Fields Institute Monographs

The Fields Institute for Research in Mathematical Sciences

Efficient Graph Representations

Jeremy P. Spinrad

American Mathematical Society

Efficient Graph Representations

This page intentionally left blank

http://dx.doi.org/10.1090/fim/019

Fields Institute Monographs

The Fields Institute for Research in Mathematical Sciences

Efficient Graph Representations

Jeremy P. Spinrad

American Mathematical Society
Providence, Rhode Island

The Fields Institute for Research in Mathematical Sciences

The Fields Institute is named in honour of the Canadian mathematician John Charles Fields (1863–1932). Fields was a visionary who received many honours for his scientific work, including election to the Royal Society of Canada in 1909 and to the Royal Society of London in 1913. Among other accomplishments in the service of the international mathematics community, Fields was responsible for establishing the world's most prestigious prize for mathematics research—the Fields Medal.

The Fields Institute for Research in Mathematical Sciences is supported by grants from the Ontario Ministry of Education and Training and the Natural Sciences and Engineering Research Council of Canada. The Institute is sponsored by McMaster University, the University of Toronto, the University of Waterloo, and York University, and has affiliated universities in Ontario and across Canada.

2000 *Mathematics Subject Classification.* Primary 05C62, 05C17, 05C50, 05C85, 05-00, 05-02, 68R10, 68W01, 68P05, 68Q30, 68-01.

For additional information and updates on this book, visit
www.ams.org/bookpages/fim-19

Library of Congress Cataloging-in-Publication Data

Spinrad, Jeremy P.
Efficient graph representations / Jeremy P. Spinrad.
p. cm. — (Fields Institute monographs ; 19)
Includes bibliographical references and index.
ISBN 0-8218-2815-0 (acid-free paper)
1. Representations of graphs. I. Title. II. Series.

QA166.242.S65 2003 2003045106
511′.–dc21 CIP

∞ The paper used in this book is acid-free and falls within the guidelines
established to ensure permanence and durability.
This publication was prepared by The Fields Institute.
Visit the AMS home page at `http://www.ams.org/`

10 9 8 7 6 5 4 3 2 1 08 07 06 05 04 03

Contents

This page intentionally left blank

Glossary

I should note that these definitions sometimes may be in conflict with general use of a term. If a term is used in a specific manner in this book, and this specific usage is simple to explain, I choose to include the definition which applies to this book rather than a more general definition which might be harder to understand. For example, this book does not discuss multigraphs, and in some cases the definition I choose to include would not correspond to standard definitions on multigraphs. I also do not worry about the fact that my definition of order notation symbols is nonstandard if negative functions are allowed, since these are used only for time analyses in this book.

Adjacent: Vertex x is adjacent to vertex y if there is an edge from x to y.

Adjacency lists: A form of graph representation in which neighbors of vertex i are stored in a linked list. In general, this list is assumed to be unordered, in the sense that vertex j may precede vertex k on an adjacency list even though $j < k$. However, it is possible to sort all adjacency lists in $O(n+m)$ time, if this is desired. Technically, an adjacency list is a matrix of linked lists, thus allowing the list of neighbors of vertex i to be accessed in constant time.

Adjacency matrix: A form of graph representation in which a graph is stored as an $n \times n$ matrix. The entry (i, j) of the adjacency matrix is 1 if i has an edge to j, and 0 otherwise.

Anti-hole: A hole (chordless induced cycle of length at least 4) in the complement graph.

Arboricity: The arboricity of G is the minimum number of forests such that every edge of G is contained in one of these forests.

Arc: A connected subset of a circle. Also used as an alternate name for edge, especially in a tree.

Asteroidal triple: A trio of mutually nonadjacent vertices x, y, z with the property that for each pair of vertices in the trio, there is a path between these vertices which avoids all neighbors of the third vertex in the trio. Graphs without asteroidal triples are also called AT-free.

AT-free: A graph is AT-free if it contains no asteroidal triples.

Augmented adjacency matrix: Entries are the same as in the adjacency matrix, except with 1s along the main diagonal rather than 0s.

Autonomous set: Alternative name for module.

Balanced k-module: A subset S of vertices induces a balanced k-module if there is a partition S_1, S_2 of S such that $|S|/3 \leq |S_1| \leq 2|S|/3$, and S_1 can be partitioned into at most k subsets $S_{1,i}$ in such a way that every vertex of S_2 is either adjacent to every vertex of $S_{1,i}$, or to no vertex in $S_{1,i}$.

Bandwidth: The bandwidth of an ordering of the vertices of G is the maximum difference between endpoint positions of any edge. The NP-complete bandwidth problem asks whether there is any ordering of the vertices which has bandwidth at most k.

Berge graph: A graph without any odd holes or odd anti-holes of length greater than three; these have recently been shown to be equivalent to perfect graphs, resolving a famous open question.

Biconvex graph: Bipartite graph with the property that there are orderings of each color class satisfying for all x, the neighbors of x occur consecutively in the ordering.

Bipartite adjacency matrix: Matrix representation of a bipartite graph. Rows correspond to vertices from one color class, columns to vertices from the other color class, with a 1 in row i column j if vertex i is adjacent to vertex j, and 0 otherwise.

Bipartite graph: A graph with the property that vertices can be partitioned into sets X, Y such that every edge goes between a vertex in X and a vertex in Y. The sets X and Y are called color classes of G.

Bisimplicial edge: An edge (x, y) of a bipartite graph such that $\mathrm{N}(x) \cup \mathrm{N}(y)$ induces a complete bipartite graph.

Bit: A single digit in a binary (0/1) representation of an object.

Block: In a partition of a set, each subset is called a block.

Bounded substitution diameter decomposition: A graph has substitution decomposition diameter k if every prime node in the decomposition tree has at most k children. If a class of graphs has substitution decomposition diameter at most k for k fixed, then the class has bounded substitution decomposition diameter.

Bounded tolerance graph: Tolerance graph such that no tolerance value is larger than the length of the interval. This implies that if an interval corresponding to vertex x contains the interval corresponding to y, the vertices must be adjacent.

Boxicity: The minimum dimension d such that a graph can be represented as an intersection graph of d-dimensional rectangles, with all rectangles oriented along the axes. Boxicity 1 graphs correspond to interval graphs.

Brittle graph: A graph G is brittle if there is an elimination scheme for G which successively removes either a vertex which is not a midpoint of any P_4, or not an endpoint of any P_4.

C_i: A chordless cycle on i vertices.

Certificate: A certificate of a property is a proof that the property holds. Certificates are important in the theory of NP-completeness; NP can be defined in terms of existence of polynomial size certificates. Certificates are used in a number of ways in this book; in algorithms, the existence of a certificate of size $\mathrm{f}(n))$ can be a separate problem from finding an $\mathrm{O}(\mathrm{f}(n)$ algorithm to solve the problem, and our discussion of robust algorithms deals with issues which arise if we are given a certificate that a graph is in a class as opposed to when such a certificate is not given.

Chain graph: A bipartite graph with the property that for each pair of vertices u, v from the same color class, either $\mathrm{N}(u) \subseteq \mathrm{N}(v)$ or $\mathrm{N}(v) \subseteq \mathrm{N}(u)$.

Child: In a rooted tree, children of v are neighbors of v other than the parent of v.

Chord: In a cycle, an edge which goes between vertices which are not consecutive on the cycle. On a circle, a straight line connecting two points on the circle.

Chordal graph: A graph with no induced (i.e. chordless) cycle of length > 3.

Chordal bipartite graph: A bipartite graph with no chordless cycles of length greater than 4. An important characterization is that G is chordal bipartite if and only if the bipartite adjacency matrix has a Γ-free ordering.

Chromatic number: The minimum number of colors necessary to color a graph.

Circle graph: Intersection graph of chords of a circle.

Circle order: Containment graph of disks in the plane.

Circuit: Alternative name for cycle.

Circular-arc graph: Intersection graph of arcs of a circle.

Circular permutation graph: Intersection graphs of curves connecting points on two concentric circles of different diameter, such that no pair of curves intersects at more than one point.

Circular 1s property: A circular ordering of a set, which obeys a set of restrictions requiring particular subsets to appear consecutively in the circular order.

Classically NP-complete: A problem P is classically NP-complete on a restricted domain D of inputs (such as a class of graphs) if every problem in NP can be mapped in polynomial time and space to an instance of P from D, such that the answer to the original problem is the same as the answer to the instance mapped to.

Claw-free graph: A graph with no induced subgraph equal to $K_{1,3}$.

Clique: A set of mutually adjacent vertices.

Clique cutset: A cutset which is also a clique.

Clique separator: Alternate name for clique cutset.

Clique separator decomposition: A form of graph decomposition which takes an arbitrary clique cutset C, decomposes the graph into $S \cup C$ for each connected component S of $G - C$, and decomposes each subgraph recursively.

Clique problem: In decision form, the problem of whether an input graph G has a clique of size equal to input number k. Also used in optimization form to denote the problem of determining the cardinality of the maximum clique in a graph. The decision problem is NP-complete.

Clique tree representation: An intersection model for a chordal graph G. There is a tree T, such each node of T corresponds to a maximal clique of G, and cliques containing each individual vertex v correspond to a connected subtree of T. G has a clique tree representation if and only if G is chordal, and this representation can be found in linear time.

Clique-width: The minimum number of labels needed to construct a graph from one vertex graphs using the operations union, addition of all edges between vertices with label i and vertices with label j, and the relabeling operation which gives label j to all vertices with current label i.

Clone: A clone c of vertex v with respect to a set of vertices S is a vertex such that $N(c) \cap S - v = N(v) \cap S - v$.

Closure: The k-closure of a graph is formed by repeatedly adding edges between nonadjacent pairs x, y of vertices such that degree(x) + degree(y) is at least k, until no further edges can be added. In a variant called k'-closure, edges are added between nonadjacent x, y such that $|N(x) \cup N(y)|$ is at least k.

Co-: For a graph class C, a graph is a co-C graph if the complement is in C.

Co-NP: A decision problem P is in co-NP if the problem of deciding whether the answer to P is no is in NP; i.e., no answers have a polynomial size certificate.

Cograph: In the original definition, a graph which can be built from single vertex graphs using the operations of complement and union. An important characterization is that these are exactly the graphs without any P_4.

Coloring: An assignment of numbers (called colors) to vertices of a graph, such that no edge connects vertices with the same color. The question of whether it is possible to color a graph with k colors is NP-complete even for $k = 3$. In the optimization version of the coloring problem, the goal is to color a graph with the minimum number of different colors.

Color class: In this book, this refers to one of the subsets of vertices used to partition a bipartite graph. More generally, a color class is a set of vertices given the same color by a coloring assignment.

Comparability graph: An undirected graph such that all edges can be assigned directions, such that in the directed graph whenever $a \to b$ and $b \to c$, then $a \to c$.

Complement: The complement of G, written as $\overline{G}$, has the same vertex set as G, and an edge from x to y in $\overline{G}$ if and only if there is no edge from x to y in G.

Complete graph: Alternative name for clique.

Complete bipartite graph. A bipartite graph which has an edge between each pair of vertices from different color classes.

Completely k-decomposable: A graph is completely k-decomposable with respect to a decomposition if every induced subgraph containing more than k vertices k is decomposable.

Completion problem: The completion problem for a class of graphs C asks for the minimum number of edges necessary to add to an input graph G, so that the augmented graph is in C.

Composition sequence: A class of graphs has a composition sequence if there is a countable sequence $G_1, .., G_i, ...$ of graphs in the class such that each G_i is an induced subgraph of G_{i+1}, and every graph in the class is a subgraph of some G_i in the sequence.

Connected component: Maximal connected subgraph.

Connected graph: A graph such that there is a path between each pair of vertices.

Connected matrix: A matrix such that no pair of rows and columns induces I_2.

Consecutive 1s property: An ordering of a set which obeys restrictions requiring elements in various subsets to occur consecutively.

Construction problem: The problem of constructing a form of graph representation.

Containment graph: The containment graph of a set of objects has a vertex corresponding to each objects, and a directed edge from x to y if object x contains object y.

Containment representation: A representation of a graph as a containment graph.

Contraction: The graph formed by contraction of an edge (x, y) replaces the pair of vertices x, y by a new vertex z, with $\mathrm{N}(z) = N(x) \cup N(y)$.

Convex graph: Bipartite graph such that vertices of one color class can be ordered so that for every vertex v from the other color class, $\mathrm{N}(v)$ occurs consecutively in the ordering.

Convex fan: A polygon P in which there is a single endpoint v with the property that every other endpoint w of P is either a neighbor of v on P, or the line segment connecting v and w is entirely inside of P.

Convex vertex: Given a model for a visibility graph, a vertex is convex if it maps to an endpoint corresponding to a polygon angle of less than 180 degrees.

Cook's theorem: The theorem proving the NP-completeness of CNF-satisfiability. Almost all NP-completeness proofs eventually derive from Cook's theorem.

Cotree: A representation of a cograph. Vertices correspond to leaves of a tree, and internal nodes of the tree are labeled with 0 and 1. Two vertices u, v of G are adjacent if and only if the least common ancestor of u and v in the cotree has label 1. G can be represented by a cotree if and only if G is a cograph.

Counting problem: The problem of determining the number of graphs in a class.

Covering graph: The undirected graph formed by removing directions from the transitive reduction of a partial order.

Cowly perfect: Literal translation of one of my favorite names for a graph class, vachement parfait, which unfortunately did not find a place in this book. Those interested in correct names for graph classes could debate whether or not such a class should be bull-free.

Cutset: A set S of vertices such that G - S is disconnected.

Cycle: A sequence of vertices $v_1, v_2, ..., v_k$ such that for i in the range $1..k$-1, v_i is adjacent to v_{i+1}, and v_k is adjacent to v_1.

Cycle-free partial order: A partial order such that the underlying graph is chordal.

Decision problem: A problem such that the answer is either yes or no. The term NP-complete applies by definition to decision problems.

Degree: For the purposes of this book (in which self-loops are never used), the number of edges out of a vertex.

Deletion problem: The deletion problem for a class C of graphs asks for the minimum number of edges necessary to delete from an input graph G so that the resulting graph is in C.

Dense: Having many edges. Often used in an informal sense, but sometimes used to mean number of edges is $\Theta(n^2)$.

Descendant: A vertex d in a rooted tree is a descendant of v if the path from the root to d goes through v.

Deterministically k-decomposable: A graph is deterministically decomposable with respect to a decomposition if for every possible choice of recursive decomposition steps, every prime component has size at most k.

Diameter: The diameter of a graph is the maximum distance between any pair of vertices.

Digraph: Directed graph.

Dilworth Number: The maximum cardinality of a set S of vertices in a graph such that for all pairs of vertices x, y in S, some neighbor of x is not in $\mathrm{N}[y]$ and some neighbor of y is not in $\mathrm{N}[x]$.

Dimension: The minimum number of linear extensions of a partial order which give the partial order as their intersection. In other words, the minimum number of lists of vertices of a partial order P such that $x < y$ if and only if x precedes y in all lists.

Disk intersection graph: Intersection graph of disks in the plane, where a disk denotes a circle plus the interior of a circle; thus, if one disk contains the other, the corresponding vertices are adjacent.

Distance: The distance between x and y is the length of the shortest bath between x and y.

Distance hereditary graph: A graph such that for every pair of vertices x, y, all chordless paths from x to y have the same length. Relevance to this book comes in part from an elimination scheme characterization; a connected graph G is distance hereditary if and only if it can be reduced to a single vertex by repeated elimination of twins or pendant vertices.

Distance labeling: Generalization of implicit representation to the distance function. Each vertex is assigned a label, and the distance between x and y can be computed using only the label information stored at x and y.

Dominating pair: A pair of vertices x, y is a dominating pair if for all vertices z, every path from x to y contains at least one neighbor of z.

Dominating set: A set S of vertices such that every vertex of V - S is adjacent to at least one vertex in S. Deciding whether a graph has a dominating set of size input number k is NP-complete. In the optimization version, it is necessary to find the smallest cardinality dominating set.

Domination graph: A graph G with the property that for every induced subgraph H of G, there is a pair of vertices x, y such that $\mathrm{N}(x) \subseteq \mathrm{N}[y]$ in H.

Dot-product representation: Representation of a graph in which each vertex is associated with a vector, and vertices are adjacent if and only if the dot-product of the corresponding vectors is at least 1. The dot-product dimension is the minimum number of elements in vectors which can be used to represent G. Representing graphs with constant dot-product dimension implicitly is open.

Doubly chordal graph: A graph is doubly chordal if there is an ordering of the vertices $v_1 v_2 .. v_n$ such that each v_i is simplicial in the graph induced by $v_i .. v_n$, and v_i has a maximal neighbor in the graph induced by $v_{i+1} .. v_n$.

Doubly convex graph: Alternate name for biconvex graph.

Dually chordal graph: Graph which can be ordered $v_1 .. v_n$ such that each v_i has a maximal neighbor in the graph induced by $v_i .. v_n$.

Edge: A connection between two vertices. This can be written as an unordered pair or as u - v for an undirected graph, and as an ordered pair or $u \rightarrow v$ for a directed graph.

Efficient representation: This is an informal term used throughout the book. In general, it means that this form of representation has important advantages over standard methods for representing a class of graphs, even though the representation might not be space optimal, and adjacency testing might not take constant time or be distributed to bits stored at vertices.

Elimination scheme: An ordering $v_1 v_2 .. v_n$ of vertices of a graph, such that each vertex v_i satisfies some particular property P in the graph induced by $v_i .. v_n$.

Envelope: The envelope of a set of line segments consists of the union of line subsegments on the infinite face of the union of the segments.

EPT graph: Intersection graph of paths in a tree, where we say two paths intersect if they share a common edge (i.e. two paths which intersect only at a vertex must correspond to nonadjacent vertices).

External visibility graph: A graph in which vertices correspond to endpoints of a polygon, and two vertices are adjacent if the line segment connecting them is entirely outside the polygon.

f-diagram: Structure used in a representation of co-comparability graphs. Vertices correspond to curves connecting points on two parallel lines, with curves intersecting (at least once) if and only if the corresponding vertices are adjacent. This representation is possible if and only if G is a co-comparability graph.

Feedback vertex set: A set S is a feedback vertex set if G - S is acyclic. The feedback vertex set problem asks for the minimum cardinality feedback vertex set; the decision version of feedback vertex set is NP-complete.

Ferrers digraph: A directed graph such that for all pairs x, y, either $\mathrm{N}[x]$ contains $\mathrm{N}(y)$ or $\mathrm{N}[y]$ contains $\mathrm{N}(x)$.

Ferrers dimension: The Ferrers dimension of a directed graph G is the minimum number of Ferrers digraphs which yield G as their intersection.

Fill-in scheme: A method for constructing a graph by repeatedly adding edges to the current graph using some set of rules.

Generalized implicit graph question: The open question as to whether every hereditary class of graphs with $2^{f(n)}$ members has a generalized implicit representation.

Generalized implicit representation: For a graph class with $2^{f(n)}$ graphs on n vertices, a representation which stores $\mathrm{O}(f(n)/n)$ bits at each vertex, such that adjacency between x and y can be tested using only the bits stored at x and y.

Global information: Information representing a graph is global if it can be accessed from more than one specific vertex of the graph.

Graph: An ordered pair (V, E), where V is a set of vertices and E is a set of edges. Edges are unordered pairs of vertices.

Graph class: A (usually infinite) set of graphs.

Graph isomorphism: The problem of determining whether two graphs are isomorphic. No polynomial algorithm for graph isomorphism is known, and graph isomorphism is not known to be NP-complete.

Greedy algorithm: An algorithm which adds to its current solution sequentially according to some criterion, and never backtracks by removing an element from its current solution.

Greedy coloring: The assignment of colors to vertices according to an order, which assigns the next vertex v the smallest color number which has not already been assigned to a neighbor of v. In the context of circular-arc graph coloring, the order corresponds to next arc in the circular-arc model.

Grid: Grid is used in two difference senses in this book. If a geometrically defined class of graphs is defined, placing the model on a grid means assigning integer coordinates to all objects used in the definition. The grid is also a special graph in which endpoints have labels (b, c) for b, c in some ranges $1..i$,$1..j$ and two vertices (b, c) and (d, e) are adjacent if and only if $b = d$ and c differs from e by 1, or $c = e$ and b differs from d by 1. A graph is a grid graph if it is an induced subgraph of the grid.

Grid intersection graph: Intersection graph of a set of line segments in the plane, where each line segment is either horizontal or vertical.

Hamilton cycle: A Hamilton cycle in a graph is a cycle which goes through every vertex exactly once. Also called Hamilton circut. The problem of deciding whether a graph has a Hamilton cycle is NP-complete, as is deciding whether a graph has a similarly defined Hamilton path.

Height: In this book, the height of a poset is the length of a longest path (or size of a maximum chain) in the poset. Thus, we would say a bipartite poset has height 1, while this is called height 2 in some texts.

Helly property: A set has the Helly property if for all subsets S such that each pair of objects in S has a common intersection, then the entire set S has at least one common intersection point. For intersection representations of graphs, this means that all cliques share a common point in the representation.

Hereditary: A class of graphs is hereditary if for every G in the class and every vertex v of G, G - v is also in the class.

Hole: Chordless cycle of length at least 4. This book allows holes to be of even length, while some papers reserve the name for odd length chordless cycles.

Homogeneous Pair: Vertex sets A, B such that $|A| + |B| > 2$, A is a module of $G - B$, and B is a module of $G - A$. Also known as a 2-module.

Homogeneous set: Alternative name for module.

House: The specific graph on 5 vertices consisting of a cycle on 4 vertices, and a 5th vertex adjacent to 2 neighbors in the cycle.

Hypergraph: A hypergraph differs from a graph in that each edge (called a hyperedge) is a subset of any number of vertices, rather than being a subset of exactly 2 vertices.

Identity matrix: A square matrix with 1s on the main diagonal, and 0s everywhere else.

Immediate predecessor/successor: In a partial order, these are vertices which have edges to/from v in the transitive reduction.

Implicit graph conjecture: The conjecture that every hereditary class of graphs which has $2^{O(n \log n)}$ graphs on n vertices must have an implicit representation.

Implicit representation: Graph representation which stores O(log$n$) bits per vertex, such that adjacency between x and y can be tested using only the bits stored at x and y.

Indegree: The number of edges into a vertex of a directed graph.

Independent set: A set of mutually nonadjacent vertices. Determining whether a graph has an independent set of size input parameter k is NP-complete. In the optimization problem, you ask for the maximum cardinality independent set.

