Graphs and Combinatorics 4, 307–321 (1988)

Graphs and Combinatorics

Algorithmic Aspects of Intersection Graphs and Representation Hypergraphs

Martin Charles Golumbic

IBM Israel Scientific Center, Haifa, Israel

Abstract. Let $\mathscr{R}$ be a family of sets. The intersection graph of $\mathscr{R}$ is obtained by representing each set in $\mathscr{R}$ by a vertex and connecting two vertices by an edge if and only if their corresponding sets intersect. Of primary interest are those classes of intersection graphs of families of sets having some specific topological or other structure. The "grandfather" of all intersection graphs is the class of interval graphs, that is, the intersection graphs of intervals on a line.

The scope of research that has been going on in this general area extends from the mathematical and algorithmic properties of intersection graphs, to their generalizations and graph parameters motivated by them. In addition, many real-world applications involve the solution of problems on such graphs.

In this paper a number of topics in algorithmic combinatorics which involve intersection graphs and their representative families of sets are presented. Recent applications to computer science are also discussed. The intention of this presentation is to provide an understanding of the main research directions which have been investigated and to suggest possible new directions of research.

1. Introduction

In 1969, Victor Klee challenged the readers of the **American Mathematics Monthly** with the question "What are the intersection graphs of arcs in a circle?" [47]. However, his article did much more than simply ask this one question about circular-arc graphs. Klee summarized most of the known results of the day on interval graphs and other related classes of intersection graphs, and implicitly offered a larger challenge. That challenge was to investigate a great variety of problems on graphs obtained from the intersection of special families of sets.

Besides progress on his specific question [21, 23, 35, 43, 45, 49, 54, 64, 66–70], Professor Klee must take a great deal of satisfaction at the volume and scope of research that has been going on in this general area over the years. This extends from the mathematical and algorithmic properties of intersection graphs, to their generalizations and graph parameters motivated by them. In addition, many real-world applications involve the solution of problems on such graphs. These include applications to computer storage optimization, genetics, mobile radio frequency assignment, fleet maintenance, traffic phasing, archaeology, task assignment, ship building, file organization, pavement deterioration analysis, circuit routing and protein sequencing. [2, 13, 22, 27, 31, 46, 48, 53, 58]. In this paper we will survey a number of topics in algorithmic combinatorics which involve intersection graphs and their representative families of sets. In the process we raise analogous questions such as "What are the representation hypergraphs of arcs in a circle?" We will also discuss some recent applications to computer science. The intention of this article is to provide an understanding of the main research directions which have been investigated and to suggest possible new directions of research. The interested reader will find [31] and [33] a good place to begin further study in this area.

2. Intersection and Other Families of Graphs

Let $\mathscr{R} = \{S_1, \ldots, S_n\}$ be a family of nonempty subsets of a set S. The subsets are not necessarily distinct. We will call S the *host* and the subsets S_i the *objects*. For example, we often think of the host S as a line, a circle, a tree or the plane and the objects as intervals, arcs, paths, or certain polygons. In addition, there may or may not be certain *constraints* placed on the objects such as not allowing one subset to properly contain another or requiring that each subset satisfy a special property. The *intersection graph* of $\mathscr{R}$ has vertices $v_1, \ldots, v_n$ with v_i and v_j joined by an edge if and only if $S_i \cap S_j \neq \varnothing$.

We call the pair $H = (S, \mathscr{R})$ an *intersection representation hypergraph* for G or more simply a *representation*. When $\mathscr{R}$ is a family of intervals on a line, G is an *interval graph* and H is an *interval hypergraph*. If we add the constraint that no interval may properly contain another, then the intersection graphs obtained are the *proper interval graphs*. *Circle graphs* are the intersection graphs of chords of a circle. Figure 1 shows a family of chords of a circle and its intersection graph. When $\mathscr{R}$ is allowed to be an arbitrary family of sets, the class obtained as intersection graphs is all undirected graphs.

