Combinatorial methods

Áron Erdélyi

2021.09.22.

Contents

1 Basic graph theory

1.1 Basic definitions

Definition 1.1 (Undirected Simple Graph). *A graph is a pair $G = (V, E)$ where*

- *V is the set of vertices,*
- *E is the set of edges.*

*Each edge connects exactly two vertices. The number $|V|$ of vertices is called the **order** of G.*

Definition 1.2 (Subgraph). *Graph $G' = (V', E')$ is a subgraph of $G = (V, E)$, if G' is a graph, moreover $V' \subseteq V$ and $E' \subseteq E$. This relation is denoted by $G' \subseteq G$.*

Definition 1.3 (Induced subgraph). *Graph $G' = (V', E')$ is an induced subgraph of $G = (V, E)$, if $V' \subseteq V$ and $E' = \{e \in E \mid e \subseteq V'\}$.*

Definition 1.4 (Neighborhood of vertex).

$$N_G(v) = \{v' \in V(G) \mid vv' \in E(G)\}.$$

Definition 1.5 (Degree). *In a simple undirected graph:*

$$d(v) = |\{e \in E(G) \mid v \in e\}| = |N_G(v)|.$$

Definition 1.6 (Regular graphs). *Graph G is k–regular if $d(v) = k \ \forall v \in V(G)$.*

Definition 1.7 (Path). *A subgraph $v_0, e_1, v_1, \ldots, e_k, v_k$ is a path from v_0 to v_k in G if the vertices $v_0, v_1, \ldots, v_k$ are all distinct, and if $k > 1$ then $e_i = \{v_{i-1}, v_i\}$ and $e_i \in E(G) \ \forall i \in \{1, 2, \ldots, k\}$. The **length** of a path is the number of it's edges.*

Definition 1.8 (Connected graph). *Graph G is connected if there exists a path from v to v' for every $v, v' \in V(G)$.*

Definition 1.9 (Component). *The induced subgraph $G[V']$ is a component of G if it is connected, and $\forall v \in V(G) \backslash V'$, the subgraph $G[V' \cup v]$ is disconnected.*

Definition 1.10 (Cycle). *A subgraph $v_0, e_1, v_1, \ldots, e_k, v_k$ is a cycle in G if $v_0 = v_k$, $e_i = \{v_{i-1}, v_i\}$, $e_i \in E(G) \ \forall i \in \{1, 2, \ldots, k\}$ and $v_i \neq v_j$ for any $i, j \in \{1, 2, \ldots, k\}$ where $i \neq j$.*

Definition 1.11 (Bipartite graph). *Graph $G = (V, E)$ is bipartite if there exists $A, B \subseteq V$ such that $A \cup B = V$, $A \cap B = \emptyset$, and $E \subseteq \{v_a v_b \mid v_a \in A, \ v_b \in B\}$.*

Definition 1.12 (Tree). *Graph G is a tree if it is connected and contains no cycle as a subgraph.*

Definition 1.13 (Complement). *The complement of graph G is defined as*

$$\overline{G} = (V(G), \{\{v_1, v_2\} \mid v_1, v_2 \in V(G), \ v_1 v_2 \notin E(G)\}).$$

1.2 Special significant graphs

Definition 1.14 (Empty graph, E_n).

$$E_n = (\{1, 2, \ldots, n\}, \emptyset).$$

Definition 1.15 (Path graph, P_n).

$$P_n = (\{1, 2, \ldots, n\}, \{\{v_i, v_{i+1}\} \mid \forall i \in \{1, 2, \ldots, n-1\}\}).$$

Definition 1.16 (Cycle graph, C_n). *For $n \geq 3$,*

$$C_n = (\{1, 2, \ldots, n\}, \{\{v_i, v_{i+1}\} \mid \forall i \in \{1, 2, \ldots, n-1\}\} \cup \{\{1, n\}\}).$$

Definition 1.17 (Complete graph, K, n).

$$K_n = (\{1, 2, \ldots, n\}, \{\{v_1, v_2\} \mid v_1, v_2 \in \{1, 2, \ldots, n\}\}).$$

Definition 1.18 (Complete bipartite graph, $K_{p,q}$).

$$K_{p,q} = (\{a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q\}, \{\{a_i, b_j\} \mid 1 \leq i \leq p, \ 1 \leq j \leq q\}).$$

1.3 Graph parameters

Definition 1.19 (Maximum degree of G, Δ).

$$\Delta(G) = \max_{v \in V(G)} d(v).$$

Definition 1.20 (Minimum degree of G, δ).

$$\delta(G) = \min_{v \in V(G)} d(v).$$

Definition 1.21 (Clique). *An induced subgraph $G[V']$ is called a clique in G if it is a complete graph.*

Definition 1.22 (Clique number, ω).

$$\omega(G) = \max_{G[V'] \text{ is a clique in } G} |V'|.$$

Proposition 1.1.

$$\omega(G) \leq \Delta(G) + 1$$

Definition 1.23 (Clique covering). *The induced subgraphs $G[V_1], G[V_2], \ldots, G[V_k]$ form a clique covering of G if $\bigcup_{i=1}^{k} V_i = V(G)$, and $G[V_i]$ is a clique forall $i \in \{1, 2, \ldots, k\}$.*

Definition 1.24 (Clique covering number, θ).

$$\theta(G) = \min_{G[V_1],\ldots,G[V_k] \text{ is a clique covering in } G} k.$$

Definition 1.25 (Independent vertex set). *A set $V' \subseteq G(V)$ of vertices is independent if $\forall v_1, v_2 \in V', \{v_1, v_2\} \notin E$.*

Definition 1.26 (Independence number, α).

$$\alpha(G) = \max_{V' \text{ is an independent set in } G} |V'|.$$

Proposition 1.2.

$$\alpha(G) \leq \theta(G).$$

Definition 1.27 (Transversal). *A set $V' \subseteq V(G)$ is a transversal of G if $\forall e \in E(G): \; e \cap V' \neq \emptyset$.*

Definition 1.28 (Transversal number, τ).

$$\tau(G) = \min_{V' \text{ is a transversal set in } G} |V'|.$$

Theorem 1.1. *For every graph G*

$$\tau(G) + \alpha(G) = |V(G)|.$$

Proof. If V' is a transversal, then $V(G) \backslash \{V'\}$ is an independent set. Assume by contradiction that it is not independent,

$$\exists v_1 v_2 \in V(G) \backslash V', \quad v_1 v_2 = e_1 \in E(G).$$

This would mean that e_1 is not covered by V', so V' is not a transversal. This means that

$$\alpha(G) \geq |V(G) \backslash V'| = |V(G)| - |V'|, \quad \forall V' \text{ transversal.}$$

If $|V'| = \tau(G)$, then $\alpha(G) \geq |V(G)| - \tau(G)$. If I is an independent set, then $V(G) \backslash I$ is a transversal. Assume by contradiction that it is not a transversal, then

$$\exists v_3, v_4 \in I, \quad e_2 = v_3 v_4 \in E(G),$$

then I is not independent.

$$\tau(G) \geq |V(G) \backslash I| = |V(G)| - |I|, \quad \forall I \text{ independent set.}$$

If $|I| = \alpha(G)$, then $\tau(G) \geq V(G) - \alpha(G)$. □

Corollary 1.1. *By the proof, if I is a maximal independent vertex set, then $V(G)\backslash I$ is a minimal transversal.*

Definition 1.29 (Vertex coloring)**.** *A mapping $\varphi : V \to \{1, 2, \ldots, k\}$ is called a coloring with k colors, or a k–coloring of G.*

Definition 1.30 (Proper vertex coloring)**.** *A coloring φ is called a proper coloring of G if $\{v_1, v_2\} \in E(G) \implies \varphi(v_1) \neq \varphi(v_2)$.*

Definition 1.31 (Chromatic number, χ)**.**

$$\chi(G) = \min_{G \text{ has a proper } k\text{–coloring}} k.$$

Proposition 1.3.

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

Definition 1.32 (Matching)**.** *A set $E' \subseteq E(G)$ is a matching if $\forall e_1, e_2 \in E' : \ e_1 \cap e_2 = \emptyset$.*

Definition 1.33 (Matching number, ν)**.**

$$\nu(G) = \max_{E' \text{ is a matching in } G} |E'|.$$

Theorem 1.2.

$$\nu(G) \leq \tau(G).$$

Definition 1.34 (Edge coloring)**.** *A mapping $\varphi : E \to \{1, 2, \ldots, k\}$ is called an edge coloring with k colors, or k–coloring of edges.*

Definition 1.35 (Proper edge coloring)**.** *An edge coloring φ is called a proper edge coloring of G, if $\forall e_1, e_2 \in E(G)$ with $e_1 \cap e_2 \neq \emptyset$ we have $\varphi(e_1) \neq \varphi(e_2)$.*

Definition 1.36 (Chromatic index, χ')**.**

$$\chi'(G) = \min_{G \text{ has a proper } k\text{–coloring of edges}} k.$$

2 Interval systems

2.1 Basic definitions

Definition 2.1 (Finite closed interval)**.**

$$[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}.$$

Definition 2.2 (Interval system)**.**

$$\mathcal{J} = \{I_i = [a, b] \mid i \in \mathcal{K}\},$$

where $\mathcal{K}$ is a set of indeces.

2.2 Helly's theorem

Theorem 2.1 (Helly's theorem, general)**.** *Let $\mathcal{F}$ be a set system containing only closed, bounded, convex sets in $\mathbb{R}^d$. If any $d+1$ members of $\mathcal{F}$ have a nonempty intersection, then the whole system has a nonempty intersection.*

If $d = 1$, we are talking about intervals.