Induced subgraph: The subgraph induced by a set S of vertices in a graph G has S as its vertex set, with two vertices adjacent in the induced subgraph if and only if these vertices are adjacent in G.

Induced visibility graph: A graph which is an induced subgraph of some visibility graph.

Intersection class: A graph class which corresponds exactly to intersection graphs of a particular type of object.

Intersection graph: The intersection graph of a set of objects has a vertex corresponding to each object, with x and y adjacent if and only if the corresponding objects have a nonempty intersection.

Interval dimension: The interval dimension of a partial order P is the minimum number of interval orders which yield P as their intersection.

Interval filament graph: The intersection graph of a set of curves connecting endpoints on a fixed line L, where all curves are required to stay above L in the plane, and between the two endpoints.

Interval graph: Intersection graph of intervals on the line.

Interval number: The interval number of G is the smallest i such that G can be represented using i intervals for each vertex, with vertices x and y adjacent if and only if at least one of x's intervals has a nonempty intersection with at least one of y's intervals.

Interval order: A partial order is an interval order if elements can be associated with intervals on the line, such that $x < y$ if and only if the interval associated with x lies entirely to the left of the interval associated with y.

Inversion-free: An inversion-free ordering of a graph is an ordering $v_1 v_2 .. v_n$ of the vertices, such that for all $i < j < k < l$, if there is an edge from v_i to v_k and v_j to v_l, there must be an edge from v_i to v_l. A graph is inversion-free if it admits an inversion-free ordering.

Isomorphic: Two graphs G_1, G_2 are isomorphic if there is a one-to-one, onto mapping f from vertices of G_1 to vertices of G_2, such that x is adjacent to y in G_1 if and only if $f(x)$ is adjacent to $f(y)$ in G_2. Graph isomorphism is a famous open problem, in the sense that it is not known to be either polynomially solvable or NP-complete. A class of graphs is isomorphism-complete if solving the isomorphism problem on the class with a polynomial time algorithm would imply a polynomial time algorithm for general graph isomorphism.

Isooriented: A set of objects whose principal axes are either mutually parallel or perpendicular.

Join decomposition: A form of graph decomposition based on splits; a split is a partition of the vertex set into S_1, S_2 such that both S_1 and S_2 have at least 2 vertices, and edges which go from S_1 to S_2 form a complete bipartite graph. If G has a split S_1, S_2, G is decomposed into $S_1 \cup m$, $S_2 \cup m$, where m is a marker vertex which has edges to exactly those vertices which have neighbors in the other partition class. The two sets $S_1 \cup m$ and $S_2 \cup m$ are then decomposed recursively.

k-tree: A graph which can be constructed from a k-clique by repeatedly adding a vertex adjacent exactly to some k-clique in the current tree.

k-module: A set M of vertices which can be partitioned into at most k subsets $M_1 .. M_k$ such that each M_i is a module in $G - M \cup M_i$.

K_i: A clique on i vertices.

$K_{i,j}$: A complete bipartite graph with i vertices in one color class, and j in the other.

Labeled graph: A labeled graph has names attached to vertices. For example, if we have two graphs, one of which corresponds to the path 1,2,3 and the other to the path 1,3,2, these are different labeled graphs (they have different answers to the question of whether vertex 1 is adjacent to 2), although the two graphs are isomorphic.

Leaf: In an undirected tree, a vertex of degree 1. In a rooted tree, a vertex with no children.

Lexicographic breadth first search: An important restriction of breadth first search, designed originally for recognizing chordal graphs. Initially, all vertices are placed in a single set. The algorithm repeatedly removes a vertex v from the last current set, and divides all subsets into neighbors of v and nonneighbors of v, placing the subset of nonneighbors immediately before the subset of neighbors.

Line graph: The line graph of G has a vertex for each edge of G, and an edge between two vertices of the line graph if and only if the corresponding edges have a common endpoint. H is a line graph if and only if H is the line graph of some graph G.

Line-of-sight graph: Given a set of objects, a line-of-sight graph has a vertex for each object and an edge between vertices if some line segment between these objects avoids all other objects. In some cases, restrictions are placed on the segments, such as requiring the line segment to be horizontal.

Linear extension: For a partial order P, a total ordering T of elements of P such that whenever $x < y$ in P, $x < y$ in T.

Linear matrix: A 0/1 matrix which does not contain any induced 2×2 induced submatrix with all values equal to 1.

Linear time: Time proportional to the size of the input plus the size of the output.

Linked list: A fundamental data structure in computer science. If items are stored in a linked list, reaching the ith item in the list requires stepping through each of the i-1 preceding items in the list.

Literal: In a satisfiability problem, a literal is either x or $\overline{x}$ for some variable x.

Local complementation: Local complementation of G at v changes a graph by replacing the subgraph induced by neighbors of v with its complement.

Local information: Information associated with specific vertices of a graph.

Local Structure: Alternative name for implicit representation.

log: $\log x$ is defined (in the tradition of computer science) to be the base 2 logarithm of x.

Matching: A set of edges is a matching if each vertex is an endpoint of at most one edge in the set. Computing a maximum cardinality matching can be done in polynomial time, but finding an $O(n^2)$ algorithm for matching would be considered an extraordinary achievement, so that if you can show a problem is as hard as maximum matching you have given some evidence of the difficulty of finding a linear time algorithm.

Maximal: A set S is maximal with respect to a property if there is no set S' which properly contains S and has the property.

Maximal neighbor: A maximal neighbor of v is a neighbor x of v such that for all neighbors w of v, $N(w)$ is a subset of $N[x]$.

Maximum: A set S is maximum with respect to a property if there is no set with larger cardinality which has the property.

Maximum cut: An NP-complete graph problem, asking whether there is a partition V_1, V_2 of the vertices such that at least k edges have one endpoint in V_1 and the other endpoint in V_2. Also known as max cut and simple max cut.

Minimal: A set S is minimal with respect to a property if there is no proper subset of S which has the property.

Minimal fill problem: The problem of finding a set S of edges to add to G so that $G \cup S$ is chordal, and there is no proper subset S' of S such that $G \cup S'$ is chordal. A set S of edges is a minimal fill set if and only if every edge of S is the unique chord in some 4-cycle of $G \cup S$.

Minimal separator: Unfortunately, this term has several different meanings in the field. In this book, a set S is a minimal separator if there is some pair of vertices a, b such that a and b are in different components of $G - S$, and for every proper subset S' of S, a and b are in the same component of $G - S$. Note that this is not identical to a cutset S such that no proper subset of S is a cutset, which is an alternative usage of the term minimal separator.

Minimum: A set S is minimum with respect to a property if there is no set which has smaller cardinality with the property. Every minimum set is minimal for a property, but a set can be minimal and not minimum.

Minimum fill problem: The problem of minimizing the number of edges which can be added to G so that the graph is chordal. The decision version is NP-complete.

Model: Many of the graph classes discussed in this book are described in terms of relations (intersection, containment, visibility, and others) between a set of objects. In such cases, we say that we are given a model if we are given the objects, rather than being given the graph in a standard format such as adjacency lists or an adjacency matrix.

Modular decomposition: Alternate name for substitution decomposition.

Module: A set M of vertices such that for every vertex v of V - M, v is either adjacent to every vertex of M, or v is adjacent to no vertex of M.

N(v), N[v]: $\mathrm{N}(v)$ is the set of vertices adjacent to v, and $\mathrm{N}[v]$ includes both v and $\mathrm{N}(v)$.

N-free poset: A partial order such that the transitive reduction contains no induced N, where an N is a set of four vertices u, v, w, x with edges (w, u), (x, u), (x, v), forming a shape roughly equal to the letter N if u and v are placed on top with w and x on bottom.

Neighborhood: The set of vertices which can be reach from a vertex using a single edge.

NLC-width: The NLC-width of a graph G is the minimum number k of labels which can be used to construct G from single vertex labeled graphs using the union and relabel operations. The relabel operation assigns all vertices with current label i the label j. The union operation takes two disjoint graphs G_1,G_2 and a bipartite graph where each color class contains a vertex for each label, creating a single graph by adding an edge between vertex i_1 from G_1 and vertex j_2 from G_2 if there is an edge of the bipartite graph between the copy of the label of i_1 from color class 1 and the label of j_2 from color class 2.

Node: Alternative name for vertex, used especially in trees.

Nondetermistically k-decomposable: A graph is nondeterministically k-decomposable with respect to a decomposition if there exists a sequence of decomposition steps which decomposes the graph into prime components of size at most k.

NP: A decision problem is in NP if there is a polynomial size certificate for every instance of the problem for which the answer is yes.

NP-complete: A problem X is NP-complete if it is in NP, and every problem in NP can be polynomially transformed to X. This is a large topic which cannot be treated fully in this book; readers unfamiliar with the term are advised to consult [206] for more information. In general, NP-completeness of a problem is used as strong evidence of the intractability of the problem; at the least, a polynomial algorithm to solve an NP-complete problem would require an enormous breakthrough.

o: $\mathrm{f}(n)$ is $\mathrm{o}(\mathrm{g}(n))$ if for every constant c there exists a constant n_0 such that for all $n > n_0$, $f(n) < cg(n)$. Intuitively, this means that $\mathrm{f}(n)$ grows more slowly than $\mathrm{g}(n)$ even if constants are ignored. For the purposes of this book, the simpler definition that $\mathrm{f}(n)$ is $\mathrm{o}(\mathrm{g}(n))$ whenever $\mathrm{f}(n)$ is $\mathrm{O}(\mathrm{g}(n))$ but $\mathrm{g}(n)$ is not $\mathrm{O}(\mathrm{f}(n))$ will be equivalent for all functions studied here, though the definitions are not equivalent in general. A typical use is posing the question of finding an $\mathrm{o}(\mathrm{f}(n))$ algorithm to solve a problem, where the fastest known algorithm has running time $\Theta(\mathrm{f}(n))$.

O: $\mathrm{f}(n)$ is $\mathrm{O}(\mathrm{g}(n))$ if there exist constants c, n_0 such that for all $n \geq n_0, f(n) \leq cg(n)$. Intuitively, $\mathrm{g}(n)$ grows at least as quickly as $\mathrm{f}(n)$, if constant factors are ignored.

Optimization problem: A problem in which the goal is to find a maximum or minimum set. Many problems have a decision (yes/no) or optimization version of the problem.

Order notation: Mathematical notation designed to compare growth rates of functions, while ignoring constant factors. In this book, we will use the symbols o, O, Ω, and Θ, which correspond to $<, \leq, \geq, =$ if we ignore constant factors.

Ordered chordal graph: Given a graph G and an ordering O of the vertices, G is ordered chordal if there is no chordless cycle C of length at least 4 such that vertices occur in the same order in C as in O.

Orientation: Assignment of directions to edges of an undirected graph.

Outdegree: The number of edges out of a vertex in a directed graph.

Outerplanar graph: A graph with a planar embedding such that every vertex is on the outer face.

Overlap graph: The overlap graph of a set of objects has a vertex for each object, and an edge between vertices if and only if the objects have a nonempty intersection, and neither object contains the other.

P_i: A chordless path on i vertices.

Parallel module: A module which induces a disconnected graph.

Parent: In a rooted tree, the next vertex on the path to the root is called the parent.

Partial k-tree: A (not necessarily induced) subgraph of a k-tree.

Partially complemented representation: A form of representation in which for each vertex v, you are given either a list of neighbors of v, or a list of nonneighbors of v.

Partial order: Normally, a partial order on a set of elements is a relation which is reflexive, transitive, and antisymmetric. Since this book deals graphs without self-loops, the terms partial order and transitive graph are sometimes viewed as equivalent.

Partition: Division of a set into subsets, such that each element is in exactly one subset. The subsets are often referred to as blocks of the partition.

Partitioning: In general, the process of subdividing blocks of a partition based on a particular rule. In this book, we focus on partitions which correspond to subsets of the vertices of a graph, and a subdivision rule which divides blocks into neighbors and nonneighbors of some vertex x which is not in the block.

Path: A sequence of vertices $v_1, v_2, .., v_n$ such that each v_i is adjacent to v_{i+1}.

Path graph: Intersection graphs of undirected paths in undirected trees. Differs from EPT graphs in the paths which share a common tree node but no common tree edge correspond to adjacent vertices in a path graph, and nonadjacent vertices in an EPT graph.

Path orderable graph: A graph which admits an ordering $v_1 v_2 .. v_n$ such that for all sets of 3 independent vertices with $v_i < v_j < v_k$ in the ordering, every path from v_i to v_k goes through at least one neighbor of v_j.

Pathwidth: The pathwidth problem tries to minimize the size of the maximum clique over all interval completions of a graph. Determining whether a G has an interval completion with maximum clique size k for input variable k is NP-complete.

Pendant vertex: A vertex of degree 1.

Perfect elimination ordering/scheme: These terms, (perfect elimination scheme and perfect elimination ordering are used interchangeably) refer to an elimination scheme defined by successive removal of a simplicial vertex. A graph is chordal if and only if it has a perfect elimination scheme.

Perfect elimination bipartite graph: A graph in which all edges can be removed by repeatedly removing both endpoints of a bisimplicial edge.

Perfect graph: A graph G in which for every induced subgraph H of G, the chromatic number of H is equal to the size of the largest clique in H.

Perfect matching: A matching is perfect if every vertex is an endpoint of some edge in the matching.

Perfectly orderable graph: A graph G which has an ordering O with the property that for every induced subgraph H of G, greedy coloring of H which colors vertices in the order which the vertices appear in O produces an optimal coloring.

Permutation diagram: Intersection model (as described below) of a permutation graph.

Permutation graph: Intersection graph of line segments, where each line segment connects a point on line L_1 with a point on the parallel line L_2.

Phylogeny problem: The problem of finding the tree describing the evolution of distinct species from a common ancestor species. In the perfect phylogeny problem, distinct species are assigned sets of attributes, and you want an evolutionary tree such that each internal node is given a set of attributes, and the set of nodes with a given attribute form a connected subtree of the evolutionary tree.

PI graph: Intersection graphs of triangles such that each triangle has one endpoint on line L_1, and two endpoints on the parallel line L_2.

PI graph*: Intersection graph of triangles such that all endpoints are on two parallel lines L_1, L_2. Differs from PI graphs in that some triangles can have two endpoints on L_1, while others have two endpoints on L_2.

Planar graph: A graph which can be drawn in the plane such that no pair of edges crosses each other.

Polygon-circle graph: Intersection graph of a set of polygons embedded in a circle.

Poset: Alternate name for partial order.

Postorder: A postorder traversal of a tree visits all children of a node before visiting the node itself.

PQ-tree: A tree which is used to represent possible orderings of a set; elements of the set are stored as leaves of the tree. The set of leaf descendants of a particular internal node i must occur consecutively in the ordering. Internal nodes are labeled either P or Q; children of a P node can be permuted in any order with respect to the final output list, while children of a Q node are assigned an ordering, and must appear in either this order or in reverse order in the output list.

Prime: Indecomposable with respect to a particular form of graph decomposition.

Probe interval graph: A graph in which each vertex corresponds to an interval, and each vertex is labeled either as a probe or nonprobe. Two vertices are adjacent if the corresponding intervals have a nonempty intersection, and at least one of the vertices is a probe. These arose from an application in computational biology.

Projective plane: Although the theory of projective planes is quite a broad subject, they are used in only one way in this book. We use the fact that for infinitely many values n, it is possible to create a 0/1 matrix M with n rows and n columns, such that each row has $\Omega(n^{1/2})$ 1 entries, and each pair of rows r_1, r_2 has exactly 1 common column c such that $M[r_1, c] = M[r_2, c] = 1$.

Proper circular-arc graph: Intersection graph of a set of arcs on a circle, such that no arc contains another arc in the set.

Proper interval graph: Intersection graph of a set of intervals on the line, such that no interval contains any other interval. The class is equivalent to unit interval graphs.

Pruning sequence: An elimination scheme defined by successive removal of pendant vertices or one vertex from a pair of twins. G has a pruning sequence if and only if G is distance hereditary.

Pseudoline: Pseudolines connect points in the plane via curves, and have the property that each pair of pseudolines cross each other at most once.

Queue: A data structure which allows the operations insertion and removal of the vertex which has been in the queue for the longest amount of time.

Quotient graph: The quotient graph of a prime module M with respect to substitution decomposition has one vertex for each maximal proper submodule of M, with two vertices adjacent in the quotient graph if there are edges between the corresponding maximal submodules in the original graph.

Random graph: For this book, a random graph is the result of choosing each edge to be in the graph or not in the graph with equal probability.

Realizer: A set of linear extensions of a partial order P which produce P as their common intersection.

Recognition problem: The problem of determining whether a graph is a member of a class of graphs.

Reconstructible vertex: A vertex is reconstructible from a set of information if adjacencies to all other vertices can be deduced from this information. This is used in a more specific manner when dealing with AT-free co-AT-free graphs, where we specify a set of rules which allow the determination of adjacencies to a vertex.

Rectangle number: The rectangle number of G is the minimum number k such that every vertex can be associated with k isooriented rectangles, and x is adjacent to y if and only if some rectangle associated with x has a nonempty intersection with some rectangle associate with y.

Rectangle overlap graph: Intersection graph of isooriented rectangular regions in the plane. We note that a rectangle is considered as a region of space, so that if one rectangle contains another they are considered to have a nonempty intersection.

Reduced matrix: The reduced M' matrix of a Γ-free matrix m is a $0/1$ matrix such that 1 entries of M' correspond to entries of M which are 1, and are not redundant 1 entries.

Redundant 1: A redundant 1 in a Γ-free matrix is a 1 entry which would create a Γ if the entry was changed to 0. These 1s are called redundant, since it is unnecessary to store the positions of redundant 1s in our representation of Γ-free matrices.

Refinement: The process of taking a partition of a set, and getting a new partition by dividing sets into subsets.

Representation problem: The problem of finding a form of representation for a class of graphs which satisfies some condition (e.g finding an implicit representation, or finding a space-optimal representation which allows constant time adjacency testing). This is separate from the construction problem, in that it deals only with the existence of such a representation rather than construction of representations for individual graphs.

Representative graph: Alternate name for quotient graph.

Robust algorithm: A robust algorithm for solving a problem P on an input class C is required to always answer P correctly when the input is in class C, and when the input is not in C may either answer P correctly or answer that the input is not in C.

Rooted tree: A rooted tree is a directed graph, which has an underlying tree and a special vertex r called the root. Directions are assigned so that every vertex of the tree is reachable via a directed path from r.

Sandwich problem: Given a set of edges E_1 and a set of edges E_2 the sandwich problem for a class of graphs asks whether there is any graph G in the class such that G contains every edge from E_1, and every edge in G comes from E_2.

Satisfiability: The problem of determining whether a Boolean formula given in conjunctive normal form has a truth assignment such that the entire formula is evaluated as true. This was the first problem shown to be NP-complete.

Semiorder: A partial order such that each element e can be mapped to a number $\mathrm{w}(e)$, and there is a fixed threshold t such that $x < y$ if and only if $w(y) - w(x) > t$.

Separator: Alternate name for cutset.

Series module: A module which induces a subgraph M such that the complement of M is disconnected.

Series-parallel graph: Although there are slightly different definitions co-existing in the literature, in this book a graph is series-parallel if all edges can be removed by a sequence of removal of degree 1 vertices, and removal of degree 2 vertices v with neighbors w, x followed by addition of an edge w, x if such an edge does not exist already.

Series-parallel poset: A partial order which can be constructed from single element partial orders using the operations disjoint union and join, where join makes every element of $P_1 <$ every element of P_2. Equivalent to partial orders with underlying cographs.

Sibling: Vertices in a rooted tree which have a common parent.

Sign pattern: A sign pattern for a sequence $p_1, , .., p_k$ of polynomial functions is a sequence $s_1, .., s_k$ of 1s and -1s such that there is some assignment to variables which makes every p_i positive if and only if $s_i = 1$.

Simple matrix: A matrix such that no pair of columns is identical.

Simple vertex: A vertex v such that neighbors of v induce a clique, and for each pair of neighbors w, x of v either $\mathrm{N}(v) \subseteq \mathrm{N}[w]$ or $\mathrm{N}(w) \subseteq \mathrm{N}[v]$.

Simplicial vertex: A vertex v such that $\mathrm{N}[v]$ induces a clique.

Sink: A vertex in a directed graph or partial order which has no outedges.

Skew partition: A skew partition of a graph is a partition of the vertex set into four nonempty subset A, B, C, D such that there is an edge from every vertex of A to every vertex of B, and there is no edge from any vertex of C to any vertex of D. Recurrsive decomposition using skew partition finds a skew partition A, B, C, D, and recursively decomposes the subgraphs induced by A, B, C, by A, B, D, by A, C, D, and by B, C, D.

Source: A vertex in a directed graph or partial order which has no inedges.

Space optimal: A representation of a graph class which has $f(n)$ graphs on n vertices is space optimal if it uses $O(\log(f(n))$ bits to store each graph in the class.

Sparse graph: A graph with a smaller number of edges. The term is informal; in some cases sparse may mean $o(n^2)$, in other cases it could be as restrictive as $O(n)$.

Sphericity: The sphericity of G is the smallest dimension such that G can be represented as the intersection graph of d-dimensional spheres. Every graph is the intersection graph of spheres in some dimension [360]; this contrasts with the fact that there are partial orders which are not containment graphs of spheres in any dimension. [187]

Spider graph: Alternate name for polygon-circle graph.

Split: A partition of vertices of a graph into S_1, S_2 is a split if it meets the following conditions. 1) S_1 and S_2 have at least two vertices. 2) Let S_{1in} of vertices in S_1 with at least one neighbor in S_2, and let S_{2in} be the set of vertices in S_2 with at least one neighbor in S_1. Every vertex of S_{1in} must be adjacent to every vertex of S_{2in}. Note that this is a special case of the original definition of split in [138, 139], which was defined for directed graphs as well as undirected graphs.

Split decomposition: Alternate name for join decomposition.

Split graph: G is a split graph if the vertices of G can be partitioned into S, K, where S induces an independent set in G and K induces a clique in G; i.e., G can be obtained from a bipartite graph by making one color class a clique. Split graphs are exactly those graphs which are both chordal and co-chordal.

Stable set: Alternate name for independent set.

Stack: A data structure which allows the operations insert, and remove the object which has been in the set for the shortest period of time.

Star: $K_{1,i}$ for any value of $i > 1$.

Steiner tree: The Steiner tree problem takes a graph G and a subset S of vertices as input, and asks for the minimum subtree T of G which connects all vertices in S. The decision version of the Steiner tree problem is NP-complete, even in the unweighted case.

Star cutset: A star cutset of a graph is a set C such that G - C is disconnected, and some vertex c in C is adjacent to every other vertex of C.

