Graph Classes:
Interrelations, Structure, and Algorithmic Issues

Martin Milanič

UP IAM and UP FAMNIT

Rogla, 19 May 2013

Outline

1. Algorithmic Graph Problems and Graph Classes
2. Perfect Graphs and Their Subclasses
3. Numerically Defined Graph Classes

ALGORITHMIC GRAPH PROBLEMS AND GRAPH CLASSES.

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a **"substructure"** of G satisfying certain criteria

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a **"substructure"** of G satisfying certain criteria and/or optimizing a given objective function

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a **"substructure"** of G satisfying certain criteria and/or optimizing a given objective function (or determine that such a substructure does not exist).

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a **"substructure"** of G satisfying certain criteria and/or optimizing a given objective function (or determine that such a substructure does not exist).

"substructure" could be any of the following:

- a subset of vertices,
- a subset of edges,
- a partition of the vertex set,
- a partition of the edge set,
- etc.

Algorithmic Graph Problems

Many algorithmic graph problems are of the following form:

- Given an input graph $G = (V, E)$, find a **"substructure"** of G satisfying certain criteria and/or optimizing a given objective function (or determine that such a substructure does not exist).

"substructure" could be any of the following:

- a subset of vertices,
- a subset of edges,
- a partition of the vertex set,
- a partition of the edge set,
- etc.

Typically, every **search problem** of the above form also has a corresponding **decision problem**.

Examples

CLIQUE

Input: Graph $G = (V, E)$, integer k
Question: Does G contain a clique of size k?

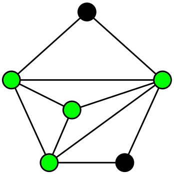
clique: a set of pairwise adjacent vertices

Examples

CLIQUE

Input:	Graph $G = (V, E)$, integer k
Question:	Does G contain a clique of size k?

clique: a set of pairwise adjacent vertices



STABLE SET

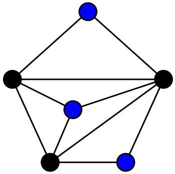
Input: Graph $G = (V, E)$, integer k
Question: Does G contain a stable set of size k?

stable (independent) set: a set of pairwise non-adjacent vertices

STABLE SET

Input:	Graph $G = (V, E)$, integer k
Question:	Does G contain a stable set of size k?

stable (independent) set: a set of pairwise non-adjacent vertices



Examples

VERTEX COVER

Input: Graph $G = (V, E)$, integer k

Question: Does G contain a vertex cover of size k?

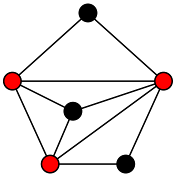
vertex cover: a set of vertices hitting all edges

Examples

VERTEX COVER

Input:	Graph $G = (V, E)$, integer k
Question:	Does G contain a vertex cover of size k?

vertex cover: a set of vertices hitting all edges



Examples

DOMINATING SET

Input: Graph $G = (V, E)$, integer k

Question: Does G contain a dominating set of size k?

dominating set: a set of vertices such that every vertex not in the set has a neighbor in it

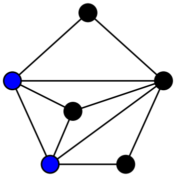
Examples

DOMINATING SET

Input: Graph $G = (V, E)$, integer k

Question: Does G contain a dominating set of size k?

dominating set: a set of vertices such that every vertex not in the set has a neighbor in it



Examples

COLORABILITY

Input: Graph $G = (V, E)$, integer k

Question: Does G admit a k-coloring?

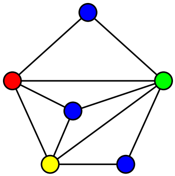
k**-coloring**: a partition of V into k (pairwise disjoint, possibly empty) stable sets

Examples

COLORABILITY

Input: Graph $G = (V, E)$, integer k
Question: Does G admit a k-coloring?

k**-coloring**: a partition of V into k (pairwise disjoint, possibly empty) stable sets



Complexity

All these problems

- Clique
- Stable Set
- Vertex Cover
- Dominating Set
- Colorability

(as well as hundreds of others) are NP-complete.

Coping With Intractability

There are several approaches to coping with the intractability of NP-hard problems:

- polynomial algorithms for particular input instances,
- approximation algorithms,
- heuristics, local optimization,
- "efficient" exponential algorithms (e.g., 1.5^n instead of 2^n),
- randomized algorithms,
- parameterized complexity
  (fixed-parameter tractable (FPT) algorithms $O(f(k)n^{O(1)})$),
- etc.

Coping With Intractability

There are several approaches to coping with the intractability of NP-hard problems:

- polynomial algorithms for particular input instances,
- approximation algorithms,
- heuristics, local optimization,
- "efficient" exponential algorithms (e.g., 1.5^n instead of 2^n),
- randomized algorithms,
- parameterized complexity
  (fixed-parameter tractable (FPT) algorithms $O(f(k)n^{O(1)})$),
- etc.