Theorem 2.2 (Helly's theorem on intevrals)**.** *Let $\mathcal{I}$ be an interval system, where any 2 intervals share a point.*

$$\exists p \in \mathbb{R}: \ \forall I_i \in \mathcal{I}: \ p \in I_i.$$

Proof. Let $L = a_i$ be the rightmost left endpoint, and $R = b_j$ be the leftmost right endpoint of the intervals in $\mathcal{I}$.

1. If they are the endpoints of the same interval, then

$$[L, R] \subseteq \bigcap_{I_k \in \mathcal{I}} I_k.$$

2. If they are endpoints of different intervals in the set, then by assumption

$$[a_i, b_i] \cap [a_j, b_j] \neq \emptyset,$$

and by the definition of a_i and b_j

$$a_j \leq a_i, \quad \leq \quad b_j \leq b_i.$$

This means that $L \leq R$, then

$$[L, R] \subseteq \bigcap_{I_k \in \mathcal{I}} I_k.$$

□

2.3 Transversals and matchings

Definition 2.3 (Transversal of an interval system)**.** *A transversal of an interval system $\mathcal{I}$ is a set $T \subseteq \mathbb{R}$ such that*

$$\forall I_i \in \mathcal{I}: \ I_i \cap T \neq \emptyset.$$

In other words T contains at least one point in every interval.

Definition 2.4 (Transversal number of an interval system, τ)**.**

$$\tau(\mathcal{I}) = \min_{T \textit{ is a transversal set of } \mathcal{I}} |T|.$$

Definition 2.5 (Matching of an interval system)**.** *A subsystem $\mathcal{M}$ of $\mathcal{I}$ is called a matching if*

$$\forall I_i, I_j \in \mathcal{M}: \ i \neq j \implies I_i \cap I_j = \emptyset.$$

In other words the intervals in $\mathcal{M}$ are pairwise disjoint.

Definition 2.6 (Matching number of an interval system, ν)**.**

$$\nu((I)) = \max_{\mathcal{M} \text{ is a matching of } \mathcal{I}} |\mathcal{M}|.$$

Theorem 2.3. *For any set system $\mathcal{S}$,*

$$\nu(\mathcal{S}) \leq \tau(\mathcal{S}).$$

Proof. Let $\mathcal{M}$ be a maximal matching in $\mathcal{S}$. A transversal T covers $\mathcal{S}$, so it also covers $\mathcal{M}$. The pairwise disjoint elements of $\mathcal{M}$ require seperate covering points in T. This implies

$$\nu(\mathcal{S}) = |\mathcal{M}| \leq |T|.$$

This holds for every transversal, even the smallest one: $\nu(\mathcal{S}) \leq \tau(\mathcal{S})$. □

Algorithm 2.1. *Algorithm to determine τ and ν for interval systems:*

1. *Arrange the intervals in a list in the order of increasing right ends. Let $T = \emptyset$, $\mathcal{M} = \emptyset$.*
2. *Take the first interval and put it into $\mathcal{M}$.*
3. *Take the right end b_j of the first interval and put it into T.*
4. *Delete all intervals from the list that contain b_j.*
5. *If the list is not empty, go to Step 2, otherwise stop.*

Theorem 2.4. *For every interval system $\mathcal{I}$:*

$$\nu(\mathcal{I}) = \tau(\mathcal{I}).$$

Proof. Let T and M be the transversal and matching returned by Algorithm 2.1, $|T| = |M|$. By definition $\tau\mathcal{I} \leq |T|$ and $|M| \leq \nu(\mathcal{I})$. From Theorem 2.3 we know that $\nu(\mathcal{I}) \leq \tau(\mathcal{I})$.

$$\tau(\mathcal{I}) \leq |T| = |M| \leq \nu(\mathcal{I}) \leq \tau(\mathcal{I}) \implies \tau(\mathcal{I}) = \nu(\mathcal{I}).$$

□

2.4 Decomposition into intersecting subsystems

Definition 2.7 (Intersecting subsystem)**.** *A set system is called intersecting if any two members of it have a nonempty intersection.*

$k(\mathcal{S})$ is the ***minimum number of intersecting subsystems*** *the $\mathcal{S}$ system can be decomposed into.*

Theorem 2.5. *For any set system $\mathcal{S}$,*

$$\nu(\mathcal{S}) \leq k(\mathcal{S}) \leq \tau(\mathcal{S}).$$

Proof. No two disjoint sets can belong to the same intersecting. This implies

$$\nu(\mathcal{S}) \leq k(\mathcal{S}).$$

Let T be a minimal transversal ($|T| = \tau(\mathcal{S})$). $\forall x \in T$ the sets containing the point is an intersecting subsystem. Because T covers every set, taking this for all $x \in T$ se get a decomposition into intersecting subsystems of size $|T|$. A minimal decomposition may be smaller, giving us

$$k(\mathcal{S}) \leq \tau(\mathcal{S}).$$

□

Corollary 2.1. *For any interval system $\mathcal{I}$,*

$$\nu(\mathcal{I}) = k(\mathcal{I}) = \tau(\mathcal{I}).$$

Proof.

$$\nu(\mathcal{I}) \leq k(\mathcal{I}) \leq \tau(\mathcal{I}) = \nu(\mathcal{I})$$

□

Algorithm 2.2. *Algorithm to determine τ and ν and minimal decomposition of $\mathcal{I}$ into intersecting subsystems for interval systems:*
Replace Step 4 of Algorithm 2.1 with

4. *If $b_i \in I_j$, the put I_j into $K(b_i)$, and delete them from the list.*

2.5 Decomposition to matchings

Definition 2.8 (Proper coloring of set systems)**.** *A proper coloring of a set system $\mathcal{S}$ is a coloring of the sets, such that $\forall A, B \in \mathcal{S}$ if $A \cap B \neq \emptyset$, the colors of A and B are different.*

Proposition 2.1. *Sets in a matching can have the same color.*

Proper coloring defines a decomposition into matchings. A proper coloring with the minimum amount of color is the same as a minimal decomposition into matchings.

Algorithm 2.3. *Algorithm to determine χ for interval systems:*

1. *Arrange the intervals in a list in the order of increasing left ends: $I_1, I_2, \dots, I_n$. Let $i = 0$.*
2. *Assign I_i the smallest possible color that is, to the smallest integer which has not been assigned to any intervals intersecting I_i. If $i < n$ $i := i + 1$, otherwise stop.*

Definition 2.9 (Maximum degree of an interval system, Δ)**.** *The maximum degree of an $\mathcal{I}$ interval system is the maximum number of intervals in $\mathcal{I}$ sharing a point. Denoted by $\Delta(\mathcal{I})$.*

Definition 2.10 (Minimum degree of an interval system, q)**.** *The minimum degree of an $\mathcal{I}$ interval system is the minimum number of matchings $\mathcal{I}$ can be decomposed into. Denoted by $q(\mathcal{I})$.*

Theorem 2.6. *For any interval system $\mathcal{I}$*

$$\Delta(\mathcal{I}) = q(\mathcal{I}).$$

Proof. Proved in two steps:

1. There are $\Delta(\mathcal{I})$ intervals sharing a point, which musht have different colors.

$$q(\mathcal{I}) \geq \Delta(\mathcal{I}).$$

2. Since the intervals are ordered according to their left endpoints, for every index pair $i < j$ the intevral I_i meets $I_j = [a_j, b_j]$ if and only if I_i contains a_j. Since a_j is incident to at most $\Delta(\mathcal{I})$ intervals, one of them being I_j itself, when it gets colored, at most $\Delta(\mathcal{I}) - 1$ intervals intersecting it have been colored previously. This means that at most $\Delta(\mathcal{I})$ colors are applied in a minimal coloring:

$$q(\mathcal{I}) \leq \Delta(\mathcal{I}).$$

Consequently,

$$q(\mathcal{I}) = \Delta(\mathcal{I}).$$

□

3 Sequetial coloring

3.1 Intercestion graph of an interval system

Definition 3.1 (Intersection graph of a set system)**.** *The intersection graph $G(\mathcal{S})$ of a set system $\mathcal{S}$ has one vertex for each set $S_i \in \mathcal{S}$ moreover two vertices v_i and v_j are adjacent in $G(\mathcal{S})$ if and only if the corresponding members S_i and S_j of $\mathcal{S}$ have a nonempty intersection.*

Proposition 3.1. *Any simple graph can be obtained as an intersection graph.*

Definition 3.2 (Interval graph)**.** *A graph which is an intersection graph of some interval system is called an interval graph.*

Proposition 3.2. *Not every simple graph can be an interval graph.*

Proposition 3.3. *Connections between the parameters of an interval system $\mathcal{I}$ and its intersection graph $G(\mathcal{S})$:*

1. *Maximum number of intersecting subgraphs in $\mathcal{I}$ are the same as the maximum clique number in $G(\mathcal{I})$:*
$$\Delta(\mathcal{I}) = \omega(G(\mathcal{I})).$$
2. *Minimal number of decompositions to intersecting subsystems in $\mathcal{I}$ is the same as the minimum number of covering with cliques in $G(\mathcal{I})$:*
$$k(\mathcal{I}) = \theta(G(\mathcal{I}))$$
3. *The maximum number of mathcings in $\mathcal{I}$ is the same as the maximum number of independent vertex sets in $G(\mathcal{I})$:*
$$\nu(\mathcal{I}) = \alpha(G(\mathcal{I})).$$
4. *The minimum number of decomposition into matchings in $\mathcal{I}$ is the same as the minimum propper vertex coloring in $G(\mathcal{I})$:*
$$q(\mathcal{I}) = \chi(G(\mathcal{I})).$$

Proof. Proofs of the connections:

1. The maximum number of intervals in $\mathcal{I}$ that share a point is the same as the maximum number of intervals that are pairwise intersecting, which is the maximum number of vertices in $G(\mathcal{I})$ that are pairwise adjacent.
2. Decomposition of $\mathcal{I}$ into intersecting subsystems is the same as covering of $G(\mathcal{I})$ with the minimum number of cliques.
3. A maximum mathcing in $\mathcal{I}$ is the maximum cardinality of independent subsystems, which is equal to the maximum cardinality of an independent vertex set in $G(\mathcal{I})$.
4. The minimum cardinality of a decomposition of $\mathcal{I}$ into matchings is the number of colors used in a proper coloring of the intervals which is equal to the minimum number of colors needed for a proper coloring of $G(\mathcal{I})$.