String graph: Intersection graph of curves in the plane.

Strong perfect graph conjecture: Famous problem in graph theory, stating that G is perfect if and only if G is a Berge graph. A proof of the conjecture has just been announced. [102]

Strongly chordal graph: Although properly speaking this is a characterization rather than the original definition of the class, we will say a graph is strongly chordal if it has an elimination scheme defined by successive removal of simple vertices.

Strongly connected graph: A directed graph in which there is a path from every vertex to every other vertex.

Substitution decomposition: A form of graph decomposition which recursively decomposes a graph into connected components if G is disconnected, connected components of $\overline{G}$ if $\overline{G}$ is disconnected, and maximal proper submodules of G if both G and $\overline{G}$ are connected.

Threshold graph: A graph in which each vertex v van be assigned a weight $\mathrm{w}(v)$ and two vertices x, y are adjacent if $w(x) + w(y) > t$ for some fixed threshold t.

Tolerance graph: A graph in which every vertex can be assigned both an interval and a tolerance, such that x and y are adjacent if and only if the intersection of the intervals exceeds the minimum of the two tolerances. Interval graphs correspond to tolerance graphs with tolerance 0. No implicit representation is known for this class.

Topological sort: An ordering of vertices of a directed acyclic graph, such that for every directed edge (x, y), x precedes y in the ordering.

Total interval number: The total interval number of a graph G is the smallest number of intervals such that each interval is associated with exactly one vertex (a single vertex can be associated with multiple intervals), and x is adjacent to y if and only if some interval associated with x has an intersection with some interval associated with y.

Totally balanced: A hypergraph is totally balanced if every cycle of length greater than 2 has an edge containing at least 3 vertices of the cycle. A well-known theorem is that a hypergraph is totally balanced if and only if every subhypergraph is a hypertree.

Transitive closure: Given a directed graph G, the transitive closure of G is a graph on the same vertex set, with an edge from x to y in the transitive closure if and only if there is a path from x to y in G.

Transitive graph: A directed graph G such that whenever there are edges (x, y) and (y, z), (x, z) is also an edge.

Transitive orientation: An assignment of directions to edges of a comparability graph, such that the resulting direction of edges is transitive.

Transitive reduction: Given a directed acyclic graph G, the transitive reduction of G is a graph on the same vertex set, with an edge from x to y if x has an edge to y in G, and there is no path of length greater than 1 from x to y in G.

Trapezoid graph: Intersection graph of trapezoids, where each trapezoid has two endpoints on a line L_1 and two endpoints on a parallel line L_2. Note that L_1 and L_2 are the same for all trapezoidal objects.

Tree: A connected acyclic graph.

Treewidth: The original definition of treewidth deals with a form of tree decomposition. For this book, it is easier to understand a characterization; G has treewidth k if and only if G is a partial k-tree. The fact that minor closed classes which do not contain all planar graphs must have bounded treewidth has deep consequences; see [412]. Determining the treewidth of a graph is NP-complete, although determining whether a graph has treewidth k is polynomial for fixed k. The treewidth problem can also be viewed as minimizing the maximum clique size over all chordal completions of a graph.

Triangle-extendible: A graph is triangle-extendible if the vertices can be ordered so that for every triangle $u < v < w$, every vertex which comes after w and is adjacent to both v and w is also adjacent to u; in other words, for every K_4 - a single edge, the nonadjacent vertices are not both first and last in the ordering.

Twins: A pair of vertices x, y such that $\mathrm{N}(x) = \mathrm{N}(y)$ if x and y are nonadjacent, and $\mathrm{N}[x] = \mathrm{N}[y]$ if x and y are adjacent.

Two-pair: A pair of vertices x, y such that every chordless path between x and y has length 2. Equivalently for a connected graph, removing $\mathrm{N}(x) \cap \mathrm{N}(y)$ disconnects x and y.

Underlying graph: The underlying graph of a directed graph or poset is the graph formed by removing directions from edges.

Uniform cost assumption: The assumption that arithmetic operations involving numbers of polynomial size takes constant time. This is a standard simplifying assumption made in analysis of algorithms.

Unit circular-arc graph: Intersection graphs of arcs with identical lengths in the circle.
Unit disk graph: Intersection graph of disks with equal diameters in the plane.

Unit interval graph: Intersection graphs of intervals of identical size in the line. These are equivalent to proper interval graphs.

Universal graph: A graph G is (vertex-induced) universal for a class C of graphs if every graph in C is an induced subgraph of G. The concept of a universal graph in the non-induced sense also exists in the literature, but does not apply to this book.

Universal vertex: A vertex which is adjacent to every other vertex in the graph.

Vertex cover: A set S of vertices such that every edge of the graph has at least one endpoint in S. Deciding whether G has a vertex cover of input size k is NP-complete. In the optimization version, the problem asks for the size of the smallest vertex cover.

Vertex expansion: The operation of replacing vertex v by a clique C, each member of which has the same adjacency to other vertices of the graph as v.

Vertex multiplication: The operation of replacing vertex v by an independent set I, each member of which has the same adjacency to other vertices of the graph as v.

Visibility graph: A graph in which vertices can be mapped to corners of a polygon, such that vertices are adjacent if they are either neighbors on the polygon, or the line segment connecting the two vertices is entirely inside the polygon.

von Emde Boas trees: A data structure which allows insertion, deletion, and search for predecessor in the current set over a set with elements in the range $1..n$. The trees use $\mathrm{O}(\mathrm{loglog} n)$ time per operation rather than the $\mathrm{O}(\mathrm{log} n)$ time per operation of such structures as 2-3-trees designed to handle these operations on sets with no upper bound on range.

Warren's theorem: This important theorem, discussed in chapter 4, was designed to give an upper bound on the number of sign patterns of a set of polynomials. In this book, it is used to give an upper bound on number of graphs in a class of graphs. For various classes of graphs, vertices can be made to correspond with sets of variables, and adjacency between vertices x and y is determined by the sign of a polynomial equation on the variables corresponding to x, y, and possibly other vertex variables.

Weak order: A partial order in which every element v can be assigned a number $\mathrm{w}(v)$, and $x < y$ if $\mathrm{w}(x) < \mathrm{w}(y)$. In other words, these are orders which can be derived from total orders by substituting independent sets for elements.

Weakly chordal graph: A graph G with no induced cycles of length greater than 4 in either G or $\overline{G}$.

Weakly triangulated graph: Alternate name for weakly chordal graph.

Well-covered graph: A graph in which all maximal independent sets have the same cardinality.

Width: The width of a partial order is the maximum independent set (also known as antichain in this context) in the underlying graph.

0/1 matrix multiplication: The result of performing matrix multiplication $A \times B$ for matrices A and B with all entries equal to 0 or 1, and replacing all nonzero entries of the result by 1s.

α: If A and B are n by n matrices, the time of the fastest matrix multiplication algorithm for computing $A \times B$ known is written as $O(n^\alpha)$. When this book was written, α was approximately 2.376.

Γ: A pair of rows $r_1 < r_2$ and columns $c_1 < c_2$ in a 0/1 matrix, such that the submatrix induced by these columns has 1s in every entry except position r_2, c_2.

Ω: Intuitively, $f(n)$ is $\Omega(g(n))$ means that $f(n)$ grows at least as quickly as $g(n)$, if constants are ignored. There are several differing precise definitions, which are not equivalent in general but are equivalent for all functions in this book. My preferred definition is that $f(n)$ is $\Omega(g(n))$ if there is a constant c such that $f(n) \geq g(n)$ for infinitely many values of n, but for this book it is sufficient to say that $f(n)$ is $\Omega(g(n))$ whenever $g(n)$ is $O(f(n))$.

Θ: $f(n)$ is $\Theta(g(n))$ if $f(n)$ is $O(g(n))$ and $g(n)$ is $O(f(n))$. This is our notion of equality in order notation, i.e. if constants are ignored.

2-join: Fundamentally, a partition of the vertex set such that edges between the set can be divided into two complete bipartite graphs. There are several variants, depending on whether these complete bipartite graphs are allowed to share vertices, and on conditions for ruling out trivial partitions. Thus, there are definitions which require only that each set contains at least 3 vertices, while others do not allow one side of the partition to be an induced path. Some definitions also require a path within each side of the partition connecting the different complete bipartite graphs.

This page intentionally left blank

Survey of Results on Graph Classes

This chapter was suggested by one of the referees, who noted that many algorithms for dealing with specific classes of graphs are discussed in the book, but that these are difficult for a reader interested in a single class of graphs to discover. The number of graph classes and results on these classes are truly overwhelming, so rather than try to organize these in the form of a table with graph classes on one side and all possible optimization problems on the other, I have chosen to give a very brief sketch of what I believe to be some of the most important results and open problems in algorithms and representation for a number of classes of graphs.

An algorithmic survey on variations of the dominating set problem is given in [329]; I will restrict my treatment here to the basic dominating set problem unless there is a particular reason to cover other domination problems. Let me note that proving NP-completeness of minimum fill was a longstanding open problem before being shown in [491]. There is far less work showing that minimum fill is NP-complete for a class than is the case for other problems, given the difficulty of showing NP-completeness at all; if there is no comment on the fill problem, this generally means that there is no polynomial time algorithm to find the minimum fill on this class of graphs, and the class does not seem to be near the border of what we know to be polynomially solvable.

There is an extensive body of literature dealing with parallel algorithms on these graph classes; since my area of research is sequential algorithms, I will leave a survey on parallel algorithms to some other author. As opposed to algorithms discussed in the rest of the text, I did not try to verify that all the algorithms listed here are correct.

There are many important resources available for finding algorithms on graph classes. The book [74] has a table dealing with best time complexities for recognizing a much larger set of graph classes than is discussed here; corrections to this table are currently kept at

`www.informatik.uni-rostock.de/~ab/survey/errata.html.`

At the same site, a very nice program for questions of inclusions between classes of graphs can be found at

`www.informatik.uni-rostock.de/~gdb/isgci/isgci.html.`

A table of graph class inclusions and counterexamples specializing in subclasses of perfect graphs is available on-line at

`www.informatik.hu-berlin.de/~hougardy/paper/classes.html.`

A survey dealing with polynomial algorithms for some fundamental NP-complete optimization problems on various classes of graphs appeared in [282], and some updates appeared in [283]. An ambitious attempt to maintain an on-line reference list of best known algorithms for problems on graph classes was attempted and some partial results can be seen at

`http://web.cs.ualberta.ca/~stewart/GRAPH/index.html.`

This project became much too large for a single person to maintain; the current plan is to distribute the work to individuals with good knowledge of a particular class of graphs. If you are interested in helping with this project, it would be a great help to all of us!

AT-FREE GRAPHS: There are $2^{\Theta(n^2)}$ graphs without asteroidal triples, so representation is not an issue. A straightforward algorithm can recognize AT-free graphs in $O(n^3)$ time. Using fast matrix multiplication as a subroutine, this can be reduced to $O(n^{2.79})$ time [332]. It will be difficult to reduce the cost to less than the cost of a single matrix multiplication, since the problem is as hard as recognizing triangle-free graphs, as discussed in the text. Polynomial algorithms on asteroidal triple-free graphs include an $O(n^3)$ algorithm for weighted independent set [316], based on the algorithm of [81], an $O(n^3)$ Steiner tree algorithm [26], and an $O(n^6)$ unweighted dominating set algorithm [330]; weighted dominating set is NP-complete on the subclass of co-comparability graphs. Clique and clique cover are NP-complete on this class [81]; these problems separate AT-free graphs in complexity from co-comparability graphs. Treewidth and minimum fill are also NP-complete on AT-free graphs, as a consequence of the original NP-completeness reduction for these problems [24, 491], but are polynomially solvable for the subclass of AT-free co-AT-free graphs [316]. Bandwidth is NP-complete for the subclass of co-bipartite graphs; an approximation algorithm for AT-free graphs graphs with performance ratio 2 is given in [308]. Max cut is also NP-complete for co-bipartite graphs [47], and co-bipartite graphs are isomorphism-complete. Chromatic number and Hamilton cycle are open for AT-free graphs at this time. Feedback vertex set, which is polynomial for co-comparability graphs, does not seem to have been studied for AT-free graphs. A very simple linear time algorithm for finding a dominating pair in an AT-free graph is given in [125]; finding a robust linear time algorithm for this problem is open. Finding a simple intersection representation which exactly characterizes the class is an open problem.

BOUNDED TOLERANCE GRAPHS: The problem of recognizing bounded tolerance graphs in polynomial time is open. An implicit representation can be constructed in polynomial time, thanks to the fact that every bounded tolerance graph is a trapezoid graph. Although most algorithms on these graphs are based upon their containment in larger graph classes, several papers have been written solving problems on bounded tolerance graphs if the model is given as input. There are papers for solving the dominating set and Hamilton cycle problems on bounded tolerance graphs more efficiently ($O(n^2)$ for dominating set and $O(n\log n)$ for Hamilton cycle) than on the larger class of co-comparability graphs if a model is given [3, 4], and a paper giving an $O(n^3 log^2 n)$ algorithm for dominating set in the complement of bounded tolerance graphs [296]. The last paper mentioned is an example of a problem which can be solved efficiently if a model is given as input, but is open if the input is given in adjacency matrix form. I know of no NP-completeness results specifically for bounded tolerance graphs; thus, we know only results implied from containment of such classes as interval graphs and permutation graphs. Isomorphism is open for bounded tolerance graphs, both with the model given and when it is not.

BOXICITY: Recognition of boxicity k graphs for $k > 1$ is NP-complete. Boxicity k graphs have an implicit representation by definition, but finding an implicit representation in polynomial time is open. Clique can be solved in polynomial time (O($n$log$n$) time given the model for boxicity 2 [276]) on this class even if the model is not given, since the number of maximal cliques is polynomially bounded. Clique cover [200], 3-coloring [337] and independent set [408] have been shown to be NP-complete on boxicity 2 graphs. Containment of grid graphs and trees shows that Hamilton cycle, dominating set, Steiner tree, and bandwidth are NP-complete for boxicity 2 graphs. An O(log$n$) approximation algorithm for independent set on boxicity graphs is given in [6]; [174] notes that this achieves the same approximation ratio for weighted independent set. Feedback vertex set is NP-complete on graphs with bounded boxicity, since every planar graph has boxicity at most 3 [458]; this may be open for boxicity 2 graphs. As far as I could determine, the complexity of such problems as max cut, isomorphism, and treewidth is unknown for graphs of bounded boxicity. However, it is possible that some classes for which these problems are known to be difficult have bounded boxicity; for example, I am not sure whether such classes as degree 3 graphs and line graphs can have arbitrarily large boxicity.

CHORDAL GRAPHS: Chordal graphs can be recognized in linear time, as discussed in the text. An intersection representation as subtrees of a tree can also be found in linear time. Many problems are solvable efficiently in polynomial time from the perfect elimination scheme or clique tree model. Weighted clique, weighted independent set, coloring, and clique cover can all be solved in linear time in this fashion. Many problems which are NP-complete on chordal graphs, such as dominating set, Steiner tree, bandwidth [306] max cut [47] and Hamilton cycle [115] are shown to be NP-complete on the smaller class of split graphs; graph isomorphism is also isomorphism-complete on both classes [54]. Treewidth and minimum fill are trivial on chordal graphs. Treewidth can be solved in polynomial time if there are no chordless cycles of length larger than k for any fixed k [52]; maximum weight induced paths can also be found in polynomial time on this superclass of chordal graphs [214]. The weighted feedback vertex problem is solvable in polynomial time for chordal graphs; this comes from the fact that feedback vertex set on a chordal graph corresponds to finding a minimum set of vertices containing at least one vertex from each triangle, and this is a special case of a covering problem solved for chordal graphs in [121].

CHORDAL BIPARTITE GRAPHS: As discussed in the text, chordal bipartite graphs can be recognized in $O(\min\{n^2, mlogn\})$ time. Existence and construction of an implicit representation are open for this class. NP-complete problems on chordal bipartite graphs include dominating set, Hamilton cycle (these two problems can be solved in polynomial time on the subclass of convex graphs [150, 378]) Steiner tree [379, 378], and bandwidth; chordal bipartite graphs are also isomorphism-complete [380]. The feedback vertex problem, which is polynomial for convex graphs [347], does not seem to have been studied for chordal bipartite graphs. Treewidth can be solved in $O(m^\alpha)$ time, where α is the exponent for matrix multiplication [305]; the related problem of pathwidth (corresponding to minimizing the maximum clique size in an interval completion, rather than a chordal completion, of the input graph) is NP-complete for chordal bipartite graphs [303]. Minimum fill can be solved in $O(n^4)$ time [94]. Many problems, such as matching, computing neighborhood containments, independent set, shortest paths can be solved more simply and efficiently on chordal bipartite graphs than on general bipartite graphs using the Γ-free matrix characterization, as discussed in the text.

CIRCLE GRAPHS: Circle graphs can be recognized in $O(n^2)$ time [442]. This algorithm also constructs the model, which is an implicit representation. Since neighbors of a vertex form a permutation graph, clique is clearly solvable in polynomial time; the best known complexity is $O(n\log n+\min\{m, n\omega\})$ for the unweighted case [364], and $O(n\log n+\min\{n^2, m\log\log n\})$ for the weighted case [20], assuming the model is given as part of the input. It is also not difficult to devise polynomial time independent set algorithms for this class, with $O(n^2)$ the best known time bound. For the unweighted case this is attributed to Buckingham in a 1981 manuscript at Stewart's website listed in the introduction, but I could not find a published version achieving the same time bound earlier than [20], which achieves this bound for the weighted case as well. Dominating set is NP-complete for this class [295]; a $2+\epsilon$ approximation algorithm is given in [277]. The k-coloring problem is NP-complete for $k > 3$ [208, 471]; an $O(n\log n)$ algorithm for the 3-coloring problem on circle graphs is given in [472]. Note that there is some disagreement in the literature on relationship between clique size and chromatic number in circle graphs; the paper [471] claims that chromatic number is at most twice the clique size for circle graphs, but more recent papers by experts in the field give 2^ω as the best known upper bound on chromatic number in terms of clique size [319]. The Hamilton cycle problem is NP-complete on circle graphs [147]; note that this is given as solvable in polynomial time in [282], but this is a case of a misplaced column entry in the table. Bandwidth is NP-complete, since every tree is a circle graph. Schäffer gave Johnson an outline of a Steiner tree algorithm for circle and circular-arc graphs (and thus the problem is listed as polynomially solvable in [282]); [40] will be the first generally available polynomial algorithm for Steiner tree on circle graphs. Other polynomial time solvable problems on circle graphs include minimum fill and treewidth, which can be solved in $O(n^3)$ time [302, 311], and isomorphism, which can be solved in $O(nm)$ time [265]. I was unable to find any work on clique cover for circle graphs; since both clique cover and circle graphs are heavily studied, this may be an interesting open problem.

CIRCULAR-ARC GRAPHS: This is a well-studied class, and many more algorithms have been designed specifically for circular-arc graphs than I will list here. Circular-arc graphs can be recognized by an algorithm which also constructs the model in linear time [365]. Maximum cardinality independent set can be solved in $O(n)$ time, if the input is given as a circularly ordered list of endpoints [225, 363], as can minimum cardinality dominating set [271]. The best known time bound for the weighted dominating set problem is $O(n+m)$ [92]. I know of no paper which specifically solves the weighted independent set problem for circular-arc graphs ([188] solves the problem in $O(n^2)$ time for the superclass of circle trapezoid graphs), though it is obviously solvable from the model in $O(ln)$ time where l is the minimum load on the circle, i.e. the number of arcs passing through the point on the circle which is covered by the fewest arcs. Maximum clique is solvable given the model in $O(n)$ time [14]. Weighted clique can be solved in $O(n\log n+m\log\log n)$ time [433].

Hamilton cycle can be solved in $O(n^2 logn)$ time on circular-arc graphs [432] (note that the linear and $O(n^2)$ time Hamilton circuit algorithms cited as forthcoming in [346] were flawed) and clique cover in $O(n)$ time [271]. Isomorphism can be solved in $O(nm)$ time [265]; at one point, I was part of a paper claiming an $O(n^2)$ bound for this, but our algorithm relied on a result from another paper which may be in error. Coloring is NP-complete on circular-arc graphs, as discussed in the text. It is possible to find a 3/2 optimal coloring [292], and to optimally solve the k-coloring problem for constant k in $O(n)$ time [208]; there is also a randomized polynomial algorithm with an approximation ratio of 1+1/e if the number of colors is not $O(\log n)$ [335]. Bandwidth is NP-complete for circular-arc graphs; an approximation algorithm with (optimal) performance ratio 2 is given in [334]. Other problems solved for circular-arc graphs include an $O(n\log n)$ algorithm for matching when the model is given [348], and $O(n^3)$ algorithms for minimum fill [311] and treewidth [451]. Although I have generally limited myself to coverage of sequential algorithms, note that unweighted independent set, clique cover, and dominating set can be solved optimally given the model in parallel, using $O(\log n)$ time on $O(n/\log n)$ processors [404]. Maximum cut seems to be open for circular-arc graphs, as well as the subclass of interval graphs. The status of Steiner tree is slightly unclear; Schäffer sketched a proof that this is polynomially solvable (for both circle and circular-arc graphs), and thus it appeared as polynomial in the table of [282], though no algorithm solving the problem appears in the general literature. I found no papers dealing with feedback vertex set in circular-arc graphs, but the problem can be solved in polynomial time by reduction to feedback vertex set on interval graphs. Pathwidth is open for circular-arc graphs.

CLIQUE-WIDTH K GRAPHS: The recognition problem can be solved in polynomial time for clique-width 1 (trivial), 2 (equals cograph recognition), and 3 [122], and is open in general and for other fixed values of k. A polynomial time algorithm which constructs an $O(\log n)$ clique-width decomposition tree for any graph with clique-width at most k for fixed k is given in [281]. For any fixed k, an implicit representation can be found in polynomial time using balanced k-homogeneous sets, as discussed in the text. The key result on this class is that any problems posed in monadic second order logic with quantification over vertex sets, but no quantification over edge sets, can be solved in linear time on clique-width k graphs if the decomposition tree is given as part of the input [135]. Problems solvable in this way include independent set, dominating set, i-colorability for fixed i, and various other NP-complete and polynomially solvable problems for general graphs. Certain problems which are not expressible in this form of logic can also be solved on clique-width k graphs; such problems include Hamilton cycle and max cut [478], and chromatic number when the number of colors may be arbitrarily large [313]. If the graph is given in adjacency matrix form, independent set and i-colorability can still be solved in polynomial time [57], but the question of whether all problems posed in the restricted second order monadic logic by robust algorithms remains open. Most other problems have not been investigated for graphs with bounded clique-width. Thus, for example, the complexity of isomorphism on graphs of bounded clique-width would seem to be an open problem, both with the decomposition tree given and for robust algorithms.