Graph Classes

A **graph class** = a set of graphs closed under isomorphism.

Graph Classes

A **graph class** = a set of graphs closed under isomorphism.

Examples:

- Planar graphs
- Connected graphs
- Trees
- Forests
- Bipartite graphs
- 3-colorable graphs
- Perfect graphs
- Cayley graphs
- vertex-transitive graphs

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X,

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X, then Π is also NP-hard for graphs in Y.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X, then Π is also NP-hard for graphs in Y.
  - If $X \subseteq Y$ and a problem Π is polynomially solvable for graphs in Y,

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X, then Π is also NP-hard for graphs in Y.
  - If $X \subseteq Y$ and a problem Π is polynomially solvable for graphs in Y, then Π is also polynomially solvable for graphs in X.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X, then Π is also NP-hard for graphs in Y.
  - If $X \subseteq Y$ and a problem Π is polynomially solvable for graphs in Y, then Π is also polynomially solvable for graphs in X.
- **Characterizations** of graphs in a given class.

Main Questions

In the study of graph classes the following questions are of central interest:

- **Computational complexity of a given problem** within a particular class.
  - Typically either polynomial or NP-hard.
  - Development of algorithmic techniques.
- **Relationships** between different graph classes.
  - If $X \subseteq Y$ and a problem Π is NP-hard for graphs in X, then Π is also NP-hard for graphs in Y.
  - If $X \subseteq Y$ and a problem Π is polynomially solvable for graphs in Y, then Π is also polynomially solvable for graphs in X.
- **Characterizations** of graphs in a given class.
- Computational complexity of **recognizing** graphs in a given class.

Recognition Problems

For a given graph class X we can define the following problem:

RECOGNITION OF GRAPHS IN X	
Input:	Graph G.
Question:	Is $G \in X$?

Recognition Problems

For a given graph class X we can define the following problem:

RECOGNITION OF GRAPHS IN X	
Input:	Graph G.
Question:	Is $G \in X$?

Examples:

- If $X =$ the class of all 3-colorable graphs, the recognition problem is NP-complete.

Recognition Problems

For a given graph class X we can define the following problem:

RECOGNITION OF GRAPHS IN X	
Input:	Graph G.
Question:	Is $G \in X$?

Examples:

- If $X =$ the class of all 3-colorable graphs, the recognition problem is NP-complete.
- If $X =$ the class of planar graphs, the recognition problem is solvable in linear time.

www.graphclasses.org

Information System on Graph Classes and their Inclusions

- ISGCI home
- The Java application
- All classes
- References
- Smallgraphs
- About ISGCI
- Screenshots
- News
- FAQ
- Contact
- Impressum

Database contents

1437 classes
167680 inclusions
updated 2013-03-13

Latest news

2013-03-01 Smallgraphs available in graph6 format.

2012-09-08 The ISGCI database is included in Sage.

2012-01-14 'Find relation' in the Java application now gives a witness for proper inclusions.

Find class

What is ISGCI?

ISGCI is an encyclopaedia of graphclasses with an accompanying java application that helps you to research what's known about particular graph classes. You can:

- check the relation between graph classes and get a witness for the result
- draw clear inclusion diagrams
- colour these diagrams according to the complexity of selected problems
- find the P/NP boundary for a problem
- save your diagrams as Postscript, GraphML or SVG files
- find references on classes, inclusions and algorithms

Classic classes

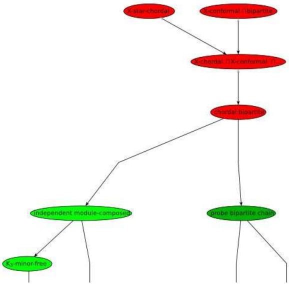
Meyniel
P_4-bipartite
P_4-reducible
P_4-sparse
bipartite
chordal
chordal bipartite
circle
circular arc
clique
cograph
comparability
distance-hereditary
even-hole-free
interval
median

Classes by definition

All classes
Chords & chordality
(De)composition
Forbidden subgraphs
(Forbidden) minors
Helly property
Hypergraphs
Intersection graphs
Matrix
Neighbourhood
Ordering
Partitionable
Perfection
Planarity
Posets
Probe graphs
Threshold
Tolerance

Problems

3-Colourability
Clique
Clique cover
Cliquewidth
Cliquewidth expression
Colourability
Cutwidth
Domination
Feedback vertex set
Hamiltonian cycle
Hamiltonian path
Independent set
Recognition
Treewidth
Weighted clique
Weighted feedback vertex set
Weighted independent set

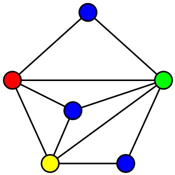


Source: `http://www.graphclasses.org/`

PERFECT GRAPHS AND THEIR SUBCLASSES.

