The Wonderful World of Chordal Graphs

Martin Charles Golumbic

University of Haifa, Israel

The 32nd European Conference on Combinatorial Optimization,

ECCO XXXII, Malta, 2019

Preamble

In **1970**, Claude Berge published the original French version his fundamental and perhaps most important book, *Graphes et Hypergraphes*.

To many graph theorists, its chapters were ***saplings*** ready to be cultivated into the vast ***forest*** that we know today.

As we enter this 50th (Jubilee) anniversary year,

we celebrate *Le Bois de Berge* with its mathematical

palms, pines and poplars,

firs, fruit and ficuses,

oaks, maples and cacti.

One of those sapling chapters was on *Perfect Graphs*.

Berge challenged us with his Perfect Graph Conjecture, and

surveyed its core subclasses:

***comparability* graphs, *interval* graphs, and *triangulated* (chordal) graphs.**

—citing Fulkerson, Gallai, Ghouila-Houri, Gilmore, Hoffman, Hojós, Lovász.

By the time the English version appeared in 1973,

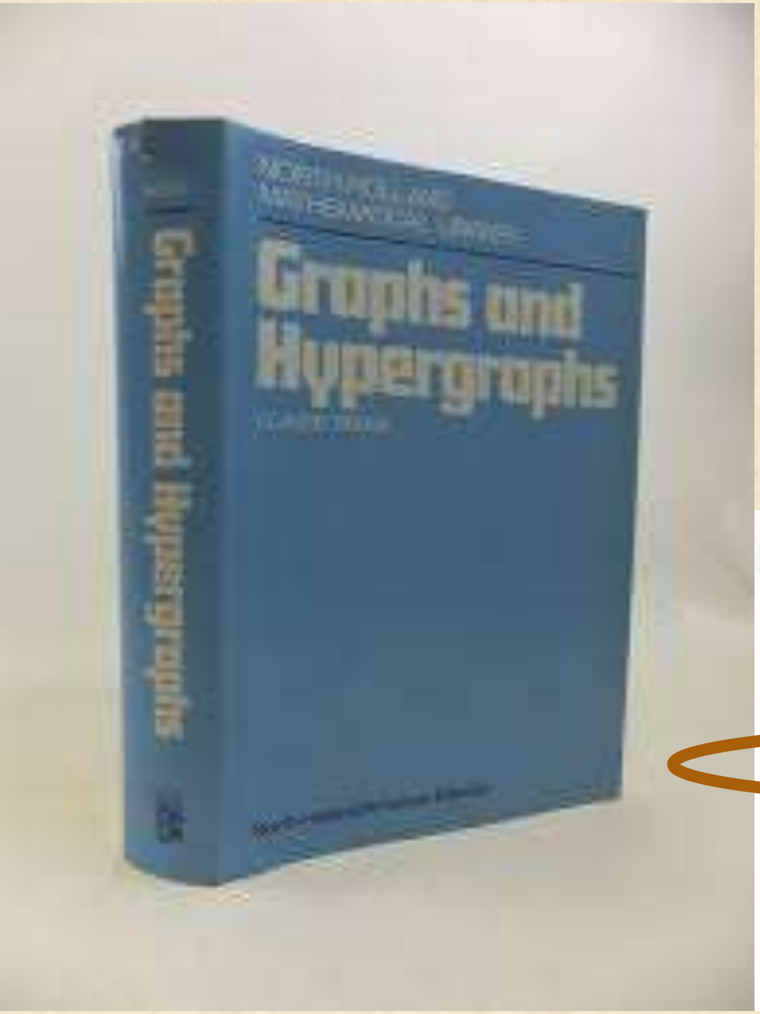
more sprouts *could have been added* to the blossoming family

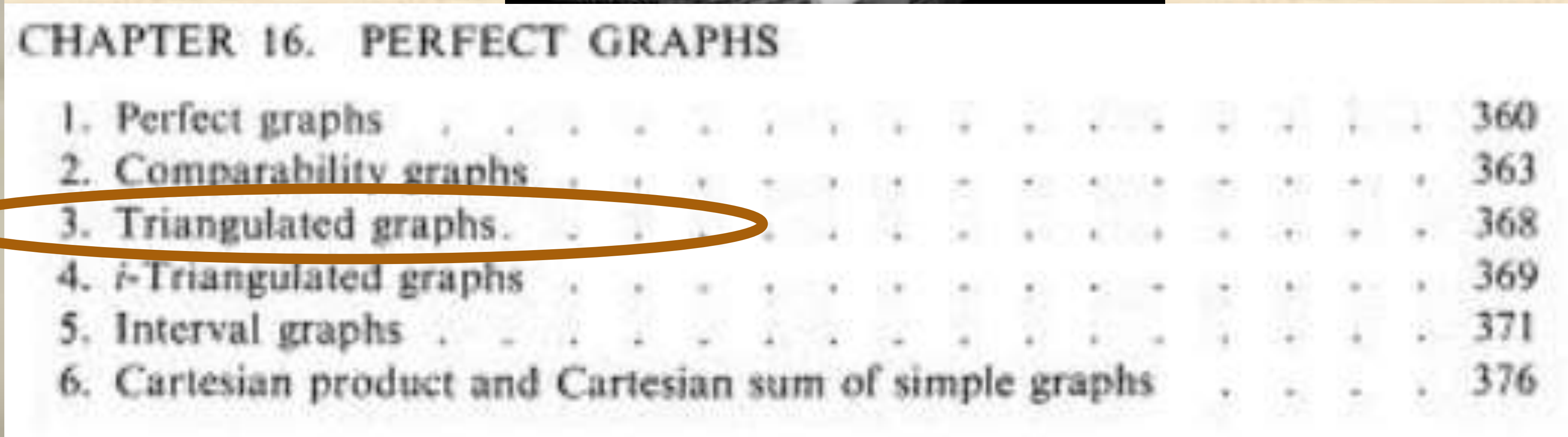
—Benzer, Dirac, Fishburn, Gavril, Roberts, Rose, Trotter.

However, these and others would wait until 1980, when my own book

Algorithmic Graph Theory and Perfect Graphs first appeared.

—a direct outgrowth of Berge’s inspiring chapter.





CHAPTER 16. PERFECT GRAPHS

1. Perfect graphs 360
2. Comparability graphs 363
3. Triangulated graphs 368
4. *i*-Triangulated graphs 369
5. Interval graphs 371
6. Cartesian product and Cartesian sum of simple graphs 376

Triangulated graphs – also known as

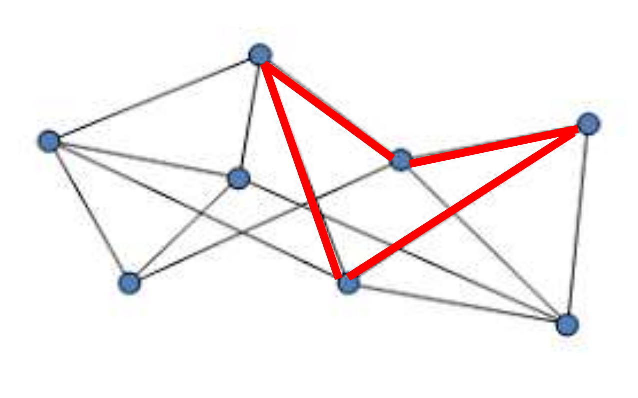
rigid circuit graphs (Dirac), *acyclic graphs* (Lekkerkerker and Boland)

But it was Fanica Gavril who coined the term *chordal graphs*.

A graph G is a ***chordal graph***, if

every cycle in G of length greater than or equal to 4 has a *chord*,

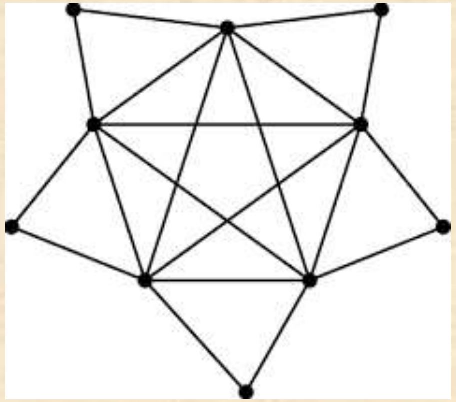
that is, an edge connecting two vertices that are not consecutive on the cycle.



NOT a chordal graph

It has many copies of C_4

This IS a chordal graph



Fanica Gavril: "I knew that these graphs occurred before as triangulated graphs, but the

term *triangulated* was also used for maximal planar graphs, implying the statement, 'some planar triangulated graphs are not triangulated graphs' (like the complete wheels). **So, I decided to call them *chordal graphs*** since every simple cycle with more than three vertices has a chord."

—personal communication with the author.

The Wonderful World of Chordal Graphs

Chordal graphs—the second most interesting and important family of graphs

after trees and before planar graphs.

Their fame is due to their

- beautiful and classical characterizations,
- diverse mathematical properties, and
- numerous applications:

combinatorial optimization, constraint programming, relational databases, perfect phylogeny, Bayesian networks for probabilistic reasoning, exploiting sparsity in large positive semidefinite matrices and recently, register allocation.

Chordal graphs -- one of the earliest families whose *structural properties* fundamentally help in *solving hard problems efficiently*, including the coloring, clique, stable set, and clique cover problems.

Chordal graphs -- lead to researchers looking carefully at the *tree structure* of graphs and hypergraphs, and developing the notion of *treewidth* and partial *k*-trees, which have many algorithmic consequences.

Lexicographic breadth first search (LexBFS) and maximum cardinality search (MCS) have their origins in *recognizing* chordal graphs.

A large *hierarchy of graph classes* have been built around chordal graphs, each with its own characterizing properties and applications.

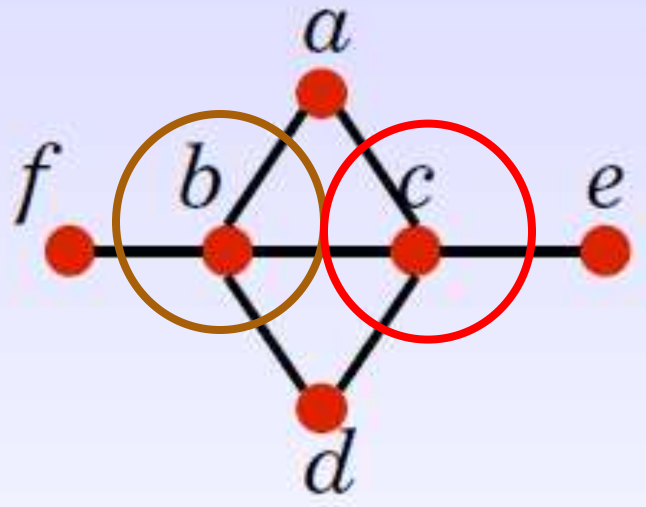
Early Results (1960s)

Vertex Separator Theorem (Dirac)
Perfect Elimination (Fulkerson & Gross)

Definition: A subset of vertices S is a ***minimal vertex separator*** of G if there exist nonadjacent $a, b \in V(G)$ such that a and b are not connected in $G - S$, and S is minimal for inclusion with this property.

We call S a *minimal* $\{a,b\}$*-separator*.