COGRAPHS: Cographs are a particularly tractable class of graphs. They can be recognized in linear time, and a cotree can be constructed in the same time bound

[128]. Since this cotree is unique, cograph isomorphism is reduced in $O(n+m)$ time to rooted tree isomorphism, which can be solved in $O(n)$ time. It is also possible to find an implicit representation in linear time. Many results on cographs are a consequence of containments in larger classes of graphs, such as permutation graphs, distance hereditary graphs, and clique-width k graphs, for which many problems are known to be linear time solvable. Nevertheless, many papers have been written about complexity of problems on cographs, since so many problems which are difficult on larger classes of graphs can be solved efficiently on cographs. Examples of tractable problems on cographs include an $O(n^2)$ max cut algorithm [47], and linear time algorithms for minimum fill [129], bandwidth [490], and pathwidth (which equals treewidth for cographs) [51]. The sandwich problem for cographs is polynomially solvable [228], contrasting with the NP-completion of the completion and deletion problems for cographs [168]. Other NP-complete problems on cographs include list coloring [279] and subgraph isomorphism [148].

COMPARABILITY AND CO-COMPARABILITY GRAPHS: Although a transitive orientation can be found in linear time, recognizing comparability graphs is as hard as recognizing triangle-free graphs, currently requiring time proportional to matrix multiplication. The maximum weighted clique and vertex coloring problems can be solved by a robust algorithm in linear time [367], and independent set can be transformed in linear time to bipartite matching once the transitive orientation is given. Many problems on comparability graphs are NP-complete since these contain the bipartite graphs; for example, Hamilton cycle, dominating set, bandwidth, feedback vertex, treewidth [24], minimum fill [301] and Steiner tree are already NP-complete on bipartite graphs, and bipartite graphs are isomorphism-complete. As far as I can determine, the max cut problem, which is trivial on bipartite graphs, is open for comparability graphs.

Since [367] produces a transitive orientation of the complement in linear time, maximum independent set and clique cover can be solved robustly on co-comparability graphs in linear time. Hamilton cycle can be solved in $O(n^3)$ time for co-comparability graphs [154]. More precisely, the algorithm takes $O(in^2)$ time, where i is the size of a maximum independent set in G; this is based on $O(i)$ calls to a Hamilton path algorithm for co-comparability graphs from [149]. The authors of [154] also explain why the algorithm for Hamilton cycle can probably be implemented to run in $O(n^2)$ time, though the full details are not worked out in the paper. Minimum weight Steiner trees in co-comparability graphs can be found in $O(n\log n+m)$ time [78]. The cardinality Steiner tree can be solved in $O(n+m)$ time, as noted independently by Colbourn and Lubiw; the algorithm is described in [333]. Although weighted dominating set is NP-complete on co-comparability graphs [93], the cardinality dominating set problem can be solved in $O(nm^2)$ time [78], and the problem remains solvable in polynomial time for integer weights up to any fixed k [93]. Note that there was a report claiming a running time for dominating set proportional to matrix multiplication, but the algorithm was apparently not correct. There is an $O(n^2m)$ algorithm for the feedback vertex set problem on co-comparability graphs [347]. Bandwidth, minimum fill, max cut and treewidth [308, 491, 47, 24] are NP-complete on co-bipartite graphs, and co-bipartite graphs are isomorphism-complete.

DISK INTERSECTION GRAPHS: Recognition of disk intersection graphs is NP-hard [258]. Existence and construction of an implicit representation are natural open problems for this class of graphs. Another natural question is whether the clique problem, which is polynomially solvable for unit disk graphs, can be solved for the more general class. Disk graphs contain both unit disk graphs and planar graphs as subclasses, implying NP-completeness of such problems as 3-coloring, independent set, clique cover, Steiner tree, feedback vertex, dominating set, Hamilton cycle, and bandwidth. If the model is given as input, polynomial time approximation schemes for weighted independent set and weighted vertex cover are given in [174]; these also apply to intersection graphs of regular polygons, in any fixed dimension. I know of no exact algorithmic results designed specifically for disk intersection graphs; thus problems which are solvable on planar graphs, such as max cut, minimum fill, and isomorphism, seem to be open for disk intersection graphs. Treewidth also does not seem to have been studied for disk graphs, and is singled out as an interesting open problem for the subclass of planar graphs in [42].

Distance hereditary graphs: Distance hereditary graphs can be recognized in linear time, and an implicit representation can be constructed in the same time bound. However, as noted in [399], no natural intersection model is known for the class, despite the fact that this is an intersection class as discussed in the text. Many problems are solvable efficiently thanks to the pruning sequence, and the fact that these have clique-width at most 3. For example, maximum weighted clique, maximum weight independent set, coloring, and clique cover can be solved in $O(n)$ time if the pruning sequence is given as input, and thus in linear time if the graph is given in adjacency list form [246]. Linear time algorithms for the dominating set problem are given in [386, 97]. Weighted Steiner tree is solvable in linear time for nonnegative edge weights [493], as are the treewidth and minimum fill problems [82], and feedback vertex set [288]. Hamilton cycle can be solved in $O(n^2)$ time on distance hereditary graphs [385]. Isomorphism can be tested in polynomial time for distance hereditary graphs, since every distance hereditary graph is a circle graph. In fact, I believe that isomorphism can be tested in linear time on this class, by reduction to tree isomorphism, but the algorithm only appeared in a technical report, and the report does not give an explicit time bound [29], see also [28]. Since every tree is distance hereditary, bandwidth is NP-complete for the class. Max cut is polynomial on the class since every distance hereditary graph has clique-width at most 3, but I know of no specific algorithm for distance hereditary graphs.

Domination graphs: The most interesting problem for domination graphs is recognition, which remains open. Algorithms on domination graphs generally come either from the definition of the class, or from the fact that every domination graph is weakly chordal, while NP-completeness results follow from containment of classes such as chordal graphs, trapezoid graphs and tolerance graphs.

Dot product k graphs: The recognition problem is open for every $k > 1$, as is the problem of existence and construction of an implicit representation. No algorithmic work has been done for $k > 1$, i.e. threshold graphs.

Doubly chordal graphs: Doubly chordal graphs can be recognized in linear time, since they are exactly equal to chordal and dually chordal graphs. Although several algorithms have been designed specifically for doubly chordal graphs, the best current bounds for problems discussed in this survey come from results on the superclasses of chordal graphs or dually chordal graphs.

DUALLY CHORDAL GRAPHS: Dually chordal graphs can be recognized in linear time [67]. Many problems are NP-complete on dually chordal graphs, since adding a universal vertex makes any graph dually chordal. Thus, for example, clique, independent set, coloring, clique cover, Hamilton cycle, treewidth, and many other problems can easily be seen to be NP-complete on this class. For some problems, adding a universal vertex would make the problem trivial; the paper [67] shows that a variety of domination and location problems, including unweighted dominating set and Steiner tree (the weighted problems are obviously NP-complete, by adding a universal vertex with appropriate weight) can be solved in linear time on dually chordal graphs. The all pairs shortest path problem can be solved in $O(n^2)$ time on dually chordal graphs [68].

EPT GRAPHS: Recognition is NP-complete for EPT graphs [227]. Although EPT graphs have an implicit representation, constructing an implicit representation in polynomial time from the adjacency matrix of an EPT graph is an open problem. Every line graph is an EPT graph, which implies that such problems as coloring, clique cover, Hamilton cycle, dominating set, and Steiner tree are NP-complete, and that EPT graph isomorphism is isomorphism-complete. The coloring problem on EPT graphs is equivalent to path coloring on undirected tree networks, and has received some attention; Tarjan [454] gave a 3/2 optimal coloring algorithm, and a 4/3 optimal algorithm with asymptotic approximation ratio 11/10 is given in [173]. Weighted clique and weighted independent set are solvable by robust algorithms, as discussed in the text. Other problems not known to be NP-complete for line graphs (these include max cut, which is polynomially solvable for line graphs [239], and bandwidth, which is open for line graphs [280]), seem to be open by way of not having been studied. As noted in [454], every EPT graph is decomposable using clique separators into line graphs; thus, if a problem solution can be constructed from the solutions on clique-separated components, the complexity on EPT graphs will be the same as the complexity on line graphs. For example, this implies that the complexity of minimum fill or treewidth on EPT graphs is equal to the complexity of these problems on line graphs.

INTERVAL FILAMENT GRAPHS: Recognizing interval filament graphs, and creating an interval filament model, are open problems. If the model is given, clique and independent set are solvable on interval filament graphs in polynomial time [211]; these problems are open for robust algorithms.

INTERVAL GRAPHS: The first linear time algorithm for recognizing interval graphs appeared in [56]; many others have appeared, and one is described in the text. The original recognition algorithm can also be used to test isomorphism of interval graphs in linear time [355]. Weighted independent set, weighted clique, coloring, and clique cover can be solved in $O(n)$ time if the intervals are given in sorted order. Hamilton cycle can also be solved in $O(n)$ time given the interval model, though this is more difficult [96]. An $O(n)$ weighted dominating set algorithm given in [92]. Bandwidth can be solved on interval graphs in $O(n\log n)$ time (for the decision problem; finding the optimal bandwidth takes $O(nlog^2 n)$ time using this algorithm) [449], and weighted feedback vertex set in $O(n+m)$ time [353]. As far as I can determine, max cut is open for interval graphs, though it can be solved in linear time for unit interval graphs [50].

INTERVAL NUMBER: Recognizing graphs with interval number k is NP-complete for any fixed $k > 1$ [481]. No algorithm is known which constructs an implicit representation of a graph with interval number 2 in polynomial time. Every circular-arc graph has interval number at most 2, so bandwidth and coloring are NP-complete on the class. Every degree 3 graph has interval number at most 2, implying NP-completeness for such problems as independent set and Hamilton cycle, and containment of line graphs implies NP-completeness for clique cover, dominating set, and Steiner tree, as well as isomorphism-completeness. Max cut [47] is also NP-complete for interval number 2. Every planar graph has interval number at most 3 [426], implying results such as NP-completeness of feedback vertex set when interval number is bounded. The real challenge for this class is to use the representation to solve problems effectively; only a few papers, such as [494], which gives a $2k$ times optimal weighted independent set algorithm for graphs with interval number k, attempt to do so. The clique problem seems to be open for graphs with interval number 2, and perhaps for graphs with bounded interval number as well.

INVERSION-FREE GRAPHS: Although these are not a well known class of graphs, I include them in this survey due to the interesting open problems, as described in the text section on robust algorithms. Recognition of the class is open. The clique problem can be solved if a model is given as input, but is open if one is given only a promise that the input is inversion-free. Solving the clique problem with a robust algorithm would also give a robust algorithm for finding a maximum clique in a visibility graph.

K-POLYGON GRAPHS: It is possible to test whether a graph is a k-polygon graph in $O(4^k n^2)$ time, making this polynomial when k is fixed, but recognition is NP-complete if k is allowed to vary [170]. Since k-polygon graphs are a subset of circle graphs, it is possible to construct an implicit representation in polynomial time. In contrast with circle graphs, dominating set (and a variety of other domination problems) are polynomial for any fixed k [169]. Natural problems to study include those which are polynomial for permutation graphs, but open or NP-complete for circle graphs; these include the Hamilton cycle, coloring, and clique cover problems. An approximation algorithm for bandwidth with performance ratio $2k^2$ (the authors note that an improvement to $4/3k^2$ has been communicated to them) for k-polygon graphs is given in [334]; I believe it is open whether bandwidth is NP-complete for every fixed k.

PATH GRAPHS: Path graphs can be recognized in $O(n + m)$ time [144]; the natural intersection model is also produced. This result is not well known, since the paper was published in a proceedings which is relatively difficult to find; the paper [422] gives an $O(nm)$ time algorithm in a more widely distributed journal. Every path graph is chordal, so problems such as clique, coloring, independent set, and clique cover can be solved in linear time. Dominating set [55], max cut [47], Hamilton cycle [38], and bandwidth are NP-complete on path graphs. Hamilton cycle is also NP-complete on the subclass of directed path graphs [381] and open for rooted directed path graphs; dominating set is polynomial on rooted directed path graphs [55]. Note that the math review article of the paper [163] is a bit ambiguous, and might lead the reader (i.e., I read it this way) to believe dominating set is polynomial on path graphs; the class of graphs in the paper is actually the dually chordal graphs. Path graphs are isomorphism-complete [54]; the subclass of directed path graphs is also isomorphism-complete, while isomorphism of rooted directed path graphs can be tested in polynomial time [25]. Path graphs are separated from split and chordal graphs by the minimum dominating clique problem, which can be solved in polynomial time on path graphs but is NP-complete on split graphs [331]. I was unable to find any work on the Steiner tree problem restricted to path graphs, but I am told that this should be NP-complete as a simple xtension of a proof that connected dominating set is NP-complete for the class [328].

Perfect graphs: Recognition of perfect graphs is a famous open problem. The difficult, but polynomial, algorithms for the clique, coloring, clique cover, and independent set [237] are robust algorithms for perfect graphs. Perfect graphs include many subclasses, including classes for which such problems as Hamilton cycle, dominating set, Steiner tree, feedback vertex, bandwidth, treewidth, and minimum fill are NP-complete. With the exception of the important paper cited above, there are few results solving NP-complete problems on perfect graphs in general; an $O(n^3)$ combinatorial algorithm for 3-coloring on perfect graphs is given in [470]. One area of algorithmic research is recognition of perfect graphs when restricted to various subclasses. Examples include recognition of perfect graphs without certain induced subgraphs, such as perfect claw-free graphs [111], perfect $K_4 - e$-free graphs [197], perfect bull-free graphs [405], paw-free perfect graphs [389], and dart-free perfect graphs [108]. Other classes for which perfect graph recognition is polynomial include planar graphs [266] and 2-split graphs, i.e. graphs which can be partitioned into two split graphs [262]. Open problems in this area are collected at

`www.cs.rutgers.edu/~chvatal/perfect/problems.html.`

Perfectly orderable graphs: Recognizing perfectly orderable graphs is NP-complete [369]. There are linear time ($O(nm)$ in the weighted case) [259] algorithms for solving chromatic number and maximum clique if a perfect ordering is given as part of the input; if the order is not given, it is open to solve these problems without resorting to more difficult and general algorithms for perfect graphs. Many NP-completeness results follow from the fact that such classes as chordal graphs, comparability graphs, and other classes are perfectly orderable. See [261] for more information on perfectly orderable graphs.

Permutation graphs: Permutation graphs can be recognized in linear time [367], by an algorithm which constructs the natural implicit representation. Independent set and clique can be solved in $O(n\log\log n)$ time if the model is given as input; these are equivalent to finding the longest increasing subsequence of a permutation. The well known algorithm $O(n\log n)$ for finding a longest increasing subsequence can be modified to run in $O(n\log\log n)$ time if input numbers are in the range $1..n$ by using von Emde Boas trees, and can also be easily modified to handle the weighted case. Since permutation graphs are perfect, coloring and clique cover can be solved in the same time bound; a maximum cardinality matching algorithm with the same running time is given in [406]. An $O(n)$ algorithm for finding the minimum cardinality dominating set problem when given the permutation diagram is contained in [98]; an $O(n+m)$ algorithm for the weighted problem is given in [407]. Isomorphism of permutation graphs can be tested in $O(n^2)$ time [447]. Steiner tree can be solved in $O(n)$ time, since every permutation graph is a trapezoid graph, and in $O(n\log n)$ time in the weighted case [275], if the permutation diagram is given. A linear time algorithm for finding a Hamilton cycle in a permutation graph is given in [153]. There is an $O(nt)$ algorithm to determine whether a permutation graph has treewidth (which equals pathwidth for permutation graphs) at most t [48]. The weighted feedback vertex set problem can be solved in $O(mn)$ time for permutation graphs [343]. The complexity of bandwidth on permutation graphs is an open problem; it is also open for the subclass of bipartite permutation graphs, but can be solved in polynomial time for chain graphs [309]. Max cut also seems to be open for this class of graphs; the the reduction given in [16] intended to show that the problem is NP-complete for permutation graphs is not polynomial.

PI GRAPHS: Recognition of PI graphs in polynomial time is open. Nevertherless, an implicit representation can be constructed in polynomial time, since every PI graph is a trapezoid graph. Although there have been papers written solving algorithmic problems on PI graphs, I know of no problem for which the best algorithm on PI graphs is faster than the time of the best current known algorithm for trapezoid graphs. I also know of no results which explicitly separate the complexity on triangle graphs from trapezoid graphs, and NP-completeness results come only as a consequence of the fact that these contain such classes as permutation graphs and interval graphs.

PROBE INTERVAL GRAPHS: Probe interval graphs can be recognized in linear time if the division of the vertices into probes and nonprobes is given [286, 287], but is open if the partition is not given. No algorithms have been specifically designed for probe interval graphs, so any known algorithms and NP-completeness results come only from containments between graph classes.

SPLIT GRAPHS: Split graphs can be recognized in linear time, and in $O(n)$ time if the degree sequence is give as input [223]. Split graphs have the same complexities as chordal graphs on all problems described under chordal graphs, and often seem to be at the core of algorithms and proofs of difficulty for chordal graphs. The two classes are separated in complexity for a number of problems such as triangle packing, i.e. the problem of finding the maximum number of vertex disjoint triangles. The triangle packing problem is polynomial for split graphs, but NP-complete for chordal graphs [239]; pathwidth is also NP-complete on chordal graphs, and polynomially solvable on split graphs [240]. The sandwich problem (given a set of required and optional edges, is there a set of optional edges which can be included to give a graph in the class) is polynomial for split graphs, in contrast to most of the other classes discussed here; specifically, the sandwich problem is NP-complete for comparability graphs, permutation graphs, co-comparability graphs, circle graphs, interval graphs, circular-arc graphs, path graphs, chordal graphs, and co-chordal graphs, and was left as an open problem for chordal bipartite graphs and strongly chordal graphs in [228]. However, the split graph completion problem and split graph deletion problems are NP-complete [383], while the number of edges which need to be added or deleted to a given graph to obtain a split graph can be computed in polynomial time [249]. Although bandwidth is NP-complete, there is a 2-optimal approximation algorithm for split graphs [306].

STRONGLY CHORDAL GRAPHS: Strongly chordal graphs can be recognized in $O(n^2)$ or $O(m\log n)$ time. No implicit representation is known for this class. Strongly chordal graphs are both chordal and dually chordal, so such problems as independent set, clique, coloring, dominating set, and Steiner tree can be solved in linear time. Unlike both of these classes, weighted dominating set is polynomial on strongly chordal graphs; the time bound is linear if the strong elimination ordering is given as part of the input [182], and thus is $O(n^2)$ or $O(m\log n)$ if given in adjacency list form. The same time bounds (with elimination order given or not) apply to the feedback vertex set problem, as a consequence of [95] and the correspondence of feedback vertex set and a covering problem as mentioned under chordal graphs. Hamilton cycle is NP-complete on strongly chordal graphs [378].

I believe that strongly chordal graphs are isomorphism-complete by a simple reduction from the isomorphism completeness of chordal bipartite graphs shown in [380]; note that this does not follow directly from the result of [25] showing that directed path graph isomorphism is isomorphism-complete. This paper distinguishes between rooted directed path graphs, in which all edge directions emanate from the root, and directed path graphs, in which edges can be directed both towards the root and away from the root. The first class is a subclass of strongly chordal graphs, while the second class is not (it contains the 4-sun). Unfortunately, both classes are sometimes referred to as directed path graphs, and the paper [282] gives directed path graphs in the first sense as a subclass of strongly chordal graphs, leading to potential confusion. [25] shows that isomorphism is polynomial for the rooted case. Bandwidth is NP-complete for this class, since every tree is strongly chordal. As far as I could determine, max cut is open for strongly chordal graphs.

TOLERANCE GRAPHS: Recognition of tolerance graphs is an open problem, as is existence of an implicit representation. Although it is not obvious, recognition of tolerance graphs is in NP [255]. Most algorithms on tolerance graphs are a consequence of containment in weakly chordal graphs, but a few algorithms have been designed to work on tolerance graphs when a representation is given as part of the input. For example, the clique and coloring problems can be solved in $O(n^2)$ time on tolerance graphs, if the tolerance representation is given [232]. Independent set and clique cover can be solved in linear time given a tolerance representation, using the fact that a co-perfect ordering comes directly from the tolerance representation [229], and that independent set and clique cover can be solved in linear time given a co-perfect ordering [109]. An $O(n^5)$ algorithm for minimum fill on the class of multitolerance graphs, which includes both tolerance and trapezoid graphs, is given in [393]. Bandwidth is NP-complete for tolerance graphs [272]; this is the only example I know of a problem proved to be NP-complete on tolerance graphs which is not known to be NP-complete on a subclass. Many algorithmic problems seem to be open for tolerance graphs, whether the model is given or not. I might choose dominating set and Hamilton cycle as interesting examples, both with the model given and for robust algorithms; these problems are solvable in polynomial time on bounded tolerance graphs and NP-complete on weakly chordal graphs.

TRAPEZOID GRAPHS: Trapezoid graphs can be recognized in $O(n^2)$ time [357], by an algorithm which constructs the natural implicit intersection model. Trapezoid graphs are co-comparability graphs, but a number of algorithms have been designed to solve problems more efficiently than on the larger class. Given the model, the chromatic number, clique cover, weighted independent set, and weighted clique problems can be solved in $O(n\log\log n)$ time [188]; k-coloring can also be solved in $O(kn)$ time [140]. It is also possible to solve the Steiner tree problem in $O(n)$ time [344] given the trapezoidal representation. Minimum weight dominating sets can be found in $O(nm)$ time [345], and the treewidth and minimum fill problems can be solved in $O(n^2)$ time [49]. In general, algorithms for trapezoid graphs received less attention than algorithms for many of the other graph classes. I found no specific papers dealing with Hamilton cycle, unweighted domination, or isomorphism when restricted to trapezoid graphs; in all these cases, I feel that behavior of trapezoid graphs should be closer to the subclass of permutation graphs than the superclass of co-comparability graphs. In particular, having worked on the related recognition problem for trapezoid graphs, I feel that there is a polynomial time isomorphism algorithm, though none has yet been developed. Along these lines, [190] remarks that the $O(nm)$ algorithm for solving the weighted feedback vertex set problem on permutation graphs [343] can easily be extended to trapezoid graphs. Although [308] gives a 2-approximation algorithm for trapezoid graphs which is simpler and faster than the algorithm for AT-free graphs, the complexity of bandwidth on trapezoid graphs seems to be an open problem; max cut is also apparently open.