On one hand, research has been directed towards intersection graphs of families having some specific topological or other structure. These include circular-arcs, paths in trees, chords of circles, cliques of graphs, and others. On the other hand, certain well known classes of graphs have *subsequently* been characterized in terms

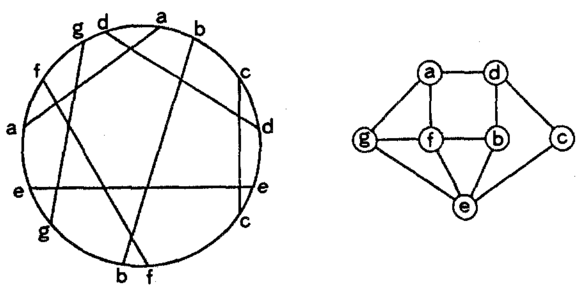


Fig. 1. A family of chords of a circle and its intersection graph

of intersection graphs. For example, triangulated graphs (every cycle of length ≤ 4 has a chord) are the intersection graphs of subtrees of a tree [6, 24, 72], and complements of comparability (cocomparability) graphs are the intersection graphs of curves of continuous functions [40]. This is particularly interesting since interval graphs, the "grandfather" of intersection graphs, had already been characterized as those graphs which are both triangulated and cocomparability graphs [28].

In addition to the variety of graph problems mentioned here, the classes of representation hypergraphs are themselves of interest. For example, the hypergraphs of subtrees of trees have been found to have a connection with the theory of relational databases. This application will be described in section 3.

Of course, intersection is only one type of *interaction* between the objects which gives rise to a graph obtained from a representation H. Other types of interaction, such as *overlap* (S_i and S_j intersect but neither contains the other), *containment* (either $S_i \subset S_j$ or $S_j \subset S_i$), or a *measured intersection* (see [38, 39]), give yet other kinds of graphs from H. Any of our classes of graphs can be described by filling in a table as in Table 1 (section 3).

Overlap graphs of intervals on a line ARE the intersection graphs of chords of a circle

Certain equivalences between such classes have been shown to hold. For example, the overlap graphs of intervals on a line are equivalent to the intersection graphs of chords of a circle. The minimum coloring problem is NP-complete even when restricted to this class of overlap graphs, but the maximum clique and the maximum stable set problems are polynomial for them (see [31, Chapter 11]). A *clique* (resp., *stable set*) is a subset of the vertices every two (resp., no two) of which are connected by an edge of the graph. The complexity of the recognition problem, that is, determining whether an arbitrary graph is the overlap graphs of intervals, has recently been shown to be polynomial [5, 20], solving a problem which had been open for 10 years.

An application requiring the coloring of such graphs is suggested by the diagram of chords on the circle (Fig. 1). The chords may represent airline routes all to be utilized at the same time of day. An assignment of altitudes to the flight paths which will insure disaster free flying can be obtained by properly coloring the intersection graph. A further use of coloring these graphs is in sorting a permutation using stacks in parallel (see [31]). Next we will discuss an application of overlap graphs to optimal macro substitution [34, 32].

A compiler or interpreter for a microcomputer system may be regarded as a byte sequence which resides in main memory. Due to restrictions on the size of main memory, it may be desirable to *compact* this byte sequence. One technique is to define a set of *macro substitutions* which allows occurrences of specified byte subsequences to be replaced by single bytes. The subsequences are restored dynamically during run time by use of an associated table.

Figure 2 shows a sequence of hexidecimal digits of length 36. Since the digits E and F do not appear, they may be used to indicate macros. Choosing $E = 6A2$ and

Original Sequence: 6A2C43B960D606A21C786A243B96A23C6A25

Macro Table: E = 6A2 F = 43B96

Abbreviated Sequence: ECF0D60E1C78EFA23CE5

Fig. 2

$F = 43B96$ the original sequence may be reduced to length 20. Notice that when two macros *overlap*, only one can be replaced. This overlapping, therefore, restricts how the macro table may be applied.

The problem to be solved is to choose an optimal set of macro substitutions and an order for performing the substitutions which minimizes the total length of the abbreviated byte sequence and the associated table. The solution relies on the efficient calculation of maximum weighted stable sets in overlap graphs of intervals. It has been shown that an optimal substitution for a given set of macros can be found in $O(m(k+n) + k^4n^2)$ where n is the length of the original sequence, k is the maximum length of a macro and m is the number of macros. Thus, an overall optimal choice of macros can be found in $O(c_{m,k}n^{m+2})$ where $c_{m,k} \sim \left(\frac{ek}{m}\right)^m \frac{k^4}{\sqrt{2\pi m}}$ which is, in terms of the length of the input sequence, a polynomial whose degree depends on the constant m. In this model we allowed the embedding of macros in other macros. If one chose to forbid embedding, a similar model can be designed which uses interval graphs instead of interval overlap graphs. The same optimal solution techniques can be used, and the respective complexities will be lowered by one factor of kn.

