Dagstuhl Seminar 99231

Graph Decompositions and Algorithmic Applications

06.06.-11.06.1999

organized by
A. Brandstädt (University of Rostock, Rostock, Germany)
S. Olariu (Old Dominion University, Norfolk, USA)
J.P. Spinrad (Vanderbilt University, Nashville, USA)

Contents

Talks ordered by speaker

1 Preface

There are many notions of graph decomposition which arise in the literature. Some decompositions involve decomposing a graph using separators of special types (balanced or polynomially bounded, star cutsets, clique cutsets), others involve identification of special sets (substitution or splits), while others involve tree decomposition (treewidth, cliquewidth, branchwidth) or tree composition (Cartesian product, lexicographic product).

These decompositions are of fundamental importance for solving optimization and recognition problems on classes of graphs. For example, substitution decomposition is closely related to such problems as solving problems expressible in monadic second order logic quantifying over vertices and/or edges and comparability graph recognition and optimization. Treewidth and its generalizations are of special importance due to the Robertson-Seymour results on tree decomposition and existential proof of existence of algorithms. Clique cutsets and star cutsets are fundamental tools used in the study of chordal and perfect graphs. Particular tools for working with these decompositions, such as partition refinement and lexicographic breadth first search, have recently been improved and generalized in this context.

This seminar was designed to bring together researchers working on a variety of aspects of graph decomposition. Talks were given studying special classes of graphs, new decomposition techniques and optimization algorithms, and data structures which allow faster decomposition algorithms.

We had 37 participants from Austria, Brazil, Canada, France, Germany, Hungary, Italy, The Netherlands, Norway, Republic of China, Switzerland and USA. During the seminar 25 lectures were given. Moreover, two evening sessions presented open problems.

Jens Gustedt, editor of the electronic journal DMTCS (`http://dmtcs.loria.fr/`) proposed to the organizers to edit a special volume of this journal devoted to our Dagstuhl seminar.

Schloß Dagstuhl and its stuff provided a very convenient and stimulating environment. The organizers wish to thank all those who helped to make the seminar a fruitful research experience.

A. Brandstädt
S. Olariu
J.P. Spinrad

2 Talks

Algorithms for General Overlap Graphs

Eowyn Cenek, University of Waterloo, Canada
Lorna Stewart, University of Alberta, Canada

Let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a finite collection of nonempty sets. The intersection and overlap graphs of $\mathcal{S}$ have:

- vertices $v_1, v_2, \ldots, v_n$,
- edges: v_i and v_j are connected by an edge iff $i \neq j$ and S_i, S_j satisfy:
  for the intersection graph: $S_i \cap S_j \neq \emptyset$,
  for the overlap graph: $S_i \cap S_j \neq \emptyset$ and $S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$.

We show the following:
Given an overlap representation $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, of size bounded by a polynomial in n, for graph $G = (V, E)$:

(1) $\alpha(G)$ can be computed in polynomial time, provided the maximum weight independent set problem is polynomially solvable for the intersection graph of subsets of $\mathcal{S}$.

(2) $\omega(G)$ can be computed in polynomial time, provided $\mathcal{S}$ satisfies the Helly property.

The overlap graphs of subtrees in a tree satisfy the conditions for both (1) and (2).

The algorithms for (1) and (2) follow the methods of:
F. Gavril, Algorithms for a maximum clique and a maximum independent set of a circle graph, *Networks*, 3 (1973), 261–273.

Bipartite–Perfect Graphs

Van Bang Le, University of Rostock, Germany

Two graphs G and H with the same vertex set are P_4–isomorphic if four vertices induce a P_4 in G if and only if they induce a P_4 in H.

Let $\mathcal{C}$ be a class of graphs. A graph G is called $\mathcal{C}$–perfect if G is P_4–isomorphic to a member in $\mathcal{C}$. This definition is motivated by the Semi–Strong Perfect Graph Theorem: if $\mathcal{C}$ is a class of perfect graphs, then $\mathcal{C}$–perfect graphs are perfect.