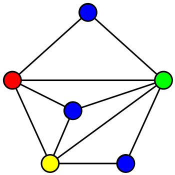
χ and ω

$\chi(G)$: chromatic number of G = the smallest k such that G admits a k-coloring

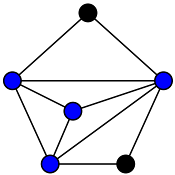


χ and ω

$\chi(G)$: chromatic number of G = the smallest k such that G admits a k-coloring



$\omega(G)$: clique number of G = the maximum size of a clique in G.



Perfect Graphs

Trivially:

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

Perfect Graphs

Trivially:

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

Definition

A graph G is perfect, if

$$\chi(H) = \omega(H)$$

holds for every induced subgraph H of G.

Perfect Graphs

Trivially:

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

Definition

A graph G is perfect, if

$$\chi(H) = \omega(H)$$

holds for every induced subgraph H of G.

The class of perfect graphs is hereditary (closed under vertex deletions).

Perfect Graphs

Theorem (Lovász 1972, Perfect Graph Theorem)

A graph G is perfect if and only if its complement $\overline{G}$ is perfect.

Perfect Graphs

Theorem (Lovász 1972, Perfect Graph Theorem)

A graph G is perfect if and only if its complement $\overline{G}$ is perfect.

Examples of non-perfect graphs:

- odd cycles of order at least 5: $C_5, C_7, C_9, \ldots$

Perfect Graphs

Theorem (Lovász 1972, Perfect Graph Theorem)

A graph G is perfect if and only if its complement $\overline{G}$ is perfect.

Examples of non-perfect graphs:

- odd cycles of order at least 5: $C_5, C_7, C_9, \ldots$
- their complements: $\overline{C_5}, \overline{C_7}, \overline{C_9}, \ldots$

Berge Graphs

Berge graph: a $\{C_5, C_7, \overline{C_7}, C_9, \overline{C_9}, \ldots\}$-free graph.

Claude Berge, 1926–2002, a French mathematician

Source: `http://www.ecp6.jussieu.fr/GT04/`

The Strong Perfect Graph Theorem

Berge graph: a $\{C_5, C_7, \overline{C_7}, C_9, \overline{C_9}, \ldots\}$-free graph.

Clearly, every perfect graph is Berge.

The Strong Perfect Graph Theorem

Berge graph: a $\{C_5, C_7, \overline{C_7}, C_9, \overline{C_9}, \ldots\}$-free graph.

Clearly, every perfect graph is Berge.

Conjecture (Berge 1963)

A graph G is perfect if and only if it is Berge.

The Strong Perfect Graph Theorem

Berge graph: a $\{C_5, C_7, \overline{C_7}, C_9, \overline{C_9}, \ldots\}$-free graph.

Clearly, every perfect graph is Berge.

Conjecture (Berge 1963)

A graph G is perfect if and only if it is Berge.

Strong Perfect Graph Theorem (Chudnovsky, Robertson, Seymour, Thomas 2002)

A graph G is perfect if and only if it is Berge.

The Strong Perfect Graph Theorem

Berge graph: a $\{C_5, C_7, \overline{C_7}, C_9, \overline{C_9}, \ldots\}$-free graph.

Clearly, every perfect graph is Berge.

Conjecture (Berge 1963)

A graph G is perfect if and only if it is Berge.

Strong Perfect Graph Theorem (Chudnovsky, Robertson, Seymour, Thomas 2002)

A graph G is perfect if and only if it is Berge.

Total length of the proof $\approx$ 150 pages.

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

- COLORABILITY,

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

- COLORABILITY,
- STABLE SET,

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

- COLORABILITY,
- STABLE SET,
- CLIQUE.

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

- COLORABILITY,
- STABLE SET,
- CLIQUE.

These results are due to Grötschel-Lovász-Schrijver (1984) and are not combinatorial.

Algorithmic Aspects of Perfect Graphs

The following problems are solvable in polynomial time for perfect graphs:

- COLORABILITY,
- STABLE SET,
- CLIQUE.

These results are due to Grötschel-Lovász-Schrijver (1984) and are not combinatorial.