Containment graphs

A binary relation $<$ on a set V is a *strict partial order* if it is irreflexive and transitive. The *comparability graph* of a poset $P = (X, <)$ is the undirected graph $G = (X, E)$ where $xy \in E$ if and only if x and y are comparable in P, (i.e., either $x < y$ or $y < x$). An undirected graph G is a *comparability graph* if it is the comparability graph of some poset. Equivalently, G is a comparability graph (or *transitively orientable graph*) if there exists an orientation F of its edges such that for all $x, y, z \in V$, $[xy \in F$ and $yz \in F] \Rightarrow xz \in F$. Such a transitive orientation F is a strict partial order.

An undirected graph $G = (V, E)$ is a *containment graph* if to each $v_i \in V$ we can associate a nonempty subset S_i of a set S such that $v_iv_j \in E$ if and only if either $S_i \subset S_j$ or $S_j \subset S_i$. Similarly, we call a (strict) partially ordered set $P = (V, <)$ a *containment poset* if to each $v_i \in V$ we can associate a nonempty subset S_i of S such that $v_i < v_j$ if and only if $S_i \subset S_j$. Obviously, G is the comparability graph of P. The following elementary result holds.

Theorem 2.1. *G is a comparability graph if and only if G is a containment graph.*

The class of *interval containment graphs* were first studied by Dushnik and Miller [11] in their work on the dimension of partial orders. A *linear order* is a poset in which every two elements are comparable. Let $P = (X, <)$ be a partial order. A *realizer* of P of size k is a collection of linear orders $L_1 = (X, <_1), \ldots, L_k = (X, <_k)$ such that $x < y$ if and only if $x <_i y$ for all i, that is, $P = L_1 \cap \cdots \cap L_k$. Every partial order can be obtained as the intersection of some number of linear orders. The *dimension* of a partial order P, denoted by $\dim(P)$, is the size of the smallest possible realizer of P. Such a realizer is called a *minimum realizer* of P. The partial order dimension has been shown to be an invariant of the comparability graph as stated in the following result.

Theorem 2.2 [65]. *If two partial orders P and P′ have the same comparability graph, then* $\dim(P) = \dim(P')$.

Therefore, for a comparability graph G we define its *partial order dimension* to be the common dimension of all transitive orientations of G. The *complement* of a graph G is the graph $\bar{G}$ obtained by replacing the edges of G by non-edges and the non-edges by edges. We can now state the characterization of interval containment graphs.

Theorem 2.3[11]. *The following conditions are equivalent:*
(i) *A graph G is the containment graph of intervals on a line,*
(ii) *G is the comparability graph of a poset of dimension* ≤ 2,
(iii) *G and its complement* $\bar{G}$ *are both comparability graphs.*

The graphs which satisfy the preceding theorem are equivalent to the class of *permutation graphs* which are the intersection graphs of permutation diagrams. The containment graphs and posets of rectilinear boxes (with sides parallel to the axes) in d-dimensional Euclidean space are also characterized in terms of partial order dimension. The generalization of Theorem 2.3 to higher dimensions is the following.

Theorem 2.4 [41]. *Let G be a comparability graph. The following conditions are equivalent:*
(i) *G is the containment graph of boxes in d-space,*
(ii) *the partial order dimension of G is at most 2d,*
(iii) *every transitive orientation of G is a containment poset of boxes in d-space.*

On one hand, the complexity of deciding whether an arbitrary comparability graph is the containment graph of boxes in d-space is NP-complete for any fixed $d \geq 2$. This follows from Theorem 2.5 and [73]. On the other hand, interval containment graphs, the case $d = 1$, can be recognized in polynomial time [12, 31, 55, 62]. For further excursions into the world of computational geometry, see [56] and, in particular, its chapters on the complexity of intersection algorithms for objects in the plane and on the geometry of rectangles. Very few containment graph problems seem to have been studied previously. We conclude this section by posing the question "What are the containment graphs of circles in the plane?"

3. Triangulated Graphs and Acyclic Database Schemes

A *relational database scheme* may be regarded as a hypergraph $H = (X, \mathscr{R})$ where the vertex set X represents a set of *attributes* and each hyperedge $E \in \mathscr{R}$ is a subset of attributes called a *relation* and may be thought of as labelling the columns of a table. For example, the relational database scheme with the three relations

$$E_1 = \{\text{class-name, interaction, objects, host, constraints}\}$$

$$E_2 = \{\text{class-name, problem, paper-no, result}\}$$

$$E_3 = \{\text{paper-no, authors, title, reference}\}$$

might be used to design a database of information about containment graphs and intersection graphs. Tables 1-3 illustrate a small portion of such a database.