Problem. Suppose that the recognition problem for $\mathcal{C}$ can be solved in polynomial time. Can you recognize $\mathcal{C}$–perfect graphs in polynomial time, too?

If $\mathcal{C}$ is the class of bipartite graphs, $\mathcal{C}$–perfect graphs can be recognized in linear time.

This follows from a good characterization of bipartite–perfect graphs that intensively makes use of the concepts of homogeneous sets and p–connected graphs, and the fact that tree–perfect graphs can be recognized in linear time, proved by Brandstädt and the author.

Clique–Width and Graph Decompositions that refine Modular Decomposition

Bruno Courcelle, Bordeaux–1 University, LaBRI, France,
`http://dept-info.labri.u-bordeaux.fr/~courcell/ActSci.html`

Certain hard problems have efficient solutions on special classes of graphs, especially those having a hierarchical decomposition bounded by some "width–parameter".

We review such decompositions: tree–decompositions, modular decomposition, bi-clique–decomposition with relevant parameters: tree–width, modular width, clique–width. All of them can be defined in terms of a few basic graph operations.

We review monadic–second order logic as a language able to specify decision problems, counting problems and optimization problems. All these problems can be solved by induction on the expression trees representing the given graphs. The parameter (tree–width etc.) corresponds to limiting the number of graph operations representing the given graphs.

A major open problem is the complexity of clique–width $\leq k$ for $k \geq 4$. We discuss cases (esp. uniformly k–sparse graphs) where the clique–width hierarchy collapses into the tree–width hierarchy.

References: see homepage (works by Courcelle, Olariu, Makowsky, Rotics).

Product Graphs

Wilfried Imrich, Montanuniversität Leoben, Leoben, Austria

The talk begins with a survey of structure theorems for the main associative products: The Cartesian, the strong, the cardinal and the lexicographic one. The emphasis is on prime factor decomposition of connected graphs with respect to these products. Uniqueness (and nonuniqueness) results are mentioned.

This prepares for the second part, in which the complexity of algorithms that decompose a given graph with respect to these products are discussed: Polynomial algorithms for the prime factorization (PFD) of connected graphs with respect to the Cartesian and the strong product and of nonbipartite connected graphs with respect to the Cartesian product. For the lexicographic product PFD is equivalent to the graph isomorphism problem.

One algorithm for PFD of graphs with respect to the Cartesian product is then taken as the basis for methods to embed graphs isometrically into hypercubes and for the recognition of median graphs. (Currently the best algorithm for recognizing median graphs has complexity $O(m^{1.41})$, where m is the number of edges of the graph being investigated; the same as that for recognizing triangle–free graphs by Alon, Yuster and Zwick).

Graph decompositions and factorizing permutations

Part I: Michel Habib, University of Montpellier, France
Part II: Christophe Paul, University of Bordeaux, France

To build an efficient recognition algorithm for a given class of graphs it is very useful to have some decomposition tool such as elimination ordering (i.e. simplicial scheme provided by LexBFS for chordal graphs) or a decomposition tree (i.e. modular decomposition).

We propose a new algorithm to recognize P_4–indifference graphs. P_4–indifference graphs are those graphs admitting an ordering $<$ of the vertices such that for every P_4 $a - b - c - d$ of the graph: $a < b < c < d$ or $d < c < b < a$. Our algorithm relies on modular decomposition and has linear time complexity.

Then we define the notion of a factorizing permutation of the vertices of a graph associated with a tree decomposition. In such a permutation the strong modules of the graph must appear as intervals. We show how to use this idea by presenting two algorithms. The first one is for cograph recognition, the second computes modular decomposition. Both are very simple and run in $O(n + m \log n)$ time complexity.