□

Theorem 3.1. *For any interval graph G*
$$\chi(G) = \omega(G), \quad \theta(G) = \alpha(G).$$

Proof. For any interval system $\mathcal{I}$ we have seen that
$$\tau(\mathcal{I}) = k(\mathcal{I}) = \nu(\mathcal{I}), \quad q(\mathcal{I}) = \Delta(\mathcal{I}).$$
From this we get
$$\chi(G(\mathcal{I})) = q(\mathcal{I}) = \Delta(\mathcal{I}) = \omega(G(\mathcal{I})), \quad \theta(G(\mathcal{I})) = k(\mathcal{I}) = \nu(\mathcal{I}) = \alpha(G(\mathcal{I})).$$

□

3.2 Sequential coloring

3.2.1 Bounds for the chromatic number

Proposition 3.4 (Lower bound for the chromatic number)**.** *For any graph G*

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

Proof. The vertices of a clique are pairwise adjacent, meaning that a maximum clique requires $\omega(G)$ colors. □

Algorithm 3.1 (Upper bound for the chromatic number – First Fit coloring)**.** *For a graph $G = (V, E)$ let us consider e vertex order $v_1, v_2, \ldots, v_n$. We color the vertices in this order. $\forall v_i \in V$ gets the smallest not forbidden color. This method returns a proper coloring of G, and gives an upper bound to $\chi(G)$.*

Proposition 3.5. *For any graph G*

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

Proof. At the coloring of $v_i \in V(G)$ colors that are less than $d(v_i)$ are forbidden. For v_i $d(v_i) + 1$ colors are enough. To color all vertices $\max d(v_i) + 1 = \Delta(G) + 1$ colors are enough. □

Depending on the vertex order the First Fit coloring can apply $\chi(G)$ colors, or much more colors than necessary.

Definition 3.3 (Backward degree)**.** *Given a graph G and a vertex order $v_1, v_2, \ldots, v_n$ let $d^-(v_i)$ denote the number of neighbors of v_i which precede it:*

$$d^-(v_i) = |\{v_j \mid v_i v_j \in E(G),\ j < i\}|.$$

Definition 3.4 (Coloring number, *col*)**.** *The coloring number $col(G)$ of graph G is the minimum of the maximum value of $d^-(v_i)$ over all vertex orders:*

$$col(G) = \min_{\textit{vertex orders}} \max\{d^-(v_i) + 1 \mid 1 \leq i \leq n\}.$$

Theorem 3.2. *For every graph G,*

$$\chi(G) \leq col(G)$$

Proof. Consider an optimal vertex order $v_1, v_2, \ldots, v_n$ of G and color the vertices using the First Fit algorithm. As every vertex v_i is preceded by $d^-(v_i)$ of its neighbors, when we color v_i not more than $d^-(v_i)$ colors are forbidden for v_i. Then v_i gets a color which is not greater than $d^-(v_i)+1$. By definition $d^-(v_i) + 1 \leq col(G)$ for every v_i hence First Fit yields a coloring with at most $col(G)$ colors and we conclude $\chi(G) \leq col(G)$. □

Proposition 3.6. *For any graph G, $col(G)$ is a better upper bound than $\Delta(G) + 1$.*

Proof.

$$col(G) = col(G) = \min\max\{d^-(v_i) + 1\} \leq col(G) = \min\max\{\Delta(G) + 1\} = \min\{\Delta(G) + 1\} = \Delta(G) + 1.$$

□

For graph G, $col(G)$ can be exactly $\chi(G)$, or much larger than $\chi(G)$.

Theorem 3.3. *For any graph the coloring number can be deretmined in polynomial time.*

Proof. We construct the following order of the n vertices of G:

1. Choose a vertex of minimum degree in G and let it be called v_n. This will be the last vertex n the order.
2. Then for every $i = n - 1, n - 2, \ldots, 1$ select a vertex of minimum degree in the subgraph induced by the remaining vertices $V(G) \backslash \{v_j \mid j > i\}$. Let it be called v_i.

This procedure results in a vertex order $v_1, v_2, \ldots, v_n$. We prove that this is an optimal one.

Consider any optimal order of the vertices. If it is $v_1, v_2, \ldots, v_n$, there is nothing to prove. Otherwise select the largest index i where the two orders differ. In the original order we have v_i and in the optimal one we have v_k with $k < i$. In the optimal order place v_i after v_k. By this change $d^-(v_i)$ may increase but it cannot be higher than $d^-(v_k)$ was before the modification, since v_i has minimum degree in $V(G) \setminus \{v_j \mid j > i\}$. Otherwise for a vertex v_l with $l > i$ the degree $d^-(v_l)$ does not change, while for a v_l with $l < i$, $d^-(v_l)$ either decreases or remains the same. Consequently, $\max d^-$ does not increase and the order remains optimal. Repeating this procedure, at each turn the largest index where the order ant the optimal one differ will be smaller by at least 1, and finally the optimal one is transformed into $v_1, v_2, \ldots, v_n$ preserving the optimality.

Minimum search in a set of size n, $n - 1$ times is of order $O(n^2)$, which is polynomial time. □

4 Chordal graphs

4.1 Chordal graphs and simplicial order

Definition 4.1 (Chordal graph). *A graph is chordal if ir contains no induced cycle of length greater than 3.*

Definition 4.2 (Simplicial vertex). *A vertex v is simplicial in G if and only if any two of its neighbors are adjacent.*

Definition 4.3 (Simplicial order). *A simplicial order of G is an order $v_1, v_2, \ldots, v_n$ of its vertices such that for every $1 \leq i \leq n-1$, vertex v_i is simplicial in the subgraph induced by $v_{i+1}, v_{i+2}, \ldots, v_n$.*

Theorem 4.1. *A graph has a simplicial order if and only if every induced subgraph of it has a simplicial vertex.*

Proof. If every induced subgaph og G has a simplicial order then nothing blocks us to choose an arbitrary simplicial vertex in G and furhter one in the subgraph induced by the remaining vertices, and so on. Finally we have a simplicial order definitely.

On the other hand assuming a simplicial order $v_1, v_2, \ldots, v_n$ of G for every induced subgraph $G' \subseteq G$, the vertex of G' which has the smallest index in the order above is surely simplicial in G'. □

Algorithm 4.1. *Computation of a simplicial order on a graph G:*

1. *For $i = 1, 2, \ldots, n$:*
   Let G_i be the subgraph induced by $V(G)\backslash\{v_j : j < i\}$.
   - *If there is no simplicial vertex in G_i then G has no simplicial order. Stop.*
   - *Otherwise let v_i be an arpitrary simplicial vertex of G_i.*
2. *If all the n iterations have been executed, the output is $v_1, v_2, \ldots, v_n$, which is a simplicial order of G.*

Theorem 4.2. *For any graph G, the following statements are equivalent:*

1. *G is chordal;*
2. *G has a simplicial order;*
3. *G is the intersection graph of a collection of subtrees of some tree T.*

Corollary 4.1. *Every interval graph is chordal.*

Proof. An interval system can be viewed as a set of subpaths of a path. Then, it is a collection of subtrees of a special tree. Thus interval graphs which are precisely their intersection graphs. Consequently they are chordal graphs. □

Algorithm 4.2. *Building a subtree representation of a chordal G, we proceed in inverse simplicial order: $v_n, \ldots, v_1$.*

1. *Vertex v_n is represented by the one–vertex subtree $\{x_1\}$ of the tree T consisting of only this vertex.*
2. *v_i is simplicial in the graph spanned by $v_i, v_{i+1}, \ldots, v_n = G_i$.*
   - *If v_i has neighbors in G_i, they form a clique, which means that $\exists x_j$ common point of the trees.*
     - *If only those subtrees contain x_j that represent a neighbor of v_i, then $T_i = \{x_j\}$.*
     - *If $\exists k$ such that $x_j \in T_k$, but $\{v_i, v_k\} \notin E(G)$, then $T_i = \{x_l\}$, where x_l is a new vertex, connected to x_j as a leaf. Moreover x_l is added to every tree representing a neighbor of v_i.*
   - *If v_i has no neighbors in G_i, then $T_i = \{x_l\}$, where x_l is a new vertex, a leaf on an arbitrary vertex of the tree, and the other subtrees ar unchanged.*

4.2 Algorithms for chordal graphs

Proposition 4.1. *For every graph G*

$$\alpha(G) \leq \theta(G),$$

where $\alpha(G)$ is the independence number, and θ is the clique covering number of G.

Proof. If $v_1, v_2 \notin E(G)$, then v_1 and v_2 cannot be covered by the same clique. This implies that to cover a maximum independent vertex set, more than or equal to $\alpha(G)$ cliques required. From this we get that to cover the vertex set $V(G)$, that is a greater set than the maximum independent vertex set, more than or equal to $\alpha(G)$ cliques are required. □

Algorithm 4.3. *Determination of α and θ:*

1. *Compute a simplicial order of the vertices, $I = \emptyset,\ K = \emptyset$.*
2. *If v_i is the first vertex in the simplicial order, put v_i into I.*
3. *$K_i = v_i \cup N(v_i)$ is a clique, put it into K.*
4. *Delete the vertices of K_i from the simplicial order.*
5. *If the list is not empty, go to Step 2.*

Theorem 4.3. *For every chordal graph*

$$\alpha(G) = \theta(G),$$

and these parameters can be computes in polinomial time.