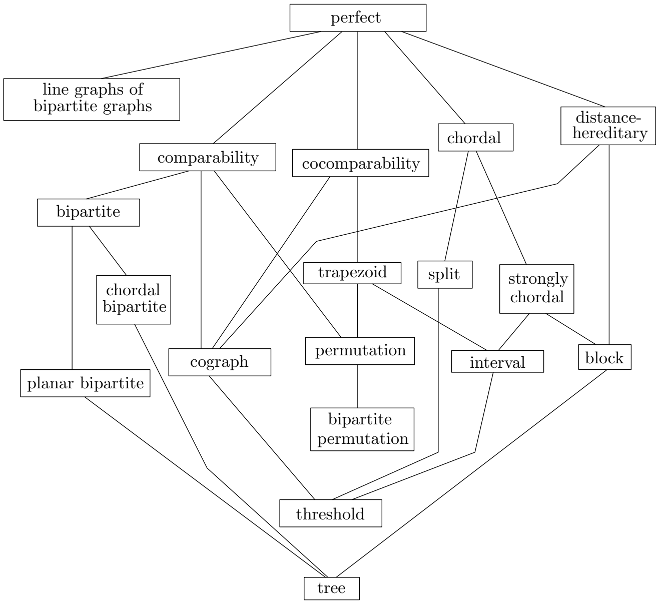
Existence of combinatorial algorithms for the COLORABILITY, STABLE SET and CLIQUE problems on perfect graphs is an open problem.

Recognizing Perfect Graphs

Theorem (Chudnovsky, Cornuéjols, Liu, Seymour, Vušković 2005)

There is a polynomial-time algorithm for recognizing Berge graphs.

Hasse Diagram of Some Classes of Perfect Graphs



CHORDAL GRAPHS.

Definition

A graph is chordal if every cycle on at least 4 vertices contains a chord.

chord: an edge connecting two non-consecutive vertices of the cycle.

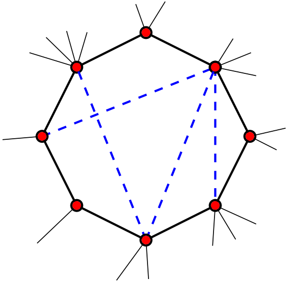
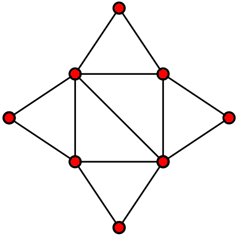


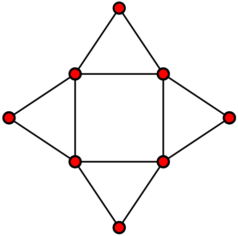
Figure: A cycle with four chords.

Chordal Graphs

Example:



chordal



not chordal

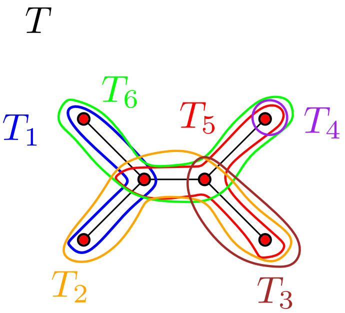
Properties of Chordal Graphs

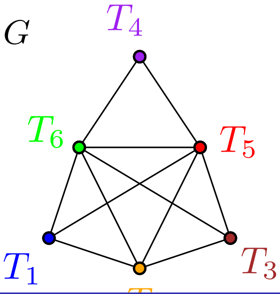
A graph is chordal if and only if it is $\{C_4, C_5, \ldots\}$-free.

Theorem (Gavril, 1974)

Chordal graphs are precisely the vertex-intersection graphs of subtrees in a tree.

Example:



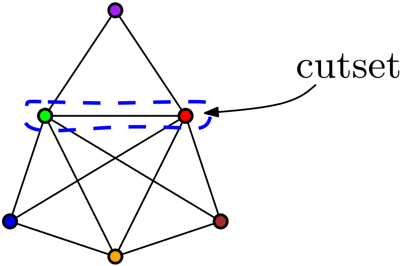


Chordal Graphs: Structural Properties

A cutset: a set of vertices $X \subseteq V$ such that the graph $G - X$ is disconnected.

Theorem (Dirac, 1961)

Every minimal cutset in a chordal graph is a clique.



Chordal Graphs: Algorithmic Aspects

Theorem

Every chordal graph contains a simplicial vertex.

simplicial vertex: a vertex whose neighborhood is a clique

Theorem

Every chordal graph contains a simplicial vertex.

simplicial vertex: a vertex whose neighborhood is a clique

If G is chordal and $v \in V(G)$ then $G - v$ is chordal.

Chordal Graphs: Algorithmic Aspects

Theorem

Every chordal graph contains a simplicial vertex.

simplicial vertex: a vertex whose neighborhood is a clique

If G is chordal and $v \in V(G)$ then $G - v$ is chordal.

With iterative deletion of simplicial vertices, it is easy to develop polynomial-time algorithms for the following problems on chordal graphs:

Chordal Graphs: Algorithmic Aspects

Theorem

Every chordal graph contains a simplicial vertex.

simplicial vertex: a vertex whose neighborhood is a clique

If G is chordal and $v \in V(G)$ then $G - v$ is chordal.