Branchwidth

Ton Kloks, Vrije Universität Amsterdam, Amsterdam, The Netherlands
Jan Kratochvil, Charles University Prague, Prague, Czech Republic
Haiko Müller, Universität Jena, Jena, Germany

A branchdecomposition for a graph $G = (V, E)$ is a pair (T, τ), where T is a binary tree with $|E|$ leaves and τ is a 1–1 mapping from the leaves of the tree to the edges of the graph. For each vertex in the graph consider the leaves corresponding with edges containing that vertex. These leaves uniquely define a subtree of T.

For each edge in the tree T we define the *order* as the number of subtrees containing this edge of T. The maximum order over all edges is the width of the branchdecomposition (T, τ). The branchwidth of G, $bw(G)$ is the minimum width over all branchdecompositions of G. There is a close relation with treewidth: For every graph G holds:

$$bw(G) \leq tw(G) + 1 \leq \frac{3}{2}\, bw(G).$$

We present two results on branchwidth:

1. Branchwidth is $\mathbb{NP}$–complete for split graphs.
2. Branchwidth is polynomial for interval graphs, permutation graphs and closely related classes like trapezoid graphs, and d–dimensional cocomparability graphs.

Our interest in the branchwidth is twofold: first of all, the difference in the width of tree–decomposition and branchdecomposition is of great importance for the running time of the algorithm. Secondly, branchwidth is polynomial for planar graphs. The treewidth problem is open.

Clique–width, NLC–width and Efficient Algorithms

Egon Wanke, Universität Düsseldorf, Düsseldorf, Germany

We compare the class of clique–width and NLC–width bounded graphs and introduce a very general method for the design of polynomial time algorithms for $\mathbb{NP}$–complete graph problems. Clique–width and NLC–width bounded graphs are recursively defined node labeled graphs. We especially consider graph problems that cannot be defined in monadic second order logic with quantifications only over vertices and vertex sets. Two examples are the Hamilton circuit problem and the simple max cut problem.

Asteroidal sets in graphs

Ton Kloks, Vrije Universität Amsterdam, Amsterdam, The Netherlands
Dieter Kratsch, Universität Jena, Jena, Germany
Haiko Müller, Universität Jena, Jena, Germany

A set $A \subseteq V$ is called asteroidal set of a graph $G = (V, E)$ if for each $a \in A$ there is a connected component of $G - N[a]$ containing all vertices of $A \setminus \{a\}$. The maximum cardinality of an asteroidal set of G is denoted by $an(G)$, and is called the asteroidal number of G.

We generalize our algorithm for AT–free graphs (those are graphs with asteroidal number at most 2) to graphs of bounded asteroidal number. This way we obtain algorithms computing the treewidth, minimum fill–in and the vertex–ranking number of a graph in time $O(n^5 r + kr^{k+1}(n+m)n \log n)$, and the maximum size of an independent set, independent dominating set and efficient dominating set in time $O(n^{k+2})$. Here n, m, k and r denote the number of vertices, edges, asteroidal number and number of minimal separators of the input graph.

Computing the asteroidal number of a graph is hard, the corresponding decision problem is $\mathbb{NP}$–complete, even if we restrict to 3–connected, 3–regular triangle–free planar graphs. We obtain polynomial time algorithms for circular cocomparability graphs ($O(n^3)$), claw–free graphs ($O(n^3)$), HHD–free graphs ($O(n^{3.5})$).

Approximating Bandwidth by Mixing Layouts of Interval Graphs

Dieter Kratsch, Universität Jena, Jena, Germany
Lorna Stewart, University of Alberta, Canada

We examine the bandwidth problem in circular–arc graphs, chordal graphs with a bounded number of leaves in the clique tree, and k–polygon graphs (fixed k). All of these graph classes admit efficient approximation algorithms which are based on exact or approximate bandwidth layouts of related interval graphs.

Specifically, we obtain a bandwidth approximation for circular–arc graphs that has performance ratio 2 and executes in $O(n \log^2 n)$ time, or performance ratio 4 while taking $O(n)$ time. For chordal graphs with not more than k leaves in the clique tree, we obtain a performance ratio of $2k$ in time $O(k(n+m))$, and our algorithm for k–polygon graphs has performance ratio $2k^2$ and runs in time $O(n^3)$.