Proof.

$$|I| = \theta(G) \leq |K| = \leq \alpha(G) \leq \theta(G).$$

Computation time of the simplicial order is $O(n^2)$, and the complexity of the algorithm above is also $O(n^2)$, hence the computation of $\alpha(G)$ and $\theta(G)$ is of order $O(n^2)$. □

Definition 4.4 (Forward degree of a vertex)**.** *Let $v_1, v_2, \ldots, v_n$ an order of the vertices of G, then in relation to this the forward degree of v_i is*

$$d^+(v_i) = |\{v_j \mid v_j \in N(v_i),\ j > i\}|.$$

Theorem 4.4. *For any chordal graph G and simplicial order $v_1, v_2, \ldots, v_n$,*

$$\omega(G) = \max_{i=1,\ldots,n} \{d^+(v_i) + 1\}.$$

Proof. Consider a chordal graph G and a simplicial order $v_1, v_2, \ldots, v_n$. Then for every $1 \leq i \leq n$, vertex v_i is simplicial in the subgaph induced by $\{v_i, v_{i+1}, \ldots, v_n\}$. Thus, a clique of $d^+(v_i) + 1$ vertices surely occurs in G implying that

$$\omega(G) \geq d^+(v_i) + 1$$

holds for every i. Then,

$$\omega(G) \geq \max_{1 \leq i \leq n} \{d^+(v_i) + 1\}$$

holds as well.

On the other hand, for a clique of ω vertices consider the vertex v_i which is the earliest one among them in the order. For this vertex, $d^+(v_i) = \omega(G) - 1$ holds and we have

$$\omega(G) \leq \max_{1 \leq i \leq n} \{d^+(v_i) + 1\}.$$

□

Algorithm 4.4. *To compute $\omega(G)$ and $\chi(G)$ of a chordal graph G:*

1. *Consider a simplicial order of G.*
2. *Apply the First Fit coloring according to the inverse order.*

Theorem 4.5. *For any chordal graph G*

$$\omega(G) = \chi(G) = col(G)$$

and can be computed in polynomial time.

Proof. In the algorithm, when v_i is colored

- $d^+(v_i)$ neighbors are already colored.
- These neighbors form a clique, $d^+(v_i)$ colors are forbidden, plus one color is needed. The algorithm will use $\max_{1\leq i\leq n}\{d^+(v_i)+1\}$ colors.

By definition $\chi(G) \leq \max\{d^+(v_i)+1\}$, and for any graph $\omega(G) \leq \chi(G)$, and for any chordal graph $\omega(G) = \max\{d^+(v_i)+1\}$, meaning that

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

Like before, the computation complexity of the algorithm is of order $O(n^2)$. □

5 Tree decompositions

Definition 5.1 (Tree decomposition)**.** *Let $G = (V, E)$ be a graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set E. A tree decomposition of G is a pair $(T, \mathcal{S})$, where $T = (X, F)$ is a tree graph with node set $X = \{x_1, x_2, \ldots, x_m\}$ and edge set F, and $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ is a set system over V, indexed according to the nodes of T, that satisfies the following three requirements:*

1. *Every vertex $v_i \in V$ of G occurs in some set $S_k \in \mathcal{S}$.*
2. *The two ends of any edge $v_i v_j$ of G occur together in some set $S_k \in \mathbf{S}$.*
3. *If $v_i \in S_{k'}$ and $v_i \in S_{k''}$ for two indices k', k'', then $v_i \in S_k$ also holds whenever the node x_k is on the $x_{k'} - x_{k''}$ path in T.*

Definition 5.2 (Width of a tree decomposition)**.** *The width of a tree decomposition*

$$w(T, \mathcal{S}) = \max_{S_k \in \mathcal{S}} \{|S_k| - 1\}.$$

Definition 5.3 (Tree width)**.** *The tree width of a graph G is the smallest width of its tree decompositions.*

$$tw(G) = \min_{(T,\mathcal{S}) \text{ is a tree dexomposition of } G} \{\max_{S_k \in \mathcal{S}} \{|S_k| - 1\}\}.$$

Theorem 5.1. *For any graph G*

$$tw(G) = \min_{G \subseteq H \text{ chordal}} \{\omega(H) - 1\}.$$

Corollary 5.1. *For any graph G*

- *$tw(G) = 0$ iff G has no edges.*
- *$tw(G) = 1$ iff G has at least 1 edge but no cycles.*
- *$tw(G) \geq 2$ iff G has at least 1 cycle.*

Corollary 5.2. *If G is chordal*

$$tw(G) = \omega(G) - 1$$

5.1 Creating a tree decomposition

Let $G = (V, E)$ be an arbitrary (simple, undirected) graph.

1. Finding a chordal subgraph H of G. Edges have to be inserted into the graph as long as it contains chordless cycles longer than 3.
2. Finding a tree representation. This can be done in the way described in the previous chapter.
3. Finding the sets S_k. Formally this step can be done by setting $\mathcal{S} = \{S_1, \ldots, S_m\}$, where

$$S_k := \{v_i \mid x_k \in T_i\}$$

   for all $1 \leq k \leq m$. That is in the set assigned to x_k we list the vertices of G with the indices of subtrees containing x_k.

Lemma 5.1. *The pair $(T, \mathcal{S})$ constructed above satisfies the requirements of a tree decomposition.*

Proof. Proved in three parts:

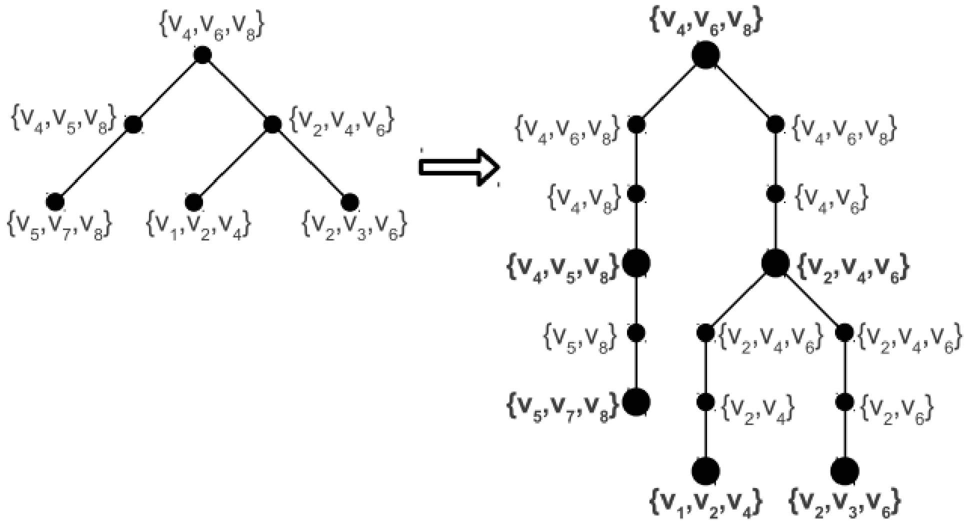
1. Every tree representing the vertices is nonempty, therefore each vertex of G occurs in at least one S_i, verifying the first condition.
2. If $v_i v_j$ is an edge in G, then it is an edge oh H as well. The definition of intersection graph then implies that T_i and T_j share a vertex, say x_k. And then v_i and v_j occur together in S_k according to the construction. This ensures that the second condition holds.
3. Finally, the occurrences of any v_i in the sets S_k correspond to the nodes of x_k which are contained in the subtree T_i. That is, those occurrences form a subtree of T, implying that the entire path connecting any two of them is inside the set of occurrences. This verifies the third condition.

□

5.2 Nice tree decomposition

Definition 5.4 (Nice tree decomposition)**.** *Suppose that $(T, \mathcal{S})$ is a tree decomposition of G. It is called a nice tree decomposition if the following further conditions are met, too:*

1. *Viewing T as a rooted tree every node of T has at most two children.*
2. *Each node is one of the following types:*
   - ***start node**: x_k has no children;*
   - ***forget node**: x_k has exactly one child, say $x_{k'}$, and $S_k = S_{k'} \setminus \{v_i\}$ for some $v_i \in V$;*
   - ***introduce node**: x_k has exactly one child, say $X_{k'}$, and $S_k = S_{k'} \cup \{v_i\}$ for some $v_i \in V$;*
   - ***join node**: x_k has exactly two children, say $x_{k'}$ and $x_{k''}$, and $S_k = S_{k'} = S_{k''}$.*



5.3 Largest independent set

For each node x_k we compute values in a table. The rows correspond to subsets S of S_k that are independent vertex sets in G.

The first column specifies the independent subset $S \subseteq S_k$. The second column contains the maximal cardinality of independent sets I in G_k for which $I \cap S_k = S$. The third column contains a possivle subset S' at the child $x_{k'}$ of x_k for which there is an independent set I attaining the maximal $\alpha(G)$ value in the 2. column so that $I \cap S_k = S$ and $I \cap S_{k'} = S'$.

5.3.1 Computation steps

- **start node**: no predecessors. α in the table x_k row $S = |S| = |I|$.
- **forget node**: $S_k = S_{k'} - \{v_i\}$, S can originate from 2 possible subsets of $S_{k'}$. S' is the one where α is maximal in table $x_{k'}$. α in the table x_k row $S = \alpha$ in the table $x_{k'r}$ rows S'.
- **introduce node**: $S_k = S_{k'} \cup \{v_i\}$.
  - If $v_i \notin S$, then $S' = S$. α in the table x_k row $S = \alpha$ in the table $x_{k'}$ row S'.
  - If $v_i \in S$, then $S' = S \setminus \{v_i\}$, which means v_i is an additional element in I. α in the table x_k row $S = \alpha$ in the table x_k row $S' + 1$
- **join node**: $S_k = S_{k'} = S_{k''}$, meaning that $S' = S$, $\forall S \subseteq S_k$ independent sets. α in table x_k row $S = (\alpha$ in table $x_{k'}$row $S) + (\alpha$ in table $x_{k''}$ row $S) - |S|$.