With iterative deletion of simplicial vertices, it is easy to develop polynomial-time algorithms for the following problems on chordal graphs:

- CLIQUE,
- COLORABILITY,
- STABLE SET.

Chordal Graphs: Algorithmic Aspects

Theorem

Every chordal graph contains a simplicial vertex.

simplicial vertex: a vertex whose neighborhood is a clique

If G is chordal and $v \in V(G)$ then $G - v$ is chordal.

With iterative deletion of simplicial vertices, it is easy to develop polynomial-time algorithms for the following problems on chordal graphs:

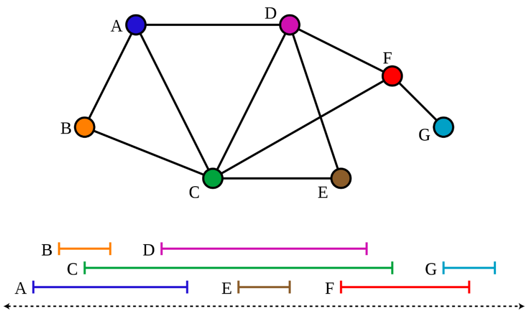
- CLIQUE,
- COLORABILITY,
- STABLE SET.

On the other hand, the DOMINATING SET problem is NP-complete on chordal graphs.

INTERVAL GRAPHS.

Definition

A graph is an interval graph if its vertices can be put into one-to-one correspondence with a set of intervals on the real line such that two vertices are connected by an edge if and only if their corresponding intervals have nonempty intersection.



Source: http://en.wikipedia.org/wiki/Interval_graph

Algorithmic Aspects

Theorem (Booth and Lueker 1976)

Interval graphs can be recognized in linear time.

Algorithmic Aspects

Theorem (Booth and Lueker 1976)

Interval graphs can be recognized in linear time.

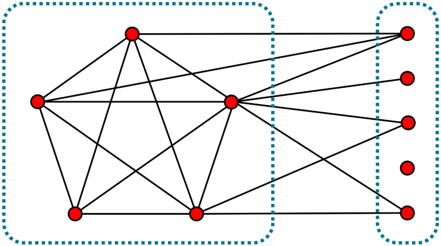
Other algorithmic problems on interval graphs:

- COLORABILITY: In P.
- CLIQUE: In P.
- STABLE SET: In P.
- DOMINATING SET: In P.

SPLIT GRAPHS.

Definition

A graph is split if there exists a partition of its vertex set into a clique and a stable set.

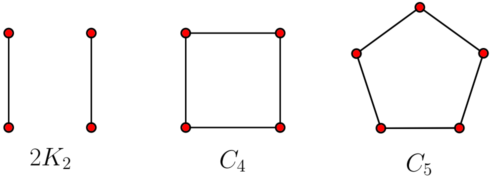


Source: http://en.wikipedia.org/wiki/Split_graph

Forbidden Induced Subgraphs

Theorem (Földes and Hammer, 1977)

A graph is split if and only if it is $\{2K_2, C_4, C_5\}$-free.



Other Properties

Theorem

Split graphs are precisely the vertex-intersection graphs of subtrees of a star.

Other Properties

Theorem

Split graphs are precisely the vertex-intersection graphs of subtrees of a star.

Theorem

Let $d_1 \geq d_2 \geq \ldots \geq d_n$ be the degree sequence of a graph G. Also, let $m = \max\{i \ : \ d_i \geq i - 1\}$. Then, G is a split graph if and only if $\sum_{i=1}^{m} d_i = m(m-1) + \sum_{i=m+1}^{n} d_i$.

Algorithmic Aspects

Split graphs can be recognized in linear time.

Algorithmic Aspects

Split graphs can be recognized in linear time.

Other algorithmic problems on split graphs:

- COLORABILITY: In P.
- CLIQUE: In P.
- STABLE SET: In P.

Algorithmic Aspects

Split graphs can be recognized in linear time.

Other algorithmic problems on split graphs:

- COLORABILITY: In P.
- CLIQUE: In P.
- STABLE SET: In P.
- DOMINATING SET: NP-complete.

Threshold Graphs.

