Contents I

1 Intersection-Graphs
   - Basics
   - Problems

2 Interval-graphs
   - Introduction
   - Colouring
   - Independent Sets and Cliques

3 Permutation-Graphs
   - Introduction
   - Colouring

4 Arc-Graphs

5 Circle-Graphs
   - Introduction
   - Colouring
   - Construction
   - Colourings
   - Independent Sets and Cliques

6 Concluding Remarks
   - Segment-graphs
   - Disk-graphs
   - Overview
Basics

- A graph consists of nodes, which are “connected” by some relation.
- Often we have objects, for which some relation exists.
- Possible relations:
  - Objects have some common property.
  - Objects are neighbours.
  - Objects have some limited distance.
  - Objects intersect.
- We define intersection-graphs using the later relation.
Definition

A graph \( G = (V, E) \) is called intersection-graph of a set \( M \) of objects, iff \( G = (V, E) \) is isomorphic to \( H = (M, \{\{a, b\} \mid a \cap b \neq \emptyset\}) \). \( M \) is called the intersection representation of \( G \).

Possible families of objects are:

- Intervals on a line.
- Arc of a circle.
- Chords of a circle.
- Circles in the plane.
- Parallelograms between two lines.
- And lots more.

By using different classes of object we get different graph classes.
Colouring

**Definition**

- A graph $G = (V, E)$ is $k$-colourable iff:
  - $\exists f : V \mapsto \{1, \ldots, k\} : \forall (a, b) \in E, f(a) \neq f(b)$.
  - The function $f$ is called a colouring of $G$.

**Definition**

- $\chi(G)$ is the chromatic number $\chi(G)$ of $G$, iff
  - $G$ is $\chi(G)$-colourable, but is not $(\chi(G) - 1)$-colourable.
Colouring Problems

Definition

The graph-to-colour problem is the following:
Input: G a graph
Output: Optimal colouring of G.

Definition

The colouring problem is the following:
Input: k ∈ \mathbb{N} and a graph G
Output: Is G k-colourable?

Definition

The k-colouring problem is the following:
Input: G a Graph
Output: Is G k-colourable?
**Independent Set**

**Definition**

- A graph $G = (V, E)$ contains an independent set of size $k$, iff
- $\exists S \subset V : |S| = k \land \forall a, b \in S, a \neq b : (a, b) \notin E.$

**Definition**

- $\alpha(G)$ denotes the size of the largest independent set:
- $G$ contains an independent set of size $\alpha(G)$, but no independent set of size $\alpha(G) + 1$. 
Definitions

**Definition**

Let $G = (V, E)$ be a graph.

\[
\alpha(G) = \max\left\{ |V'| ; \ V' \subset V \land \forall a, b \in V' : (a, b) \notin E \right\}
\]

\[
\omega(G) = \max\left\{ |V'| ; \ V' \subset V \land \forall a, b \in V' : (a, b) \in E \right\}
\]

\[
\chi(G) = \min\left\{ k ; \ \exists V_1, V_2, \ldots, V_k : \bigcup_{i=1}^{k} V_i = V \land \right. \\
                      \left. \forall i : 1 \leq i \leq k : \forall a, b \in V_i : (a, b) \notin E \right\}
\]

\[
\overline{\chi}(G) = \min\left\{ k ; \ \exists V_1, V_2, \ldots, V_k : \bigcup_{i=1}^{k} V_i = V \land \right. \\
                      \left. \forall i : 1 \leq i \leq k : \forall a, b \in V_i : (a, b) \in E \right\}
\]

**More notations:**

\[
\omega(G) = \overline{\alpha}(G),
\]

\[
\alpha(G) = \overline{\omega}(G) = \beta_0(G),
\]

\[
\kappa(G) = \overline{\chi}(G)
\]
Bounds

Lemma

- $\chi(G) \geq \omega(G)$
- $\chi(G) \geq n/\alpha(G)$
- $\chi(G) = \max\{\chi(B) \mid B\text{ is a block in } G\}$

Block is another name for 2-connected component.

Theorem

Let $G = (V, E)$ be a graph with maximal node degree of $d$ ($d = \Delta(G)$). Then $\chi(G) \leq d + 1$ holds.