If we had such a database, we could answer the following sample queries.

Query A: Find all titles by Berge and Hoffman.

Query B: Find all results on interval graphs.

Query C: Find all references to recognition problems on trees.

Query D: Find all references on containment graphs.

Query A can be answered immediately by scanning table E_3, and Query B by scanning E_2. However, Query C is somewhat more complicated since references are in E_3, problems in E_2 and hosts in E_1. We have to *join* these relations in order to answer Query C. This is illustrated in Table 4 where we first join E_1 and E_2 and

Table 1. The table describing relation E_1

Class-name	Interaction	Objects	Host	Constraints
Path graphs	Intersection	Paths	Tree	—
Edge-path graphs	Edge intersection	Paths	Tree	—
Local edge-path graphs	Edge intersection	Paths	Tree	All paths must share a vertex
Circular-arc graphs	Intersection	Arcs	Circle	—
Proper circular-arc graphs	Intersection	Arcs	Circle	No proper containment allowed
Interval graphs	Intersection	Intervals	Line	—
Circle containment graphs	Containment	Circles	Plane	—
Triangulated	Intersection	Subtrees	Tree	—
Cocomparability graphs	Intersection	Curves	Plane	Each curve is the graph of a function
Rectangle containment graphs	Containment	Rectilinear rectangles or boxes	Plane	—

Table 2. The table describing relation E_2

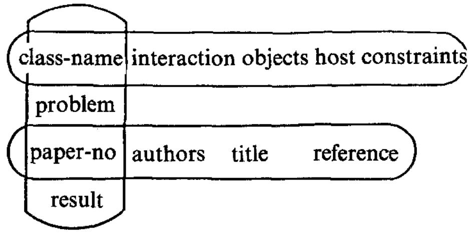
Class-name	Problem	Paper-no	Result
Path graphs	Recognition	25	Polynomial time
Edge-path graphs graph	Maximum clique	36	$O(t^3n)$ for n paths in a tree with t edges
Edge-path graphs	Recognition	37	NP-complete
Local edge-path graphs	Minimum coloring	36	NP-complete
Circular-arc graphs	Minimum coloring	21	NP-complete
Circular-arc graphs	Recognition	70	$O(n^3)$
Proper circular-arc graphs	Minimum coloring	54	$O(n^2 \log n)$
Proper circular-arc graphs	Minimum coloring	64	$O(n^{3/2} \log n)$
Triangulated	Recognition	19	polynomial time
Triangulated	Recognition	60	$O(n + e)$
Triangulated	Recognition	63	$O(n + e)$
Rectangle containment graphs	Recognition	41	NP-complete

Table 3. The table describing relation E_3

Paper-no	Authors	Title	Reference
25	Gavril	Algor. on circular arcs	Discrete Math....
36	Golumbic & Jamison	The edge intersec...	J. Comb. Theo....
37	Golumbic & Jamison	Edge and vertex inter...	Discrete Math....
21	Garey, et al.	The complexity of...	SIAM J. Discrete...
54	Orlin, et. al.	An $O(n^2)$ algorithm...	SIAM J. Discrete...
64	Teng & Tucker	An $O(qn)$ algorithm...	Discrete Math....
19	Fulkerson & Gross	Incidence matrices...	Pacific J. Math....
60	Rose, et. al.	Algorithmic aspects...	SIAM J. Comput....
63	Tarjan & Yannakakis	Simple linear-time...	SIAM J. Comput....
41	Golumbic & Scheinerman	Containment graphs,...	Ann. N.Y. Acad....

Table 4. A portion of the join $E_1 \times E_2 \times E_3$

Class-name	Host	Problem	Paper-no	Reference
Path graphs	Tree	Recognition	25	Discrete Math....
Edge-path graphs	Tree	Recognition	36	Discrete Math....
Triangulated graphs	Tree	Recognition	19	Pacific J. Math....
Triangulated graphs	Tree	Recognition	60	SIAM J. Comput....
Triangulated graphs	Tree	Recognition	63	SIAM J. Comput....



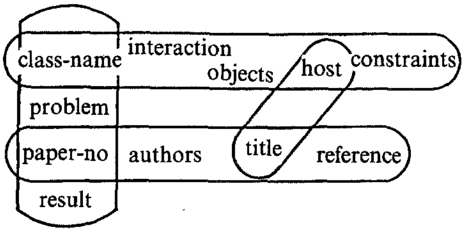


Fig. 3. Our acyclic database scheme E_1, E_2, E_3 and the (undesirable) cyclic scheme E_1, E_2, E_3, E_4

then join it with E_3 to obtain a table, denoted by $E_1 \times E_2 \times E_3$, which contains our answer. Similarly, for Query D.