TREEWIDTH k: Although computing the treewidth is NP-complete, it is possible to determine whether a graph has treewidth at most k for fixed k in linear time [43], and to find the corresponding tree decomposition. Every problem which can be posed in second order monadic logic can be solved for graphs of bounded treewidth [132]; the class of problems solvable is extended in [59], and algorithms have also been designed for many problems which do not fit in a general framework of solvable problems. There are too many problems which have been solved on graphs of fixed treewidth to deal with here; [44] is a survey devoted to this issue. Problems such as clique, independent set, traveling salesman, and dominating set can be solved in $O(n)$ time on graphs of bounded treewidth. Max cut is $O(n)$ on graphs of bounded treewidth [47]; note that even the weighted case is solvable in linear time, while weighted max cut is NP-complete on any class which contains arbitrarily large cliques. Isomorpism can be tested in polynomial time for graphs of fixed treewidth k, though k appears in the exponent of the running time [45]. Bandwidth is one of the better known problems which is NP-complete on the class, since it remains NP-complete for trees.

UNIT DISK GRAPHS: Unit disk graph recognition is NP-hard [77]; membership in NP is open. Existence and construction of an implicit representation is an open problem. k-coloring for every $k > 2$ is NP-complete for unit disk graphs [234], as is independent set [112]. Dominating set, Hamilton cycle, bandwidth, and Steiner tree are NP-complete for the subclass of grid graphs [112, 278, 157, 207]. Clique is solvable by a robust algorithm in polynomial time, as discussed in the text. Polynomial time approximation schemes (these achieve approximation ratios of $1+\epsilon$ for any ϵ, with running time polynomial in the size of the input and $1/\epsilon$) for independent set, vertex cover and dominating set on unit disk graphs are presented in [273]; extensions for independent set and vertex cover are discussed under disk graphs. Good approximation algorithms for bandwidth and various other vertex ordering problems on intersection graphs of randomly generated unit disks are presented in [156]. [362] gives a coloring algorithm with approximation ratio 3 for unit disk graphs; this can be improved to a ratio of 2 if the clique size is fixed and a model is given as part of the input [233]. There is a considerable body of work dealing with unit-disk graphs as an idealized model for radio transmission, solving broadcasting problems of various types; there is also a body of work using unit disk graphs for map labeling. Since I am not expert in these areas, I will not try to pick out the important results on these subjects. Although there is a great deal of work dealing with algorithms for specialized problems on unit disk graphs, the complexity of many of the classical graph problems on unit disk graphs seems to be open. Thus, I was not able to determine the complexity of such problems as clique cover, isomorphism, treewidth, minimum fill, feedback vertex set, or max cut when restricted to unit disk graphs, whether the model was given as part of the input or not. Interested readers should note that unit disk graphs are studied under many different names (one common name is geometric graphs) in the literature.

VISIBILITY GRAPHS: Visibility graph recognition is a well known open problem, both in the case when an ordering of vertices on the outside of the polygon is specified, and when any ordering is permissible. No implicit representation is known for this class of graphs. If the model is given, the clique problem can be solved in polynomial time for visibility graphs; this is open for robust algorithms, and is discussed in the text. Independent set and dominating set are NP-complete for visibility graphs, and isomorphism is as hard as for general graphs [349]. Clique cover is also NP-complete for visibility graphs [137]; a logarithmic approximation algorithm is given in [167]. The complexity of coloring visibility graphs is an open problem. Although every visibility graph has a Hamilton cycle, it is open whether you can find a Hamilton cycle in a visibility graph when the input is given in adjacency matrix form.

WEAKLY CHORDAL GRAPHS: Weakly chordal graphs can be recognized in $O(m^2)$ time using two fundamentally different algorithms [36, 256], both of which are discussed in the text. The best known algorithms for independent set, clique cover, coloring, and clique run in $O(nm)$ time for the unweighted case [256] and $O(n^4)$ time for the weighted case [446]. NP-complete problems include dominating set, max cut, Steiner tree, bandwidth, and Hamilton cycle, which are NP-complete on chordal graphs. Treewidth and minimum fill-in can be solved in $O(n^6)$ time on weakly chordal graphs [64]. Feedback vertex does not seem to have been studied for this class. Weakly chordal graphs seem much more general than chordal graphs, but there are relatively few problems known to be tractable on chordal graphs but NP-hard on weakly chordal graphs; the most natural problem I could find of this form is testing whether an input graph is perfectly orderable [260], which is trivial on chordal graphs.

WELL COVERED GRAPHS: Well-covered graphs are co-NP-complete to recognize. As is shown in the text, the independent set problem, which is trivial on well-covered graphs given a promise that the input is in the class, is intractable for robust algorithms. Many problems, such as clique, coloring, Steiner tree, minimum fill, max cut, Hamilton cycle, clique cover, and dominating set are NP-complete on this class, and well-covered graphs are isomorphism-complete, all of which are shown in [419]. It is easy to show that other problems are NP-complete, since adding pendant vertices adjacent to each vertex of G makes any graph well-covered; for example, it is easy to see that treewidth and minimum fill are NP-complete in this way. Although I would not classify it as a significant open problem, it is interesting to note that this transformation cannot be used to show that bandwidth is NP-complete. Therefore, despite the fact that bandwidth is NP-complete on most of the classes above and most optimization problems are hard for well-covered graphs, bandwidth has not been studied for this class and cannot be trivially shown to be NP-complete. Distinctions between complexity of problems on well-covered graphs and problems on subclasses such as very well-covered graphs (in which all maximal independent sets have size $n/2$, and for which recognition, clique cover, dominating set, and Hamilton cycle are polynomial) are studied in [418].

Bibliography

[1] N. Abbas, L.K. Stewart, Biconvex Graphs: Ordering and Algorithms, Discrete Applied Mathematics 103, 1-19, 2000

[2] J. Abello, H. Lin, S. Pisupati, On Visibility Graphs of Simple Polygons, Congressus Numerantium 90, 119-128, 1992

[3] G.S. Adhar, Domination in Bounded Interval-Tolerance Graphs, 7th International Conference on Parallel and Distributed Systems, 29-34, 2000

[4] G.S. Adhar, Optimal Parallel Algorithms for Cut-Vertices, Bridges and Hamilton Path in Bounded Interval-Tolerance Graphs, 8th International Conference on Parallel and Distributed Systems, 91-98, 2001

[5] P.K. Agarwal, N. Alon, B. Aronov, S. Suri, Can Visibility Graphs be Represented Compactly?, Discrete Computational Geometry 12, 347-365, 1994

[6] P.K. Agarwal, M. van Kreveld, S. Suri, Label Placement by Maximum Independent Set in Rectangles, Computational Geometry: Theory and Applications 11, 209-218, 1998

[7] P. Agarwal, M. Sharir, S. Toledo, Applications of Parametric Searching in Geometric Optimization, Journal of Algorithms 17, 292-318, 1994

[8] A. Agarwal, S. Suri, The Biggest Diagonal in a Simple Polygon, Information Processing Letters 35, 13-18, 1990

[9] P.K. Agarwal, M. Sharir, P. Shor, Sharp Upper and Lower Bounds for the Length of General Davenport-Schinzel Sequences, Journal of Combinatorial Theory Series A 52, 1989 228-274

[10] A.A. Ageev, V.L. Beresnev, Polynomially Solvable Special Cases of Simple Plant Location Problem, Proceedings of the First IPCO Conference, Waterloo University Press, 1-6, 1990

[11] A. V. Aho, J. E. Hopcroft, J. D. Ullman, The Design and Analysis of Computer Algorithms Addison-Wesley Reading, MA 1974

[12] V.E. Alekseev, On the Entropy Values of Hereditary Classes of Graphs, Discrete Mathematical Applications 3, 191-199, 1993

[13] V.E. Alekseev, D.V. Korobitsyn, Complexity of Some Problems on Hereditary Classes of Graphs, Diskret. Mat. 4, 34-40, 1992

[14] S.G. Akl, B.K. Bhattacharya, Computing Maximum Cliques of Circular Arcs in Parallel, Parallel Algorithms and Applications 12, 305-320, 1997

[15] O. Alevizos, J.-D. Boissonnant, F.P. Preparata, An Optimal Algorithm for the Boundary of a Cell in a Union of Rays, Algorithmica 5, 573-590, 1990

[16] H.H. Ali, N.A. Sherwani, A. Boals, On the Max Cut Problem in Permutation Graphs, University of Nebraska Omaha Report UNO-CS-TR-90-6, 1990

[17] N. Alon, E.R. Scheinerman, Degrees of Freedom Versus Dimension for Containment Orders, Order 5, 1988 11-16

[18] R.P. Anstee, M. Farber, Characterizations of Totally Balanced Matrices, Journal of Algorithms 5, 215-230, 1984

[19] R.P. Anstee, Z. Füredi, Forbidden Submatrices, Discrete Mathematics 62, 225-243, 1986

[20] A. Apostolico, M.J. Atallah, S. E. Hambrusch, New Clique and Independent Set Algorithms for Circle Graphs, Discrete Applied Mathematics 36, 1-24, 1992 Erratum: Discrete Applied Mathematics 41, 179-180, 1993

[21] J.C. Arditti, Partially Ordered Sets and their Comparability Graphs, their Dimension and their Adjacency, *Problemes Combinatoires et Theorie des Graphes*, 1976, *Proc. Coll. Int. CNRS*, Orsay, France

[22] S.R. ARIKATI, C. PANDU RANGAN, An Efficient Algorithm for Finding a Two-Pair, and its Applications, Discrete Applied Mathematics 31, 71-74, 1991

[23] E.M. ARKIN, D. HALPERIN, K. KEDEM, J.S.B. MITCHELL, N. NAOR, Arrangement of Segments that Share Endpoints: Single Face Results, Discrete Computational Geometry 13, 257-270, 1995

[24] S. ARNBORG, D.G. CORNEIL, A. PROSKUROWSKI, Complexity of Finding Embeddings in a k-tree, SIAM Journal on Algebraic Discrete Methods 8, 277-284, 1987

[25] L. BABEL, I.N. PONOMARENKO, G. TINHOFER, The Isomorphism Problem for Directed Path Graphs and for Rooted Directed Path Graphs, Journal of Algorithms 21, 542-564, 1996

[26] H. BALAKRISHNAN, R. RAJARAMAN, C. PANDU RANGAN, Connected Domination and Steiner Set on Asteroidal Triple-Free Graphs, Lecture Notes in Computer Science 709, WADS 93, 131-141, 1993

[27] H. BANDELT, H.M. MULDER, Distance-Hereditary Graphs, Journal of Combinatorial Theory Series B 41, 182-208, 1986

[28] H.J. BANDELT, A. D'ATRI, M. MOSCARINI, H.M. MULDER, A. SCHULTZE, Operations on Distance-Hereditary Graphs, Instituto di Analisi di Sistimi ed Informatica, Research Report 209, 1988

[29] A. D'ATRI, M. MOSCARINI, H.M. MULDER, On the Isomorphism Problem for Distance-Hereditary Graphs, Econometric Institute Technical Report EI9241/A, Rotterdam School of Economics, 1992

[30] G. DI BATISTA, P. EADES, R. TAMASSIA, I.G. TOLLIS, Graph Drawing, Prentice Hall, Upper Saddle River, New Jersey, 1998

[31] C. BENZAKEN, P.L. HAMMER, D. DE WERRA, Threshold Characterization of Graphs with Dilworth Number Two, Journal of Graph Theory 9, 245-267, 1985

[32] S. BENZER, On the Topology of the Genetic Fine Structure, Proc. Nat. Acad. Sci. USA 45, 1607-1620, 1959

[33] V.L. BERESNEV, A.I. DAVYDOV, On Matrices with Connectedness Properties, Upravlyaemye Sistemy 19, 3-13, 1979

[34] C. BERGE, Farbung von Graphen, deren Samtliche bzw. deren Ungerade Kreise Starr sind, Wiss. Zeitschrift Martin-Luther-Univ. Halle-Wittenburg, 1961, 114

[35] C. BERGE, Hypergraphs, North Holland, Amsterdam, 1989

[36] A. BERRY, J.-P. BORDAT, P. HEGGERNES, Recognizing Weakly Triangulated Graphs by Edge Separability, Nordic Journal of Computing 7, 164-177, 2000

[37] A. BERRY, A. SIGAYRET, C. SINOQUET, Improving Phylogenetic Data by Correcting the 2-Pair Edge-Addition Ordering, manuscript, 2001

[38] A.A. BERTOSSI, M.A. BONUCCELLI, Hamiltonian Circuits in Interval Graph Generalizations, Information Processing Letters 23, 195-200, 1986

[39] B.K. BHATTACHARYA, D. KALLER, An $O(m+n\log n)$ Algorithm for the Maximum Clique Problem in Circular-Arc Graphs, Journal of Algorithms 25, 336-358, 1997

[40] B. BHORE, S. PEMMARAJU, Minimum Cardinality Steiner Forests in Circle Graphs, manuscript, 2002

[41] A. BLUM, New Approximation Algorithms for Graph Coloring, Journal of the ACM 41, 470-516, 1994

[42] H.L. BODLAENDER, Treewidth: Algorithmic Results and Techniques, Technical Report UU-CS-1997-31, Universiteit Utrecht, 1997

[43] H.L. BODLAENDER, A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth, SIAM Journal on Computing 25, 1305-1317, 1996

[44] H.L. BODLAENDER, Treewidth: Algorithmic Techniques and Results, MFOCS 97, Lecture Notes in Computer Science 1295, 19-36, 1997

[45] H. BODLAENDER, Polynomial Algorithms for Graph Isomorphism and Chromatic Index on Partial k-Trees, Journal of Algorithms 11, 631-643, 1990

[46] H.L. BODLAENDER, M.R. FELLOWS, T. WARNOW, Two Strikes Against Perfect Phylogeny, ICALP 92, Lecture Notes in Computer Science 623 273-283, 1992

[47] H.L. BODLAENDER, K. JANSEN, On the Complexity of the Maximum Cut Problem, Nordic Journal of Computing 7, 14-31, 2001

[48] H.L. BODLAENDER, T. KLOKS, D. KRATSCH, Treewidth and Pathwidth of Permutation Graphs, SIAM Journal on Discrete Mathematics 8, 606-616, 1995

[49] H.L. BODLAENDER, T. KLOKS, D. KRATSCH, H. MÜLLER, Treewidth and Minimum Fill-In on d-Trapezoid Graphs, Journal of Graph Algorithms and Applications 2, 1-23, 1998

[50] H.L. BODLAENDER, T. KLOKS, R. NIEDERMEIER, Simple Max-Cut for Unit Interval Graphs and Graphs with few P_4s, manuscript, 2002

[51] H.L. BODLAENDER, R.H. MÖHRING, The Pathwidth and Treewidth of Cographs, SIAM Journal on Discrete Mathematics 6, 181-188, 1993

[52] H.L. BODLAENDER, D.M. THILIKOS, Treewidth for Graphs with Small Chordality, Discrete Applied Mathematics 79, 45-61, 1997

[53] J.A. BONDY, V. CHVÁTAL, A Method in Graph Theory, Discrete Mathematics 15, 111-135, 1976

[54] K.S. BOOTH, C.J. COLBOURN, Problems Polynomially Equivalent to Graph Isomorphism, Report CS-77-04, Computer Science Department, University of Waterloo, 1980

[55] K.S. BOOTH, J.H. JOHNSON, Dominating Sets in Chordal Graphs, SIAM Journal on Computing 11, 191-199, 1982

[56] K.S. BOOTH, G.S. LUEKER Testing for the Consecutive Ones Property, Interval Graphs, and Planarity Using PQ-Tree Algorithms, Journal of Computer and System Science 13, 335-379, 1976

[57] R. BORIE, J. JOHNSON, V. RAGHAVAN, J. SPINRAD, Robust Algorithms for Optimization Problems on Graphs of Bounded Clique-Width, in preparation, 2002

[58] R. BORIE, R.G. PARKER, C.A. TOVEY, A Deterministic Decomposition of Recursive Graph Families, SIAM Journal on Discrete Mathematics 4, 481-501, 1991

[59] R.B. BORIE, R.G. PARKER, C.A. TOVEY, Automatic Generation of Linear-Time Algorithms from Predicate Calculus Descriptions on Recursively Constructed Graph Families, Algorithmica 7, 555-582, 1992

[60] R.B. BORIE, J.P. SPINRAD, Construction of a Simple Elimination Scheme for a Chordal Comparability Graph in Linear Time, Discrete Applied Mathematics 91, 287-292, 1999

[61] P. BOSE, H. EVERETT, S. FEKETE, M.E. HOULE, A. LUBIW, H. MEIJER, K. ROMANIK, G. ROTE, T. SHERMER, S. WHITESIDES, C. ZELLE, A Visibility Representation for Graphs in Three Dimensions, Journal of Graph Algorithms and Applications 2, 1-16, 1998

[62] A. BOUCHET, Circle Graph Obstructions, Journal of Combinatorial Theory Series B 60, 107-144, 1994

[63] A. BOUCHET, Reducing Prime Graphs and Recognizing Circle Graphs, Combinatorica 7, 243-254, 1987

[64] V. BOUCHITTÉ, I. TODINCA, Treewidth and Minimum Fill-In: Grouping the Minimal Separators, SIAM Journal on Computing 31, 212-232, 2001

[65] V. BOUCHITTÉ, I TODINCA, Listing all Potential Maximal Cliques of a Graph, STACS 2000, Lecture Notes in Computer Science 1770, 503-515, 2000

[66] A. BRANDSTÄDT, Classes of Bipartite Graphs Related to Chordal Graphs, Discrete Applied Mathematics 32, 51-60, 1991

[67] A. BRANDSTÄDT, V.D. CHEPOI, F.F. DRAGAN, The Algorithmic Use of Hypertree Structure and Maximum Neighborhood Orderings, Discrete Applied Mathematics 82, 43-77, 1998

[68] A. BRANDSTÄDT, V.D. CHEPOI, F.F. DRAGAN, Clique r-Domination and Clique r-Packing Problems on Dually Chordal Graphs, SIAM Journal on Discrete Mathematics 10, 109-127, 1997

[69] A. BRANDSTÄDT, V.D. CHEPOI, F.F. DRAGAN, V. VOLOSHIN, Dually Chordal Graphs, SIAM Journal on Discrete Mathematics 11, 437-455, 1998

[70] A. BRANDSTÄDT, F.F. DRAGAN, On the Linear and Circular Structure of (claw,net)-free Graphs, manuscript, 1999

[71] A. BRANDSTÄDT, F.F. DRAGAN, E. KÖHLER, Linear Time Algorithms for Hamiltonian Problems on (claw,net)-free Graphs, SIAM Journal on Computing 30, 1662-1677, 2000

[72] A. BRANDSTÄDT, T. KLEMBT, V.B. LE, S. MAHFUD, T. SZYMCZAK, On the Structure and Clique Width of Graphs Defined by Two Forbidden 4-Vertex Graphs, manuscript in preparation, 2002

[73] A. BRANDSTÄDT, V.V. LOZIN, On the Linear Structure and Clique-Width of Bipartite Permutation Graphs, Rutgers Research Report RRR 29-2001, 2001

[74] A. BRANDSTÄDT, V.B. LE, J. SPINRAD Graph Classes: A Survey, SIAM, Philadelphia, 1999

[75] A. BRANDSTÄDT, S. MAHFUD, Maximum Weight Stable Set on Graphs without Claw and Co-Claw (and Similar Graphs Classes) can be Solved in Linear Time, Information Processing Letters, in press, 2002

[76] A. BRANDSTÄDT, J. SPINRAD, L. STEWART, Bipartite Permutation Graphs are Bipartite Tolerance Graphs, Congressus Numerantium 58, 165-174, 1987

[77] H. BREU, D.G. KIRKPATRICK, Unit Disk Graph Recognition is NP-hard, Computational Geometry 9, 3-24, 1998

[78] H. BREU, Algorithmic Aspects of Unit Disk Graphs, PhD Thesis, Computer science, University of British Columbia, 1996, Technical Report UBC CS TR-96-15

[79] G. BRIGHTWELL, On the Complexity of Diagram Testing, Order 10, 297-303, 1993

[80] G. BRIGHTWELL, D.A. GRABLE, H.J. PRÖMEL, Forbidden Induced Partial Orders, Discrete Mathematics 201, 53-80, 1999

[81] H. BROERSMA, T. KLOKS, D. KRATSCH, H. MÜLLER, Independent Sets in Asteroidal Triple-Free Graphs, SIAM Journal on Discrete Mathematics 12, 276-287, 1999

[82] H. BROERSMA, E. DAHLHAUS, T. KLOKS, A Linear Time Algorithm for Minimum Fill-In and Treewidth for Distance Hereditary Graphs, Discrete Applied Mathematics 99, 367-400, 2000

[83] H. BUER, R.H. MOHRING, A Fast Algorithm for the Decomposition of Graphs and Posets, Journal of Mathematics and Operations Research 8, 170-184, 1983

[84] P. BUNEMAN, A Characterization of Rigid Circuit Graphs, Discrete Mathematics 9, 205-212, 1974

[85] R.E. BURKARD, On the Role of Bottleneck Monge Matrices in Combinatorial Optimization, Operations Research Letters 17, 53-56, 1995

[86] L. CAI, The Complexity of Colouring Parameterized Graphs, University of Hong Kong Technical Report CS-TR-1999-07, 1999

[87] K.B. CAMERON, Polyhedral and Algorithmic Ramifications of Antichains, PhD Thesis, University of Waterloo, 1982

[88] K. CAMERON, J. EDMONDS, Existentially Polytime Theorems, Polyhedral Combinatorics, DIMACS Series on Discrete Mathematics and Theoretical Computer Science 1, 83-100, 1990

[89] K. CAMERON, J. EDMONDS, Some Graphic Uses of an Even Number of Odd Holes, Annals Inst. Fourier (Grenoble) 49, 815-827, 1999

[90] M.C. CARLISLE, E.L. LLOYD, On the k-coloring of Intervals, Discrete Applied Mathematics 59, 225-235, 1995

[91] E. CENEK, L. STEWART, Overlap Graphs of Trees, in preparation

[92] M.S. CHANG, Efficient Algorithms for the Domination Problems on Interval and Circular-Arc Graphs, SIAM Journal on Computing 27, 1671-1694, 1998

[93] M.S. CHANG, Weighted Domination on Cocomparability Graphs, Discrete Applied Mathematics 80, 135-148, 1997