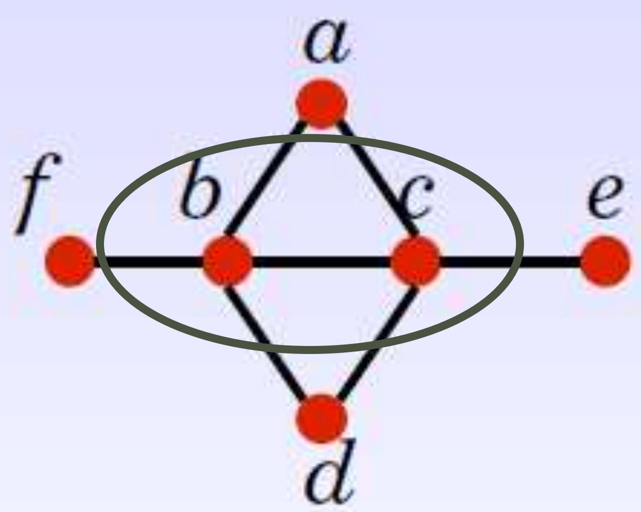
Example of Minimal Vertex Separators



3 minimal separators $\{b\}$ for f and a, $\{c\}$ for a and e and $\{b, c\}$ for a and d.

Note: A minimal vertex separator MAY CONTAIN another minimal vertex separator.

Example of Minimal Vertex Separators



3 minimal separators $\{b\}$ for f and a, $\{c\}$ for a and e and $\{b, c\}$ for a and d.

Note: A minimal vertex separator MAY CONTAIN another minimal vertex separator.

Dirac [1961]

Theorem 1: A graph is chordal iff
every minimal vertex separator is a complete graph.

Definition: A vertex v is ***simplicial*** if its neighborhood is complete.

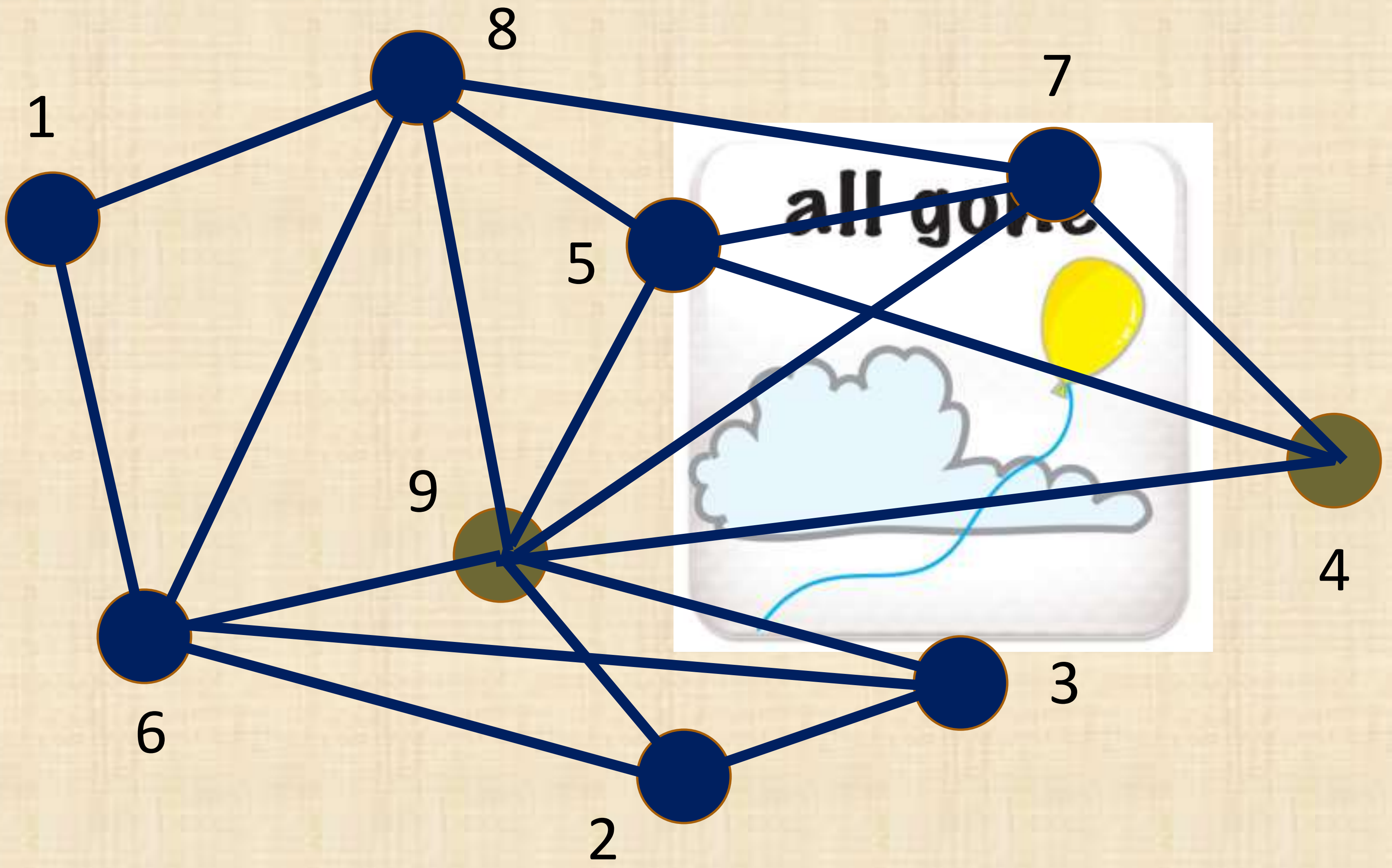
Theorem 2: Every chordal graph has two non-adjacent simplicial vertices -- unless it is complete, where every vertex is simplicial.

Fulkerson & Gross [1965]

Theorem 3: A graph is chordal iff
it has a ***perfect elimination ordering*** (PEO): $[v_1, v_2, \dots, v_n]$
where v_i is simplicial in the induced subgraph $G_i = G[v_i, \dots, v_n]$.

Moreover, every simplicial vertex of a chordal graph G can be the first vertex of a perfect elimination ordering.

Example: Finding a PEO – Recognizing Chordal Graphs



Important Remark: A chordal graph has at most n maximal cliques.

Proof: Take a perfect elimination ordering (PEO): $[v_1, v_2, \dots, v_n]$.

For each v_i , its forward neighborhood

$C_i = N[v_i] \cap \{v_i, \dots, v_n\}$ is a clique.

For any maximal clique C of G,

let v_j be the first vertex of C in the PEO.

Then $C = C_j$, so there are at most n maximal cliques of G.

Algorithmic and Structural Aspects (1970s)

Computational Complexity

A naïve implementation to find a PEO and recognizing chordal graphs can be done in $O(n^4)$. This can easily be improved to $O(n^3)$. **However,**

Theorem 4: Rose, Tarjan and Lueker [1976]

LexBFS (Lexicographic Breadth-first Search) or
MCS (Maximum Cardinality Search)

will find a PEO of a chordal graph in $O(n + m)$ time.

Moreover, applied to any graph, the ordering produced can be tested as a PEO in $O(n + m)$ time,
thus also *recognizing chordal graphs*.

Corollary: Given a PEO, ***MINCOLOR, MAXCLIQUE, MAXSTABLE*** and several other optimizations can be done in $O(n + m)$ time on chordal graphs.

LexBFS Algorithm:

Breadth First Search giving a ***Lexicographic***-priority for choosing the next vertex to explore

Input: Graph $G = (V, E)$ and a distinguished (starting) vertex x of G

Output: An ordering σ of the vertices of G

for each vertex y in $V \setminus \{x\}$ **do** label(y) $\leftarrow$ null;

for $i \leftarrow |V|$ **downto** 1 **do**

- pick an unnumbered vertex y with *lexicographically largest label*;
- $\sigma^{-1}(y) \leftarrow |V| + 1 - i$; {*assign next number* to y} ;
- for each unnumbered vertex z in $N(y)$ **do** *append* i to label(z).

LexBFS Algorithm:

Breadth First Search giving a ***Lexicographic***-priority for choosing the next vertex to explore

Theorem. If G is chordal, then σ the *reverse* of any LexBFS ordering is a perfect elimination ordering.

An alternate view of LexBFS:

BFS uses a ‘*stack*’

LexBFS uses ‘*partition refinement*’ on that stack to obtain a linear-time implementation.

Algorithmic and Structural Aspects (1970s)

Subtrees of a Tree

Theorem 5: Buneman [1974], Gavril [1974], Walters [1972, 1978]

A graph G is chordal iff it is the intersection graph of subtrees of a tree T.

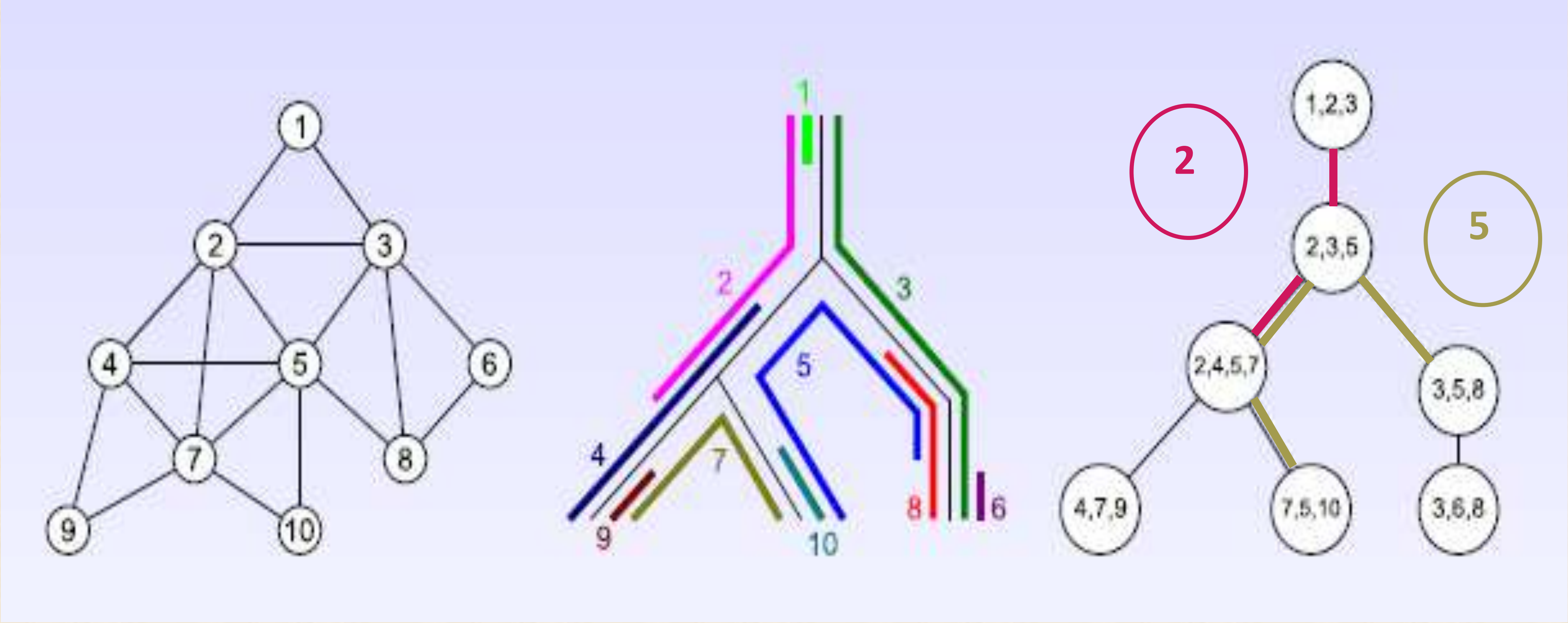
$v \in V(G) \rightarrow T_v$ such that $uw \in E(G) \Leftrightarrow T_u \cap T_w \neq \emptyset$

Moreover, the tree T can be chosen such that each *node of* T corresponds to a *maximal clique* of G,

such that the subtree T_v associated with vertex $v \in V(G)$ is induced precisely by those maximal cliques in G that contain v.