Definition

A graph $G = (V, E)$ is threshold if

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that for every vertex set $S \subseteq V$,

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that for every vertex set $S \subseteq V$,

$$S \text{ is stable} \quad \text{if and only if} \quad w(S) := \sum_{v \in S} w(v) \leq t\,.$$

Definition

Definition

A graph $G = (V, E)$ is threshold if

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and

Definition

A graph $G = (V, E)$ is **threshold** if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that

Definition

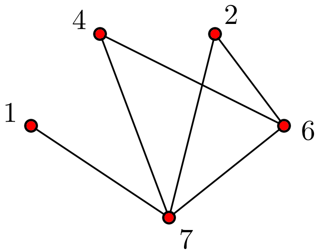
Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that for every vertex set $S \subseteq V$,

Definition

A graph $G = (V, E)$ is threshold if there exist positive real vertex weights $w(v)$ for all $v \in V$ and a threshold $t \in \mathbb{R}$ such that for every vertex set $S \subseteq V$,

$$S \text{ is stable} \quad \text{if and only if} \quad w(S) := \sum_{v \in S} w(v) \leq t\,.$$

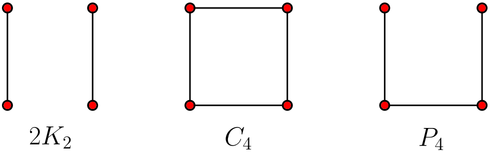


$t = 7$

Forbidden Induced Subgraphs

Theorem (Chvátal, Hammer 1977)

A graph is threshold if and only if it is $\{2K_2, C_4, P_4\}$*-free.*



Further Characterizations

Theorem

A graph G is threshold if and only if can be constructed from the one-vertex graph by repeated applications of the following two operations:

- *Addition of a single isolated vertex to the graph.*
- *Addition of a single dominating vertex to the graph.*

Further Characterizations

Theorem

A graph G is threshold if and only if can be constructed from the one-vertex graph by repeated applications of the following two operations:

- *Addition of a single isolated vertex to the graph.*
- *Addition of a single dominating vertex to the graph.*

Threshold graphs can be recognized in linear time.

Further Characterizations

Theorem

A graph G is threshold if and only if can be constructed from the one-vertex graph by repeated applications of the following two operations:

- *Addition of a single isolated vertex to the graph.*
- *Addition of a single dominating vertex to the graph.*

Threshold graphs can be recognized in linear time.

Other algorithmic problems on threshold graphs:

- COLORABILITY: In P.
- CLIQUE: In P.
- STABLE SET: In P.
- DOMINATING SET: In P.

Numerically Defined Graph Classes.

A General Framework

A General Framework

Let $\mathcal{P}$ denote a **property** meaningful for vertex or edge subsets of a graph.

A General Framework

Let $\mathcal{P}$ denote a **property** meaningful for vertex or edge subsets of a graph.
For example, $\mathcal{P}$ could be any of the following:

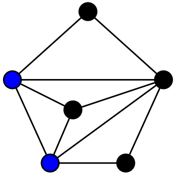
- a matching,
- a clique,
- a stable set,
- a dominating set,
- a total dominating set,
- etc.

Total Dominating Sets

total dominating set: a set of vertices such that every vertex has a neighbor in it

Total Dominating Sets

total dominating set: a set of vertices such that every vertex has a neighbor in it



A General Framework

Problem

Given a graph G, does G admit
positive integer weights $\boldsymbol{w}$ on its vertices (or edges) and a set T such that
$\mathcal{P}(S)$ holds if and only if $w(S) \in T$?

A General Framework

Problem

Given a graph G, does G admit
positive integer weights $\mathbf{w}$ *on its vertices (or edges) and a set* T *such that*
$\mathcal{P}(S)$ *holds if and only if* $w(S) \in T$*?*

Several graph classes can be defined with a suitable choice of property $\mathcal{P}$ and restriction on the set T.

Example:
$\mathcal{P}$ = a stable set; T $=$ an interval unbounded from below:

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

A General Framework – Examples

Example:

$\mathcal{P}$ = a stable set; T $=$ an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T $=$ a single number:

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T = a single number:
equistable graphs (Payan 1980)

A General Framework – Examples

Example:

$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T = a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; T = an interval unbounded from above:

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; $T =$ an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; $T =$ a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; $T =$ an interval unbounded from above:
domishold graphs (Benzaken-Hammer 1978)

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; T $=$ an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T $=$ a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; T $=$ an interval unbounded from above:
domishold graphs (Benzaken-Hammer 1978)

$\mathcal{P}$ = a total dominating set; T $=$ an interval unbounded from above:

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T = a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; T = an interval unbounded from above:
domishold graphs (Benzaken-Hammer 1978)

$\mathcal{P}$ = a total dominating set; T = an interval unbounded from above:
total domishold graphs (Chiarelli-M. 2013)

A General Framework – Examples

Example:
$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T = a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; T = an interval unbounded from above:
domishold graphs (Benzaken-Hammer 1978)

$\mathcal{P}$ = a total dominating set; T = an interval unbounded from above:
total domishold graphs (Chiarelli-M. 2013)

$\mathcal{P}$ = a perfect matching; T = a single number:

A General Framework – Examples

Example:

$\mathcal{P}$ = a stable set; T = an interval unbounded from below:
threshold graphs (Chvátal-Hammer 1977)

$\mathcal{P}$ = a maximal stable set; T = a single number:
equistable graphs (Payan 1980)

$\mathcal{P}$ = a dominating set; T = an interval unbounded from above:
domishold graphs (Benzaken-Hammer 1978)

$\mathcal{P}$ = a total dominating set; T = an interval unbounded from above:
total domishold graphs (Chiarelli-M. 2013)

$\mathcal{P}$ = a perfect matching; T = a single number:
all graphs

EQUISTABLE GRAPHS.