[94] M.S, CHANG, Algorithms for Maximum Matching and Minimum Fill-In on Chordal Bipartite Graphs, Lecture Notes in Computer Science 1178, Algorithms and Computation 96, 146-155, 1996

[95] M.S. CHANG, Y.H. CHEN, G.J. CHANG, J.H. YAN, Algorithmic Aspects of the Generalized Clique-Transversal Problem on Chordal Graphs, Discrete Applied Mathematics 66, 189-203, 1996

[96] M.S. CHANG, S.L. PENG, J.L. LIAW, Deferred-Query: An Efficient Approach for Some Problems on Interval Graphs, Networks 34, 1-10, 1999

[97] M.S. CHANG, S.C. WU, G.J. CHANG, H.G. YEH, Domination in Distance-Hereditary Graphs, Discrete Applied Mathematics 116, 103-113, 2002

[98] H.S. CHAO, F.R. HSU, R.C.T. LEE, An Optimal Algorithm for Finding the Minimum Cardinality Dominating Set on Permutation Graphs, Discrete Applied Mathematics 102, 159-173, 2002

[99] B. CHAZELLE, A Polygon Cutting Theorem, Proceedings of the 23d Symposium on Foundations of Computer Science 339-349, 1982

[100] L. CHEN, Y. YESHA, Efficient Parallel Algorithms for Bipartite Permutation Graphs. Networks 22, 29-39, 1993

[101] N. CHIBA, T. NISHIZEKI, Arboricity and Subgraph Listing Algorithms, SIAM Journal on Computing 14, 210-223, 1985

[102] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, The Strong Perfect Graph Theorem, manuscript. 2002
[103] V. Chvátal, A Combinatorial Theorem in Plane Geometry, Journal of Combinatorial Theory Series B, 39-41, 1975
[104] V. Chvátal, Perfectly Ordered Graphs, Annals of Discrete Mathematics 21, 63-65, 1984
[105] V. Chvátal, Recognizing Decomposable Graphs, Journal of Graph Theory 8, 51-53, 1984
[106] V. Chvátal, Star-Cutsets and Perfect Graphs, Journal of Combinatorial Theory Series B 39, 189-199, 1985
[107] V. Chvátal, Which Claw-Free Graphs are Perfectly Orderable?, Discrete Applied Mathematics 44, 39-63, 1993
[108] V. Chvátal, J. Fonlupt, L. Sun, A. Zemirline, Recognizing Dart-Free Perfect Graphs, SIAM Journal on Computing 31, 1315-1338, 2002
[109] V. Chvátal, C.T. Hoàng, N.V.R. Mahadev, D. de Werra, Four Classes of Perfectly Orderable Graphs, Journal of Graph Theory 11, 481-485, 1987
[110] V. Chvátal, N. Sbihi, Bull-free Berge Graphs are Perfect, Graphs and Combinatorics 3, 127-139, 1987
[111] V. Chvátal, N. Sbihi, Recognizing Claw-Free Perfect Graphs, Journal of Combinatorial Theory Series B 44, 154-176, 1988
[112] B.N. Clark, C.J. Colbourn, D.S. Johnson, Unit Disk Graphs, Discrete Mathematics 86, 165-177, 1990
[113] O. Cogis, A Characterization of Digraphs with Ferrers Dimension 2, Res. Report 19, CNRS Paris, 1979
[114] O. Cogis, On the Ferrers Dimension of a Digraph, Discrete Mathematics 38, 47-52, 1982
[115] C.J. Colbourn, L.K. Stewart, Dominating Cycles in Series Parallel Graphs, Ars Combinatorica 19A, 107-112, 1985
[116] P. Colley, A. Lubiw, J. Spinrad, Visibility Graphs of Towers, Computational Geometry 7, 161-172, 1997
[117] M. Conforti, G. Cornuejols, A. Kapoor, K. Vuskovic, Even-Hole-Free Graphs, Part II: Recognition Algorithm, Journal of Graph Theory 40, 238-266, 2002
[118] D. Coppersmith, S. Winograd, Matrix Multiplication via Arithmetic Progressions, Journal of Symbolic Computation 9, 1-6, 1990
[119] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT Press, Cambridge MA, 1989
[120] D.G. Corneil, personal communication, 1999
[121] D.G. Corneil, J. Fonlupt, The Complexity of Generalized Clique Covering, Discrete Applied Mathematics 22, 109-118, 1988/89
[122] D.G. Corneil, M. Habib, J.M. Lanlignel, B. Reed, U. Rotics, Polynomial Time Algorithm for the 3-Clique-Width Problem, Lecture Notes in Computer Science 1776, LATIN 2000, 126-134, 2000
[123] D. G. Corneil, P. K. Kamula, Extensions of Permutation and Interval Graphs, Congressus Numerantium 58, 267-276, 1987
[124] D.G. Corneil, H. Kim, S. Natarajan, S. Olariu, A.P. Sprague, Simple Linear Time Recognition of Unit Interval Graphs, Information Processing Letters 55, 99-104, 1995
[125] D. Corneil, S. Olariu, L. Stewart, Linear Time Algorithms for Dominating Pairs in Asteroidal Triple-Free Graphs, SIAM Journal on Computing 28, 1284-1297, 1999
[126] D. Corneil, S. Olariu, L. Stewart, Asteroidal Triple-Free Graphs, SIAM Journal on Discrete Mathematics 10, 399-430, 1997
[127] D. Corneil, S. Olariu, L. Stewart, The Ultimate Interval Graph Recognition Algorithm?, SODA, 1998, corrected manuscript 2000
[128] D.G. Corneil, Y. Perl, L.K. Stewart, A Linear Recognition Algorithm for Cographs, SIAM Journal on Computing 14, 926-934, 1985
[129] D.G. Corneil, Y. Perl, L.K. Stewart, Cographs: Recognition, Applications, and Algorithms, Congressus Numerantium 43, 249-258, 1984
[130] G. Cornuejols, X. Liu, K. Vuskovic, Recognizing Three Classes of Perfect Graphs, in preparation, 2002
[131] C. Coullard, A. Lubiw, Distance Visibility Graphs, International Journal of Computational Geometry Applications 2, 349-362, 1992

[132] B. Courcelle, The Monadic Second-Order Logic of Graphs I: Recognizable Sets of Finite Graphs, Information and Computation 85, 12-75, 1990

[133] B. Courcelle, The Monadic Second-Order Logic of Graphs XIV: Uniformly Sparse Graphs and Edge-Set Quantifications, Theoretical Computer Science, to appear

[134] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle Rewriting Graph Grammars, Journal of Computer and System Science 46, 218-270, 1993

[135] B. Courcelle, J.A. Makowsky, U. Rotics, Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width, Theory of Computing Systems 33, 125-150, 2000

[136] A. Cournier, M. Habib, A New Linear Algorithm for Modular Decomposition, Lecture Notes in Computer Science 787, 68-84, 1994

[137] J.C. Culberson, R.A. Reckhow, Covering Polygons is Hard, Journal of Algorithms 17, 2-44, 1991

[138] W.H. Cunningham, Decomposition of Directed Graphs, SIAM Journal on Algebraic Discrete Methods 3, 214-228, 1982

[139] W.H. Cunningham, J. Edmonds, A Combinatorial Decomposition Theory, Canadian Journal of Mathematics 22, 734-765, 1980

[140] I. Dagan, M.C. Golumbic R.Y. Pinter, Trapezoid Graphs and their Coloring, Discrete Applied Mathematics 21, 35-46, 1988

[141] E. Dahlhaus, Chordale Graphen im besonderen Hinblick auf parallele Algorithmen, Habilitation Thesis, Universität Bonn, 1991

[142] E. Dahlhaus, Sequential and Parallel Algorithms on Compactly Represented Chordal and Strongly Chordal Graphs, Lecture Notes in Computer Science 1200, 487-498, 1997

[143] E. Dahlhaus, Parallel Algorithms for Hierarchical Clustering and Applications to Split Decomposition and Parity Graph Recognition, Journal of Algorithms 36, 205-240, 2000

[144] E. Dahlhaus, G. Bailey, Recognition of Path Graphs in Linear Time, Theoretical Computer Science (Ravello 1995), 201-210, World Sci. Publishing, River Edge, NJ, 1996

[145] E. Dahlhaus, J. Gustedt, R.M. McConnell, Efficient and Practical Algorithms for Sequential Modular Decomposition, Journal of Algorithms 41, 360-387, 2001

[146] E. Dahlhaus, J. Gustedt, R.M. McConnell, Partially Complemented Representations of Digraphs, Discrete Mathematics and Theoretical Computer Science, to appear

[147] P. Damaschke, The Hamilton Circuit Problem for Circle Graphs is NP-complete, Information Processing Letters 32, 1-2, 1989

[148] P. Damaschke, Induced Subgraph Isomorphism for Cographs is NP-complete, Lecture Notes in Computer Science 484, WG 90, 72-78, 1991

[149] P. Damaschke, J. Deogun, D. Kratsch, G. Steiner, Finding Hamiltonian Paths in Cocomparability Graphs using the Bump Number, Order 8, 383-391, 1991/92

[150] P. Damaschke, H. Müller, D. Kratsch, Domination in Convex and Chordal Bipartite Graphs, Information Processing Letters 36, 231-236, 1990

[151] X. Deng, P. Hell, J. Huang, Linear-Time Representation Algorithms for Proper Circular-Arc Graphs and Proper Interval Graphs, SIAM Journal on Computing 25, 390-403, 1996

[152] J.S. Deogun, T. Kloks, D. Kratsch, H. Müller, On the Vertex Ranking Problem for Trapezoid, Circular-Arc and Other Graphs, Discrete Applied Mathematics 98, 39-63, 1999

[153] J.S. Deogun, C. Riedesel, Hamiltonian Cycles in Permutation Graphs, Journal of Combinatorial Mathematics and Combinatorial Computing 27, 161-200, 1998

[154] J.S. Deogun, G. Steiner, Polynomial Algorithms for Hamilton Cycle in Cocomparability Graphs, SIAM Journal on Computing 23, 520-552, 1994

[155] U. Derigs, O. Goecke, R. Schrader, Bisimplicial Edges, Gaussian Elimination, and Matchings in Bipartite Graphs, Report 84332-OR, Institut für Operations Research, Universität Bonn

[156] J. Díaz, M.D. Penrose, Approximating Layout Problems on Random Geometric Graphs, Journal of Algorithms 39, 78-116, 2001

[157] J. Díaz, M.D. Penrose, J. Petit, M.J. Serna, Linear Orderings of Random Geometric Graphs, Lecture Notes in Computer Science 1665, WG 99, 291-302, 1999

[158] P.F. Dietz, Intersection Graph Algorithms, Ph.D. Thesis, Comp. Sci. Dept., Cornell University, 1984

[159] R.P. Dilworth, A Decomposition Theorem for Partially Ordered Sets, Annals of Mathematics Series 2, 161-166, 1950

[160] G.A. Dirac, On Rigid Circuit Graphs, Abh. Math. Sem. Univ. Hamburg 25, 71-76, 1961.

[161] D. DOBKIN, H. EDELSBRUNNER, Searching for Empty Convex Polygons, Algorithmica 5, 561-571, 1990
[162] J.P. DOIGNON, A. DUCAMP, J. C. FALMAGNE, On Realizable Biorders and the Biorder Dimension of a Relation, Journal of Mathematical Psychology 28, 77-109, 1984
[163] F.F. DRAGAN, K.F. PRISAKAR, V.D. CHEPOI, The Location Problem on Graphs and the Helly Problem, Diskret. Mat. 4, 67-73, 1992
[164] F.F. DRAGAN, V.I. VOLOSHIN, Incidence Graphs of Biacyclic Hypergraphs, Discrete Applied Mathematics 68, 259-266, 1996
[165] G. DURAN, A. GRAVANO, J. SPINRAD, A.C. TUCKER, Polynomial Time Recognition of Unit Circular-Arc Graphs, manuscript, 2002
[166] B. DUSHNICK, E. W. MILLER, Partially Ordered Sets, American Journal of Mathematics 63, 1941, 600-610
[167] S. EIDENBENZ, C. STAMM, Maximum Clique and Minimum Clique Partition in Visibility Graphs, Lecture Notes in Computer Science 1872, IFIP TCS 00, 200-21, 2000
[168] E.S. ELMALLAH, C.J. COLBOURN, The Complexity of Some Edge Deletion Problems, IEEE Transactions on Circuits and Systems 35, 354-362, 1988
[169] E.S. ELMALLAH, L.K. STEWART, Independence and Domination in Polygon Graphs, Discrete Applied Mathematics 44, 65-77, 1993
[170] E.S. ELMALLAH, L.K. STEWART, Polygon Graph Recognition, Journal of Algorithms 26, 101-140, 1998
[171] P. VAN EMDE BOAS, Preserving Order in a Forest in Less than Logarithmic Time and Linear Space, Information Processing Letters 6, 80-82, 1977
[172] P. ERDÖS, D.B. WEST, A Note on the Interval Number of a Graph, Discrete Mathematics 55, 129-133, 1985
[173] T. ERLEBACH, K. JANSEN, The Complexity of Path Coloring and Call Scheduling, Theoretical Computer Science 255, 33-50, 2001
[174] T. ERLEBACH, K. JANSEN, E. SEIDEL, Polynomial-Time Approximation Schemes for General Graphs, Proceedings of 12th ACM-SIAM Symposium on Discrete Algorithms, 671-679, 2001
[175] E. ESCHEN, Circular-Arc Graph Recognition and Related Problems, Ph.D. Thesis, Dept. of Computer Science, Vanderbilt University, 1996
[176] E. ESCHEN, R. HAYWARD, J. SPINRAD, R. SRITHARAN, Weakly Triangulated Comparability Graphs, SIAM Journal on Computing 29, 378-386, 1999
[177] E. ESCHEN, J. SPINRAD, An $O(n^2)$ Algorithm for Circular-Arc Graph Recognition Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms 128-137, 1993
[178] E. ESCHEN, J. SPINRAD, Finding Triangles in Restricted Classes of Graphs, in preparation, 2002
[179] S. EVEN, Graph Algorithms, Computer Science Press, 1979
[180] H. EVERETT, D. CORNEIL, Negative Results on Visibility Graphs, Computational Geometry Theory and Applications 5, 51-63, 1995
[181] H. EVERETT, S. KLEIN, B. REED, An Algorithm for Finding Homogeneous Pairs, Discrete Applied Mathematics 72, 209-218, 1997
[182] M. FARBER, Domination, Independent Domination, and Duality in Strongly Chordal Graphs, Discrete Applied Mathematics 7, 115-130, 1984
[183] M. FARBER, Characterizations of Strongly Chordal Graphs, Discrete Mathematics 43, 173-189, 1983
[184] M. FARBER, Applications of Linear Programming Duality to Problems Involving Independence and Domination, Ph.D. Thesis, Simon Fraser University, 1981
[185] T. FEDER, P. HELL, S. KLEIN, R. MOTWANI, Complexity of Graph Partition Problems, Proceedings of the 31st Symposium on the Theory of Computing 464-472, 1999
[186] S. FELSNER, Tolerance Graphs and Orders, Journal of Graph Theory 28, 129-140, 1998
[187] S. FELSNER, P.C. FISHBURN, W.T. TROTTER, Finite Three-Dimensional Partial Orders which are not Sphere Orders, Discrete Mathematics 201, 101-132, 1999
[188] S. FELSNER, R. MULLER, L. WERNISCH, Trapezoid Graphs and Generalizations, Geometry and Algorithms, Discrete Applied Mathematics, 13-32, 1997
[189] S. FELSNER, V. RAGHAVAN, J. SPINRAD, Recognition Algorithms for Orders of Small Width and Graphs of Small Dilworth Number, manuscript, 2000
[190] P. FESTA, P.M. PARDALOS, M.G.C. RESENDE, Feedback Set Problems, Handbook of Combinatorial Optimization, Supplement Volume A, 209-258, Kluwer, 1999

[191] C.M. FIDUCCIA, E.R. SCHEINERMAN, A. TRENK, J.S. ZITO, Dot Product Representations of Graphs, Discrete Mathematics 181, 113-138, 1998

[192] C.M.H. DE FIGUEIREDO, S. KLEIN, Y. KOHAYAKAWA, B.A. REED, Finding Skew Partitions Efficiently, Journal of Algorithms 37, 505-521, 2000

[193] C.M.H. DE FIGUEIREDO, K. VUSKOVIC, Recognition of Quasi-Meyniel Graphs, Discrete Applied Mathematics 13, 255-260, 2001

[194] P. C. FISHBURN, Interval Orders and Interval Graphs, Wiley, New York, 1985

[195] P.C. FISHBURN, W.T. TROTTER, Angle Orders, Order 1, 333-343, 1985

[196] S. FOLDES, P.L. HAMMER, Split Graphs, Congressus Numerantium 19, 311-315, 1977

[197] J. FONLUPT, A. ZEMIRLINE, A Polynomial Recognition Algorithm for Perfect $K_4 - e$-Free Graphs, Rev. Maghrébine Math 2, 1-26, 1993

[198] L. FORD, D. FULKERSON, Flows in Networks, Princeton University Press, Princeton NJ, 1962

[199] J.L. FOUQUET, J.L. JOLIVET, M. RIVIERE, An $O(n^3)$ Algorithm to Recognize Bipartite P_6-free Graphs, listed as to appear Journal of Algorithms 1994!!

[200] R.J. FOWLER, M.S. PATERSON, S.L. TANIMOTO, Optimal Packing and Covering in the Plane are NP-complete, Information Processing Letters 12, 133-137, 1981

[201] R. FREIVALDS, Fast Probabilistic Algorithms, Lecture Notes in Computer Science 74, MFOCS 79, 57-69, 1979

[202] A. FRANK, T. NISHIZEKI, N. SAITO, H. SUZUKI, E. TARDOS, Algorithms for Routing Around a Rectangle, Discrete Applied Mathematics 40, 363-378, 1992

[203] D.R. FULKERSON, O.A. GROSS, Incidence Matrices and Interval Graphs, Pacific Journal of Mathematics 15, 835-855, 1965

[204] C.P. GABOR, W.L. HSU, K.J. SUPOWIT, Recognizing Circle Graphs in Polynomial Time, Journal of the ACM 36, 435-473, 1989

[205] T. GALLAI, Transitiv Orientbare Graphen, Acta Math. Acad. Sci. Hung. Tom. 18, 25-66, 1967

[206] M.R. GAREY, D.S. JOHNSON, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1978

[207] M.R. GAREY, D.S. JOHNSON, The Rectilinear Steiner Tree Problem is NP-complete, SIAM Journal on Applied Mathematics 32, 826-834, 1977

[208] M.R. GAREY, D.S. JOHNSON, G.L. MILLER, C.H. PAPADIMITRIOU, The Complexity of Coloring Circular Arcs and Chords, SIAM Journal on Algebraic Discrete Methods 1, 216-227, 1980

[209] A. GARG, R. TAMASSIA, Upward Planarity Testing, Order 12, 109-133, 1995

[210] C. GAVOILLE, D. PELEG, S. PÉRENNES, R. RAZ, Distance Labeling in Graphs, 12th ACM-SIAM Symposium on Discrete Algorithms, 210-219, 2001

[211] F. GAVRIL, Algorithms for a Maximum Clique and a Maximum Independent Set of a Circle Graph, Networks 3, 261-273, 1973

[212] F. GAVRIL, Algorithms on Circular-Arc Graphs, Networks 4, 357-369, 1974

[213] F. GAVRIL, Maximum Weight Independent Sets and Cliques in Intersection Graphs of Filaments, Information Processing Letters 73, 181-188, 2000

[214] F. GAVRIL, Algorithms for Maximum Weight Induced Paths, Information Processing Letters 81, 203-208, 2002

[215] F. GAVRIL, A Recognition Algorithm for the Intersection Graphs of Directed Paths in Directed Trees, Discrete Mathematics 13, 237-249, 1975

[216] F. GAVRIL, The Intersection Graphs of Subtrees of Trees are Exactly the Chordal Graphs, Journal of Combinatorial Theory Series B 16, 47-56, 1974

[217] F. GAVRIL, Algorithms for Maximum k-colorings and k-coverings of Transitive Graphs, Networks 17, 465-470, 1987

[218] S. GHOSH, On Recognizing and Characterizing Visibility Graphs of Simple Polygons, Lecture Notes in Computer Science 318, SWAT 88, 96-104, 1988

[219] V. GIAKOUMAKIS, I. RUSU, Weighted Parameters in $P_5,\overline{P_5}$-free Graphs, Discrete Applied Mathematics 80, 255-261, 1997

[220] P.C. GILMORE, A.J. HOFFMAN, A Characterization of Comparability Graphs and Interval Graphs, Canadian Journal of Mathematics 16, 539-548, 1964

[221] F. GLOVER, Maximum Matching in a Convex Bipartite Graph, Naval Research Logistics Quarterly 14, 313-316, 1967

[222] L. GOH, D. ROTEM, Recognition of Perfect Elimination Bipartite Graphs, Information Processing Letters 15, 179-182, 1982

[223] M.C. GOLUMBIC, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980

[224] M.C. GOLUMBIC, C.F. GOSS, Perfect Elimination and Chordal Bipartite Graphs, Journal of Graph Theory 2, 155-163, 1978

[225] M.C. GOLUMBIC, P.L. HAMMER, Stability in Circular-Arc Graphs, Journal of Algorithms 9, 314-320, 1988

[226] M.C. GOLUMBIC, R. E. JAMISON, The Edge Intersection Graphs of Paths in Trees, Journal of Combinatorial Theory Series B 38, 8-22, 1985

[227] M.C. GOLUMBIC, R.E. JAMISON, Edge and Vertex Intersection of Paths in a Tree, Discrete Mathematics 55, 151-159, 1985

[228] M.C. GOLUMBIC, H. KAPLAN, R. SHAMIR, Graph Sandwich Problems, Journal of Algorithms 19, 449-473, 1995

[229] M.C. GOLUMBIC, C.L. MONMA, W.T. TROTTER, Tolerance Graphs, Discrete Applied Mathematics 9, 157-170, 1984

[230] M.C. GOLUMBIC, U. ROTICS, On the Clique-Width of Some Perfect Graph Classes, International Journal on Foundations of Computer Science 11, 423-443, 2000

[231] M.C. GOLUMBIC, E.R. SCHEINERMAN, Containment Graphs, Posets, and Related Classes of Graphs, Annals New York Academy of Sciences 555, 192-204, 1989

[232] M.C. GOLUMBIC, A. SIANI, Coloring Algorithms for Tolerance Graphs: Reasoning and Scheduling with Interval Constraints, proceedings of AISC 2002, to appear