Theorem

Let $G = (V, E)$ be a graph. Then $\chi(G) = 2$ holds, iff $G$ contains no cycles of odd length.
First simple Example

- Time of activity of a register (construction of a compiler)
- Program segments: \( \cdots \text{Read}(A) \cdots \text{Write}(B) \cdots \)
- Living time of a variable \( A \): Maximal interval
  - Starting with a \( \text{Write}(A) \).
  - Ending by the last \( \text{Read}(A) \).
  - Such that no further \( \text{Write}(A) \) is between this two points.
- Problem: how many registers are needed?
- D.h. assign for each living time of a variable a register.
- Example: \((0, 10), (3, 7), (9, 20), (25, 50), (12, 34), (6, 16), (17, 26), (11, 46), (23, 26), (30, 46), (19, 27)\)
Definition (Interval-graphs)

- A graph $G = (V, E)$ is called **intervall-graph**, iff
- it is the intersection graph of a set of intervals on a line.
- An interval-graph is called **proper**, iff no interval is contained in an other interval.
Idea: look for independent sets.
Model and Colouring (Idea)

Idea: check the intervals from left to the right (sorted by the left endpoints):
Model and Colouring (Invariant)

Determine the invariant:

```
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

![Diagram showing the invariant](image-url)
Colouring of Interval-graphs (Algorithm)

**Theorem**

*The graph-to-colour problem is for interval-graphs in time \(O(n \log(n))\) solvable.*

1. Sort the intervals by their left endpoints.
2. Check all endpoints \(e\) from the left to the right.
3. If \(e\) is the starting point of an interval, colour it with the smallest free colour.
4. If \(e\) is the ending point of an interval \(I\) is, free the colour of \(I\).

**Invariant**

*If a node \(v\) is coloured with colour \(k\), then \(v\) is part of a \(k\)-clique.*
Example of independent set problem on interval graphs

1. Sort the intervals by their starting points.
2. Go through all starting points \( e \) from left to right.
3. Store for each interval \( I \) the size of a maximal independent set of intervals, which contain \( I \) as the rightmost interval.
Independent Set Problem for Interval-graphs

**Theorem**

*Finding a maximal independent set is solvable in time \( O(n \log(n)) \) on interval-graphs.*

1. Sweep through the start- and endpoints of intervals from left to right.
2. Store for each endpoint \( e \) the size of a maximal independent set of intervals, which is placed to the left of \( e \).
3. While sweeping from left to right do:
   1. If \( e \) is a starting point of interval \((e, f)\) and there is no endpoint to the left of \( e \), then let \( S(f) = 1 \).
   2. If \( e \) is a starting point of interval \((e, f)t\), then compute: largest endpoint \( e' \) to the left of \( e \) and let \( S(f) = S(e') + 1 \).
   3. If \( e \) is an endpoint of interval \((a, e)\), then compute: largest endpoint \( e' \) to the left of \( e \) and to the right of \( a \). If that exists, then let \( S(e) = \max(S(e'), S(e)) \).
Maximal Clique on Interval-graohs

Theorem

*Finding a maximal clique is sovable in time $O(n \log(n))$ on interval-graphs.*

Remark

Very many problems are efficient solvable on interval-graphs.
Permutation-Graphs

Definition (Permutations-Graph)

- A graph $G = (V, E)$ is called permutation-graph,
- iff he is definable by a permutation $\pi : \{1..n\} \rightarrow \{1..n\}$ in the following way:
- $G = (\{1..n\}, \{(i, j); (i - j)(\pi(i) - \pi(j)) < 0\})$.

Theorem

A permutation-graph is the intersection graph of a set of lines, which are drawn between to parallel lines.
Example and Colouring

The invariant is the same as the one on interval graphs.
Colouring Problem on Permutation-Graphs

**Theorem**

*The graph-to-colour problem is solvable in time $O(n \log(n))$ on permutation-graphs.*

Idea: Analog algorithms as on interval-graphs.