Notice from Fig. 3 that our scheme E_1, E_2, E_3 has no cycle. This turns out to be very important. For suppose the database designer had added an extra relation

$$E_4 = \{\text{host, title}\}.$$

Then, E_1, E_2, E_3, E_4, which has a cycle, would allow you to answer Query D in two different ways: $E_1 \times E_2 \times E_3$ and $E_1 \times E_4 \times E_3$. This is an open invitation to inconsistent answers to queries.

With such issues as inconsistency, computational efficiency and database distributivity in mind, a number of researchers in quite different contexts of database design have defined and studied several basic, desirable properties of relational database schemes. A list of some of these properties is given in Fig. 4. What is most remarkable is that *all of these properties have been shown to be equivalent.* Moreover, database schemes which fail to satisfy any of these properties exhibit certain unpleasant, pathological behavior. Therefore, the acyclic or tree schemes *characterize* exactly when good behavior is achieved.

Let $H = (X, \mathscr{R})$ be a hypergraph with $X = \{x_1, \ldots, x_n\}$ and $\mathscr{R} = \{E_1, \ldots, E_m\}$. The *dual* hypergraph H^* of H has vertex set $\{e_1, \ldots, e_m\}$, where e_i corresponds to E_i, and hyperedges $X_1, \ldots, X_n$ where $X_k = \{e_i | x_k \in E_i\}$. Clearly, the (hyperedges-vs-vertices) incidence matrix of H^* equals the transpose of the incidence matrix of H and $(H^*)^* = H$, (see Berge [3]). The *representative graph* of H, denoted by $L(H)$, is simply the intersection graph of $\mathscr{R}$. The *2-section graph* of H, denoted by $G(H)$, is the undirected graph with vertex set X and with two vertices x and y adjacent in $G(H)$ if there is some hyperedge $E \in \mathscr{R}$ containing both x and y. It is easy to show that $L(H^*) = G(H)$.

1. If semijoins have no effect on the database, then the universal relation assumption (URA) holds.
2. If the URA holds, then computing the natural join of all relations can be done efficiently using joins.
3. Join dependency of all relations is equivalent to a set of multivalued dependencies.
4. The recognition algorithm of M.H. Graham succeeds.
5. Every pairwise consistent database is consistent.
6. The definition of acyclicity holds.
7. The definition of tree scheme holds.

Fig. 4. Desirable properties of relational database schemes which are all equivalent

A hypergraph $H = (X, \mathscr{R})$ where $\mathscr{R} = \{E_i | i \in I\}$ satisfies the *Helly property* if for all $J \subseteq I$ we have, $E_i \cap E_j \neq \varnothing$ for all $i, j \in J$ implies that $\bigcap \{E_j | j \in J\} \neq \varnothing$. A hypergraph H is called *conformal* if each clique of $G(H)$ is contained in a hyperedge of H. (A clique of a graph G is a subset of vertices such that every pair is adjacent in G.) These two notions are dual to one another as stated in the following theorem.

Theorem 3.1. *H is conformal if and only if H^* satisfies the Helly property.*

It is well known [3, 24, 31] that if $H = (X, \mathscr{R})$ where $\mathscr{R}$ is a family of subtrees of a tree, then H satisfies the Helly property. Furthermore, $L(H)$ is a triangulated graph in this case, as mentioned in section 2. The converse of these properties also holds as the following result shows.

Theorem 3.2 [8, 17]. *The following conditions are equivalent:*
(i) *H is isomorphic to a hypergraph of subtrees of a tree,*
(ii) *H satisfies the Helly property and $L(H)$ is a triangulated graph,*
(iii) *H^* is conformal and $G(H^*)$ is a triangulated graph.*

We will present the *tree scheme* definition next since it is closest to our study of intersection graphs.

A hypergraph $H = (X, \mathscr{R})$ is called a *tree scheme* if there exists a tree T with vertices $e_1, \ldots, e_m$ corresponding to the hyperedges $E_1, \ldots, E_m \in \mathscr{R}$ such that for each $x \in X$ the subgraph T_x of T induced by $\{e_i | x \in E_i\}$ is connected, (i.e., T_x is a subtree of T). In [1] and [42] the tree T is called a join tree and a qual tree, respectively. Intuitively, this is a nice property to have since a query which involves attributes $x_1, \ldots, x_k$ can be answered efficiently and unambiguously. We join the relations corresponding to the smallest subtree T' of T containing $T_{x_1}, \ldots, T_{x_k}$ by choosing a root of T' and joining the relations in the reverse of a breadth-first ordering.