Our approximation algorithm for circular–arc graphs is optimal since there is no polynomial time bandwidth approximation algorithm for circular–arc graphs with performance ratio $2-\epsilon$ for any $\epsilon > 0$ unless $\mathbb{P} = \mathbb{NP}$ [W. Unger, FOCS '98].

Polynomial Time Recognition of Clique Width 3 Graphs

Derek Corneil, University of Toronto, Toronto, Canada
Michel Habib, University of Montpellier, France
Jean–Marc Lanlignel, University of Montpellier, France
Bruce Reed, CNRS, Paris, France
Udi Rotics, University of Toronto, Toronto, Canada

A graph can be constructed with the following operations: create a vertex with label i; take disjoint union, merging sets with the same label; for a pair of labels, add all edges between vertices of one label with vertices of the other label; relabel all vertices of one label with another, existing label. The clique width of a graph is the minimum number of labels that can be used to generate this graph.

In this talk we present an overview of an algorithm to determine if a graph has clique width at most 3. (A graph has clique width at most 2 iff it is a cograph.) Our algorithm runs in time $O(n^2 \cdot m)$ when $n = |V_G|$ and $m = |E_G|$.

The Divide–and–conquer Approach to Modular Decomposition

Ross McConnell, University of Colorado, USA
Jeremy P. Spinrad, Vanderbilt University, Nashville, USA
Jens Gustedt, INRIA Lorraine, France
Elias Dahlhaus, Universität Köln, Köln, Germany

On an adjacency–list representation of a graph, each vertex carries a list of its neighbors and the running time of an algorithm is measured against $n+m$, when n is the number of vertices and m is the number of edges. In this talk, we advocate an alternative data structure, a *partially complemented representation*, where each vertex carries either a list of its neighbors or a list of its non–neighbors, and the running time of an algorithm is measured against $n + m'$, where m' is the sum of lengths of the lists. Surprisingly many common graph algorithms can be modified to run in $O(n + m')$ time.

The talk explores how an $O(n+m')$ algorithm for depth–first search leads to a simple $O(n + m \log n)$ divide–and–conquer algorithm for modular decomposition. A similar trick is a key step in the linear modular decomposition algorithm I have obtained with Dahlhaus + Gustedt, and the $O(n + m)$ bound for permutation–graph recognition I have obtained with Spinrad. The algorithms can also be modified to run in $O(n + m')$ time.

Computing Optimal Linear Layouts of Trees in Linear Time

Konstantinos Skodinis, Universsät Passau, Passau, Germany

We present a linear time algorithm for computing linear layouts of trees which are optimal with respect to vertex separation. The best algorithm known so far is given by Ellis, Sudborough and Turner and needs $O(n \cdot \log n)$ time. This result solves several other related open problems on trees as for example the one of Megiddo, Hakimi, Garey, Johnson, and Papadimitriou concerning optimal search strategies.

Tight bounds on the size of Indecomposable graphs and hypergraphs

Paola Bonizzoni, Universita di Milano, Milano, Italy
Ross McConnell, University of Colorado, USA

Recent work on efficient algorithms for prime graphs recognition and modular decomposition is based on the discovery that prime induced subgraphs are densely nested when they occur. Such a property suggests that although a prime graph cannot be decomposed by modular decomposition, it can be decomposed into a sequence of prime subgraphs which grows from a P_4 up to the prime graph itself.

This fact naturally leads to develop an incremental procedure to build and hence to recognize a prime graph. In fact, tight lower bounds on the "nesting density" of prime graphs were used by Cournier and Habib for designing a linear time algorithm for the modular decomposition of directed graphs.