**Theorem**

*Finding a maximal independent set is solvable in time $O(n \log(n))$ on permutation-graphs.*

Idea: Analog algorithms as on interval-graphs.

**Theorem**

*Finding a maximal clique is solvable in time $O(n \log(n))$ on permutation-graphs.*

Idea: Analog algorithms as on interval-graphs.
**Definition (Arc-Graph)**

- A graph $G = (V, E)$ is called arc-graph,
- iff he is the intersection graph of a set of arcs on a circle.
- A arc-graph is called **proper**, iff no arc in contained in an other arc.

**Remark**

An interval-graph is an arc-graph.

**Question:**

Are the algorithms for interval-graphs adaptable to arc-graphs.
Reasoning for the above Results

- Question, what is the reason that the above problems are efficient solvable on interval-graphs?

- Consider the “flow of information”, i.e.:

- Which information is used (stored) when the algorithms move from left to right.

- One could think, all \( k! \) colourings should be considered (stored).

- But, the colourings are exchangeable.

- Thus only the optimal colouring at each position is stored.

Question:
What is the situation on arc-graphs?
Colouring on Arc-Graphs (Idea)

- Consider the flow of information.
- What information has to be considered when moving around the circle?
- The colouring are not exchangeable because the end the colours have to match.
- Thus we may have to consider $k!$ colourings.
- If $k$ is constant, then the problem is in $\mathcal{P}$
- IF $k$ is not constant, then the problem could be in $\mathcal{NPC}$. 

---

- Consider the flow of information.
- What information has to be considered when moving around the circle?
- The colouring are not exchangeable because the end the colours have to match.
- Thus we may have to consider $k!$ colourings.
- If $k$ is constant, then the problem is in $\mathcal{P}$
- IF $k$ is not constant, then the problem could be in $\mathcal{NPC}$. 

---
Theorem

The $k$-colouring problem on arc-graphs is solvable in polynomial time.

Idea: Consider all $k!$ colourings.

1. W.l.o.g.: The graph contains no $k + 1$ clique.
2. Otherwise we search analog as on interval-graphs for the largest clique.
3. Colour an some maximal $k'$-Clique.
4. Colour the arcs in a clockwise order.
5. At most $k!$ colourings are considered (stored) during this process.
6. Check at the end if some colouring do not contradict with the first one.
7. Running time: $O(k!^2 \cdot n \log n) = O(n \log n)$
Colouring Problem on Arc-Graphs

Theorem

The colouring problem on arc-graphs NP-complete.

Idea: Reduction to the word problem for symmetric groups.

Definition

The word problem for symmetric groups is the following:
Input: $\pi \in S_k$ (Word and symmetric group) and $S_1, S_2, \ldots, S_n$ subgroups
Output: Holds: $\pi \in S_1 \odot S_2 \odot \cdots \odot S_n$
Colouring Problem on Arc-Graphs

\begin{itemize}
\item $S_1 = \{2, 4\}$
\item $S_2 = \{4, 6\}$
\item $S_3 = \{1, 3\}$
\item $S_4 = \{1, 6\}$
\end{itemize}
Circle-Graphs

Definition (Circle-Graphs)

- A graph $G = (V, E)$ is called circle-graph,
- iff it is the intersection graph of a set of chords within one circle.

Definition (Overlap-Graph)

- A graph $G = (V, E)$ is called overlap-graph,
- iff it is definable by the overlapping of a set of intervals on a line.
- Let $I$ be a set of intervals.
- Then the corresponding overlap-graph is:
  $G = (I, \{(a, b) \mid a, b \in I \land a \setminus b \neq \emptyset \land b \setminus a \neq \emptyset \land a \cap b \neq \emptyset\})$
Example
Statements on Circle-Graphs

Lemma

1. An interval-graph is an arc-graph.
2. A proper arc-graph is a circle-graph.
3. A permutation-graph is a circle-graph.
4. A graph $G$ is a circle-graph, iff $G$ is a overlap-graph.

Just show: a graph $G$ is a circle-graph, iff $G$ is a overlap-graph.