T is then called a *clique tree representation* of G.

Important Remark: A clique tree for a chordal graph has at most n nodes.

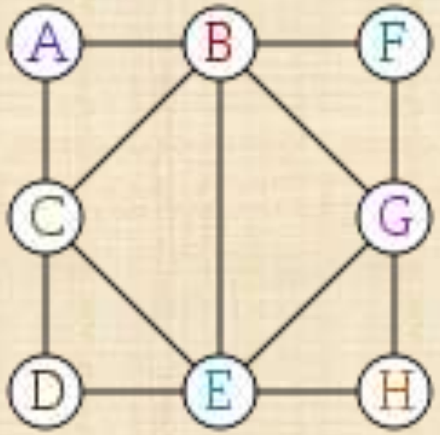


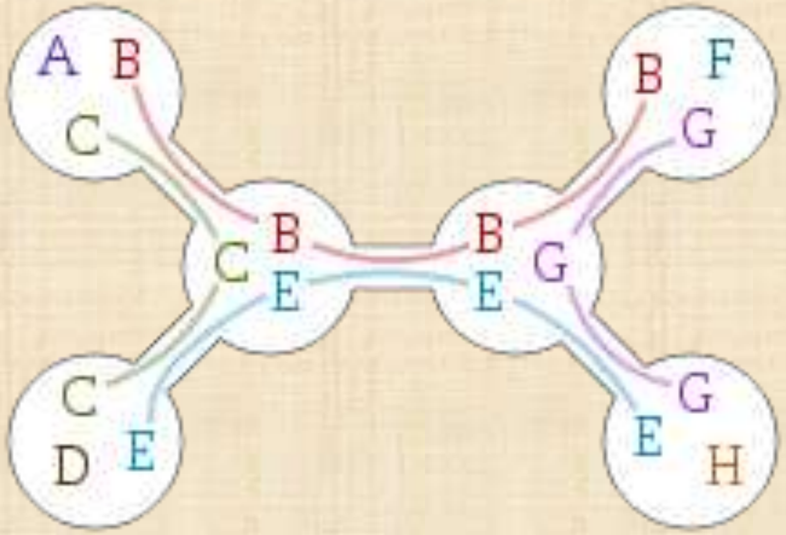
G

Intersection representation for *G*

Clique tree representation for *G*

Clique Tree Representation of a Chordal Graph





How to construct a clique tree

Use your favorite <u>maximum</u>-weight spanning tree algorithm

For a chordal graph *G*, ***the weighted clique intersection graph*** $\mathcal{WC}(G)$

vertex set: The set of maximal cliques of *G*

edge set: Two distinct cliques *K* and *K'* are connected by an edge if and only if their intersection is nonempty

edge weights: $w(K, K') = |K \cap K'|$

Theorem 6: Bernstein and Goodman [1981], Gavril [1987], Shibata [1988]
Every maximum-weight spanning tree of $\mathcal{WC}(G)$ is a clique tree of G.

Thus, your favorite efficient algorithm for MST such as Prim or Kruskal can be used (negating the edge weights) to find a clique tree, given the set of maximal cliques of a chordal graph (easy to find from a PEO).

Chordal graphs could easily fill an entire book!
— theory, applications and algorithmic aspects

Golumbic [1980] had a 22 page chapter,
Triangulated graphs

Blair and Peyton [1993] published a 29 page survey,
An introduction to chordal graphs and clique trees

Vandenberghe and Andersen [2015] published a 92 page survey,
Chordal graphs and semidefinite optimization

Many books today have large sections devoted to
chordal graphs and their offspring.

Applications

- **Solving (sparse) semidefinite matrix problems**

When the zero/non-zero pattern of a (sparse) matrix *M* is the adjacency matrix of a chordal graph *G*, then a PEO for *G* can be used for efficiently solving many problems – like Gaussian elimination preserving sparsity.

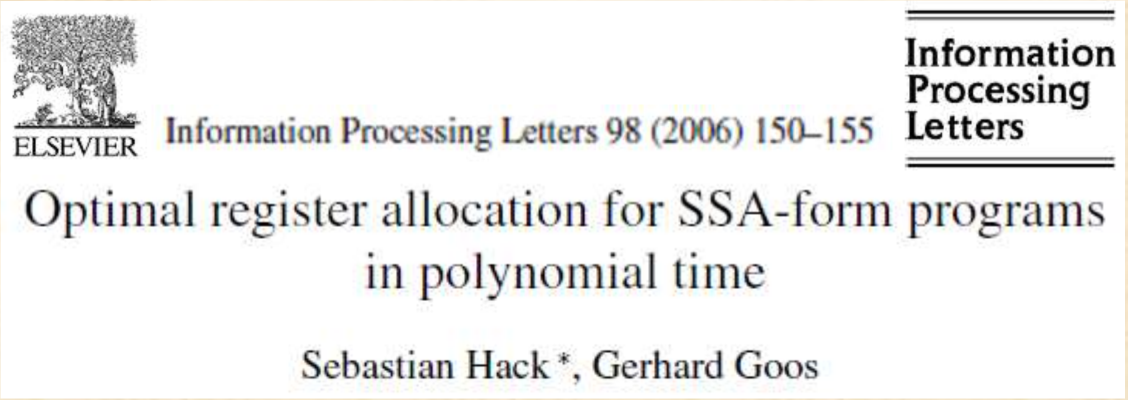
- **Acyclic Database Schemes**

When a Relational Database Scheme R can be designed so its tables form an acyclic hypergraph *H*, (the hypergraph analogue of chordal graphs), then queries can be solved very efficiently.

More Applications

- **Register allocation in compilers**

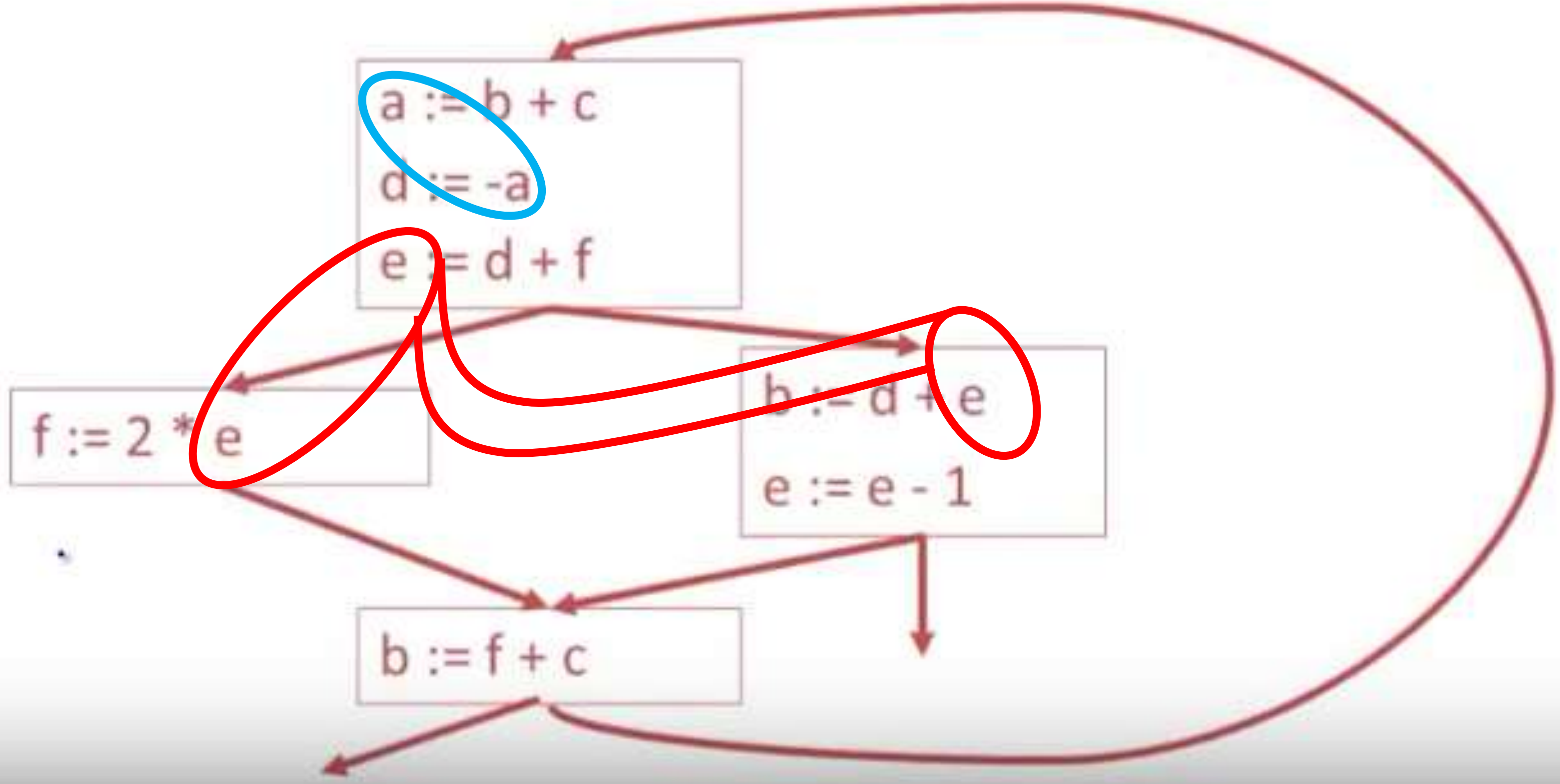
SSA-form programs (Static Single Assignment) are an intermediate representation in a compiler, which requires that each variable is assigned exactly once – *usually by renaming*.


ELSEVIER Information Processing Letters 98 (2006) 150–155

Information Processing Letters

Optimal register allocation for SSA-form programs in polynomial time

Sebastian Hack *, Gerhard Goos



"live area" of a variable:
a variable is *defined*
then *used* …
until its value "*dies*"

Since the 1980's, all register allocation is done by coloring the interference graphs of live areas.

Sabastian Hack:

The interference graph of every SSA program is chordal, independent of its control flow structure.

Essentially, the fact that in SSA, *every variable dominates all its uses* and the fact that *dominance is a tree-like partial order* (all dominators of a node are linearly ordered) yields a ***perfect elimination order***.