6 Bipartite graphs

Proposition 6.1. *G is a bipartite graph iff G contains no cycles of odd length.*

Theorem 6.1. *Bipartite graphs can be recognized in $O(n+c)$ steps.*

Proof. Apply Breath–First search (BFS). For every vertex it determines the distance from the root layers – vertices with identical distances from the root.

The edges of G cannot connect vertices from layers at distance of more than 1. □

Proposition 6.2. *G is a bipartite graph iff there is no edge connecting vertices belonging to the same layer of the BFS tree.*

Proof. Two parts:

1. If there is an edge uw connecting vertices in the same layer
   - Let v be the lowest common ancestor of u and w in the BFS tree.
   - u and w are both of distance d from v.
   - The paths vu and vw in the BFS tree with the edge uw form an odd cycle of length $2d+1$.
2. if there is an odd cycle, let u and w be one of the highest and lowest points in the tree. The cycle provides two paths in G connecting u and w one of them is of even length, the other is odd.

   It is not possible that all of the edges are connecting neighboring layers, so there must be a forbidden edge.

The BFS is of complexity $O(n+e)$. □

6.1 Maximum matchings in bipartite graphs

Theorem 6.2 (König)**.** *For any bipartite graph G*

$$\tau(G) = \nu(G)$$

and optimal sets can be determined efficiently.

Theorem 6.3. *][Hall's marrige theorem] If $G = (A, B, E)$ a bipartite graph, where $|A| = |B|$ and the Hall–condition holds for A:*

$$|X| \leq N(X),$$

there is a perfect mathcing in G.

Theorem 6.4. *In the bipartite graph $G = (A, B, E)$ there is a matching covering the vertex set A iff Hall's condition holds for A.*

6.2 Edge coloring of graphs

Proposition 6.3. *If $G = (A, B, E)$ is a k–regular bipartite graph, there exists a perfect matching.*

Proof. $E(G) = k \cdot |A| = k \cdot |B|$ which means that $|A| = |B|$. The edges coming from X go to at least $|X|$ vertices in B, meaning that Hall's condition holds. □

Theorem 6.5. *If $G = (A, B, E)$ is a k–regular bipartite graph, then*

$$\chi'(G) = k.$$

Proof. Induction by k:

- $k = 1$: G is a matching, 1 color is enough.
- $k \to k+1$: We assume that k–regulars can be colored by k colors. If G is $k+1$–regular, there exists a perfect matching M. $G \backslash \{M\}$ is a k–regular bipartite graph. By using a k coloring for the edges of $G \backslash \{M\}$ and one other color for the perfect matching. This is a coloring of G with $k+1$ colors.

We have seen that for the edge coloring of a k–regular bipartite graph k colors are enough, meaning $\chi'(G) \leq k$. In general $\chi'(G) \geq \Delta(G) = k$. This gives us

$$\chi'(G) = k = \Delta(G).$$

□

Theorem 6.6. *For an arbitrary bipartite graph $G = (A, B, E)$,*

$$\chi'(G) = \Delta(G).$$

Proof. Since $\chi' \geq \Delta$ holds for every graph, it is sufficient to prove that G has an edge coloring with Δ colors. The crucial point is the extension of G by some edges and vertices to obtain a Δ–regular bipartite graph G'. Then, by the previous theorem there is an edge coloring of G' with Δ colors. Finally we delete some appropriately chosen edges and vertices, and we obtain a proper edge coloring of G with Δ colors.

The extension of G can be obtained in several ways. For example:

1. If $|A| > |B|$, extend B with $|A| - |B|$ new vertices. If $|B| > |A|$, do it the other way around.
2. While we have nonadjacent vertex pairs (a_i, b_j) with $a_i \in A$ and $b_j \in B$ and with degrees $d(a_i) < \Delta$ and $d(b_j) < \Delta$, extend G by the edge $a_i b_j$.
3. If Step 2 cannot be applied, the vertices with degree smaller than Δ form a complete bipartite graph whose partite classes are S_A and S_B. It is clear that $|S_A| \leq \Delta$ and $|S_B| \leq \Delta$. Then, put Δ new vertices into A and B each (these form vertex sets N_A and N_B) and create some edges between S_A and N_B such that every vertex in S_A has degree Δ and the degrees in N_B differ by at most one- A similar procedure is executed for S_B and N_A.
4. Finally, take vertex v from N_A of degree smaller than Δ and connect it to $\Delta - d(v)$ vertices of N_B such that the degrees in N_B differ by at most one. This ensures that all such vertices from N_A can be treated and when all the degrees in N_A become equal to Δ, then also the degrees in N_B equal Δ, and we have a Δ–regular bipartite graph G' with subgraph G.

□

6.3 Stable matchings

Definition 6.1 (Stable matching)**.** *A stable matching in a graph G is a matching M such that for every edge $uv \in E(G) \setminus M$ either*

1. *u has a neighbor u' such that $uu' \in M$ and u prefers u' to v, or*
2. *v has a neighbor v' such that $vv' \in M$ and v prefers v' to u.*

Theorem 6.7 (Stable marriage theorem)**.** *For any bipartite graph and any preference list of the vertices there exists a stable matching in G.*

Proof. Consider a graph G with partite classes A and B and with preference lists on its vertices. Each phase of the algorithm consists of two steps:

1. Every unmatched vertex $a \in A$ marks the edge connecting it to its neighbour with the highest preference.
2. If there are more than one marked edges in the case of some $b \in B$ the most preferred by b is kept, the others are unmarked.

When the algorithm terminates, we have some edges marked. Let M be the set of these edges. It is clear that M is a matching, we prove that this is stable. We have two cases for an edge $a_i b_j \notin M$:

- If $a_i b_j$ was not marked in any phases and the algorithm terminated, then a_i is paired with a vertex b_k which has a higher preference than b_j.
- If $a_i b_j$ was marked in some phase but then was rejected by b_j, then b_j has a more preferred pair and again $a_i b_j$ is not a blocking edge.

Therefore, the algorithm produces a stable matching for every bipartite graph. □

7 The Max–Cut problem

Definition 7.1 (Cut)**.** *Let $G = (V, E)$ be a graph and $X \cup Y = V$ a partition of its vertex set into two classes. The cut generated by (X, Y) is the set $F \subseteq E$ of edges which have one end in X and the other end in Y.*

The number of edges in the cut (X, Y) is denoted by

$$e(X, Y) = |F|.$$

The maximum cut is denoted by

$$mc(G) = \max_{(X,Y) \text{ is a cut of } G} e(X, Y).$$

Proposition 7.1. *For any graph G*

$$mc(G) = |E| \quad \Leftrightarrow \quad G \text{ is bipartite.}$$

Theorem 7.1. *For every graph G*

$$mc(G) \geq \frac{|E|}{2}.$$

Proof 1: Local optimum. Consider an arbitrary vertex position $V = X \cup Y$. If a vertex $x \in X$ has more neighbours in X than in Y,

$$X := X \backslash \{x\}, \quad Y := Y \cup \{x\}.$$

Similar steps can be performed for $y \in Y$.

If $e(X, Y)$ cannot be further increased, then every vertex has at least half of its edges in the cut. □

Proof 2: Finding a solution online. Consider tan arbitrary order of vertices $v_1, v_2, \ldots, v_n$. For every vertex v_i we make a decision, according to the subgraph induced by $\{v_1, \ldots, v_i\}$. Initially $X = Y = \emptyset$.

For $i \in \{1, 2, \ldots, n\}$ if v_i has at least as many neighbors in Y as in X, then $X := X \cup \{v_i\}$. Otherwise $Y := Y \cup \{v_i\}$.

Let d_j^- denote the number of neighbors v_i of v_j where $i < j$. When we decided the partition of v_j, the size of the cut has increased by at least $\frac{d_j^-}{2}$ edges.

$$mc(G) \geq e(X, Y) \geq \sum_{j=2}^{n} \frac{d_j^-}{2} / \frac{1}{2} \cdot \sum_{j=1}^{n} d_j^- = \frac{|E|}{2}.$$

□

8 Locally restricted colorings

8.1 Precoloring extension problem

We have a graph $G = (V, E)$, color bound $k \in \mathbb{N}$, partial coloring $\varphi_W : W \to \{1, 2, \ldots, k\}$ that is a proper vertex coloring of the subgraph $G[W]$ induced by the precolored set $W \subset V$ in G.

The question is that does G have a proper vertex coloring φ with at most k colors, which extends φ_W?

Special cases:

- If $k < \chi(G)$, then no.
- If $W = \emptyset$ or W induces a complete subgraph in G the answer is yes iff $k \geq \chi(G)$.
- If G is a bipartite graph and
  - $k = 2$: the answer can be determined efficiently,
  - $k \geq 3$: the answer is hard to decide.
- If $\overline{G}$ is a bipartite graph and $k \in \mathbb{N}$ the problem is equivalent to finding a maximal matching in $\overline{G}$. It can be done efficiently.

8.2 List coloring

Definition 8.1 (List coloring)**.** *Let $G = (V, E)$ be a graph, and let $\mathcal{L} = \{L_v \mid v \in V\}$ be a collection of sets which specify the colors allowed for every vertex v. A list coloring of G is a color assignment $\varphi : V \to \bigcup_{v \in V} L_v$ such that*

- *$\varphi(v) \in L_v$ for all $v \in V$;*
- *$\varphi(u) \neq \varphi(v)$ whenever $uv \in E$.*

If such a φ exists, we say that G is list colorable.