- Chord $A$ from $r \cdot e^{i \cdot a}$ to $r \cdot e^{i \cdot a'}$ becomes interval $A' = (a, a')$ $(0 \leq a < a' < 2 \cdot \pi)$.
- Chord $B$ from $r \cdot e^{i \cdot b}$ to $r \cdot e^{i \cdot b'}$ becomes interval $b' = (b, b')$ $(0 \leq b < b' < 2 \cdot \pi)$.
- The chord $A$ crosses $B$, iff $a < b < a' < b'$ oder $b < a < b' < a'$.
- Interval $A'$ has an overlap with $B$, iff $a < b < a' < b'$ oder $b < a < b' < a'$. 
A graph $G$ is a circle-graph, iff $G$ is a overlap-graph
Colouring of Circle-Graphs (Idea)

- What is the flow of information?
- Crossing chords “limit” the flow of information.
- But: information about the colouring of pairs of chords could an idea.
- Thus, the 4-colouring problem on circle-graphs could be NP-complete.
- And the 3-colouring on circle-graphs could still be in \( \mathcal{P} \).
Colouring Problems (Overview)

**Theorem**

The 4-colouring problem on circle-graphs is NP-complete.

**Theorem**

The 3-colouring problem on circle-graphs is solvable in time $O(n \log(n))$. 
4-Colouuring Problem on Circle-Graphs

- Reduction from the 3-SAT Problem.
- For a given 3-SAT formula $\mathcal{F}$ we construct a circle-graph $G$.
- It has to hold: $\mathcal{F}$ satisfiable $\iff$ $G$ 4-colourable.
- Problem: Coding of logical values by the colouring of cords.
- Idea: Each pair of chord $(a, b)$ codes a logical value of $v$.
- Holding: $v \iff f(a) = f(b)$ for a colouring $f$.
- Construct some kind of “circuit”.

Component Negation I \((x = \neg y)\)
The Negation

Negation II: $x = \neg y$

Combination of Colours

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$a$</th>
<th>$b$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>2,3</td>
<td>4,4</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td>1,1</td>
<td>2,3</td>
<td>4,4</td>
<td>1,3</td>
<td></td>
</tr>
<tr>
<td>1,1</td>
<td>2,4</td>
<td>3,3</td>
<td>1,4</td>
<td></td>
</tr>
<tr>
<td>1,1</td>
<td>2,3</td>
<td>4,4</td>
<td>2,3</td>
<td></td>
</tr>
<tr>
<td>1,1</td>
<td>2,4</td>
<td>3,3</td>
<td>2,4</td>
<td></td>
</tr>
<tr>
<td>1,1</td>
<td>3,4</td>
<td>2,2</td>
<td>3,4</td>
<td></td>
</tr>
</tbody>
</table>
Some Simple Components

Negation II:  
\[ x = \neg y \]

Equality:  
\[ x = y \]

Static XOR:  
\[ x = y \oplus e \]
Equality: \((x = y = z)\)

- \(\neg y \Rightarrow a_1 \Rightarrow a_2 \Rightarrow \neg z\)
- \(\neg y \Rightarrow b_1 \Rightarrow \neg x\)
- \(y \Rightarrow \neg a_1 \Rightarrow \neg a_2 \Rightarrow z\)
- \(y \Rightarrow \neg a_2 \Rightarrow b_2 \Rightarrow \neg b_1 \Rightarrow x\)
- A colouring is possible in all cases.
Equality: \((x = x' \land y = y')\)

\[\begin{array}{cccc}
  y & y_1 & y_2 & y_3 \\
1,1 & 1,2 & 1,1 & 1,2 & 1,1 \\
1,1 & 1,2 & 1,1 & 1,2 & 2,2 \\
1,1 & 1,2 & 1,1 & 1,3 & 3,3 \\
1,1 & 1,2 & 1,1 & 1,4 & 4,4 \\
1,2 & 1,1 & 1,2 & 1,1 & 1,2 \\
1,2 & 1,1 & 1,2 & 1,1 & 1,3 \\
1,2 & 1,1 & 1,2 & 1,1 & 1,4 \\
1,2 & 1,1 & 1,2 & 2,2 & 2,3 \\
1,2 & 1,1 & 1,2 & 2,2 & 2,4 \\
1,2 & 1,1 & 1,3 & 3,3 & 3,4 \\
\end{array}\]
Equality \((x = y = z)\)
More Simple Components

Weak Or:
\[ \neg x \land \neg z \Rightarrow \neg y \]

Weak Negation:
\[ \neg x \Rightarrow y \text{ and } \neg y \Rightarrow x \]

True:
\[ x = true \]
Or \((x \lor y = z)\)

\[\neg x \land \neg y \Rightarrow \neg x_3 \land \neg y_1 \Rightarrow \neg z_3 \Rightarrow \neg z\]

\[x \Rightarrow \neg x' \Rightarrow z_1 \Rightarrow z\]

\[y \Rightarrow \neg y' \Rightarrow z_4 \Rightarrow z\]

A colouring is possible in all cases.
Static Simple Clause

\[ x_2 = x'_2 \text{ and } (x_1 \oplus e_1) \lor (x_2 \oplus e_2) \lor (x_3 \oplus e_3) = true \]
Multiple Equality \((x_i = y_i)\) [with Transport \((z_0 = z_k)\)]
Clause \( (x_i = y_i \text{ und } c_i \text{ satisfied}) \)
Formula (all $c_i$ are satisfied)
Colouring problems

Theorem