Therefore, they can be optimally colored efficiently.

But, the real problem of register allocation is *not just finding a coloring*, but first doing *live-range splitting* to lower the register pressure – maximal number of simultaneously live variables = size of a maxclique.

SSA insures that the chromatic number = size of a maxclique, since the graph is chordal.

Many Newer Papers on Chordal Graphs

Contents lists available at SciVerse ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

ELSEVIER

(2013)

An efficient representation of chordal graphs

Lilian Markenzon[a,*], Christina Fraga Esteves Maciel Waga[b], Paulo Renato da Costa Pereira[c], Clícia Valladares Peixoto Friedmann[d], Abel Rodolfo Garcia Lozano[d]

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

ELSEVIER

(2013)

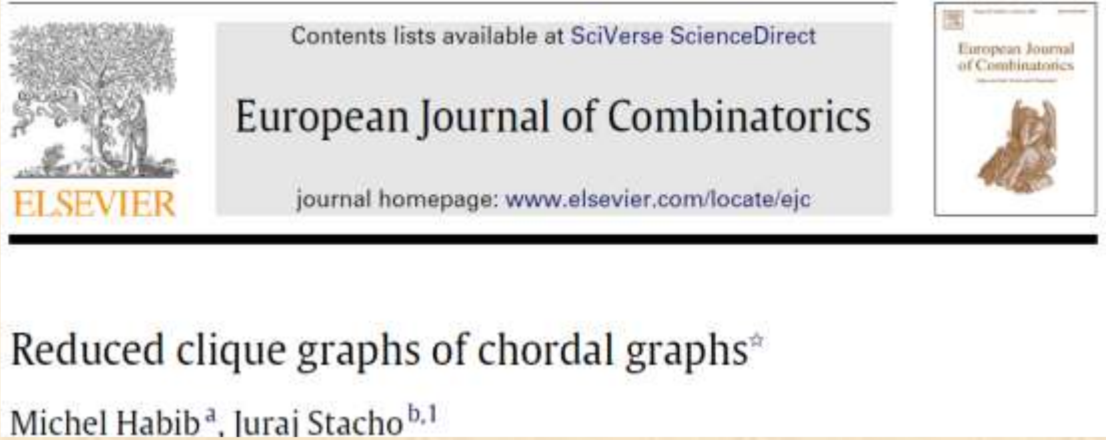
Note

A Dirac-type characterization of k-chordal graphs

CrossMark

R. Krithika[a,*], Rogers Mathew[b], N.S. Narayanaswamy[a], N. Sadagopan[c]

Cliques and Chordal Graphs


Contents lists available at SciVerse ScienceDirect

European Journal of Combinatorics

journal homepage: www.elsevier.com/locate/ejc

ELSEVIER

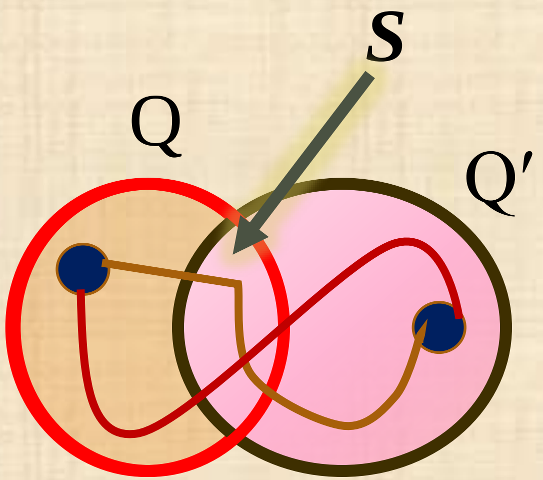
Reduced clique graphs of chordal graphs☆

Michel Habib[a], Juraj Stacho[b,1]

Maximal cliques play a central role
in the study of chordal graphs.

- A chordal graph has a *clique tree representation*
- All *minimal vertex separators* are cliques
- The (weighted) *clique intersection graph* provides much information
- The *reduced clique graph* -- Habib and Stacho, European J. of Combin. 33 (2012) 712–735
  - To characterize the *astroidal sets* in chordal graphs, and
  - To study chordal graphs that admit a tree representation with a small number of leaves, i.e., small *leafage*, a notion introduced by Lin, McKee and West [1998]
  - Determining the leafage of a chordal graph is Polynomial-time -- Habib and Stacho [2009]

The reduced clique graph



Two maximal cliques Q and Q′ of G form a *separating pair* if Q ∩ Q′ is non-empty and every path in G from a vertex of Q \ Q′ to a vertex of Q′ \ Q contains a vertex of Q ∩ Q′ .

Theorem [Habib and Stacho, 2012] A set S is a minimal vertex separator of a chordal graph G if and only if there exist maximal cliques Q and Q′ of G forming a separating pair such that $S = Q \cap Q'$.

This extends Dirac's result.

Definition: The *reduced clique graph* $\mathcal{RC}(G)$ of G is the graph whose vertices are maximal cliques of G with edges {Q, Q′} between separating pairs Q and Q′ .

Theorem [Habib and Stacho, 2012] Let G be a connected chordal graph. The reduced clique graph $\mathcal{RC}(G)$ is the union of all clique-trees of G.

Generating random chordal graphs

Motivation: The need for testing and comparing various optimization algorithms on chordal graphs – including *exact* / *heuristic* / *parameterized* algorithms

Linear time algorithm: Ekim, Shalom and Şeker [2016]

Technique: Build a 'contraction minimal' *clique tree* $\mathcal{T}$ by adding a leaf ℓ *adjacent to some existing node u chosen uniformly at random* at each iteration:

a) a non-empty set S of new subtrees consisting of only ℓ is added to $\mathcal{T}$,

b) a random proper subset of the subtrees containing u is chosen and extended by adding the node ℓ and the edge $u\ell$,

c) the graph G is extended to reflect the changes in $\mathcal{T}$ (set S are simplicial true twins).

Goal: Fair distribution of maximal cliques

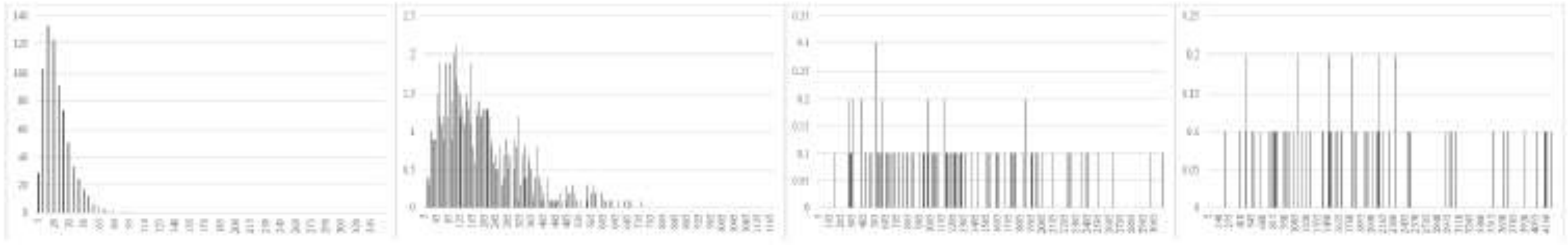
Remark: Generating chordal graphs uniformly at random is subject to further research.

Ekim, Shalom and Şeker [2016]: Experimental results provide **insight into the distribution of chordal graphs generated,** specifically, ***comparing the maximal clique sizes*** of two other methods from the literature.

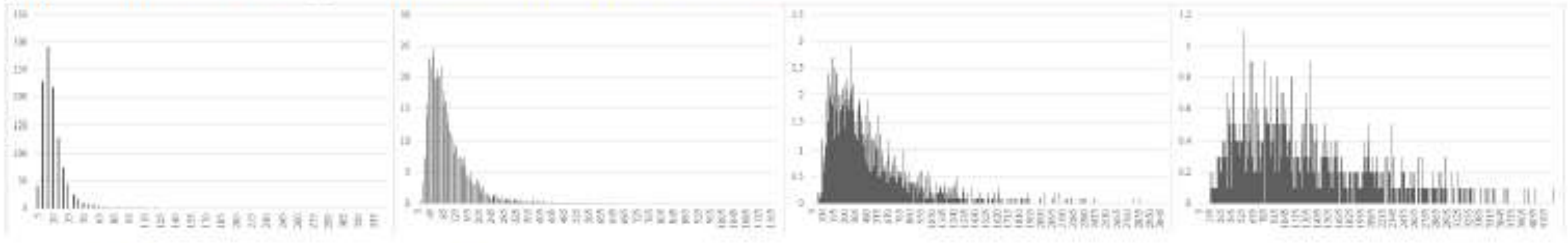
Histograms of

maximal clique sizes

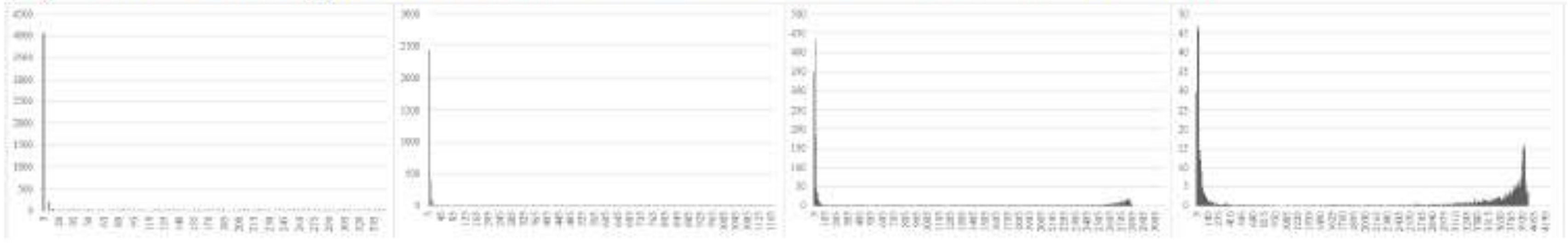
for n = 5000 and average edge densities 0.01, 0.1, 0.5, and 0.8 (from left to right).