Definition 8.2 (k–assignment)**.** *A k–assignment on a graph $G = (V, E)$ is a list assignment $\mathcal{L} = \{L_v \mid v \in V\}$ in which $|L_v| = k$ for all $v \in V$. The choice number of G is the smallest k such that G is $\mathcal{L}$–colorable for every k–assignment $\mathcal{L}$. We denote the choice number of G by $\chi_l(G)$. We also say that G is k–choosable if it is list colorable for every k–assigment.*

Proposition 8.1. *For every graph G*

$$\omega(G) \leq \chi(G) \leq \chi_l(G) \leq col(G).$$

Proof. The proof is done with two steps:

1. $\chi(G) \leq \chi_l(G)$:
   If $\chi_l(G) = k$, then G is list colorable for any k–assignment, so for $L_v = \{1, 2, \ldots, k\}$ $\forall v \in V(G)$ as well, meaning $\chi(G) \leq k = \chi_l(G)$.
2. $\chi_l(G) \leq col(G) = \min_{\text{vertex order}} \max_{i=1,\ldots,n}\{d^-(v_i) + 1\}$:
   consider a vertex order, where $\max\{d^-(v_i + 1)\} = col(G)$ to every vertex assign a list of length $L = col(G)$. This way G is L-colorable. If we color the vertices in increasing order of indices for the vertex v_i at most $d^-(v_i)$ colors are forbidden. Since the list is bigger, we can choose a suitable color for v_i. This means that G is k–choosable, where $k = col(G)$, and also $\chi_l(G) \leq k$:

$$\chi_l(G) \leq col(G).$$

□

8.3 Kernels in directed graphs

Definition 8.3 (Kernel of a directed graph)**.** *Let $D = (V, A)$ be a digraph with vertex set V and arc set A. A kernel of G is a set $? \subset V$ satisfying the following two properties:*

1. *M is independent;*
2. *for every vertey $u \in V \backslash M$ there is a $v \in M$ such that $uv \in A$.*

Not every directed graph has a kernel.

Theorem 8.1. *Two statements:*

1. *If T is an oriented tree, then T has a kernel, its unique, and can be found by an efficient algorithm.*
2. *More generally, every bipartite graph has at least one kernel, and a kernel can be found efficiently.*

Proof. The proofs in order:

1. T must have a vertex v for which the out degree is zero, otherwise it would have a cycle.

   The general step of the construction($M = \emptyset$ at the beginning): If v is a vertex with zero out–degrees, put v into M and delete all vertices u for which $uv \in A$.

2. Let D be a bipartite directional graph $V = A \cup B$. The weakly connected components (components of the graph without considering orientation) can be considered separately.

   Consider a weakly connected bipartite graph.

   - If $\exists v \in V$ with out degree zero, the general step can be performed.
   - If no such vertices are found, then all vertices have out neighbours. Since D is bipartite, their neighbours are in the other class, and the sets A and B are independent sets. Either A or B can be chosen as kernel.

   Using these steps we can efficiently determine a kernel in D.

□

Theorem 8.2. *Let $D = (V, A)$ be an orientation of the graph $G = (V, E)$ where every induced subgraph has a kernel. If $\mathcal{L} = \{L_v \mid v \in V(G)\}$ is a list assignment, where $|L_v| > d^+(v)$, $\forall v \in V(G)$, where $d^(v)$ denotes the out degree of a vertex, then G is $\mathcal{L}$–colorable.*

Proof. Select an arbitrary color c. G_c a subgraph of G, $D_c \subseteq D$, where $v \in G_c$ iff $c \in L_v$.
By assumption D_c contains a kernel M:

- assign the color c to $\forall v \in M$;
- delete the color c from all the lists L_v if $v \in D_c \backslash M$.

This does not guarantee monochromatic edges:

- M is independent,
- the color c is deleted from the list of all the uncolored vertices that had it.

In the remaining graph $D' = D \backslash M$ $|L_v| > d^+(v)$ remains true:

- If $v \in G_c \backslash M$, then $|L_v| := |L_v| - 1$, but $d^+(v)$ is decreased by at least 1.
- If $v \in G \backslash G_c$, then L_v is unchanged, and $d^+(v)$ might decrease, or remains the same.

The list L_v will never become empty. The graph G can be colored from the list by repeating the step above, while $V(G) \neq \emptyset$. If we can efficiently determine an orientation where every induced subgraph has a kernel, the proof gives us an efficient methos to find a list–coloring. □

9 Edge decomposition of graphs

Definition 9.1 (Edge decomposition of a graph)**.** *An edge decomposition of a graph G is a partition of its edge set into some subgraphs $F_1, F_2, \ldots, F_m$ where $F_i = (V_i, E_i)$, $V_i \subseteq V$ for all $1 \leq i \leq m$, the sets E_i are mutually disjoint and their union is E. In other words, each edge of G occurs in precisely one of the subgraphs.*

Theorem 9.1. *The complete graph K_n $(n \geq 2)$ is decomposable into perfect matchings iff n is even.*

Proof. Two steps:

1. A perfect matching is a set of vertex pairs. A graph with an odd number of vertices has no perfect matching.
2. For K_{2k} $(k \geq 2)$ arrange the vertices into a regular $2k-1$-gon $v_1, v_2, \ldots, v_{2k-1}$, and put the last vertex in the center.

   Consider the M_i matchings

   $$M_i = \{v_i, v_{2k}\} \cup \{v_{i-j}v_{i+j} \mid 1 \leq j \leq k-1\}.$$

   These matchings are pairwise disjoint the edges in M_i are v_iv_{2k} and all the ones that are orthogonal to it.

   M_j can be obtained from M_i by a rotation of degree $(i-j) \cdot \frac{360}{2k-1}$

□

Theorem 9.2. *The complete graph K_n on $n \geq 2$ is decomposable into Hamiltonian paths iff n is even.*

Proof. A Hamiltonian path on n vertices has $n-1$ edges, and K_n has $\frac{n(n-1)}{2}$ edges. The decomposition has to contain $\frac{n}{2}$ subgraphs, so n must be even.

The case of K_2 is trivial. For K_n $n \geq 4$ arrange the vertices in a $2k$-gon. The $2k$-gon has k long diagonals. For every long diagonal we define a Hamiltonian path P_i which can be obtained by rotation from each other.

It can be seen, that paths P_i are pairwise disjoint, and every edge is contained by exactly one of them. □

Theorem 9.3. *The complete graph K_n $(n \geq 3)$ is decomposable into Hamiltonian cycles iff n is odd.*

Proof. A cycle on n vertices has n edges and K_n has $\frac{n(n-1)}{2}$ edges. The decomposition consists of $\frac{n-1}{2}$ cycles, so n is odd.

Construction is similar to the case of Hamiltonian paths is $n \geq 5$. Arrange the vertices into an $n-1$-gon and one vertex at the center.

If the central vertex is removed, we construct the Hamiltonian paths, and then add the central vertex and connect both ends of the path to it. □

Definition 9.2 (Edge decomposition into complete bipartite graphs)**.** *The edge decomposition of a graph into complete bipartite graphs is an edge decomposition $F_1, F_2, \ldots, F_m$, where for all $1 \leq i \leq m$ $F_i \in \mathcal{F}$,*

$$\mathcal{F} = \{K_{a,b} \mid a \geq 1,\ b \geq 1\}.$$

Theorem 9.4. *If $F_1, F_2, \ldots, \mathfrak{F}_m$ is a decomposition of K_n into complete bipartite graphs, then $m \geq n-1$.*

Proof. We represent the vertices as $x_1, \ldots, x_n \in \mathbb{R}$ variables, and edge edge v_iv_j as the product x_ix_j.

$H \subseteq K_n$ subgraph will be

$$s(H) = \sum_{v_iv_j \in E(H)} x_ix_j.$$

If $F_l(A_l, B_l, E_l)$ a complete bipartite subraph, then

$$s(F_l) = \sum_{v_i \in A_l, v_j \in B_l} x_ix_j = \left(\sum_{v_i \in A_l} x_i\right) \cdot \left(\sum_{v_j \in B_l} x_j\right).$$

$$s(K_n) = \sum_{1\le i\le j\le n} x_i x_j = \frac{1}{2}\left[\left(\sum_{i=1}^{n} x_i\right)^2 - \left(\sum_{j=1}^{n} x_j^2\right)\right].$$

If $F_1, \dots, F_m$ is an edge decomposition of K_n, then

$$s(K_n) = \sum_{l=1}^{m} s(F_l),$$

since $s(G)$ is the sum of edges in G.

$$\frac{1}{2}\left[\left(\sum_{i=1}^{n} x_i\right)^2 - \left(\sum_{j=1}^{n} x_j^2\right)\right] = \sum_{l=1}^{m}\left(\sum_{v_i\in A_l} x_i\right)\cdot\left(\sum_{v_j\in B_l} x_j\right).$$

Consider the following system of $m+1$ homogenous linear equations over n variables:

$$\begin{cases} x_1 + \dots + x_n = 0 \\ \sum_{v_i\in A_1} x_i = 0 \\ \vdots \\ \sum_{v_i\in A_m} x_i = 0 \end{cases}$$

If the real numbers $x_1, \dots, x_n$ fulfill the linear equations, then the right side of the equations is zero, the first termn on the left side is zero, so the equation holds.

Consequently the system of equations has one solution. We have seen, that if a system of linear equations has exactly one soluton, then the number of equations is at leas as big as the number of variables. In this case

$$m + 1 \ge n.$$

□

Theorem 9.5. *If $F_1, F_2, \dots, F_m$ is an $\mathcal{F}$-decomposition of K_n where $m \ge 2$, $m \ge n$.*

Proof. Assume by contradiction that $m < n$. Let F_j have a vertex set V_j and let us denote $n_j := |V_j|$ for $j = 1, 2, \dots, m$. Further, for vertex v_i, let us denote by d_i the number of subgraphs F_j containing v_i.

