_t are *induced* subgraphs of G, so they are again chordal.

We use the induction assumption for G_s:
If G_s is a clique, then s is simplicial.
Otherwise, G_s contains two nonadjacent simplicial vertices.
At least one of these is therefore part of $G_s - X$. (Since X is a clique.)
This is a simplicial vertex of G as well. (Since X is a cut.)

Similarly, by applying induction for G_t we find a second simplicial vertex of G in $G_t - X$. $\square$

• A *perfect elimination order* is an order $v_1, \ldots, v_n$ of the vertices of a graph such that for every i, v_i is a simplicial vertex of $G[\{v_1, \ldots, v_i\}]$.

Theorem 7.23

A graph is chordal if and only if it admits a perfect elimination order.

Proof: $\Rightarrow$: By Lemma 7.22, G has a simplicial vertex v. The graph $G - v$ is again chordal, so by induction it admits a perfect elimination order $v_1, \ldots, v_{n-1}$. Then $v_1, \ldots, v_{n-1}, v$ is a perfect elimination order for G.

$\Leftarrow$: Let $v_1, \ldots, v_n$ be a perfect elimination order for G, and let $v = v_n$. Since v is a simplicial vertex, it cannot be part of a chordless cycle of length at least 4. (Any two neighbors of v are adjacent, which would give a chord.) $v_1, \ldots, v_{n-1}$ is a perfect elimination order for $G - v$, so by induction, $G - v$ neither contains a chordless cycle of length at least 4. It follows that there are no such cycles in G, so G is chordal. $\square$

Proposition 7.24

In polynomial time, it can be tested whether a graph is chordal, and if so, a perfect elimination order can be found.

- Using Proposition 7.17 we get:

Proposition 7.25

If G is chordal, then in polynomial time $P_G(x)$ can be computed.

- Using Observation 7.16:

Corollary 7.26

If G is chordal, then in polynomial time $\chi(G)$ can be computed.

Proposition 7.27

If G is chordal, then $\chi(G) = \deg(G) + 1 = \omega(G)$.

Proof: Let H be a subgraph of G with $\delta(H)$ maximum, so $\deg(G) = \delta(H)$.

W.l.o.g. H is an induced subgraph of G (adding edges does not decrease $\delta(H)$), so H is chordal.
Then H has a simplicial vertex v (Lemma 7.22). $\{v\} \cup N(v)$ is a clique in H, so $\omega(G) \geq \omega(H) \geq d(v) + 1 \geq \delta(H) + 1 = \deg(G) + 1$.

Furthermore, $\chi(G) \geq \omega(G)$ (Observation 7.3) and $\deg(G) + 1 \geq \chi(G)$ (Proposition 7.11), so

$$\chi(G) \geq \omega(G) \geq \deg(G) + 1 \geq \chi(G).$$

It follows that all inequalities are equalities. □

Perfect graphs

- A graph G is *perfect* if for every induced subgraph H of G, $\chi(H) = \omega(H)$.

- Since induced subgraphs of chordal graphs are chordal, Proposition 7.27 shows that chordal graphs are perfect.

Q: Do we know more perfect graphs?

Observation 7.28

Bipartite graphs are perfect.

Proposition 7.29

If G is bipartite, then its line graph $L(G)$ is perfect.

Proof: Let G be a bipartite graph, and let $H = L(G)$.
Consider any induced subgraph $H' = H[M]$ of H.
(So M is a set of *vertices* of H, and a set of *edges* of G.)
Since H' is an induced subgraph of H, it holds that $H' = L(G[M])$, so $\chi(H') = \chi'(G[M])$ and $\Delta(G[M]) \leq \omega(H')$.
Let $G' = G[M]$. G' is a subgraph of G, so it is again bipartite.
Therefore, $\chi'(G') = \Delta(G)$ (Theorem 6.7). So:

$$\chi(H') = \chi'(G') = \Delta(G') \leq \omega(H').$$

Since $\chi(H') \geq \omega(H')$, it follows that $\chi(H') = \omega(H')$.
This holds for any induced subgraph H' of H, so H is perfect. $\square$

Complements

• Recall that the *complement* $\overline{G}$ of a graph $G = (V, E)$ has vertex set V, and edge set $\{uv \mid uv \notin E\}$.

• By $\alpha(G)$ we denote the maximum size of an independent set in G.

• Cliques in G are independent sets in $\overline{G}$, so $\alpha(\overline{G}) = \omega(G)$.