(a) Results from algorithm GenerateContractionMinimal



(b) Results from algorithm ChordalGen with GrowingSubtree method



(c) Results from Andreou *et al.*'s method [1]

Another companion paper on the generation of random chordal graphs:

Seker, Heggernes, Ekim and Taskın [2017] {full version on Arxiv}

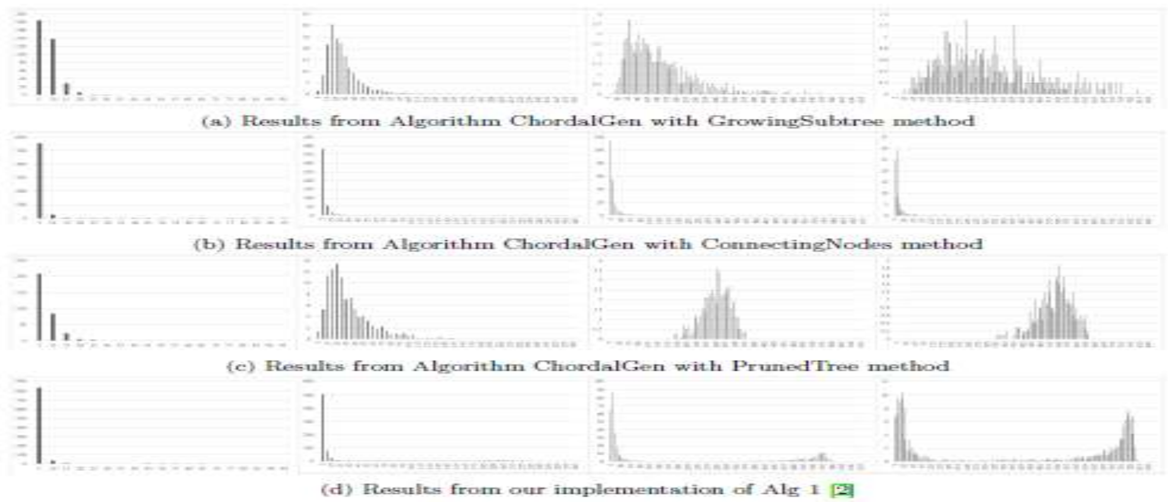


Fig. 1: Histograms of maximal clique sizes for $n = 1000$ and average edge densities 0.01, 0.1, 0.5, and 0.8 (from left to right)

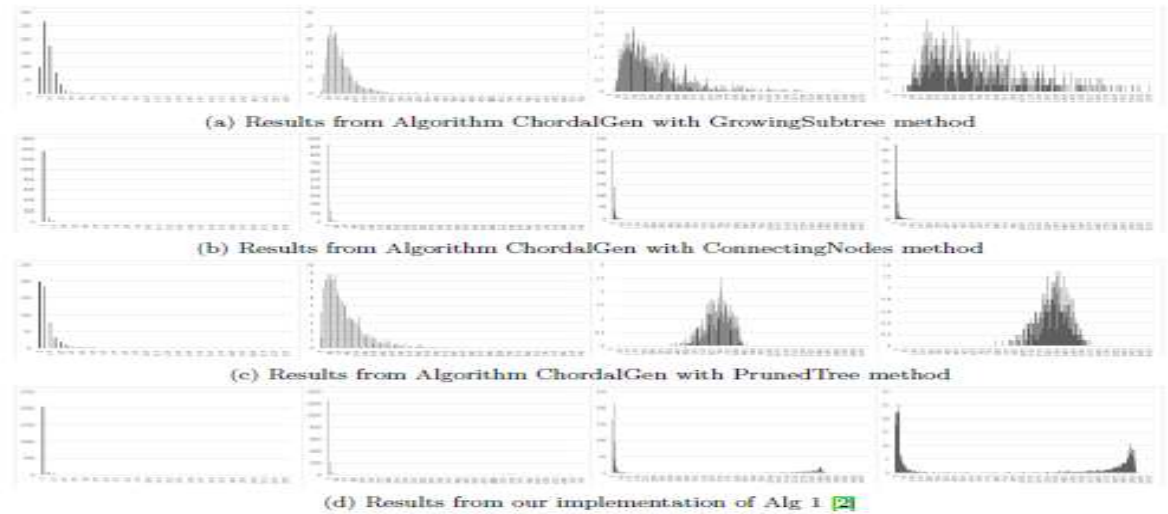


Fig. 2: Histograms of maximal clique sizes for $n = 2500$ and average edge densities 0.01, 0.1, 0.5, and 0.8 (from left to right)

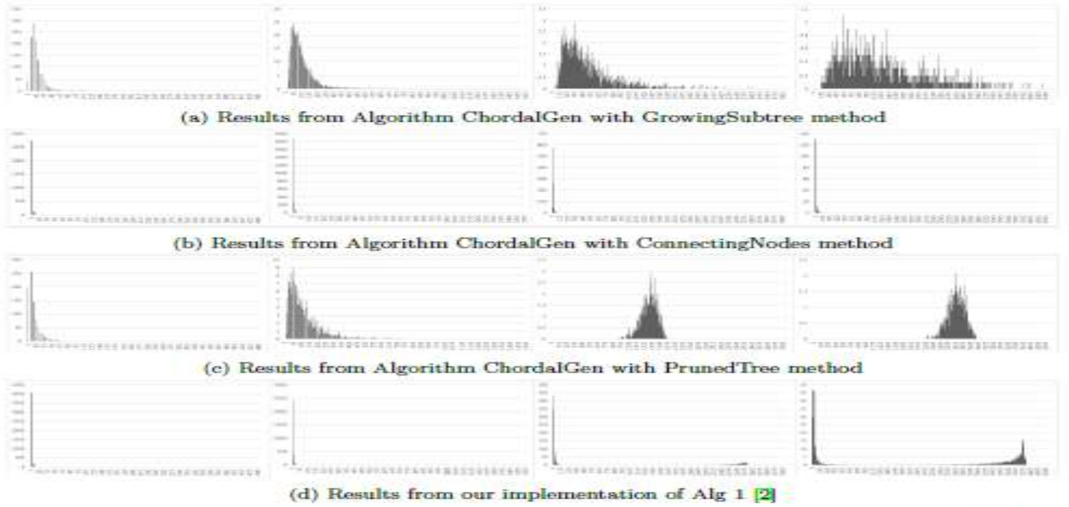


Fig. 3: Histograms of maximal clique sizes for $n = 5000$ and average edge densities 0.01, 0.1, 0.5, and 0.8 (from left to right)

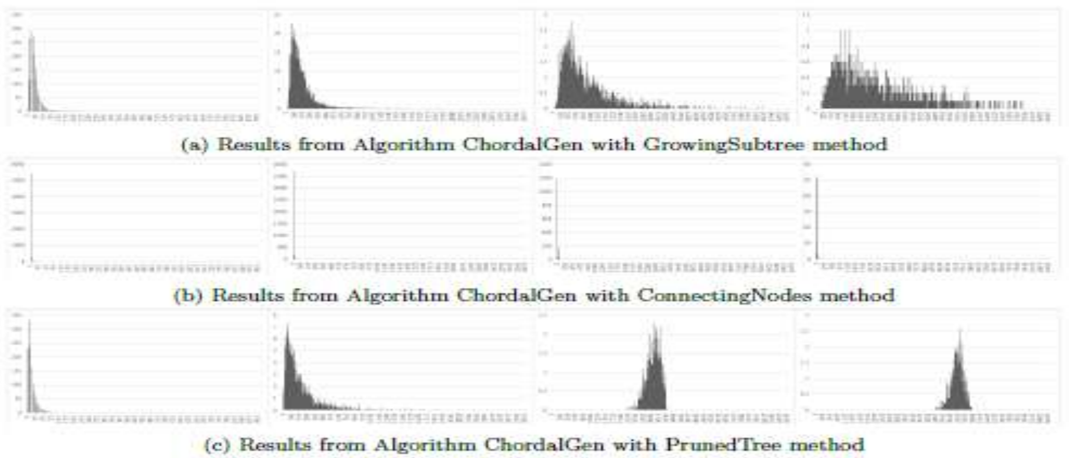
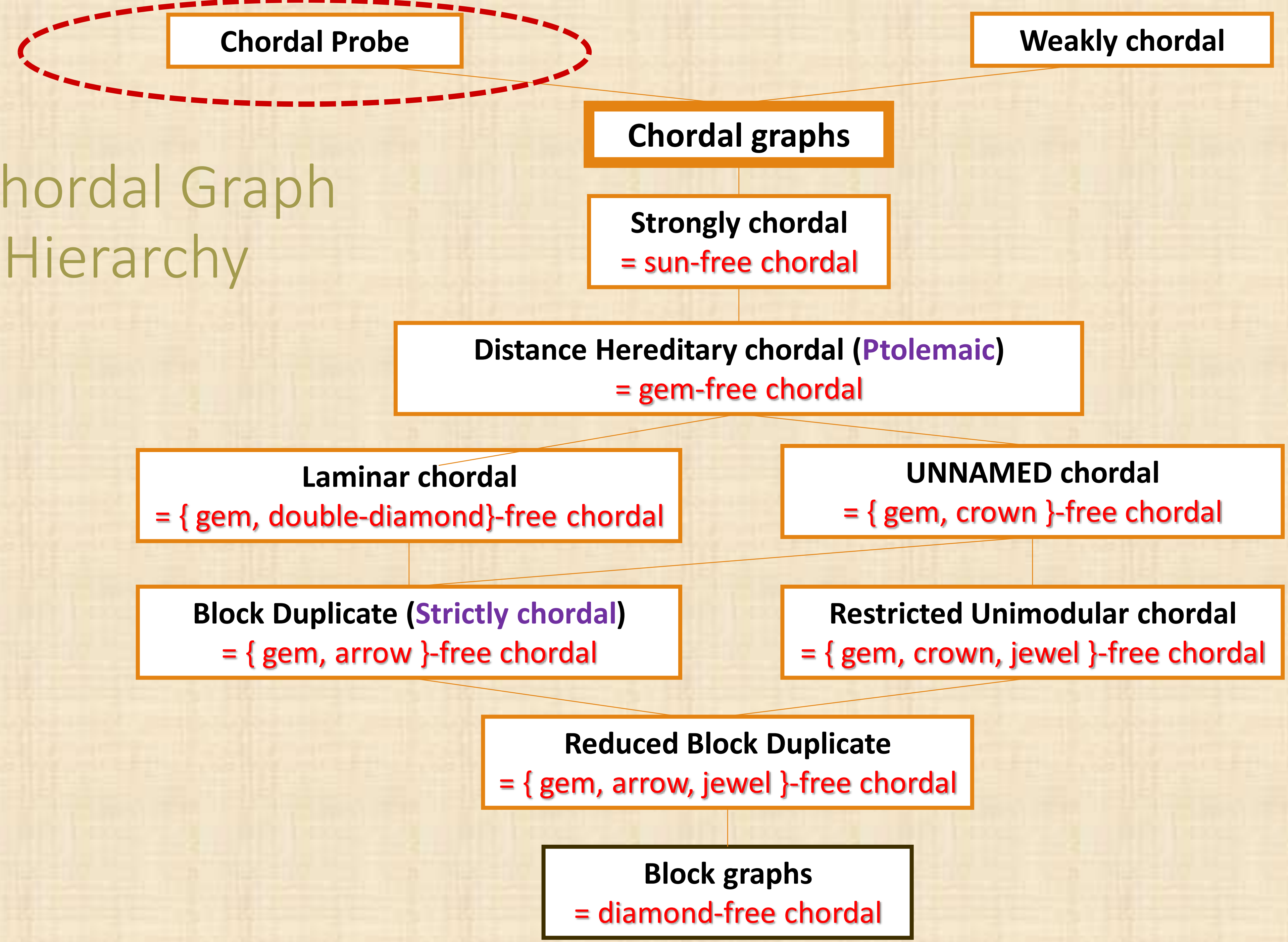


Fig. 4: Histograms of maximal clique sizes for $n = 10000$ and average edge densities 0.01, 0.1, 0.5, and 0.8 (from left to right)

A Chordal Graph Hierarchy



Chordal Probe Graphs – a superclass of chordal graphs

Dealing with Partial Information -- missing data and deducing consistency

A graph G is *chordal probe* if its vertices can be partitioned into two sets P (***probes***) and N (***non-probes***), where N is a stable set, such that G can be extended to a chordal graph by adding edges only between non-probes.