Claim: If $v_i \ne V_j$, then $d_i \ge n_j$.

Proof. If $v_l \in V_j$, then edge $v_i v_l$ is contained by exactly one subgraph F_{k_l}. For different vertices v_l the subgraphs F_{k_l} are different. If for two vertices v_{l_1} and v_{l_2} the edges $v_{l_1}v_i$ and $v_{l_2}v_i$ were covered by the same $F_{k_l} \ne F_j$, then the edge $v_{l_1}v_{l_2}$ was covered by both F_{k_l} and F_j. □

The number of subgraphs containing $v_i \ge |V_j|$, meaning $d_i \ge n_j$. By indirect assumption $n > m$, $n \cdot d_i > m \cdot n_j$, whichimplies $nm - nd_i < nm - mn_j$, so

$$\frac{1}{n(m-d_i)} > \frac{1}{m(n-n_j)} \implies \sum_{i,j,\ v_i\notin V_j} \frac{1}{n(m-d_i)} > \sum_{i,j,\ v_i\notin V_j} \frac{1}{m(n-n_j)}.$$

The left side:

$$\sum_{i=1}^{n}\sum_{v_i\notin V_j} \frac{1}{n(m-d_i)} = \sum_{i=1}^{n}(m-d_i)\frac{1}{n(m-d_i)} = \sum_{i=1}^{n}\frac{1}{n} = 1.$$

The right side:

$$\sum_{i,j,\ v_i\notin V_j} \frac{1}{m(n-n_j)} = \sum_{j=1}^{m}(n-n_j)\frac{1}{m(n-n_j)} = \sum_{j=1}^{m}\frac{1}{m} = 1.$$

By using the computed values we get $1 > 1$. The assumption $n > m$ is false. □

Proposition 9.1. *If K_n is decomposed into copies of K_p, then*

- $\binom{n}{2}$ *is a multiple of* $\binom{p}{2}$,
- $n - 1$ *is a multiple of* $p - 1$.

These conditions are called the integrality conditions.

Proof. Two steps:

1. The number of edges: $e(K_n) = \binom{n}{2} = \frac{n(n-1)}{2}$, $e(K_p) = \binom{p}{2}$. Each edge is covered exactly once, with copies of K_p implies that $\binom{n}{2}$ is divisible by $\binom{p}{2}$.
2. The vertex degrees in K_n $\forall i$ $d(v_i) = n - 1$, and the vertex degrees in K_p $\forall j$ $d(v_j) = p - 1$. If a vertex is covered by t copies of K_p it covers $t(p-1)$ edges incident to v_j and each of the $n-1$ edges is covered exactly once, meaning that $t(p-1) = n-1$ implying that $n-1$ is divisible by $p-1$.

□

Theorem 9.6. *For every $p \geq 3$ there is a threshold value $n_0(p)$ so that for every $n \geq n_0(p)$ K_n can be decomposed into copies of K_p iff the integrality conditions are met.*

Example 9.1 (Steiner Triple Systems)**.** *For $p = 3$, that is $F = K_3$, decomposition into triangles, the integrality conditions mean:*

- $\binom{n}{2}$ *is divisible by 3,*
- $n-1$ *is even.*

10 Finite projective planes

Definition 10.1 (Axioms of finite projective planes of order q)**.** *A pair $(\mathcal{P}, \mathcal{L})$ is a projective plane of order q if $\mathcal{L}$ is a set system over $\mathcal{P}$. They fulfill the following axioms:*

1. *Any two points are contained together in exactly one line.*
2. *Any two lines intersect in exactly one point.*
3. *There is a line with exactly $q+1$ points.*
4. *There are four points, no three of which are on the same line.*

Theorem 10.1. *Every projective plane of order q has the following parameters.*

1. *The number of points is q^2+q+1.*
2. *The number of lines is q^2+q+1.*
3. *Every line has exactly $q+1$ points.*
4. *Every point is incident with exactly $q+1$ lines.*

Proof. The proofs:

1. From the 3. axiom: $\exists L_0 \in \mathcal{L}$ with exactly $q+1$ points, $p_1, \ldots, p_{q+1}$.
2. From the 4. axiom: $\exists p \notin L_0$.
3. From the 1. axiom the point pairs p and p_i $\forall i \in \{1, 2, \ldots, q+1\}$ determine lines.
4. From the 2. axiom: the lines are pairwise different. If the lines determined by the point pairs $\{p, p_i\}$ and $\{p, p_j\}$ were the same line $L_1 \in \mathcal{L}$, then the intersection of the lines L_0 and L_1 was two points, p_i and p_j. There cannot be any other lines passing through p, otherwise by the 2. axiom it would also intersect the line L_0. The intersection point would be different from $p_1, \ldots p_{q+1}$.
5. $\forall p \notin L_0$ is incident to exactly $q+1$ lines.

   By switching the roles of point and line it can be proven similarly as before, that $\forall L$ that is not incident to the point p has exactly $q+1$ points.
6. From the 4. axiom: There are 4 points in a general position. Any two different pairs determine different lines.

   p and p' are both incident with $q+1$ lines. From the 1. axiom, there is exactly one line L' containing bot p and p'. Beside L' there are $2q$ lines incident to p and p', non of these lines are incident to both.

   By Statement 5 all these lines contain exactly $q+1$ points. For every point p'' there is such a line among these that does not lie on the point p''. By Statement 5, p'' is incident to $q+1$ lines. Every point is incident to exactly $q+1$ lines, and by switching the roles, every line is incident to exactly $q+1$ points.

   Each point $p_i \in L$ is incident to q lines beside L every line is among these for exactly one p_i.
7. From Statement 6 the number of lines is

   $$(q+1)q+1 = q^2+q+1$$

   .
8. Similarly every point p'' is incident to $q+1$ lines, that contain all other points, and each line has q points beside p'' and the lines have no other common point than p''.

   The number of points is $(q+1)q+1 = q^2+q+1$.

□

Proposition 10.1. *There is a finite projective plane of order q iff q is a prime power*

Theorem 10.2. *If a q is a prime power, then there exists a projective plane of order q.*

Proof. Proved by construction: Galois field of order q, denoted by $GF(q)$. This is a finite field with underlying set $\{0, 1, 2, \ldots, q-1\}$.

The so called Galois plane is a plane built upon $GF(q)$.

- Points: $(a, b, c) \sim (\lambda a, \lambda b, \lambda c)$, $\forall \lambda \neq 0$
- Lines: $[x, y, z] \sim [\lambda x, \lambda y, \lambda z]$, homogeneous coordinates.

There are q possible values in every coordinate, where $(0, 0, 0)$ and $[0, 0, 0]$ are the exception. This means that there are $q^3 - 1$ possible triplets.

Each triplet is represented $q-1$ times, so the number of point, and the number of lines equals $\frac{q^3-1}{q-1} = q^2 + q + 1$.

The point (a, b, c) is incident to the line $[x, y, z]$ iff $ax + by + cz = 0$ iff $\lambda ax + \lambda by + \lambda cz = 0$.

For a fixed (a, b, c) the equation $ax + by + cz = 0$ has q^2 solutions (2 free variables with q possible values), but $[0, 0, 0]$ is among the solutions, so only $q^2 - 1$ are valid. Each solution is contained $q-1$ times because of homogeneous coordinates, meaning $\frac{q^2-1}{q-1} = q + 1$ lines are incident to the point (a, b, c). Every point is incident to $q + 1$ lines. Similarly every line has $q + 1$ points.

The line connecting two different points (a, b, c) and $(a', b', c') + \lambda(a, b, c)$ is the solution of the system of linear equations:

$$\begin{cases} ax + by + cz = 0 \\ a'x + b'y + c'z = 0 \end{cases}$$

the number of solutions is q, but $[0, 0, 0]$ is also a solution, giving us $q - 1$ valid solutions.

Every solution is considered $q - 1$ times because of the homogeneous coordinates, giving us exactly one solution.

Any 2 different points are contained together by exactly 1 line, and similarly any 2 different lines have exactly one common intersection point.

For axiom 4, it is enough to show 4 points in general position: $(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 1)$, these are different and no 3 of them is on the same line. □

11 Extremal problems

The task is to find maximum or minimum value of a function over a given set.

11.1 Forbidden subgraphs

F is a fixed forbidden subgraph. $ex(n,F)$ is the max number of edges in a graph on n vertices that does not contain the forbidden graph F as a subgraph.

Special cases:

- $ex(n,F)=\binom{n}{2}$ if $n<|V(F)|$.
- $ex(n,K_2)=0$.
- $ex(n,P_3)=lb\left(\frac{n}{2}\right)$, where lb is the whole part.

Theorem 11.1. *If a graph G contains no K_3 as a subgraph, then*

$$|E|\leq lb\left(\frac{n^2}{4}\right),$$

and $|E|=lb\left(\frac{n^4}{4}\right)$ iff it is the complete bipartite graph $K_{ub\left(\frac{n}{2}\right),lb\left(\frac{n}{2}\right)}$

Theorem 11.2 (Turán)**.** *For $n\geq p$ the graph having the largest number of edges without containing K_p as a subgraph is obtained by:*

- *Partition of vertices into $p-1$ classes.*
- *Two vertices are connected iff they belong to different classes.*

From this

$$ex(n,K_p)=\binom{n}{2}-\sum_{i=0}^{p-1}\binom{lb\left(\frac{n+i}{p-1}\right)}{2}.$$

Theorem 11.3. *For all $p\geq 3$, $p\in\mathbb{N}$ and $n>p$,*

$$ex(n,K_p)\leq\frac{n^2}{2}-(p-1)\frac{\left(\frac{n}{p-1}\right)^2}{2}.$$

Proof. Begin by

$$\frac{n^2}{2}-(p-1)\frac{\left(\frac{n}{p-1}\right)^2}{2}=\frac{n^2}{2}-\frac{n^2}{2(p-1)}=\frac{n^2(p-2)}{2(p-1)}.$$

If a graph G does not contain K_p as a subgraph, then $\omega(G)\leq p-1$ or $\alpha(\overline{G})\leq p-1$.