In this talk, I present a notion of hypergraph which naturally generalizes that of a k–ary relation and of k–structure and leads to the definition of prime hypergraph. Then, results on tight bounds (upper and lower) on the nesting density of prime structures which generalize those obtained for graphs are shown. A characterization of non prime structures in terms of forbidden small size prime substructures is given.

On linear and circular structure of (claw,net)–free graphs

Andreas Brandstädt, Universität Rostock, Rostock, Germany
Feodor F. Dragan, Universität Rostock, Rostock, Germany

We prove that every (claw,net)–free graph contains an induced doubly dominating cycle or a dominating pair. Moreover, we present a linear time algorithm which, for a given (claw,net)–free graph, finds either a dominating pair or an induced doubly dominating cycle. Our algorithm essentially uses a vertex ordering of the graph produced by LexBFS — a kind of partition refinement technique. The existence of a dominating pair or an induced doubly dominating cycle in (claw,net)–free graphs can be used to design efficient algorithms for domination–like problems, Hamiltonian problems and the maximum independent set problem.

An Error-Tolerant Algorithm for Interval Graph Recognition

Wen-Lian Hsu, Institute of Information Science, Academia Sinica, ROC

An important problem in DNA physical mapping is to reassemble the clone fragments to determine the structure of the entire molecule. The error-free version of this problem can be modeled as an interval graph recognition problem, where an interval graph is the intersection graph of a collection of intervals. However, since the data collected from laboratories almost surely contain some errors, traditional error-sensitive recognition algorithms can hardly be applied directly. These include approaches based on lexicographical breadth-first search, maximal clique construction (and hence, the PQ-tree algorithm) and direct modular decomposition.

We present a new test here based on the idea of constructing the overlap graph of the given graph. Such a construction will lead to a modular decomposition at the end. This new algorithm has the following features:

(1) the algorithm will assemble the clones efficiently when the data are error-free.

(2) in case the error rate is small (say, less than 3%) the test can likely detect and automatically correct the following three types of errors: false positives, false negatives and chimeric clones.

(3) the test will also identify those parts of the data that are problematic, thus allowing biologists to perform further experiments to clean up the data.

Information System on Graph Class Inclusions

Andreas Brandstädt, Universität Rostock, Rostock, Germany
Van Bang Le, Universität Rostock, Rostock, Germany
Frank Siegemund, Universität Rostock, Rostock, Germany
Thomas Szymczak, Universität Rostock, Rostock, Germany

We present an information system on graph class inclusions available via internet (`http://www.informatik.uni-rostock.de/~gdb/isgci/Isgci.html`) to keep an updated knowledge base of graph classes and their inclusions. The user has the possibility to ask queries about inclusions of classes and to draw inclusion hierarchies for selected classes.

Special graph classes are interesting for several topics of discrete mathematics and theoretical computer science, such as the time complexity of algorithmic problems restricted to special graphs. Up to now, more than 300 graph classes and their inclusions have been studied.

The book of A. Brandstädt, V.B. Le, J. Spinrad, Graph Classes: A Survey, SIAM Monographs in Discrete Mathematics and Applications, Philadelphia, 1999, is a main source of information on structural properties and inclusions of graph classes. Our information system is based on the list of inclusions given in Appendix B of this book.

Minimum Average Distance Trees for Distance Hereditary Graphs

Elias Dahlhaus, Universität Köln, Köln, Germany
P. Dankelmann, University of Natal, Durban, South Africa
W. Goddard, University of Natal, Durban, South Africa
H. Swart, University of Natal, Durban, South Africa

The average distance of a graph is the sum of its distances. This concept can be extended for vertex weighted graphs. For graphs in general determining a minimum average distance spanning tree is $\mathbb{NP}$–complete. We show that this problem can be solved in linear time if the given graph is distance hereditary.

The Role of Clique Multigraphs in Intersection Graph Theory

Erich Prisner, Universität Hamburg, Hamburg, Germany

We want to illustrate how the clique multigraph – the intersection multigraph of the set of all maximal cliques – of a graph can be used for recognizing certain intersection graphs, respectively for reconstructing an intersection model. We do this by giving four examples, two old ones from the literature and two recent ones.