For example,

Every bipartite graph is chordal probe—a chordal completion:

Use the same bipartite partition $V(G) = X \cup Y$ into two stable sets, calling X probes and Y non-probes, and fill in edges to make Y into a clique.

This completion is a split graph and thus a chordal graph.

Remark: The path P_6 is not a chordal probe graph.

Chordal Probe Graphs – another view

The Probe Game

- Take a chordal graph G.
- Choose a subset of vertices N.
- Erase the edges in $N \times N$.
- Give this graph to your daughter.

HER CHALLENGE:

Can she fill in some edges in $N \times N$

to find a chordal graph?

We assume your daughter is very clever!

She might even find a chordal graph different from G.

Remark: The set N is now a stable set. All other original edges are "*mandatory*".

Theorem: The Chordal **Probe** Graph Recognition Problem is polynomial solvable.

Remark: The *Chordal **Sandwich** Graph Game/Problem* is NP-complete,

– erasing edges, declaring these as "optional", then finding a chordal fill-in.

Case 1: **Given a fixed partition of the vertices into probes and non-probes.**

Quasi-perfect elimination -- recognizing partitioned chordal probe graphs

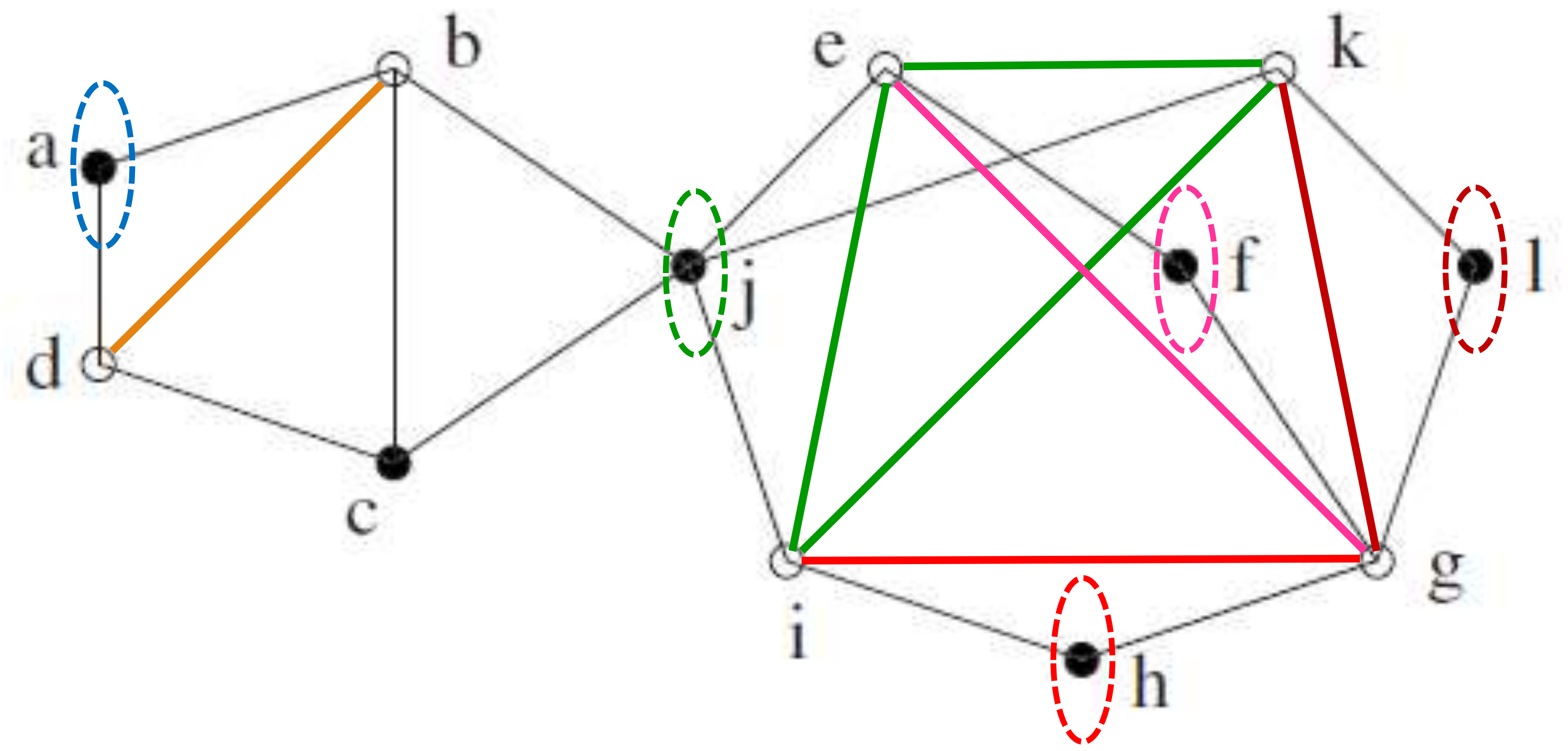
Definition: A vertex v of G is ***quasi-simplicial*** if every non-edge of $N(v)$ has both endpoints which are non-probes (Could potentially be filled-in.)

Theorem: Let $G = (P+N,E)$ be a chordal probe graph, and let v be a quasi-simplicial vertex of G.
If G' is the graph obtained by making v simplicial and removing it, then G is also chordal probe.

Observation: This allows us to define a ***quasi-PEO*** to recognize chordal probe graphs, in the same spirit as Fulkerson and Gross for chordal.

A. BERRY, M. C. GOLUMBIC, AND M. LIPSHTEYN

SIAM J. Discrete Math. [2006]



A chordal probe graph

– Probes are BLACK

– non-Probes are WHITE

Vertex *a* is quasi-simplicial and *d* is not.

However, if *a* is chosen first in a quasi-peo, saturating *N* (*a*) and removing *a* will make *d* quasi-simplicial (actually simplicial)

$\sigma = (a, d, c, b, j, h, l, f, e, k, g, i)$ is a quasi-peo.

Partitioned chordal probe graphs – additional results

- Another recognition method in BGL[2006] is based on LB-substars and LB-simplicial vertices with complexity O($|P||E|$) time.
- Similar results for the more general case where N is not required to be a stable set.

Case 2: **No partition is given** as part of the input.

Golumbic and Lipshteyn [2004]: In a chordal probe graph $G = (P + N, E)$, the probes and non-probes alternate in every chordless cycle of G.

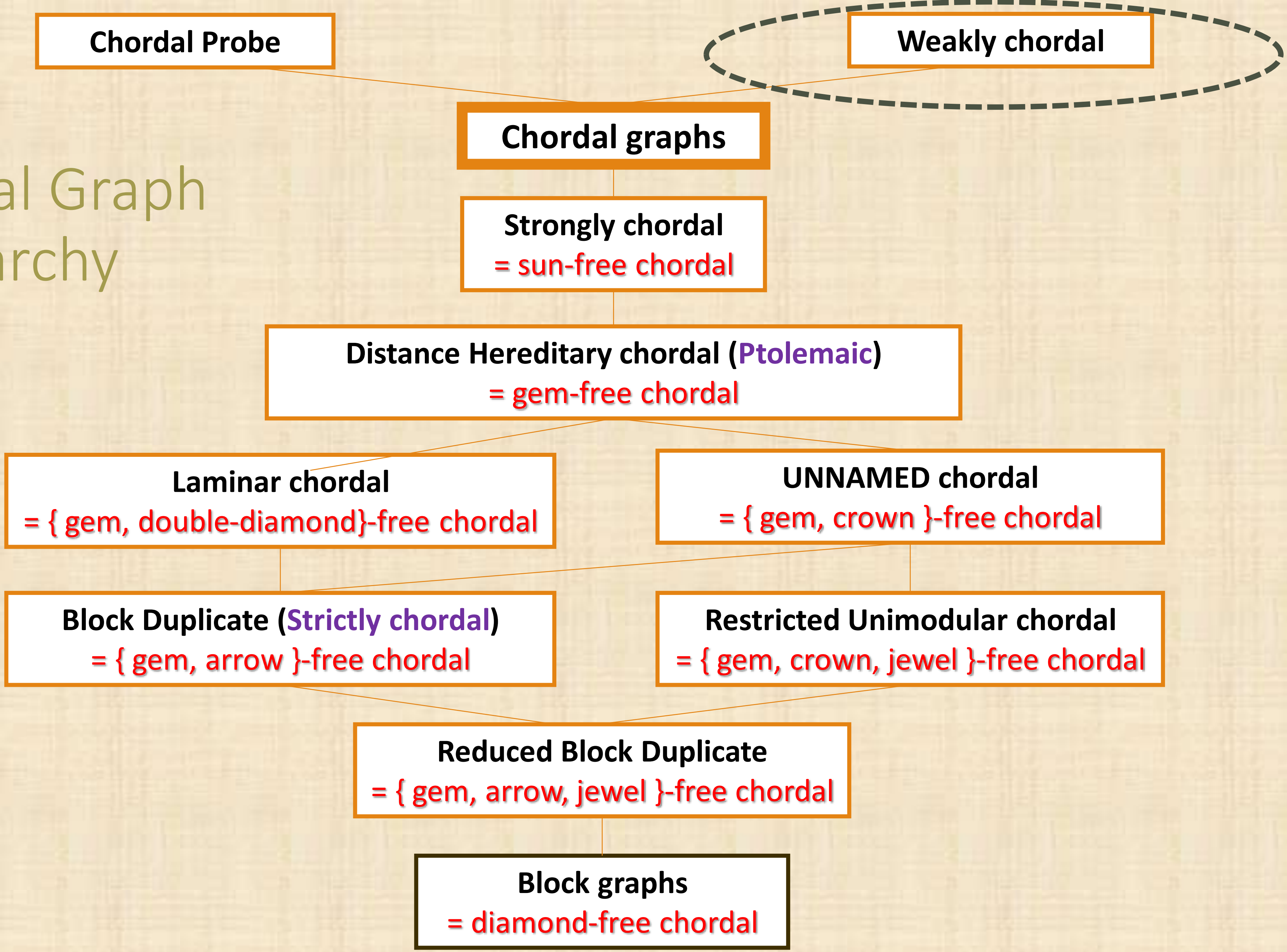
Berry, Golumbic and Lipshteyn [2006] use this property to introduce a new graph class.

Definition: A graph $G = (V, E)$ is cycle-bicolorable if each vertex can be labeled with one of two colors in such a fashion that the colors alternate in every chordless cycle.

Our results:

- Cycle-bicolorable graphs are perfect graphs.
- Both *cycle-bicolorable* graphs and their subfamily *chordal probe* graphs can be recognized in $O(|E|^2)$ time.

A Chordal Graph Hierarchy



Weakly Chordal Graphs – a superclass of chordal graphs

Introduced by Ryan Hayward [1985]

Definition: A graph is *weakly chordal* if it has no induced subgraph isomorphic to C_k or $\overline{C_k}$ for $k \geq 5$.