• For colorings of $G = (V, E)$, there is also an analogous notion for $\overline{G}$. Recall that a coloring is a partition of V into *independent sets*:

• A *k-clique cover* or *clique cover* of $G = (V, E)$ is a partition of V into sets $S_1, \ldots, S_k$ such that every S_i is a clique.
The *clique cover number* $\overline{\chi}(G)$ is the minimum k such that G admits a k-clique cover.

• k-colorings of G are k-clique covers of $\overline{G}$, so $\overline{\chi}(\overline{G}) = \chi(G)$.

• Clearly, $\overline{\chi}(G) \geq \alpha(G)$. (See also Observation 7.3)

Theorem 7.30

The complement $\overline{G}$ of a bipartite graph G is perfect.

Proof: It suffices to show that for any induced subgraph H of G, $\alpha(H) = \overline{\chi}(H)$.
To that end, we will show that for any bipartite graph H, $\alpha(H) = \overline{\chi}(H)$.

Let $H = (V, E)$ and $n = |V|$. Let S be a minimum vertex cover of H and let M be a maximum matching of H.
By Theorem 5.6, $|S| = |M|$, since H is bipartite. Let $k = |S| = |M|$.
Since S is a vertex cover, $V \setminus S$ is an independent set, so $\alpha(H) \geq |V \setminus S| = n - k$.

Let X be the set of vertices unmatched by M, so $|X| = n - 2k$.
Taking all the vertices in X as single-vertex cliques and all edges in M as two-vertex cliques gives a clique cover consisting of $|X| + |M| = n - 2k + k = n - k$ cliques.
So $\overline{\chi}(H) \leq n - k \leq \alpha(H)$.
Since $\overline{\chi}(H) \geq \alpha(H)$, this shows that $\overline{\chi}(H) = \alpha(H)$. □

- The previous proposition can actually be generalized to all perfect graphs:

Theorem 7.31 (Weak Perfect Graph Theorem (Lovász, 1972))

A graph is perfect if and only if its complement is perfect.

Proof: Omitted. □

- An *odd hole* is an induced odd cycle of length at least 5.

Theorem 7.32 (Strong Perfect Graph Theorem (Chudnovsky, Robertson, Seymour, Thomas, 2006))

A graph G is perfect if and only if neither G nor $\overline{G}$ contain odd holes.

Proof: $\Rightarrow$: If G contains an odd hole H, then $\chi(H) = 3$ and $\omega(H) = 2$, which shows that G is not perfect.
Similarly, if $\overline{G}$ contains an odd hole, then $\overline{G}$ is not perfect and therefore G is not perfect (Theorem 7.31).
Alternatively, if S induces an odd hole in $\overline{G}$ on $2k+1$ vertices, then it can easily be shown that $\chi(G[S]) > \omega(G[S])$, again showing that G is not perfect.

$\Leftarrow$: Omitted! □

- The strong perfect graph theorem implies the weak perfect graph theorem.

• The algorithmic results for chordal graphs can also be generalized to all perfect graphs:

Theorem 7.33

If G is perfect, then in polynomial time we can find a maximum clique, maximum independent set, and vertex coloring using $\chi(G)$ colors.

Proof: Omitted. □

Vertex colorings and edge colorings - summary

- Finding a coloring that uses the minimum number of colors is NP-hard in both the vertex and edge case.

- However, for edge colorings (in simple graphs) we have the best possible approximation result: in polynomial time we can find an edge coloring using at most $\chi'(G) + 1$ colors.
On the other hand, for vertex colorings, efficiently approximating $\chi(G)$ is basically hopeless.

- Bounds for χ': for simple graphs G it holds that $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ (Vizing). For bipartite (multi)graphs G, $\Delta(G) = \chi'(G)$.

- Bounds for χ: the most important bounds are $\omega(G) \leq \chi(G) \leq \deg(G) + 1$. These may however be very far apart.

- For *chordal graphs*, we can efficiently determine $\chi(G)$, and $\chi(G) = \omega(G) = \deg(G) + 1$ holds.
For the more general class of *perfect graphs*, $\chi(G)$ can also be determined efficiently, and $\chi(G) = \omega(G)$. (However the proofs are much more complex, and have been omitted.)

- Method for computing $\chi(G)$ and even the total number of k-colorings for every k: *chromatic polynomials*.
The given method to compute $P_G(x)$ may take exponential time, but is efficient for chordal graphs. (And for graphs that are 'close to chordal graphs')