Equistable Graphs

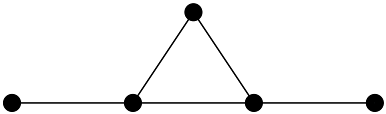
Definition

A graph $G = (V, E)$ is **equistable** if there exists a weight function $w : V \to \mathbb{N}$ and a positive integer $t \in \mathbb{N}$ such that for every $S \subseteq V$:

$$S \text{ is a maximal stable set in } G \quad \Leftrightarrow \quad w(S) = t\,.$$

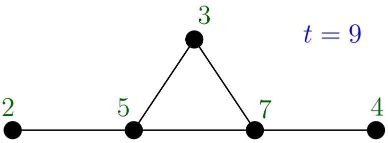
Equistable graphs: example

The following graph is equistable:



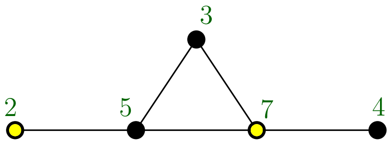
Equistable graphs: example

The following graph is equistable:



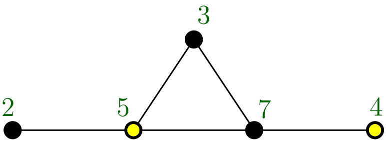
Equistable graphs: example

The following graph is equistable:



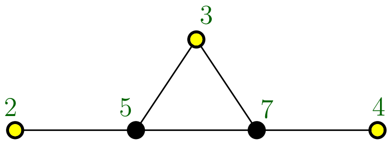
Equistable graphs: example

The following graph is equistable:



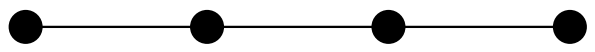
Equistable graphs: example

The following graph is equistable:



Equistable graphs: example

The following graph is not equistable:



Equistable graphs: example

The following graph is not equistable:

If

$$\begin{aligned} w_1 + w_3 &= t \\ w_2 + w_4 &= t \\ w_1 + w_4 &= t \end{aligned}$$

Equistable graphs: example

The following graph is not equistable:

If

$$\begin{aligned} w_1 + w_3 &= t \\ w_2 + w_4 &= t \\ w_1 + w_4 &= t \end{aligned}$$

then

$$w_2 + w_3 = t\,.$$

Complexity of algorithmic problems on equistable graphs:

Algorithmic Aspects of Equistable Graphs

Complexity of algorithmic problems on equistable graphs:

- RECOGNITION: OPEN.

Algorithmic Aspects of Equistable Graphs

Complexity of algorithmic problems on equistable graphs:

- RECOGNITION: OPEN.
  - In P if weights are in $\{1, \ldots, k\}$ for a fixed k (Levit-M.-Tankus 2012)

Complexity of algorithmic problems on equistable graphs:

- RECOGNITION: OPEN.
  - In P if weights are in $\{1, \ldots, k\}$ for a fixed k (Levit-M.-Tankus 2012)
- COLORABILITY: NP-complete.
- CLIQUE: NP-complete.
- STABLE SET: NP-complete.
- DOMINATING SET: NP-complete.

DOMISHOLD GRAPHS.

Domishold Graphs

Definition

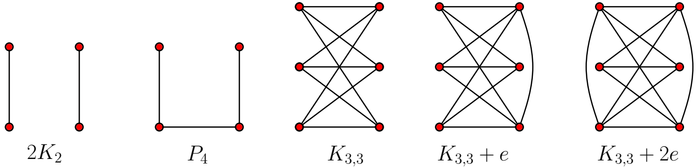
A graph $G = (V, E)$ is **domishold** if there exists a weight function $w : V \to \mathbb{N}$ and a positive integer $t \in \mathbb{N}$ such that for every $S \subseteq V$:

$$S \text{ is a dominating set in } G \quad \Leftrightarrow \quad w(S) \geq t\,.$$

Characterizations of Domishold Graphs

Theorem (Benzaken and Hammer 1978)

A graph G is domishold if and only if G is $\{2K_2, P_4, K_{3,3}, K_{3,3}+e, K_{3,3}+2e\}$-free.