Remark: A chordal graph is weakly chordal, since $\overline{C_5} = C_5$, and $\overline{C_k}$ (for $k \geq 6$) contains induced copies of C_4.

Trapeziod graphs and tolerance graphs are non-chordal subfamilies of weakly chordal graphs, (see Golumbic and Trenk [2004]).

Definition: A *two-pair* in a graph is a pair of (non-adjacent) vertices x and y such that every shortest path between them has exactly two edges.

Weakly Chordal Graphs – characterizations

Theorem: The following are equivalent.

(i) *G* is a weakly chordal graph.

(ii) Every induced subgraph of *G* is either a *clique* or *has a two-pair*.

Hayward, Hoàng and Maffray [1990]

(iii) If edges are repeatedly added between two-pairs in *G*, the result is eventually a clique. Spinrad and Sritharan [1995]

This leads to an $O(n^4)$ recognition algorithm for weakly chordal graphs.

Berry, Bordat and Heggernes [2000] established a strong structural relationship between *chordal graphs* and *weakly chordal graphs*, leading them to an another $O(n^4)$ recognition algorithm based minimal separators and LB-simplicial edges.

Example: an Analogy between *Chordal & Weakly Chordal*

EPT graphs: Edge intersection graphs of Paths in a Tree

They can have chordless cycles of all sizes, BUT ---

Theorem: The following are equivalent,

chordal $\cap$ EPT

$\equiv$ EPT on deg3 host trees

[Golumbic-Jamison, 1985]

Theorem: The following are equivalent,

weakly chordal $\cap$ EPT

$\equiv$ EPT on deg4 host trees

[Golumbic-Lipshteyn-Stern, 2005]

Finally, the so called *Chordal-Bipartite graphs*
are actually the class weakly chordal $\cap$ bipartite

What is between Chordal and Weakly Chordal Graphs?

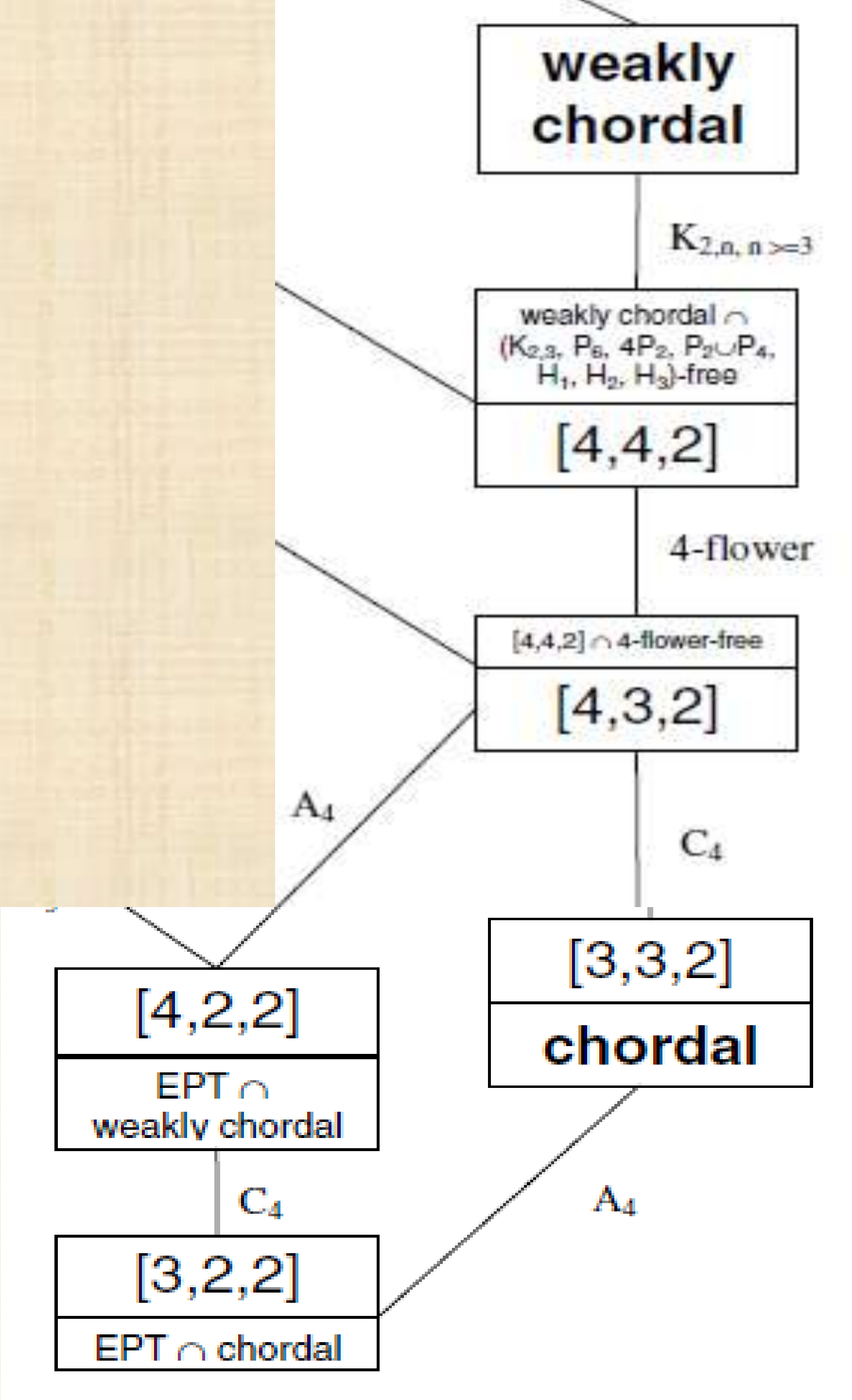
Elad Cohen, Martin Charles Golumbic,
Marina Lipshteyn and Michal Stern [2008]

[h, s, 2] graphs:

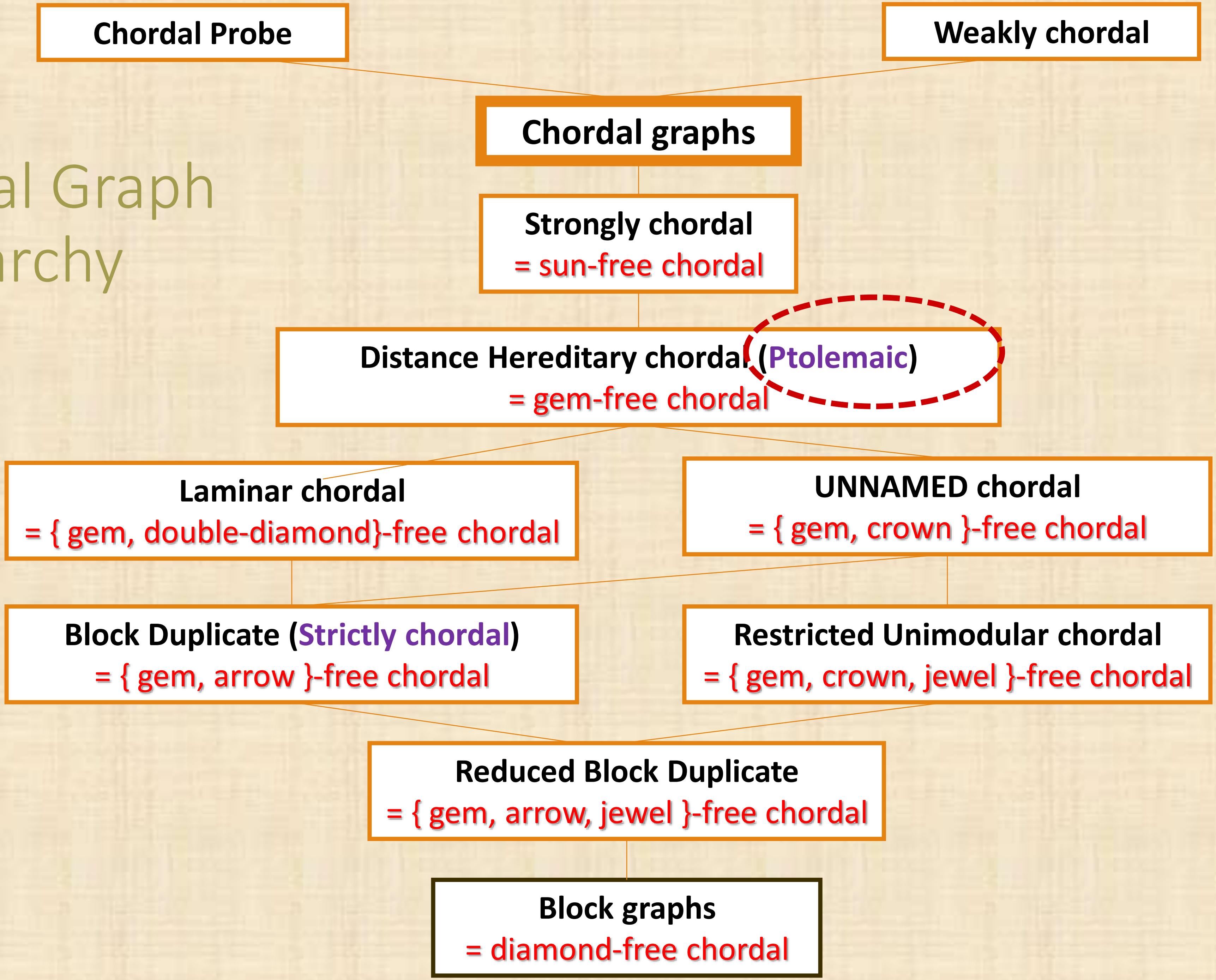
Edge intersection graphs of

subtrees of degree s

in a host tree of degree h



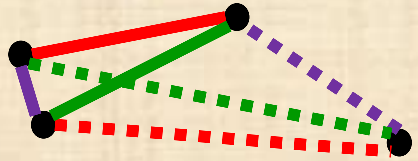
A Chordal Graph Hierarchy



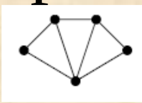
Ptolemaic Graphs *aka* Distance Hereditary Chordal Graphs

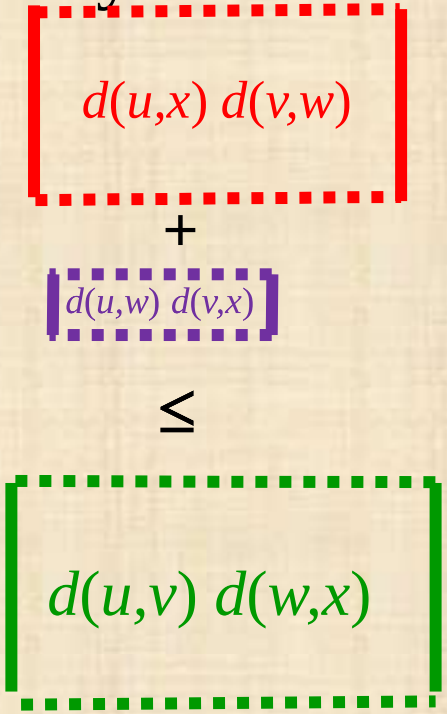
A graph is *ptolemaic* if any four vertices u, v, w, x satisfy the ptolemaic inequality:

$$d(u,v)\, d(w,x) \leq d(u,w)\, d(v,x) + d(u,x)\, d(v,w)$$



Characterizations: Ptolemaic graphs are equivalent to

1. chordal and gem-free -no 
2. chordal and distance hereditary i.e., if the distance function in every induced subgraph of G is the same as in G itself.
3. chordal and the reduced clique graph $\mathcal{RC}(G)$ and the clique-intersection graph $\mathcal{C}(G)$ are the same.
4. chordal and every pair of distinct non-disjoint maximal cliques Q and Q′ of G forms a separating pair.
5. chordal and the minimal vertex separators contained in each maximal clique is a laminar family.