The $k$-colouring problem on circle-graphs is NP-complete for $k \geq 4$.

Theorem

The $(2 \cdot k - 1)$-colouring problem on circle-graphs with clique size $k$ is NP-complete for $k \geq 3$.

Theorem

A circle-graph with clique size $k$ is always $(3 \cdot k)$-colourable.
Indepenedet Set and Clique

**Theorem**

*Findiing a maximal independent set is solvable in time $O(n \log(n))$ on circle-graphs.*

**Theorem**

*Finding a maximal clique is solvable in time $O(n \log(n))$ on circle-graphs.*
Concluding Remarks

Theorem

On an interval graph $G$ we may in time $O(n \log(n))$ compute $\chi(G)$, $\alpha(G)$ and $\omega(G)$.

Theorem

On a permutation graph $G$ we may in time $O(n \log(n))$ compute $\chi(G)$, $\alpha(G)$ and $\omega(G)$.

Theorem

The $k$-colouring problem on arc-graphs is solvable in polynomial time, but the colouring problem for arc-graphs is NP-complete.

Theorem

The 3-colouring on circle-graphs is solvable in time $O(n \log(n))$. The $k$-colouring problem on circle-graphs is NP-complete for $k \geq 4$. 
Conclusions

- Colouring (and many more problems) on interval graphs are easy.
- $k$-colouring on arc-graphs is easy.
- Colouring on arc-graphs is hard.
- 4-colouring problem on circle-graphs is hard.
- 3-colouring problem on circle-graphs is easy.
- $k$-colouring problem on circle-graphs is hard for $k > 3$. 
**g-Segment-graphs**

**Definition (g-Segment-graphs)**

- A graph \( G = (V, E) \) is called a \( g \)-Segment-graph, iff
- it is the intersection-graph of a set of chords within a regular \( g \)-polygon.

**Lemma**

We have:

1. A permutation-graph is a circle-graph.
2. A permutation-graph is a \( g \)-segment-graph.
3. A proper arc-graph is a circle-graph.
4. There are proper arc-graphs, which are not \( g \)-segment-graphs.
Disk-graphs

Definition (Disk-graphs)

- A graph $G = (V, E)$ is called disk-graph, iff
- it is a intersection-graph of a set of disks in the plane.

Definition (Unit-Disk-graphs)

- A graph $G = (V, E)$ is called unit-disk-graph, iff
- it is a intersection-graph of a set of equally sized disks in the plane.
## Overview of the Results

<table>
<thead>
<tr>
<th></th>
<th>$k$-Col.</th>
<th>Col.</th>
<th>Opt-Col</th>
<th>Ind.</th>
<th>Clique</th>
<th>Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intervall</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>Permut.</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>Circle-G.</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$-hard</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>$g$-Segment</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>Disk-G.</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$-hard</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{NP}$</td>
</tr>
<tr>
<td>Planar</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{NP}$-hard</td>
<td>$\mathcal{NP}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>$k$-Planar</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
</tbody>
</table>
Questions

- How is the colouring problem solvable on interval graphs?
- How is the colouring problem solvable on permutation graphs?
- How is the independent set problem solved on arc-graphs and cycle-graphs?
- How is the clique problem solved on arc-graphs?
- Why is the colouring-problem on arc-graphs hard?
- What is the idea of the reduction of the 4-colouring-problem on cycle-graphs?
<table>
<thead>
<tr>
<th>Legend</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>n</strong> : Not of relevance</td>
<td></td>
</tr>
<tr>
<td><strong>g</strong> : implicitly used basics</td>
<td></td>
</tr>
<tr>
<td><strong>i</strong> : idea of proof or algorithm</td>
<td></td>
</tr>
<tr>
<td><strong>s</strong> : structure of proof or algorithm</td>
<td></td>
</tr>
<tr>
<td><strong>w</strong> : Full knowledge</td>
<td></td>
</tr>
</tbody>
</table>