We start by presenting Bernstein & Goodman's Theorem on the representability of a chordal graph in a tree. We proceed by presenting a method to recognize line graphs by clique multigraphs, which relies on van Rooij/Wilf's Theorem. The third example concerns recognition of intersection graphs of linear 3–uniform hypergraphs when the input graphs have large enough minimum degree. The results of this part are joint work with Y. Metelsky, S. Suzdal, and R. Tyshkjevich. In the final example we discuss the recognition problem of line bigraphs — the intersection bigraph of the edge sets of two graphs with the same vertex set.

Prime graph structures

Alain Cournier, University of Amiens

Let $G = (V, E)$ be a graph, a subset X of V is a module of G whenever for $a, b \in X$ and $x \in V - X$, $(a, x) \in E$ (respectively $(x, a) \in E$) if and only if $(b, x) \in E$ (respectively $(x, b) \in E$). For instance, $\emptyset$, $\{x\}$ where $x \in V$, and V are modules of G, called trivial modules.

A graph is then said to be indecomposable or prime when all of its modules are trivial. We now introduce the minimal prime graphs in the following way. Given a prime graph $G = (V, E)$ and vertices $x_1, \ldots, x_k$ of G, G is said to be minimal for $x_1, \ldots, x_k$ whenever for every proper subset W of V, if $x_1, \ldots, x_k \in W$ and $W \geq 3$, then the induced subgraph $G(W)$ of G is decomposable.

In this talk, we characterize the minimal prime graphs for $k = 1$ or $k = 2$.

Approximation results for coloring problems

Klaus Jansen, IDSIA – Lugano, Switzerland

In this talk, we present appproximation results for two coloring problems. Given a graph $G = (V, E)$ with n vertices and a sequence $k_1, \ldots, k_n$ of coloring cost $k_i > 0$, the problem is to find a feasible coloring f of G with minimum total cost $\sum_{v \in V} k_{f(v)}$. We give approximability and inapproximability results for bipartite, chordal, comparability, interval, permutation and split graphs. As an example, we show that there exists no approximation algorithm A for coloring bipartite graphs with total cost $A(I) \leq O(|V|^{1/2-\epsilon})OPT(I)$, unless $\mathbb{P} = \mathbb{NP}$.

In the second part, we present approximability results for wavelength allocation of directed requests in a tree. Given a tree T and a set R of paths $r = (u, v)$ with $u, v \in V$, find a wavelength (color) $f(v)$ for each request such that each set $R_w = \{r \in R | f(r) = w\}$ consists of only edge disjoint paths for each w. The goal is to find an assignment (coloring) f with minimum number of colors. We present an approximation algorithm that computes for each instance I a coloring with $\leq \lceil \frac{5}{3} OPT(I) \rceil$ wavelengths. The second part is joint work with T. Erlebach, C. Kaklamanis and P. Persiano.

Edge Clique Graphs and Chordal Graphs

Maruiz R. Carioli, University of Rio de Janeiro, Rio de Janeiro, Brazil
Jayme L. Szwarcfiter, University of Rio de Janeiro, Rio de Janeiro, Brazil

The edge clique graph of a graph G is one having as vertices the edges of G, two vertices being adjacent if the corresponding edges of G belong to a common clique. The class has been introduced by Albertson and Collins (1989). Although many interesting properties of it have been since studied, we do not know complete characterizations of edge clique graphs of any nontrivial class of graphs. In this matter, we describe characterizations relative to edge clique graphs and some classes of chordal graphs, as starlike, starlike–threshold, split and threshold graphs. In particular, a known necessary condition for a graph to be an edge clique graph is that the sizes of all maximal cliques and intersections of maximal cliques ought to be triangular numbers. We show that this condition is also sufficient for starlike–threshold graphs.