Algorithmic Aspects of Domishold Graphs

Complexity of algorithmic problems on domishold graphs:

- RECOGNITION: In P.
- COLORABILITY: In P.
- CLIQUE: In P.
- STABLE SET: In P.
- DOMINATING SET: In P.

TOTAL DOMISHOLD GRAPHS.

Total Domishold Graphs

Definition

A graph $G = (V, E)$ is **total domishold** if there exists a weight function $w : V \to \mathbb{N}$ and a positive integer $t \in \mathbb{N}$ such that for every $S \subseteq V$:

$$S \text{ is a total dominating set in } G \quad \Leftrightarrow \quad w(S) \geq t\,.$$

Algorithmic Aspects of Total Domishold Graphs

Complexity of algorithmic problems on total domishold graphs (Chiarelli-M. 2013):

- RECOGNITION: In P.

Algorithmic Aspects of Total Domishold Graphs

Complexity of algorithmic problems on total domishold graphs (Chiarelli-M. 2013):

- RECOGNITION: In P.
- COLORABILITY: NP-complete.
- CLIQUE: NP-complete.
- STABLE SET: NP-complete.

Algorithmic Aspects of Total Domishold Graphs

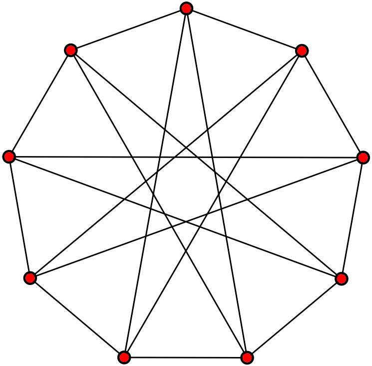
Complexity of algorithmic problems on total domishold graphs (Chiarelli-M. 2013):

- RECOGNITION: In P.
- COLORABILITY: NP-complete.
- CLIQUE: NP-complete.
- STABLE SET: NP-complete.
- DOMINATING SET: OPEN.

Circulant Graphs.

Circulants

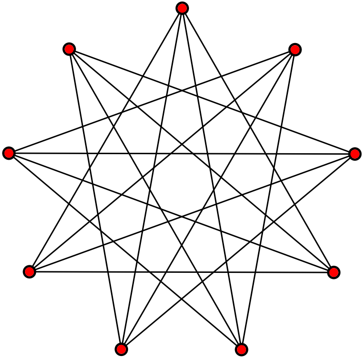
A **circulant** is a Cayley graph over a cyclic group.



$C_9(\{1,4,5,8\})$

Circulants

A **circulant** is a Cayley graph over a cyclic group.



$C_9(\{3, 4, 5, 6\})$

Proposition

A circulant $G = C_n(D)$ *is*
connected
if and only if
$\gcd(D \cup \{n\}) = 1$.

Some Graph-theoretic Properties of Circulants

Proposition

A circulant $G = C_n(D)$ *is*
connected
if and only if
$\gcd(D \cup \{n\}) = 1$.

Proposition

A connected circulant $G = C_n(D)$ *with at least two vertices is*
bipartite
if and only if
n *is even, while every* $d \in D$ *is odd.*

Algorithmic Aspects

Theorem (Evdokimov and Ponomarenko 2004)

Circulant graphs can be recognized in polynomial time.

Algorithmic Aspects

Theorem (Evdokimov and Ponomarenko 2004)

Circulant graphs can be recognized in polynomial time.

Other algorithmic problems on circulant graphs:

- COLORABILITY: NP-complete. Codenotti–Gerace–Vigna 1998
- CLIQUE: NP-complete. Codenotti–Gerace–Vigna 1998
- STABLE SET: NP-complete. Codenotti–Gerace–Vigna 1998

Algorithmic Aspects

Theorem (Evdokimov and Ponomarenko 2004)

Circulant graphs can be recognized in polynomial time.

Other algorithmic problems on circulant graphs:

- COLORABILITY: NP-complete. Codenotti–Gerace–Vigna 1998
- CLIQUE: NP-complete. Codenotti–Gerace–Vigna 1998
- STABLE SET: NP-complete. Codenotti–Gerace–Vigna 1998
- DOMINATING SET: OPEN.

CONCLUSION.

Conclusion

Graph classes form a rich field of research, with practical and theoretical applications.

Conclusion

Graph classes form a rich field of research, with practical and theoretical applications.

Methods from different branches of mathematics and computer science apply to the study of graph classes:

1. algebraic and Boolean methods,
2. combinatorial methods,
3. mathematical programming (linear programming, polyhedral combinatorics, semidefinite programming),
4. algorithm design and computational complexity analysis,
5. etc.

Questions?

Thank you!