Definition. A **laminar** family of sets: any two of the sets are either disjoint or one of them is a subset of the other.

Subclasses of Ptolemaic Graphs

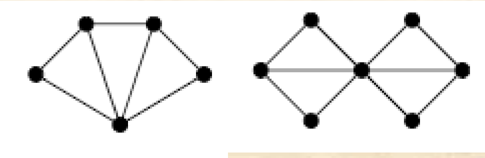
Lilian Markenzon and Christina Fraga Esteves Maciel Waga,
New results on ptolemaic graphs, DAM 2015

AC graphs: The clique-intersection graph $\mathcal{C}(G)$ of a graph G is acyclic

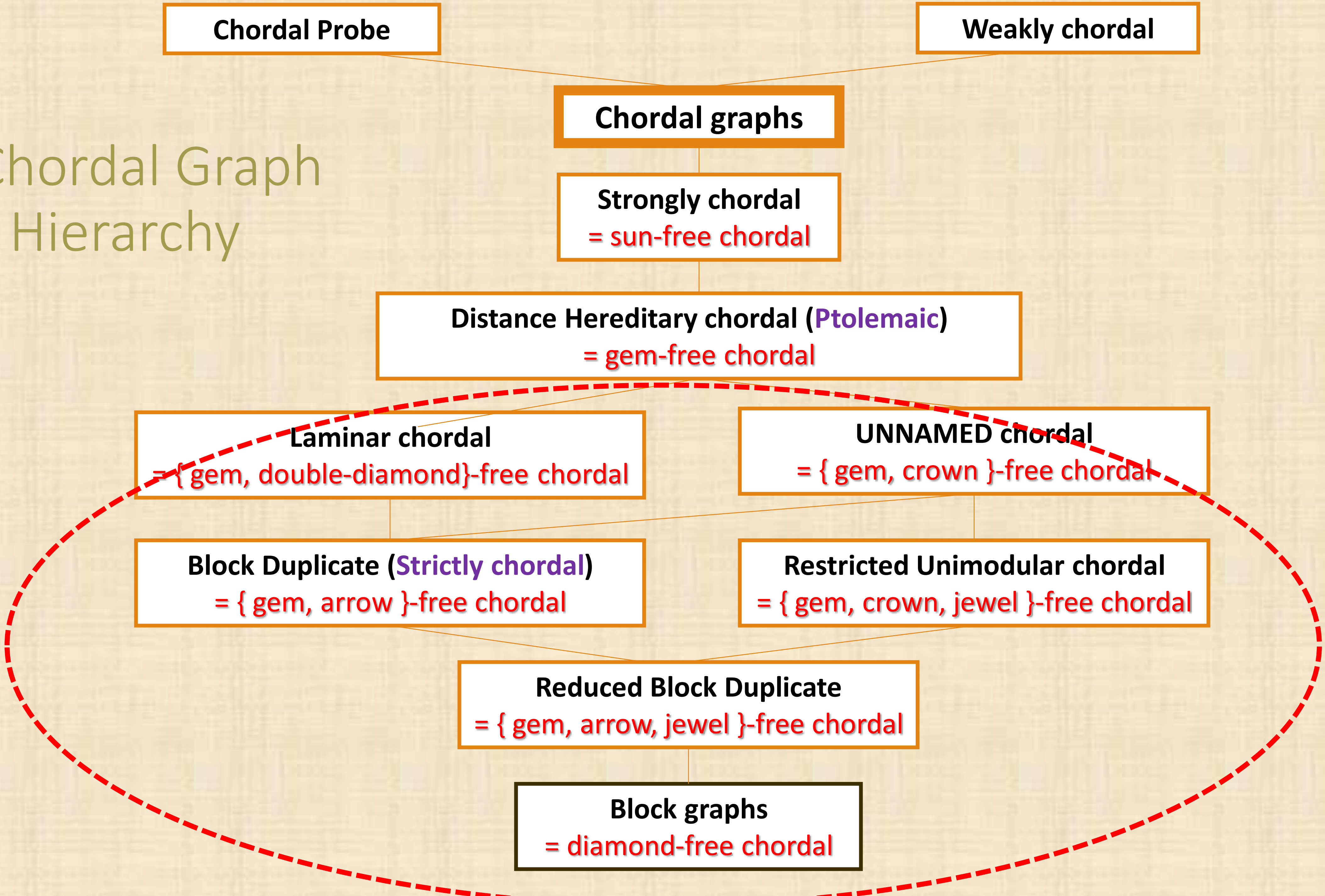
Theorem: A chordal graph is AC if and only if
every vertex in G belongs to at most two maximal cliques.

Laminar chordal graphs: The set $\boldsymbol{S}$ of all minimal vertex separators is laminar.

Theorem: A chordal graph is laminar chordal if and only if
G is {gem, double diamond}-free. no



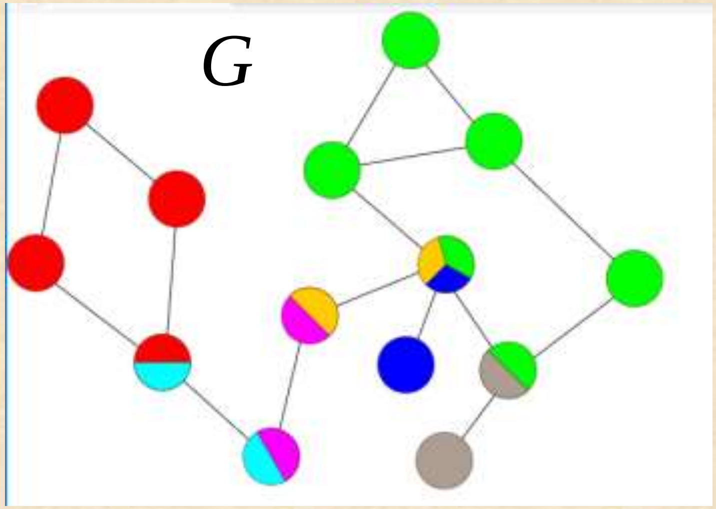
A Chordal Graph Hierarchy



Block Graphs

The **blocks** of a graph are its maximal 2-connected subgraphs.

The **cut-vertices** are the vertices belonging to more than one block.



By Zyqqh at English Wikipedia

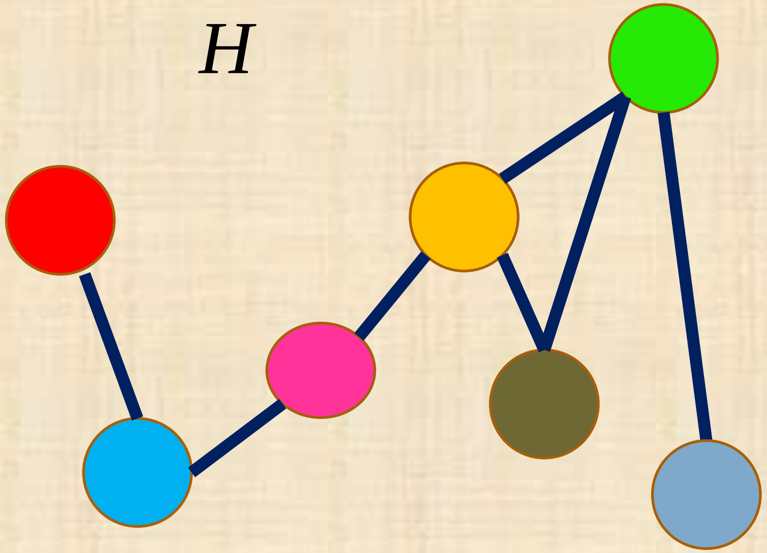
A **block graph** ***H*** is defined and characterized by

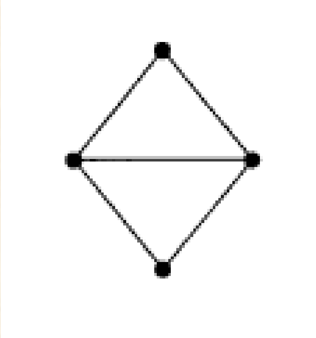
the following equivalent conditions:

(1) *H* is the intersection graph of the blocks of a graph *G*.

(2) Every block of *H* is a clique.

(3) *H* is chordal and diamond-free.





Diamond

Block Duplicate Graphs *aka* Strictly Chordal Graphs

A *block duplicate graph* is a graph obtained by adding **zero or more true twins** to each vertex of a block graph G.

Introduced by Golumbic and Peled (*Discrete Appl. Math.*, 2002)

Characterization 1: Equivalent to the {gem, dart}-free graphs.

Independently, William Kennedy (Masters Thesis, Univ. Alberta, 2005) defined *strictly chordal graphs* from hypergraph properties, proving they too are the {gem,dart}-free graphs.

Characterization 2: Let $G = (V, E)$ be a chordal graph and $\mathcal{S}$ be the set of minimal vertex separators of G. G is a block duplicate graph iff for any distinct $S, S' \in \mathcal{S}$, $S \cap S' = \emptyset$.

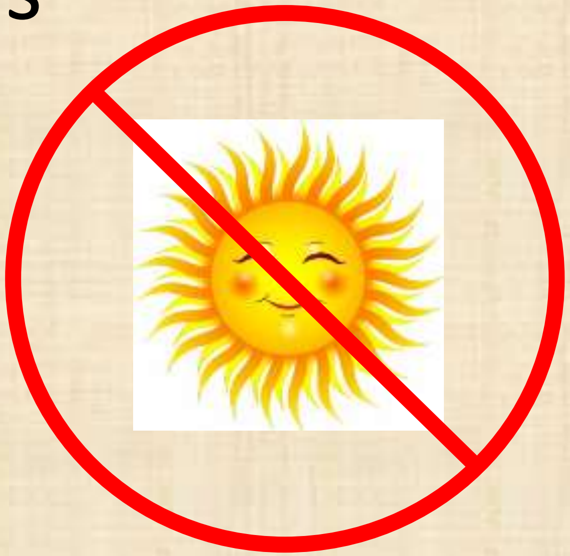
-- Markenzon and Waga (*Discrete Appl. Math.*, 2015)

Strongly Chordal Graphs

Include: Interval graphs, Leaf power graphs, Ptolemaic graphs, Block duplicate graphs,

They are the ***Dark Side*** of chordal graphs

i.e., the *sun-free* chordal graphs



More on that next year.

The Dark World of Strongly Chordal Graphs

Graphs and Optimization (GO XI) **July 5-9, 2020 in Spa, Belgium,** 2020

Martin Charles Golumbic

University of Haifa, Israel