Linear–time Register Allocation for a fixed number of registers

H. Bodlaender, University of Utrecht, Utrecht, The Netherlands
Jens Gustedt, INRIA Lorraine, France
Jan Arne Telle, University of Bergen, Bergen, Norway

We show that for any fixed number of registers there is a linear–time algorithm which given a structured (≡ goto–free) program finds, if possible, an allocation of variables to registers without using intermediate storage. Our algorithm allows far rescheduling, i.e. that straight–line sequences of statements may be reordered to achieve a better register allocation as long as data dependencies are not violated.

The main algorithmic technique used is that of bounded treewidth algorithms, applicable since the control–flow graph of a structured program is known to have treewidth at most 6 (for C programs).

Forbidden Subgraph Decomposition

Jerry Spinrad, Vanderbilt University, Nashville, USA

This talk introduces a new notion of graph decomposition, based on forbidding a fixed bipartite graph from occuring as an induced subgraph of the graph formed by edges which cross a cut.

We study graph decompositions defined by small forbidden subgraphs, and raise a number of new open problems.

Median graphs and triangle–free graphs

Wilfried Imrich, Montanuniversität Leoben, Leoben, Austria

In this talk characterization and decomposition problems of partial cubes, median graphs, semimedian graphs and their generalizations to Hamming graphs, partial Hamming graphs and quasimedian graphs are discussed. In particular the currently best characterization and embedding algorithms and their complexities are compared. Moreover, a connection with triangle–free graphs is established which allows further improvements of the recognition algorithms for median and quasimedian graphs.

Open Problems

Zsolt Tuza, Hungarian Academy of Sciences, Hungary

We discuss open problems and related results on graph colorings, mostly from algorithmic aspects. Subjects include list colorings of planar graphs, precoloring extension on interval graphs, on–line vertex ranking, excluded cycle lengths vs. chromatic number, the Acyclic Orientation game, triangle coverings, and the complexity of hypergraph 2–coloring.

3 Open Problems

3.1 Bin packing with conflicts (Klaus Jansen)

Given a graph $G = (V, E)$ and sizes $size(v) \in (0, 1]$, the problem is find a partition of V into independent sets (bins) $U_1, \ldots, U_m$ with $\sum_{v \in U_i} size(v) \leq 1$ for each $1 \leq i \leq m$. The goal is to find a partition with a minimum number of bins. There exists a asymptotically fully polynomial time approximation scheme (AFPTAS) for trees, planar graphs and graphs with constant treewidth (see SWAT'98).

Open Question: Is there a AFPTAS or is the problem MAXSNP hard for interval graphs or bipartite graphs?

3.2 Coloring of permutation (and comparability) graphs (Klaus Jansen)

Given a permutation (or comparability graph) G, the problem is to find a partition into independent sets $U_1, \ldots, U_m$ with $|U_i| \leq t$ for each $1 \leq i \leq m$. The goal is to minimize the number m of independent sets. The problem is NP-complete for permutation graphs and each constant $t \geq 6$ (see STACS'98).

Open question: What is the complexity of this problem for permutation and comparability graphs and $t = 3, 4$ and 5? This question was mentioned also as an open problem by Lonc and Möhring.

3.3 Open Problems on biclique–decompositions (Bruno Courcelle)

1. Complexity of $cwd(G) \leq k$.
2. Characterization of $cwd(G) \leq k$ by forbidden induced subgraphs (P_4 for $k = 2$). Finite sets? Structure?
3. Upper bounds:
   (a) $cwd(G) \leq f(|V_G|)$, say $f(n) = 5n/6$,
   (b) tight upper bounds in comparision with twd,
   (c) $cwd(G) \leq cwd(G - x) + 1$, $x \in V_G$,
   (d) $cwd(G) \leq cwd(G - e) + 1$, $e \in E_G$.
4. Reduction rules for $cwd(G) \leq k$.