Remark 3.1. It follows from the definitions that H is a tree scheme if and only if its dual H^* is isomorphic to a hypergraph of subtrees of a tree. Therefore, we obtain the following corollary of Theorem 3.2 which was discovered independently by [1] and [42].

Corollary 3.3. *The following conditions are equivalent*:
(i) *H is a tree scheme,*
(ii) *H is conformal and $G(H)$ is a triangulated graph.*

Let $H = (X, \mathscr{R})$ be a hypergraph and let vertices x and y be in X. A *path* of length $m\,(m \geq 1)$ from x to y in H is a sequence of distinct hyperedges $[E_1, E_2, \ldots, E_m]$ where $x \in E_1$, $y \in E_m$ and $E_i \cap E_{i+1} \neq \varnothing$ for $1 \leq i \leq m$. We also say that $[E_1, E_2, \ldots, E_m]$ is a path from E_1 to E_m. Two vertices or two hyperedges are *connected* if there is a path between them. A set of edges is *connected* if every pair of hyperedges in the set is connected. A *connected component* is a maximal connected set of hyperedges.

A closed path $C = [E_1, E_2, \ldots, E_m, E_1]$ is called a *hypercycle* of length m. (The definition of hypercycle given here differs from the definition of a cycle in a hypergraph given in Berge [3].) Let $F_i = E_i \cap E_{i+1}$, $1 \leq i \leq m$ (addition modulo m). We call $E \in \mathscr{R}$ a *chord* of C if there exist $1 \leq i < j < k \leq m$ such that $F_i \cup F_j \cup F_k \subseteq E$. A hypercycle is *chordless* if it has no chord.

We say that Graham's algorithm *succeeds* on $H = (X, \mathscr{R})$ if $\mathscr{R}$ can be reduced to the empty set by repeated application of the following two rules (in any order).
(1) *Elimination*: If x is a vertex that appears in exactly one hyperedge E_i, then delete x from E_i.
(2) *Reduction*: If $E_i \subseteq E_j$ and $i \neq j$, then delete E_i from $\mathscr{R}$.

Remark 3.2. Graham's algorithm is essentially the hypergraph version of perfect vertex elimination in graphs, the technique first mentioned in [19] to recognize triangulated graphs, and subsequently applied to Gaussian elimination in sparce matrices [59, 30, 31, Chapter 12].

Finally, we define acyclic schemes [1, 51]. A hypergraph is called *reduced* if rule (2) does not apply. The *reduction* of H, denoted by *red*(H), is obtained by applying rule (2) repeatedly until it no longer applies. Let $H = (X, \mathscr{R})$ and $Y \subset X$. The *Y-generated hypergraph*, denoted by $H[Y]$, is the hypergraph *red*$((Y, \mathscr{R}[Y]))$ where

$$\mathscr{R}[Y] = \{E \cap Y \mid E \in \mathscr{R}\}.$$

A set $A \subset X$ is called an *articulation set* for H if $A = E_1 \cap E_2$ for some pair of hyperedges $E_1, E_2 \in \mathscr{R}$ and $H[X - A]$ has more connected components than H. A *block* of H is a Y-generated hypergraph of H, for some $Y \subset X$, with no articulation set. A block is *trivial* if it has only one hyperedge.

A reduced hypergraph is called *acyclic* if all its blocks are trivial; otherwise, it is *cyclic*. An arbitrary hypergraph is *acyclic* or *cyclic* precisely if its reduction is.

Theorem 3.4. *The following conditions are equivalent*:
(i) *H is acyclic,*
(ii) *H is a tree scheme,*
(iii) *H has no chordless hypercycle of length ≥ 3,*
(iv) *Graham's algorithm succeeds on H.*

The equivalence of (i), (ii), and (iv) are proved in [1], (see also [42, 51]). The equivalence of (iii) is cited in [51, Ex. 13.37] and is attributed to Maier, Ullman and Laver. Further results can be found in [17].

Remark 3.3. The definition of acyclic hypergraph is a generalization of the notion of an ordinary undirected graph with the property that any nontrivial connected subgraph has an articulation vertex, i.e., the graphs with no 2-connected subgraphs. These, of course, are precisely trees. In retrospect, the name "acyclic hypergraph" is an unfortunate choice of terms since there are differing definitions of cycles in hypergraphs [14, 15]. Maier [51, Ex. 13.38] even asks for an example of an acyclic hypergraph with a cycle! The better names seem to be tree scheme or chordal hypergraph.