[233] A. GRÄF, Coloring and Recognizing Special Graph Classes, PhD Thesis, Johannes Gutenberg Universität, Mainz, 1994, and report 20/95 in Musikinformatik und MedienTechnik

[234] A. GRÄF, M. STUMPF, G. WEISSENFELS, On Coloring Unit Disk Graphs, Algorithmica 20, 1998

[235] R.L. GRAHAM, B.L. ROTHSCHILD, J.H. SPENCER, Ramsey Theory, Wiley, New York, 1990

[236] J.R. GRIGGS, Extremal Values of the Interval Number of a Graph, II, Discrete Mathematics 28, 37-47, 1989

[237] M. GRÖTSCHEL, L. LOVÁSZ, A. SCHRIJVER, The Ellipsoid Method and its Consequences in Combinatorial Optimization, Combinatorica 1, 169-197, 1981

[238] V. GURUSWAMI, Maximum Cut on Line and Total Graphs, Discrete Applied Mathematics 92, 217-221, 1999

[239] V. GURUSWAMI, C.PANDU RANGAN, M.S. CHANG, G.J. CHANG, C.K. WONG, The K_r-Packing Problem, Computing 66, 79-89, 2001

[240] J. GUSTEDT, On the Pathwidth of Chordal Graphs, Discrete Applied Mathematics 45, 233-248, 1993

[241] W. GUTJAHR, E. WELZL, G. WOEGINGER, Polynomial Graph Colorings, Discrete Applied Mathematics 35, 29-45, 1992

[242] M. HABIB, R. JEGOU, N-free Posets as Generalizations of Series-Parallel Posets, Discrete Applied Mathematics 3, 279-291, 1985

[243] M. HABIB, D. KELLY, R.H. MÖHRING, Interval Dimension is a Comparability Invariant, Discrete Mathematics 88, 211-229, 1991

[244] M. HABIB, M.C. MAURER, On the X-join Decomposition for Undirected Graphs, Discrete Applied Mathematics 3, 198-207, 1979

[245] H. HADWIGER, H. DEBRUNNER, V. KLEE, Combinatorial Geometry in the Plane, Holt Rinehardt and Winston, New York, 1964, p. 54

[246] P.L. HAMMER, F. MAFFRAY, Completely Separable Graphs, Discrete Applied Mathematics 27, 85-100, 1990

[247] P.L. HAMMER, N.V.R. MAHADEV, D. DE WERRA, Stability in CAN-free Graphs, Journal of Combinatorial Theory Series B 38, 23-30, 1985

[248] P.L. HAMMER, N.V.R. MAHADEV, D. DE WERRA, The Struction of a Graph: Application to CN-free Graphs, Combinatorica 5, 141-147, 1985

[249] P.L. HAMMER, B. SIMEONE, The Splittance of a Graph, Combinatorica 1, 275-284, 1981

[250] K. HAN, C.N. SEKHARAN, R. SRIDHAR, Unified All-Pairs Shortest Path Algorithms in the Chordal Hierarchy, Discrete Applied Mathematics 77, 59-71, 1997

[251] D. HAREL, R.E. TARJAN, Fast Algorithms for Finding Nearest Common Ancestors, SIAM Journal on Computing 13, 338-355, 1984

[252] I.B.-A. HARTMAN, I. NEWMAN, R. ZIV, On Grid Intersection Graphs, Discrete Mathematics 87, 41-52, 1991

[253] R. HAYWARD, Weakly Triangulated Graphs, Journal of Combinatorial Theory Series B 39, 200-209, 1985

[254] R. HAYWARD, C. HOÁNG, F. MAFFRAY, Optimizing Weakly Triangulated Graphs, Graphs and Combinatorics 5, 339-349, 1989

[255] R. HAYWARD, R. SHAMIR, Tolerance Graphs are in NP, in preparation

[256] R. HAYWARD, J. SPINRAD, R. SRITHARAN, Weakly Chordal Graph Algorithms via Handles, Proceeding of the 11th Symposium on Discrete Algorithms, 42-49, 2000

[257] P. HLINENY, The Maximal Clique and Colourability of Curve Contact Graphs, Discrete Applied Mathematics 81, 59-68, 1998

[258] P. HLINENY, J. KRATOCHVIL, Representing Graphs by Disks and Balls (A Survey of Recognition-Complexity Results), Discrete Mathematics 229, 101-124, 2001

[259] C.T. HOÀNG, Efficient Algorithms for Minimum Weighted Colouring of Some Classes of Perfect Graphs, Discrete Applied Mathematics 55, 133-143, 1994

[260] C.T. HOÀNG, On the Complexity of Recognizing a Class of Perfectly Orderable Graphs, Discrete Applied Mathematics 66, 219-226, 1996

[261] C.T. HOÁNG, Perfectly Orderable Graphs: A Survey, in Perfect Graphs, Ed. J.L. Ramirez and B.A. Reed, Wiley, Chichester, 139-166, 2001

[262] C.T. HOÀNG, V.B. LE, Recognizing Perfect 2-Split Graphs, SIAM Journal on Discrete Mathematics 13, 48-55, 2000

[263] C.T. HOÁNG, N.V.R. MAHADEV, A Note on Perfect Orders, Discrete Mathematics 74, 77-84, 1989

[264] A.J. HOFFMAN, A.W.J. KOLEN, M. SAKAROVITCH, Totally-Balanced and Greedy Matrices, SIAM Journal on Algebraic Discrete Methods 6, 721-730, 1985

[265] W.L. HSU, O(mn) Algorithms for the Recognition and Isomorphism Problems on Circular-Arc Graphs, SIAM Journal on Computing 24, 411-439, 1995

[266] W.L. HSU, Recognizing Planar Perfect Graphs, Journal of the ACM 34, 255-288, 1987

[267] W.L. HSU, Decomposition of Perfect Graphs, Journal of Combinatorial Theory, Series B 43, 70-94, 1987

[268] W.L. HSU, T.H. MA, Fast and Simple Algorithms for Recognizing Chordal Comparability Graphs and Interval Graphs, SIAM Journal on Computing 28, 1004-1020, 1999

[269] W.L. HSU, W.K. SHIH, An Approximation Algorithm for Coloring Circular-Arc Graphs, manuscript, 1996

[270] W.L. HSU, J. SPINRAD, Independent Sets in Circular-Arc Graphs, Journal of Algorithms 19, 145-160, 1995

[271] W.L. HSU, K.H. TSAI, Linear Time Algorithms on Circular-Arc Graphs, Information Processing Letters 40, 123-129, 1991

[272] L.T.Q. HUNG, M.M. SYSLO,M.L. WEAVER,D.B. WEST, Bandwidth and Density for Block Graphs, Discrete Mathematics 189, 163-176, 1998

[273] H.B. HUNT, M.V. MARATHE, V. RADHAKRISHNAN, S.S. RAVI, D.J. ROSENKRANTZ, NC-Approximation Schemes for NP- and PSPACE-Hard Problems on Geometric Graphs, Journal of Algorithms 26, 238-274, 1998

[274] T. IBARAKI, U. N. PELED, Sufficient Conditions for Graphs to have Threshold Dimension 2, Studies on Graphs and Discrete Programming, Annals of Discrete Mathematics 11, 241-268, 1981

[275] O.H. IBARRA, Q. ZHENG, Some Efficient Algorithms for Permutation Graphs, Journal of Algorithms 16, 453-469, 1994

[276] H. IMAI, T. ASANO, Finding the Connected Components and Maximum Clique of an Intersection Graph of Rectangles in the Plane, Journal of Algorithms 4, 310-323, 1983

[277] M.D. IORDACHE, A $2+\epsilon$-Approximation Scheme for Minimum Domination on Circle Graphs, Journal of Algorithms 42, 255-276, 2002

[278] A. ITAI, C.H. PAPADIMITRIOU, J.L. SZWARCFITER, Hamilton Paths in Grid Graphs, SIAM Journal on Computing 11, 676-686, 1982

[279] K. JANSEN, P. SCHEFFLER, Generalized Coloring for Tree-like Graphs, Discrete Applied Mathematics 75, 135-155, 1997

[280] T, JIANG, D. MUBAYI, A. SASTRI, D.B. WEST, Edge-Bandwidth of Graphs, SIAM Journal on Discrete Mathematics 12, 307-316, 1999

[281] O. JOHANSSON, log$n$-Approximative NLC_k-Decomposition in $O(n^{2k+1})$ Time, Lecture Notes in Computer Science 2204, WG 01, 229-240, 2001

[282] D.S. JOHNSON, The NP-Completeness Column: An Ongoing Guide, Journal of Algorithms 6, 434-451, 1985

[283] D.S. JOHNSON, The NP-Completeness Column: An Ongoing Guide: Announcements, Updates, and Greatest Hits, Journal of Algorithms 8, 438-448, 1987

[284] D.S. JOHNSON, Worst Case Behavior of Graph Coloring Algorithms, Proceedings of the 5th Southeastern Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium 10, 513-528, 1974

[285] D.S. JOHNSON, M. YANNAKAKIS, C.H. PAPADIMITRIOU, On Generating All Maximal Independent Sets, Information Processing Letters 27, 119-123, 1988

[286] J. JOHNSON, R. MCCONNELL, J. SPINRAD, Linear Time Recognition of Probe Interval Graphs, in preparation, 2002

[287] J. JOHNSON, J. SPINRAD, A Polynomial Time Recognition Algorithm for Probe Interval Graphs, Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, 477-486, 2001

[288] C.M.L. JONES, The Weighted Feedback Vertex Set Problem on Distance Hereditary Graphs, MS Thesis, Cung Cheng University, 2000

[289] D. JOSEPH, J. MEIDANIS, P. TIWARI, Determining DNA Sequence Similarity Using Maximum Independent Set Algorithms for Interval Graphs, Lecture Notes in Computer Science 621, SWAT 92, 326-337, 1992

[290] S. KANNAN, M. NAOR, S. RUDICH, Implicit Representation of Graphs, SIAM Journal on Discrete Mathematics 5, 596-603, 1992

[291] S. KANNAN, T. PROEBSTING, Register Allocation in Structured Programs, Journal of Algorithms 29, 223-237, 1998

[292] I.A. KARAPETJAN, Coloring of Arc Graphs, Akad. Nauk Armyan SSR Dokl 70, 306-311, 1980

[293] T. KASHIWABARA, S. MASUDA, K. NAKAJIMA, T. FUJISAWA, Polynomial Time Algorithms on Circular-Arc Overlap Graphs, Networks 21, 195-203, 1991

[294] M. KATZ, N. KATZ, D. PELEG, Distance Labeling Schemes for Well-Separated Graph Classes, Lecture Notes in Computer Science 1770, STACS 00, 516-528, 2000

[295] J.M. KEIL, The Complexity of Domination Problems in Circle Graphs, Discrete Applied Mathematics 42, 51-63, 1993

[296] M. KEIL, P. BELLEVILLE, Dominating the Complements of Bounded Tolerance Graphs and Trapezoid Graphs, manuscript, 2000

[297] H.A. KIERSTEAD, W.T. TROTTER, J. QIN, The Dimension of Cycle-Free Orders, Order 9, 103-110, 1992

[298] D.J. KLEITMAN, K.J. WINSTON, The Asymptotic Number of Lattices, Annals of Discrete Mathematics 6, 243-249, 1980

[299] B. KLINZ, R. RUDOLF, G. WOEGINGER, Permuting Matrices to Avoid Forbidden Submatrices Discrete Applied Mathematics 60, 223-248, 1995

[300] B. KLINZ, R. RUDOLF, G.J. WOEGINGER, On the Recognition of Permuted Bottleneck Monge Matrices, Discrete Applied Mathematics 63, 43-74, 1995

[301] T. KLOKS, Treewidth – Computations and Approximations, Springer Verlag, Lecture Notes in Computer Science 842, 1994

[302] T. KLOKS, Treewidth of Circle Graphs, Lecture Notes in Computer Science 762, Algorithms and Computation 93, 108-117, 1993

[303] T. KLOKS, H. BODLAENDER, H. MÜLLER, D. KRATSCH, Computing Treewidth and Minimum Fill-in: all you Need are the Minimal Separators, Lecture Notes in Computer Science 726, ESA 93, 260-271, 1993

[304] T. KLOKS, D. KRATSCH, Listing all Minimal Separators of a Graph, SIAM Journal on Computing 27, 605-613, 1998

[305] T. KLOKS, D. KRATSCH, Treewidth of Chordal Bipartite Graphs, Journal of Algorithms 19, 266-281, 1995

[306] T. KLOKS, D. KRATSCH, Y. LEBORGNE, H. MULLER, Bandwidth of Split and Circular Permutation Graphs, Lecture Notes in Computer Science 1928, WG 00, 243-254, 2000

[307] T. KLOKS, D. KRATSCH, H. MÜLLER, Finding and Counting Small Induced Subgraphs Efficiently, Lecture Notes in Computer Science 1017, WG 95, 14-23, 1995

[308] T. KLOKS, D. KRATSCH, H. MÜLLER, Approximating the Bandwidth for Asteroidal Triple-Free Graphs, Journal of Algorithms 32, 41-57, 1999

[309] T. KLOKS, D. KRATSCH, H. MÜLLER, Bandwidth of Chain Graphs, Information Processing Letters 68, 313-315, 1998

[310] T. KLOKS, D. KRATSCH, J. SPINRAD, On Treewidth and Minimum Fill-in of Asteroidal Triple-free Graphs, Theoretical Computer Science 175, 309-335, 1997

[311] T. KLOKS, D. KRATSCH, C.K WONG, Minimum Fill-In on Circular and Circular-Arc Graphs, Journal of Algorithms 28, 272-289, 1998

[312] T. KLOKS, H. MÜLLER, C.K. WONG, Vertex Ranking of Asteroidal Triple-free Graphs, Information Processing Letters 68, 201-206, 1998

[313] D. KOBLER, U. ROTICS, Edge Dominating Sets and Colorings on Graphs with Fixed Clique-Width, Discrete Applied Mathematics, in press, 2002

[314] M. KOEBE, Colouring of Spider Graphs, Topics in Combinatorics and Graph Theory, Physica Verlag, Heidelberg, 1990, 435-441

[315] M. KOEBE, On a New Class of Intersection Graphs, unpublished manuscript, includes results published under the same title in Fourth Czechoslovakian Symposium on Combinatorics, Graphs, and Complexity, Elsevier, 141-143, 1992, as well as a proposed recognition algorithm

[316] E. KÖHLER, Graphs without Asteroidal Triples, Ph.D. Thesis, TU-Berlin, 1999

[317] E. KÖHLER, D.G. CORNEIL, S. OLARIU, L. STEWART, On Subfamilies of AT-free Graphs, Lecture Notes in Computer Science 2204, WG 01, 241-253, 2001

[318] N. KORTE, R.H. MÖHRING, Transitive Orientation of Graphs with Side Constraints, Lecture Notes on Computer Science 246, WG 85, 1-16, 1987

[319] A. KOSTOCHKA, J. KRATOCHVÍL, Covering and Coloring Polygon-Circle Graphs, Discrete Mathematics 163, 299-305, 1997

[320] J. KRATOCHVÍL, A Special Planar Satisfiability Problem and a Consequence of its NP-completeness, Discrete Applied Mathematics 52, 233-252, 1994

[321] J. KRATOCHVÍL, k-Ramsey Classes and Dimension of Graphs, Commentationes Mathematicae Universitatis Carolinae 36, 263-269, 1995

[322] J. KRATOCHVÍL, Intersection Graphs of Noncrossing Arc-Connected Sets in the Plane, Lecture Notes in Computer Science 1190, Graph Drawing 96, 257-270, 1996

[323] J. KRATOCHVÍL, A. KUBENA, On Intersection Representations of Co-Planar Graphs, Discrete Mathematics 178, 251-255, 1998

[324] J. KRATOCHVÍL, J. MATOUSEK, NP-hardness Results for Intersection Graphs, Commentationes Mathematicae Universitatis Carolinae 30, 761-773, 1989

[325] J. KRATOCHVÍL, J. MATOUSEK, Intersection Graphs of Segments, Journal of Combinatorial Theory Series B 62, 289-315, 1994

[326] J. KRATOCHVÍL, J. NESETRIL, INDEPENDENT SET and CLIQUE Problems in Intersection-Defined Classes of Graphs, Commentationes Mathematicae Universitatis Carolinae 31, 85-93, 1990

[327] J. KRATOCHVÍL, Z. TUZA, Intersection Dimensions of Graph Classes, Graphs and Combinatorics 10, 159-168, 1994

[328] D. KRATSCH, personal communication, 2002

[329] D. KRATSCH, Algorithms, from Domination in Graphs, Monograph Textbooks in Pure and Applied Mathematics 209, 191-231, 1998

[330] D. KRATSCH, Domination and Total Domination on Asteroidal Triple-Free Graphs, Discrete Applied Mathematics 99, 111-123, 2000

[331] D. KRATSCH, Finding Dominating Cliques Efficiently in Strongly Chordal Graphs and Undirected Path Graphs, Discrete Mathematics 86, 225-238, 1990

[332] D. KRATSCH, J. SPINRAD, Between $O(nm)$ and $O(n^{\alpha})$, to appear, SODA 2003

[333] D. KRATSCH, L. STEWART, Domination of Cocomparability Graphs, SIAM Journal on Discrete Mathematics 6, 400-417, 1993

[334] D. KRATSCH, L. STEWART, Approximating Bandwidth by Mixing Layouts of Interval Graphs, SIAM Journal on Discrete Mathematics 15, 435-449, 2002

[335] V. KUMAR, An Approximation Algorithm for Circular-Arc Colouring, Algorithmica 30, 406-417, 2001

[336] L.J. LANGLEY, A Note on Bipartite Interval Tolerance Graphs, Congressus Numerantium 102, 191-192, 1994

[337] D.T. LEE, J.Y.-T. LEUNG, On the 2-Dimensional Channel Assignment Problem, IEE Transactions on Computing, C-33, 2-6, 1984

[338] J. VAN LEEUWEN, Graph Algorithms, in Handbook of Theoretical Computer Science Volume A, ed. J. van Leeuwen, 525-631, Elsevier Science Publishers, Amsterdam, MIT Press, Cambridge, MA, 1990

[339] J. LEHEL, A Characterization of Totally Balanced Hypergraphs, Discrete Mathematics 57, 59-65, 1985

[340] P.G.H. LEHOT, An Optimal Algorithm to Detect a Line Graph and Output its Root Graph, Journal of the ACM 21, 569-575, 1974

[341] C.G. LEKKERKERKER, J. BOLAND, Representation of a Finite Graph by a Set of Intervals on the Line, Fund Math. 51, 45-64, 1962

[342] H. Lerchs, On Cliques and Kernels, Technical Report, Dept. of Compueter Science, University of Toronto, 1971

[343] Y.D. LIANG, On the Feedback Vertex Set Problem in Permutation Graphs, Information Processing Letters 52, 123-129, 1994

[344] Y.D. LIANG, Steiner Set and Connected Domination in Trapezoid Graphs, Information Processing Letters 56, 101-108, 1995

[345] Y.D. LIANG, Dominations in Trapezoid Graphs, Information Processing Letters 52, 309-315, 1994

[346] Y.D. LIANG, N. BLUM, Circular Convex Graphs: Maximum Matching and Hamiltonian Circuits, Information Processing Letters 56, 215-219, 1995

[347] Y.D. LIANG, M.S. CHANG, Minimum Feedback Vertex Sets in Cocomparability Graphs and Convex Bipartite Graphs, Acta Informatica 34, 337-346, 1997

[348] Y.D. LIANG, C. RHEE, Finding a Maximum Matching in a Circular-Arc Graph, Information Processing Letters 45, 185-190, 1993

[349] Y-L LIN, S. SKIENA, Complexity Aspects of Visibility Graphs, International Journal of Computational Geometry 5, 289-312, 1995

[350] X. LIU. K. VUSKOVIC, Perfection-Preserving Decompositions, in preparation, 2002

[351] R.-D. LOU, M. SARRAFZADEH, Circular Permutation Graph Family with Applications, Discrete Applied Mathematics 40, 433-457, 1992

[352] L. LOVASZ, Normal Hypergraphs and the Perfect Graph Conjecture, Discrete Mathematics 2, 253-267, 1972

[353] C.L. LU, C.Y. TANG, A Linear-Time Algorithm for the Weighted Feedback Vertex Problem on Interval Graphs, Information Processing Letters 61, 107-111, 1997

[354] A. LUBIW, Doubly Lexical Orderings of Matrices, SIAM Journal on Computing 16, 854-879, 1987

[355] G.S. LUEKER, K.S. BOOTH, A Linear Time Algorithm for Deciding Interval Graph Isomorphism, Journal of the ACM 26, 183-195, 1979

[356] T.-H. MA, J. SPINRAD, Cycle-free Partial Orders and Chordal Comparability Graphs, Order 8, 49-61, 1991

[357] T.-H. MA, J. SPINRAD, On the 2-Chain Subgraph Cover and Related Problems, Journal of Algorithms 1994, 251-268

[358] T.-H. MA, J. SPINRAD, Transitive Closure for Restricted Classes of Partial Orders, Order 8, 175-183, 1991

[359] T.-H. MA, J. SPINRAD, An $O(n^2)$ Algorithm for the Undirected Split Decomposition, Journal of Algorithms 16, 264-282, 1994

[360] H. MAEHARA, Space Graphs and Sphericity, Discrete Applied Mathematics 7, 55-64, 1984

[361] N.V.R. MAHADEV, U.N. PELED, Threshold Graphs and Related Topics, Annals of Discrete Mathematics 56, North Holland, 1995

[362] M.V. MARATHE, H. BREU, H.B. HUNT, S.S. RAVI, D.J. ROSENKRANTZ, Simple Heuristics for Unit Disk Graphs, Networks 25, 59-68, 1995

[363] S. MASUDA, K. NAKAJIMA, An Optimal Algorithm for Finding a Maximum Independent Set of a Circular-Arc Graph, SIAM Journal on Computing 17, 41-52, 1988

[364] S. MASUDA, K. NAKAJIMA, T. KASHIWABARA, T. FUJISAWA, Efficient Algorithms for Finding Maximum Cliques of an Overlap Graph, Networks 20, 157-171, 1990

[365] R.M. MCCONNELL, Linear-Time Recognition of Circular-Arc Graphs, Proceedings of the 42d IEEE Symposium on Foundations of Computer Science, 386-394, 2001

[366] R.M. McConnell, J. Spinrad, Linear-time Modular Decomposition and Efficient Transitive Orientation of Comparability Graphs, Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms, 536-545, 1994