3.4 Graphs with fixed cliquewidth (Jerry Spinrad)

There are algorithms for solving certain $\mathbb{NP}$–complete problems in polynomial time for graphs with fixed cliquewidth K, if the composition sequence for the graph is given. Can these problems be solved in polynomial time on graphs with cliquewidth K if the input is given in the form of an adjacency matrix? Example problems include max cut, Hamiltonian circuit.

An implicit representation of a graph is an assignment of $O(\log n)$ bits to each vertex, and a test procedure which decides corretly whether two vertices x, y are adjacent based only on the bits stored at x and y. Can you find an implicit representation of graphs with cliquewidth K?

3.5 Bandwidth of bipartite permutation graphs (Ton Kloks)

Let G be a bipartite permutation graph. The following conjectures are stated:

1. There is an optimal bandwidth layout that respects the order in both color classes.
2. There is an optimal layout that starts with the leftmost vertex of one of the two bipartite color classes.

3.6 Chordal graphs for which branchwidth = treewidth (Ton Kloks)

Characterize those chordal graphs for which branchwidth is equal to treewidth. It is known that for every graph G holds

$$bw(G) \leq tw(G) \leq \frac{3}{2} \cdot bw(G).$$

Zsolt Tuza presents a solution to the above problem. He proved: Let G be a chordal graph with clique number $\omega > 2$. Then the branchwidth of G is equal to ω if and only if G contains an ω–clique K such that any three vertices of K have a common neighbor outside K.

3.7 Relative Minimal Elimination Ordering (Elias Dahlhaus)

We consider the following problem, called *Relative Minimal Elimination Ordering.* Given a graph $G = (V, E)$ which is a subgraph of the chordal graph $G' = (V, E')$, compute an inclusion minimal chordal graph $G'' = (V, E'')$, such that $E \subseteq E'' \subseteq E'$ or the perfect elimination ordering of G''. In general, it can be done in $O(nm)$ time.

The open problem is whether one can do it faster than $O(nm)$ for planar graphs, i.e. whether it can be done faster than $O(n^2)$ for planar graphs. There is some evidence that it can be done in almost linear time. For example, it is possible to find, for any planar graph G, an elimination ordering that has a subset minimal fill-in.

3.8 Some Poset Problems (Jayme Szwarcfiter)

1. Given a graph G, is it the intersection graph of the chains of a poset? Characterize the class of all graphs which are the intersection graph of the chains of a poset.

2. Given a graph G, is it the intersection graph of the antichains of a poset? Characterize the class of all graphs which are the intersection graph of the antichains of a poset.

3. Let $L = \{l_1, \ldots, l_t\}$ be a set of labels and E a set of elements of a poset P. To each element $e_i \in E$ we associate a label subset $L_i \subseteq L$.
   Question. Is there a chain decomposition of P such that each label $l \in L$ appears in at most two different chains?

3.9 Minimizing memory requirement for table computations in bounded treewidth algorithms (Jan Arne Telle)

Let T_r be a tree with root r and with weighted vertices $w : V(T) \rightarrow \mathbb{N}$. An ordering of edges $e_1, e_2, ..., e_{n-1}$ defines a *bottom–up traversal* on T_r if for every vertex, all edges in the subtree rooted at that vertex appear before the edge from that vertex to its parent. Define vertex v to be *alive in the interval* $[i, j]$ where e_i is the lowest–numbered edge containing vertex v and e_j is the highest–numbered such edge. Define the *memory requirement* of this traversal order to be the maximum, over all i between 1 and $n-1$, of the sum of weights of all vertices alive at step i.

Problem. Let T be a vertex weigthed tree. Find a root r of T and a bottom–up traversal order of T_r that minimizes the memory requirement.

For depth–first traversal orders, and for the case when all vertex weights are equal, the problem can be solved efficiently (see B. Aspvall, A. Proskurowski, J. A. Telle: Memory requirements for table computations in partial k–tree algorithms, SWAT'98, to appear in a special issue of Algorithmica on Bounded Treewidth). The general problem as described above is open.