4. Representation Hypergraphs

Let Σ be a collection of nonempty subsets of a set S. We say that a hypergraph $H = (X, \mathscr{R})$ is Σ-representable if there is a mapping $f\colon X \to S$ such that $\{f(x) | x \in E_i\} \in \Sigma$ for every $E_i \in \mathscr{R}$. For example, the tree hypergraphs, which were discussed in section 3, are obtained by letting Σ be the collection of subtrees of a tree. In this section we present a survey of several other classes including circular-arc hypergraphs and edge-path hypergraphs.

The *edge-path hypergraphs* are defined by letting S be the set of edges of a tree T and letting $\Sigma = \{P \subset S | P \text{ is a path in } T\}$. Edge-path hypergraphs have been studied in the context of the theory of matroids by Tutte [71] and Fournier [18]. Gavril and Tamari [26] have given an $O(|X|^2|\mathscr{R}|)$ time algorithm for deciding whether a given hypergraph is an edge-path hypergraph. Such an algorithm can also be used to decide whether a binary matroid is graphic. The intersection graphs of edge-paths in a tree, called *EPT graphs*, have been studied in [36, 37]. Interestingly, the recognition problem for *EPT* graphs is *NP*-complete. Further results can be found in [52].

Interval hypergraphs and circular-arc hypergraphs have an interesting matrix formulation. A matrix **M** whose entries are zeros and ones is said to have the *consecutive* 1*'s property for rows* if its columns can be permuted in such a way that the 1's in each row occur consecutively. Similarly, **M** is said to have the *circular* 1*'s property for rows* if its columns can be permuted in such a way that the 1's in each row occur in circular consecutive order, i.e., regarding the matrix as wrapped around a cylinder. The following result is immediate.

Theorem 4.1. *A hypergraph H is an interval hypergraph if and only if its incidence matrix (hyperedges-vs-vertices) has the consecutive* 1*'s property for rows. A hyper-*

graph H is a circular-arc hypergraph if and only if its incidence matrix has the circular 1's property for rows.

The efficiency of testing for consecutive 1's and circular 1's depends partly upon the sparseness of **M**. Booth and Lueker [4] have shown that an $m \times n$ (0, 1)-valued matrix **M** with f nonzero entries can be tested for the consecutive 1's property or the circular 1's property in $O(m + n + f)$ steps. An important application of interval (resp., circular-arc) hypergraphs is the consecutive (resp., circular) retrieval property in database file organization [27].

It is not hard to show that if H is an interval hypergraph, then H and its dual H^* are both acyclic hypergraphs. The converse does not hold as seen by considering the hypergraph with $X = \{1,2,3,4\}$ and $\mathscr{R} = \{\{1,4\},\{1,2,3\},\{1,3,4\},\{1,2\}\}$. It can also be observed [50] that if H is a circular-arc hypergraph such that $H = red(H)$, then H^* is a circular-arc hypergraph. Further results on interval hypergraphs are surveyed in [10].

A more general version of Theorem 3.2 given by Duchet [9, 10] relates interval graphs and interval hypergraphs, namely, H is an interval hypergraph if and only if H is Helly and $L(H)$ is an interval graph. No such version holds for circular-arcs since they do not satisfy the Helly property. Finding an analogous theorem for circular-arc hypergraphs is still an open research problem. However, Quilliot [57] has obtained some results in this direction.

A hypergraph $H = (X, \mathscr{R})$ has the *anti-Helly* property if for all $E_1, E_2, E_3 \in \mathscr{R}$ such that $E_1 \cap E_2 \cap E_3 \neq \varnothing$, there exists a permutation π of $\{1,2,3\}$ such that $E_{\pi_1} \subset E_{\pi_2} \cup E_{\pi_3}$.

Let $|X| = n$ and denote by $\bar{H} = (X, \bar{\mathscr{R}})$ the smallest hypergraph (with respect to containment) such that

$$\text{(i) } \mathscr{R} \subset \bar{\mathscr{R}} \qquad \text{(ii) } [A \subseteq X \text{ and } |A| \in \{0, 1, n-1, n\}] \Rightarrow A \in \bar{\mathscr{R}},$$

and for all $E_1, E_2 \in \mathscr{R}$ we have

$$\text{(iii) } E_1 \cap E_2 \neq \varnothing \Rightarrow E_1 \cup E_2 \in \bar{\mathscr{R}} \qquad \text{(iv) } E_1 \cup E_2 \neq X \Rightarrow E_1 \cap E_2\ \bar{\mathscr{R}}$$

$$\text{(v) } E_1 \not\subset E_2 \Rightarrow E_1 - E_2 \in \bar{\mathscr{R}}.$$

The following result holds.