[367] R.M. McConnell, J. Spinrad, Modular Decomposition and Transitive Orientation, Discrete Mathematics 201, 189-241, 1999

[368] T.A. McKee, F.R. McMorris, Topics in Intersection Graph Theory, SIAM, Philadelphia, 1999

[369] M. Middendorf, F. Pfeiffer, On the Complexity of Recognizing Perfectly Orderable Graphs, Discrete Mathematics 80, 1990, 327-333

[370] G.J. Minty, On Maximal Independent Sets of Vertices in Claw-free Graphs, Journal of Combinatorial Theory Series B 28, 284-304, 1980

[371] J. Mitas, Minimal Representation of Semiorders with Intervals of Same Length, Lecture Notes in Computer Science 831, Ordal 94, 162-175, 1994

[372] R.H. Möhring, Almost all Comparability Graphs are UPO, Discrete Mathematics 50, 63-70, 1984

[373] R.H. Möhring, F.J. Radermacher, Substitution Decomposition for Discrete Structures and Connections with Combinatorial Optimization, Annals of Discrete Mathematics 19, 257-356, 1984

[374] C.L. Monma, V.K. Wei, Intersection Graphs of Paths in a Tree, Journal of Combinatorial Theory Series B 41, 141-181, 1986

[375] M. Moscarini, Doubly Chordal Graphs, Steiner Trees, and Connected Domination, Networks 23, 59-69, 1993

[376] J.H. Muller, Local Structure in Graphs Ph.D. Thesis, School of Information and Computer Science, Georgia Institute of Technology Atlanta, GA, 1988

[377] J.H. Muller, J. Spinrad, Incremental Modular Decomposition, Journal of the ACM 36, 1-19, 1989

[378] H. Müller, Hamilton Circuits in Chordal Bipartite Graphs, Discrete Mathematics 156, 291-298, 1996

[379] H. Müller, A. Brandstädt, The NP-completeness of Steiner Tree and Dominating Set for Chordal Bipartite Graphs, Theoretical Computer Science 53, 257-265, 1987

[380] T. Nagoya, R. Uehara, S. Toda, Completeness of Graph Isomorphism Problem for Bipartite Graph Classes, IEICE Technical Report COMP2001-93, 2001

[381] G. Narasimhan, A Note on the Hamilon Circuit Problem on Directed Path Graphs, Information Processing Letters 32, 167-170, 1989

[382] G. Narasimhan, R. Manber, Stability and Chromatic Number of Tolerance Graphs, Discrete Applied Mathematics 36, 47-56, 1992

[383] A. Natanzon, R. Shamir, R. Sharan, Complexity Classification of Some Edge Modification Problems, Discrete Applied Mathematics 113, 109-128, 2001

[384] J. Nesetril, S. Poljak, On the Complexity of the Subgraph Problem, Commentationes Mathematicae Universitatis Carolinae 26, 415-419, 1985

[385] F. Nicolai, Hamiltonian Problems on Distance-Hereditary Graphs, Technical Report, Gerhard-Mercator-Universitat Gesamthochschule Duisburg SM-DU-255, 1994

[386] F. Nicolai, T. Szymczak, Homogeneous Sets and Domination: A Linear Time Algorithm for Distance Hereditary Graphs, Networks 37, 117-128, 2001

[387] C. Nomikos, Path Coloring Problems in Graphs are Non-Approximable, manuscript, 2000

[388] T. Ohtsuki, A Fast Algorithm for Finding an Optimal Ordering for Vertex Elimination on a Graph, SIAM Journal on Computing 5, 133-145, 1976

[389] S. Olariu, Paw-Free Graphs, Information Processing Letters 28, 53-54, 1998

[390] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, 1987

[391] R. Paige, R.E. Tarjan, Three Partition Refinement Algorithms, SIAM Journal on Computing 16, 973-989, 1987

[392] C.H. Papadimitriou, M. Yannakakis, Scheduling Interval-Ordered Tasks, SIAM Journal on Computing 8, 405-409, 1979

[393] A. Parra, Triangulating Multitolerance Graphs, Discrete Applied Mathematics 84, 183-197, 1998

[394] M.C. Paull, S.H. Unger, Minimizing the Number of States in Incompletely Specified Sequential Switching Functions, IRE Trans. Electronic Computers EC-8, 356-367, 1959

[395] C. Payan, Perfectness and Dilworth Number, Discrete Mathematics 44, 229-230, 1983

[396] D. PELEG, Proximity Preserving Labeling Schemes and their Applications, Journal of Graph Theory 33, 167-176, 2000

[397] U. PFERSCHY, R. RUDOLF, G.J. WOEGINGER, Monge Matrices Make Maximization Manageable, Operations Research Letters 16, 245-254, 1994

[398] M.D. PLUMMER, Well-Covered Graphs: A Survey, Quaestiones Mathematicae 16, 253-287, 1993

[399] E. PRISNER, A Journey Through Intersection Graph County, http:// www.math.uni-hamburg.de /spag/gd/mitarbeiter /prisner/ Pris/Rahmen.html

[400] H.J. PRÖMEL, A. STEGER, Excluding Induced Subgraphs III. A General Asymptotic, Random Structures and Algorithms 3, 19-31, 1992

[401] I. RABINOVITCH, The Dimension of Semiorders, Journal of Combinatorial Theory Series A 25, 50-61, 1978

[402] V. RAGHAVAN, J. SPINRAD, Robust Algorithms for Restricted Domains, Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, 460-467, 2001

[403] J.L. RAMÍREZ ALFONSÍN, B.A. REED, Perfect Graphs, Wiley, Chichester, 2001

[404] A.S. RAO, C. PANDU RANGAN, Optimal Parallel Algorithms on Circular-Arc Graphs, Information Processing Letters 33, 147-156, 1989

[405] B. REED, N. SBIHI, Recognizing Bull-Free Perfect Graphs, Graphs and Combinatorics 11, 171-178, 1995

[406] C. RHEE, Y.D. LIANG, Finding a Maximum Matching in a Permutation Graph, Acta Informatica 32, 779-792, 1995

[407] C. RHEE, Y.D. LIANG, S.K. DHALL, S. LAKSHIVARAHAN, An $O(m+n)$ Algorithm for Finding Minimum Weight Dominating Set in Permutation Graphs, SIAM Journal on Computing 25, 404-419, 1996

[408] C.S. RIM, K. NAKAJIMA, On Rectangle Intersection and Overlap Graphs, IEEE Trans. Circuits Systems/ Fund. Theory Appl. 42, 549-553, 1995

[409] R.L. RIVEST, J. VUILLEMIN, On Recognizing Graph Properties from Adjacency Matrices, Theoretical Computer Science 3, 371-384, 1976/77

[410] F.S. ROBERTS, Indifference Graphs, Proof Techniques in Graph Theory, Ed. F. Harary, Academic Press, New York, 1969, 139-146

[411] F.S. ROBERTS, Graph Theory and Its Application to Problems of Society, SIAM, 1978

[412] N. ROBERTSON, P.D. SEYMOUR, Graph Minors - A Survey, in Surveys in Combinatorics, I. Anderson, ed., Cambridge University Press, Cambridge, 153-171, 1985

[413] D.J. ROSE, Triangulated Graphs and the Elimination Process, Journal of Mathematical Analysis and Applications 32, 597-609, 1970

[414] D.J. ROSE, R.E. TARJAN, G.S. LUEKER, Algorithmic Aspects of Vertex Elimination on Graphs, SIAM Journal on Computing 5, 266-283, 1976

[415] D. ROTEM, J. URRUTIA, Circular Permutation Graphs, Networks 12, 428-437, 1982

[416] N.D. ROUSSOPOLOUS, A O(max(m,n)) Algorithm for Determining the Graph H from its Line Graph, Information Processing Letters 2, 108-112, 1973

[417] I. RUSU, J. SPINRAD, Forbidden Subgraph Decomposition, Discrete Mathematics 247, 159-168, 2002

[418] R.S. SANKARANARAYANA, Tracking the P-NP Boundary for Well-Covered Graphs, Networks 29, 29-37, 1997

[419] R.S. SANKARANARAYANA, L.K. STEWART, Complexity Results for Well Covered Graphs, Networks 22, 247-262, 1992

[420] N. SBIHI, Algorithme de Recherche d'un Stable de Cardinalité Maximum dans un Graphe sans Étoile, Discrete Mathematics 29, 53-76, 1980

[421] M. SCHAEFER, E. SEDGWICK, D. STEFANKOVIC, Recognizing String Graphs is in NP, 34th Symposium on the Theory of Computing, 1-6, 2002

[422] A.A. SCHÄFFER, A Faster Algorithm to Recognize Undirected Path Graphs, Discrete Applied Mathematics 43, 261-295, 1993

[423] E. SCHEINERMAN, Local Representation Using Very Short Labels, Discrete Mathematics 203, 287-290, 1999

[424] E.R. SCHEINERMAN, The Many Faces of Circle Orders, Order 9, 343-348, 1992

[425] E.R. SCHEINERMAN, Characterizing Intersection Classes of Graphs, Discrete Mathematics 55, 185-193, 1985

[426] E.R. SCHEINERMAN, D.B. WEST, The Interval Number of a Planar Graph: Three Intervals Suffice, Journal of Combinatorial Theory Series B 35, 224-239, 1983

[427] E. SCHEINERMAN, J. ZITO, On the Size of Hereditary Classes of Graphs, Journal of Combinatorial Theory Series B 61, 16-39, 1994

[428] W. SCHNYDER, Planar Graphs and Poset Dimension, Order 5, 323-343, 1989

[429] A. SEBÖ, Integer Polyhedra: Combinatorial Properties and Complexity, working paper, Leibniz-IMAG, France, 2001

[430] F.B. SHEPHERD, The Theta Body and Imperfection, Perfect Graphs, Wiley, Chichester, 261-291, 2001

[431] T. SHERMER, Recent Results in Art Galleries, Proceedings of the IEEE 80, 1992

[432] W.K. SHIH, T.C. CHERN, W.L. HSU, An $O(n^2 \log n)$ Algorithm for the Hamiltonian Cycle Problem on Circular-Arc Graphs, SIAM Journal on Computing 21, 1026-1046, 1992

[433] W,K. SHIH, W.L. HSU, An $O(n \log n + m \log\log n)$ Maximum Weight Clique Algorithm for Circular-Arc Graphs, Information Processing Letters 31, 129-134, 1989

[434] D.J. SKRIEN, A Relationship between Triangulated Graphs, Comparability Graphs, Proper Interval Graphs, Proper Circular-Arc Graphs, and Nested Interval Graphs, Journal of Graph Theory 6, 309-316, 1982

[435] J. SPINRAD, Nonredundant Ones and Chordal Bipartite Graphs SIAM Journal on Discrete Mathematics 8, 251-257, 1995

[436] J.P. SPINRAD, On Comparability and Permutation Graphs, SIAM Journal on Computing 14, 658-670, 1987

[437] J.P. Spinrad, Two Dimensional Partial Orders, Ph.D. Thesis, Dept. of EECS, Princeton Univ., 1982

[438] J.P. Spinrad, Doubly Lexical Ordering of Dense 0-1 Matrices, Information Processing Letters 45, 229-235, 1993

[439] J.P. SPINRAD, Circular-Arc Graphs with Clique Cover Number Two, Journal of Combinatorial Theory Series B 44, 300-306, 1988

[440] J.P. SPINRAD, Prime Testing for the Split Decomposition of an Undirected Graph, SIAM Journal on Discrete Mathematics 2, 590-599, 1989

[441] J.P. SPINRAD, Finding Large Holes, Information Processing Letters 39, 227-229, 1991

[442] J. SPINRAD, Recognition of Circle Graphs, Journal of Algorithms 16, 264-282, 1994

[443] J. SPINRAD, Dimension and Algorithms, Lecture Notes in Computer Science 831, Ordal 94, 33-52, 1994

[444] J. SPINRAD, Recognizing Quasi-Triangulated Graphs, Discrete Applied Mathematics, to appear

[445] J.P. SPINRAD, A. BRANDSTÄDT, L. STEWART, Bipartite Permutation Graphs, Discrete Applied Mathematics 18, 279-292, 1987

[446] J. SPINRAD, R. SRITHARAN, Algorithms for Weakly Triangulated Graphs, Discrete Applied Mathematics 19, 181-191, 1995

[447] J. SPINRAD, J. VALDES, Recognition and Isomorphism of Two-Dimensional Partial Orders, Lecture Notes in Computer Science 154, ICALP 1983, 676-686, 1983

[448] A.P. SPRAGUE, Recognition of Bipartite Permutation Graphs, Congressus Numerantium 112, 151-161, 1995

[449] A.P. SPRAGUE, An $O(n \log n)$ Algorithm for Bandwidth of Interval Graphs, SIAM Journal on Discrete Mathematics 7, 213-220, 1994

[450] R. SRITHARAN, A Linear Time Algorithm to Recognize Circular Permutation Graphs, Networks 27, 171-174, 1996

[451] R. SUNDARAM, K.S. SINGH, C. PANDU RANGAN, Treewidth of Circular-Arc Graphs, SIAM Journal on Discrete Methods 7, 647-655, 1994

[452] M. TALAMO, P. VOCCA, Representing Graphs Implicitly Using Almost Optimal Space, Discrete Applied Mathematics 108, 193-210, 2001

[453] R.E. TARJAN, Efficiency of a Good but not Linear Set Union Algorithm, Journal of the ACM 22, 215-225, 1975

[454] R.E. TARJAN Decomposition by Clique Separators, Discrete Mathematics 55, 221-232, 1985

[455] R.E. TARJAN, Complexity of Combinatorial Algorithms, SIAM Review 20, 457-489, 1978

[456] R.E. TARJAN, M. YANNAKAKIS, Simple Linear Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs, SIAM Journal on Computing 13, 566-579, 1984

[457] A. TARSKI, A Decision Method for Elementary Algebra, RAND Corporation, 1948
[458] C. THOMASSEN, Interval Representations of Planar Graphs, Journal of Combinatorial Theory Series B, 9-20, 1986
[459] M. THORUP, All Structured Programs have Small Tree Width and Good Register Allocation, Information and Computation 142, 159-181, 1998
[460] W.T. TROTTER, Problems and Conjectures in the Combinatorial Theory of Ordered Sets, Annals of Discrete Mathematics 41, 401-416, 1989
[461] W.T. TROTTER, Combinatorics and Partially Ordered Sets: Dimension Theory, Johns Hopkins University Press, Baltimore MD, 1992
[462] W.T. TROTTER, F. HARARY, On Double and Multiple Interval Graphs, Journal of Graph Theory 3, 205-211, 1979
[463] W. T. TROTTER JR., J. I. MOORE, Characterization Problems for Graphs, Partially Ordered Sets, Lattices, and Families of Sets, Discrete Mathematics 16, 361-381, 1976
[464] S. TSUKIYAMA, M. IDE, H. ARIYOSHI, I. SHIRAKAWA, A New Algorithm for Generating all the Maximal Independent Sets, SIAM Journal on Computing 6, 505-517, 1977
[465] A.C. TUCKER, An Efficient Test For Circular-Arc Graphs, SIAM Journal on Computing 9, 1-24, 1980
[466] A.C. TUCKER, Matrix Characterizations of Circular-Arc Graphs, Pacific Journal of Mathematics 19, 535-545, 1971
[467] A.C. TUCKER, A Structure Theorem for the Consecutive 1's Property, Journal of Combinatorial Theory Series B 12, 153-162, 1972
[468] A.C. TUCKER Structure Theorems for Some Circular-Arc Graphs, Discrete Mathematics 7, 167-195, 1974
[469] A.C. TUCKER Coloring a Family of Circular-Arc Graphs, SIAM Journal on Applied Mathematics 29, 493-502, 1975
[470] A.C. TUCKER, Coloring Perfect $(K_4 - e)$-free Graphs, Journal of Combinatorial Theory Series B 43, 151-172, 1987
[471] W. UNGER, On the k-Colouring of Circle Graphs, Lecture Notes in Computer Science 294, STACS 88, 61-72, 1988
[472] W. UNGER, The Complexity of Colouring Circle Graphs, extended abstract available at citeseer.nj.nec.com/195897.html.
[473] J. URRUTIA, Partial Orders and Euclidean Geometry, Algorithms and Order, I. Rival ed., Kluwer, Dordrecht, 387-434, 1989
[474] J. URRUTIA, F. GAVRIL, An Algorithm for Fraternal Orientation of Graphs, Information Processing Letters 41, 271-274, 1992
[475] J. VALDES, R.E. TARJAN, E.L. LAWLER The Recognition of Series-Parallel Digraphs, SIAM Journal on Computing 11, 298-314, 1978
[476] J.R. WALTER, Representations of Chordal Graphs as Subtrees of a Tree, Journal of Graph Theory 2, 265-267, 1978
[477] D.W. WANG, Y.S. KUO, A Study on Two Geometric Location Problems, Information Processing Letters 28, 281-286, 1988
[478] E. WANKE, k-NLC Graphs and Polynomial Algorithms, Discrete Applied Mathematics 54, 251-266, 1994
[479] H. WARREN, Lower Bounds for Approximation by Nonlinear Manifolds, Transactions of the AMS 133, 167-178, 1968
[480] G. WEGNER, Eigenschaften der Nerven Homologisch-Einfacher Familien in R^n, Dissertation, University of Göttingen, 1967
[481] D.B. WEST, D.B. SHMOYS, Recognizing Graphs with Fixed Interval Number is NP-complete, Discrete Applied Mathematics 8, 295-305, 1984
[482] D.B. WEST, Introduction to Graph Theory, Prentice Hall, 1996
[483] S.H. WHITESIDES, An Algorithm for Finding Clique Cut-Sets, Information Processing Letters 12 , 31-32, 1981
[484] S.H. WHITESIDES, A Method for Solving Certain Graph Recognition and Optimization Problems, with Applications to Perfect Graphs, Annals of Discrete Mathematics 21, 281-297, 1984
[485] A. WIGDERSON, Improving the Performance Guarantee for Approximate Graph Coloring, Journal of the ACM 30, 729-735, 1983

[486] T.V. WIMER, Linear Algorithms on k-terminal Graphs, PhD Thesis, Dept. of Computer Science, URI-030, Clemson University, 1987

[487] P.M. WINKLER, Proof of the Squashed Cube Conjecture, Combinatorica 3, 135-139, 1983

[488] S.K, WISMATH, Characterizing Bar Line-of-Sight Graphs, Proceedings 1st ACM Symposium on Computational Geometry, 147-152, 1985

[489] T.-H. WU, An $O(n^3)$ Isomorphism Test for Circular-Arc Graphs, Ph.D. Thesis, Applied Mathematics and Statistics, SUNY-Stonybrook, 1983

[490] J.-H. YAN, The Bandwidth Problem in Cographs, Tamsui Oxford Journal of Mathematical Sciences 13, 31-36, 1997

[491] M. YANNAKAKIS, Computing the Minimum Fill-in is NP-complete, SIAM Journal on Algebraic Discrete Methods 2, 77-79, 1981

[492] M. YANNAKAKIS, The Complexity of the Partial Order Dimension Problem, SIAM Journal on Algebraic Discrete Methods 3, 351-358, 1982

[493] H.G. YEH, G.J. CHANG, Weighted Connected Domination and Steiner Trees in Distance-Hereditary Graphs, Discrete Applied Mathematics 87, 245-253, 1998

[494] R. BAR-YEHUDA, M.M. HALLDORSSON, J. NAOR, H. SHACHNAI, I. SHAPIRA, Scheduling Split Intervals, 13th ACM-SIAM Symposium on Discrete Algorithms, 2002

[495] C.W. YU, G.H. CHEN, Efficient Parallel Algorithms for Doubly-Convex Bipartite Graphs, Theoretical Computer Science 147, 249-265, 1995

Index

Titles in This Series

19 **Jeremy P. Spinrad,** Efficient graph representations, 2003
18 **Olavi Nevanlinna,** Meromorphic functions and linear algebra, 2003
17 **Vitaly I. Voloshin,** Coloring mixed hypergraphs: theory, algorithms and applications, 2002
16 **Neal Madras,** Lectures on Monte Carlo Methods, 2002
15 **Bradd Hart and Matthew Valeriote, Editors,** Lectures on algebraic model theory, 2002
14 **Frank den Hollander,** Large deviations, 2000
13 **B. V. Rajarama Bhat, George A. Elliott, and Peter A. Fillmore, Editors,** Lectures in operator theory, 2000
12 **Salma Kuhlmann,** Ordered exponential fields, 2000
11 **Tibor Krisztin, Hans-Otto Walther, and Jianhong Wu,** Shape, smoothness and invariant stratification of an attracting set for delayed monotone positive feedback, 1999
10 **Jiří Patera, Editor,** Quasicrystals and discrete geometry, 1998
9 **Paul Selick,** Introduction to homotopy theory, 1997
8 **Terry A. Loring,** Lifting solutions to perturbing problems in C^*-algebras, 1997
7 **S. O. Kochman,** Bordism, stable homotopy and Adams spectral sequences, 1996
6 **Kenneth R. Davidson,** C*-Algebras by example, 1996
5 **A. Weiss,** Multiplicative Galois module structure, 1996
4 **Gérard Besson, Joachim Lohkamp, Pierre Pansu, and Peter Petersen Miroslav Lovric, Maung Min-Oo, and McKenzie Y.-K. Wang, Editors,** Riemannian geometry, 1996
3 **Albrecht Böttcher, Aad Dijksma and Heinz Langer, Michael A. Dritschel and James Rovnyak, and M. A. Kaashoek Peter Lancaster, Editor,** Lectures on operator theory and its applications, 1996
2 **Victor P. Snaith,** Galois module structure, 1994
1 **Stephen Wiggins,** Global dynamics, phase space transport, orbits homoclinic to resonances, and applications, 1993

The book deals with questions which arise from storing a graph in a computer. Different classes of graphs admit different forms of computer representations, and focusing on the representations gives a new perspective on a number of problems. For a variety of classes of graphs, the book considers such questions as existence of good representations, algorithms for finding representations, questions of characterizations in terms of representation, and how the representation affects the complexity of optimization problems. General models of efficient computer representations are also considered.

The book is designed to be used both as a text for a graduate course on topics related to graph representation, and as a monograph for anyone interested in research in the field of graph representation. The material is of interest both to those focusing purely on graph theory and to those working in the area of graph algorithms.

For additional information and updates on this book, visit

www.ams.org/bookpages/fim-19

ISBN 0-8218-2815-0

9 780821 828151

FIM/19

AMS *on the* Web
www.ams.org