For any graph G,

$$\alpha(G)\geq\sum_{v\in V}\frac{1}{d(v)+1}.$$

By the claim for $\overline{G}$,

$$\alpha(\overline{G})\geq\sum_{v\in V}\frac{1}{(n-1-d(v))+1}=\sum_{v\in V}\frac{1}{n-d(v)}.$$

If the graph G does not contain K_p as a subgraph then

$$p-1\geq\sum_{v\in V}\frac{1}{n-d(v)}.$$

By the inequality for the convex function $f(x)=\frac{1}{x}$,

$$\frac{\sum_{i=1}^n f(x_i)}{n}\geq f\left(\frac{\sum_{i=1}^n x_i}{n}\right)\quad n\in\mathbb{N}$$

$$\frac{\sum_{v\in V}\frac{1}{n-d(v)}}{n} \geq \frac{1}{\frac{\sum_{v\in V}(n-d(v))}{n}} = \frac{n}{n^2-\sum_{v\in V}d(v)} = \frac{n}{n^2-2e}.$$

$$p-e \geq \sum_{v\in V}\frac{1}{n-d(v)} \geq \frac{n^2}{n^2-2e}, \quad n^2-2e>0 \forall G.$$

$$(p-1)(n^2-2e) \geq n^2$$
$$(p-1)n^2-(p-1)2e \geq n^2$$
$$(p-2)n^2 \geq (p-1)2e$$
$$\frac{(p-2)n^2}{2(p-1)} \geq e.$$

This is an upper bound for the number of edges graphs that do not contain K_p as a subgraph. □

To prove the claim, that for any graph G the independence number

$$\alpha(G) \geq \sum_{v\in V}\frac{1}{d(v)+1} = s(G)$$

Proof 1: Greedy selection of vertices. At the beginning $S=\emptyset$. In each step select a vertex v of G with minimal degree:

- $S := S\cup\{v\}$,
- $G := G\backslash\{v\cup N(v)\}$.

At the end S is an independent vertex set.

$$\alpha(G) \geq 1+\alpha(G\backslash\{v\cup N(v)\}) \geq 1+s(G\backslash\{v\cup N(v)\}) \geq s(G)$$

□

Proof 2: Greedy deletion of vertices. While G has edges, delete a vertex v with maximal degree. $s(G)$ is decreased by $\frac{1}{d(v)+1}$ and increased by

$$\sum_{v_i\in N(v)}\left(\frac{1}{d(v_i)}-\frac{1}{d(v_i)+1}\right) = \sum_{v_i\in N(v)}\frac{1}{d(v_i)(d(v_i)+1)} \geq \sum_{v\in N(v)}\frac{1}{d(v)(d(v)+1)} = d(v)\frac{1}{d(v)(d(v)+1)} = \frac{1}{d(v)+1}.$$

The value of s during the algorithm does not decrease. $s(G-v)\geq s(G)$. At the end we get a set S with no edges, an independent vertex set for which the inequality also holds

$$s(S)\geq s(G), \quad s(S)=\sum_{v\in S}\frac{1}{d(v)+1} = |S|.$$

By definition $\alpha(G)\geq|S|$ and we have seen that $s(S)\geq s(G)$, which implies

$$\alpha(G)\geq s(G).$$

□

11.2 Routings

In this part we assume that G is a connected graph.

Definition 11.1 (Routing)**.** *A routing $\mathcal{R}$ in a graph G is a collection of $n(n-1)$ paths. For each ordered pair (v_i,v_j), $v_i,v_j\in V(G)$ there is a path P_{ij} starting at v_i and ending at v_j.*

Definition 11.2 (Load of a vertex)**.** *In a routing $\mathcal{R}$ of the graph G the load of the vertex $v_i\in V$ is the number of pahts P_{jk} containing v_i as an interior point. Notation: $\xi_{\mathcal{R}}(v_i)$.*

Definition 11.3 (Forwarding index)**.** *The forwarding index $\xi(G)$ of a graph G is*

$$\xi(G) = \min_{\mathcal{R}}\max_{1\leq i\leq n}\xi_{\mathcal{R}}(v_i).$$

Theorem 11.4. *For any connected graph G with n vertices and m edges*

$$\xi(G) \geq \frac{2}{n} \sum_{1\leq i\leq j\leq n} (d(v_i, v_j) - 1) \geq n - 1 - \frac{2m}{n},$$

where d is the distance of vertices v_i, v_j.

Proof. Let $\mathcal{R}$ be any routing. Any path connecting the vertices v_i and v_j has at least $d(v_i, v_j) - 1$ interior vertices.

Every pair v_i, v_j of vertices is connected by two paths P_{ij} and P_{ji} adding 2 to the load at least $d(v_i, v_j) - 1$ vertices:

$$\sum_{k=1}^{n} \xi_{\mathcal{R}}(v_k) \geq 2 \sum_{1\leq i\leq j\leq n} (d(v_i, v_j) - 1) \geq 2\left[\binom{n}{2} - m\right] = n(n-1) - 2m$$

for all routings.

$$\xi(G) = \min_{\mathcal{R}} \max_{1\leq i\leq n} \xi_{\mathcal{R}}(v_i) = max_{1\leq i\leq n}\xi_{\mathcal{R}_0}(v_i) \geq \frac{1}{n}\sum_{i=1}^{n} \xi_{\mathcal{R}_0}(v_i) \leq \frac{1}{n}\cdot 2\cdot \sum_{i=1}^{n}(d(v_i, v_j) - 1) \geq \frac{1}{n}(n(n-1) - 2m).$$

□

Theorem 11.5. *For infinitely many values of n there are graphs on n vertices, where*

$$\Delta(G) < \sqrt{n} + \frac{1}{2}, \quad \xi(G) < n.$$

Proof. We apply a similar method as the construction of Galois planes. Let q be a prime power, $n = q^2 + q + 1$.

- vertices: $(a, b, c) \neq (0, 0, 0)$ homogeneous coordinates.
- adjacent vertices: (a, b, c) and (a', b', c') where $aa' + bb' + cc' = 0$.

Claim 1: Every vertex of the so generated graph G has degree q or $q + 1$.

The equation $ax + by + cz = 0$ has q^0, but $(0, 0, 0)$ is one of them, so $q^2 - 1$ valid solutions, so every point is repserented $q - 1$ times: $\frac{q^2-1}{q-1} = q + 1$ different.

If all of them are different from (a, b, c), then it has degree $q + 1$.

If one of them is (a, b, c) it has degree q.

Claim 2: The so called generated graph G has diameter 2. For any two vertices there is a path of length at most 2 connecting them.

Consider the points (a, b, c) and (a', b', c') and the points with the same coordinates in the Galois plane $PG(2, q)$. By the axiom 1 there exists a line $[x, y, z]$ in $PG(2, q)$ that is incident to both of the points.

- The vertex (x, y, z) is adjacent to both vertices in the graph.
- Through (x, y, z) there is a path of length 2 connecting the two vertices, so their distance is at most 2.

Let us denote the vertices of G by $v_1, v_2, \ldots, v_n$.

- If $v_iv_j \in E$ let this edge be the paths P_{ij} and P_{ji} as well.
- If $v_iv_j \notin E$ by Claim 2 there exists a v_k common neighbor of v_i and v_j. Set $P_{ij} = v_iv_kv_j$ and $P_{ji} = v_jv_kv_i$.

The paths in the first case do not load any vertices. The paths in the second vertex load the vertex v_k by 2. The loading of a vertex v_k is at most $2\binom{d(v_k)}{2} \leq (q + 1)q$.

$$\xi(G) = \min\max \xi_{\mathcal{R}}(v_k) \leq (q + 1)q = q^2 + q = n - 1 \implies \xi(G) < n.$$

By Claim 1 $\Delta(G) \leq q+1$

$$q+1 = \left(q+\frac{1}{2}\right)+\frac{1}{2} = \sqrt{\left(q+\frac{1}{2}\right)^2}+\frac{1}{2} = \sqrt{q^2+q+\frac{1}{4}}+\frac{1}{2} < \sqrt{q^2+q+2}+\frac{1}{2} = \sqrt{n}+\frac{1}{2}$$

$$\Delta(G) < \sqrt{n}+\frac{1}{2}.$$

□

11.3 The Turán problem for 4–cycles

Proposition 11.1. *The graph G constructed in the proof of the previous Theorem is C_4–free.*

Proof. In the proof of Claim 2 we have seen that if $v_i \in V$ and $v_i \sim (a,b,c)$ and $v_j \in V$ and $v_j \sim (a',b',c')$, there exists a $v_k \in V$, $v_k \sim (x,y,z)$ neighbour of both v_i and v_j.

v_k can be imagined as a line $[x,y,z]$ in $PG(2,q)$ containing the points (a,b,c) and (a',b',c'). By axiom 1 there is exactly one such line, so there is exactly one common neighbor.

If G would have C_4 as a subgraph, there would be at leas 2 common neighbours. □

Corollary 11.1. *If q is a prime power and $n = q^2+q+1$, then $ex(n, C_4) \geq |E(G)|$ if G is the constructed graph.*

Proof. By Claim 1 $\forall v \in V$ $d(v) \geq q$, so $|E(G)| \geq \frac{nq}{2}$, and $n = q^2+q+1 < (q+1)^2$, so $\sqrt{n}-1 < q$. From this

$$\frac{n(\sqrt{n}-1)}{2} < |E(G)| \leq ex(n, C_4).$$

□