Theorem 4.2 [57]. *A hypergraph H is a circular-arc hypergraph if and only if* (1) *the hypergraph $\bar{H}$ satisfies the anti-Helly property and* (2) *there exists a vertex x_0 in H such that the partial hypergraph $\bar{H} - x_0$ of H (constituted by the hyperedges which do not contain x_0) is an interval hypergraph.*

Let C be either an edge or a triangle of a graph $G = (V, E)$. We say that C is *attractive* if (i) $Adj(v) \cap C \neq \varnothing$, for all $v \in V$, and (ii) $Adj(x) \cap Adj(y) \cap C \neq \varnothing$ for all $xy \in E$. (The notion here is that C has a common neighbor with every vertex and every edge.)

Theorem 4.3 [57]. *Let G be a graph such that every attractive triangle contains an attractive edge. Then G is a circular-arc graph if and only if its clique hypergraph is a circular-arc hypergraph.*

We conclude with some problems on acyclic projections of cyclic hypergraphs. A hypergraph H is called a *ring* if H is cyclic and H^* is a circular-arc hypergraph. It follows from the definitions that H is a circular-arc hypergraph if and only if either H is an interval hypergraph or H^* is a ring. A hypergraph is called *Graham reduced* if neither rule (1) nor rule (2) of section 3 applies. An *Xring* is defined in [50] to be a hypergraph which is a ring and is Graham reduced.

A hypergraph H is called a *projection* of H', denoted by $H \leq H'$, if for every hyperedge E of H there exist a hyperedge E' of H' such that $E \subseteq E'$. We say that H is an *acyclic projection* of H_2 over H_1 if H is acyclic and $H_1 \leq H \leq H_2$. The complexity of deciding whether an acyclic projection H exists for arbitrary cyclic H_1 and H_2 is an open question and has been conjectured to be NP-complete. Lustig and Shmueli [50] have given a polynomial time algorithm for solving this problem when H_1 is an Xring. Because of the important role that acyclic hypergraphs play in the theory of relational databases, identifying the existence of acyclic projections provide new options for processing queries.

References

1. Beeri, C., Fagin, R., Maier, D., Yannakakis, M.: On the desirability of acyclic database schemes, J. Assoc. Comput. Mach. **30**, 479–513 (1983)
2. Benser, S.: On the topology of the genetic fine structure, Proc. Natl. Acad. Sci. USA **45**, 1607–1620 (1959)
3. Berge, C.: Graphs and Hypergraphs. Amsterdam: North-Holland 1973
4. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. Syst. Sci. **13**, 335–379 (1976)
5. Bouchet, A.: Un algorithme polynomial pour reconnaître les graphes d'alternance. C. R. Acad. Sci. Paris Ser. I Math **300**, 569–572 (1985)
6. Buneman, P.: A characterization of rigid circuit graphs. Discrete Math. **9**, 205–212 (1974)
7. Cohen, Joel E.: The asymptotic probability that a random graph is a unit interval graph, indifference graph, or proper interval graph. Discrete Math. **40**, 21–24 (1982)
8. Duchet, P.: Propriété de Helly et problèmes de représentation, Colloq. Internat. CNRS 260, Problèmes Combinatoires et Théorie des Graphes, Orsay, France, 117–118 (1976)
9. Duchet, P.: Representations, noyaux, en théorie des graphes et hypergraphs. Doctoral Thesis, Univ. Paris VI, 1979
10. Duchet, P.: Classical perfect graphs. Ann. Discrete Math. **21**, 67–96 (1984)
11. Dushnik, B., Miller, E.W.: Partially ordered sets. Amer. J. Math. **63**, 600–610 (1941)
12. Even, S., Pnueli, A., Lempel, A.: Permutation graphs and transitive graphs. J. Assoc. Comput. Mach. **19**, 400–410 (1972)
13. Fabri, J.: Automatic storage optimization. Ann Arbor, MI: UMI Press 1982
14. Fagin, R.: Degrees of acyclicity for hypergraphs and relational database schemes. J. Assoc. Comput. Mach. **30**, 514–550 (1983)
15. Fagin, R.: Acyclic database schemes of various degrees: a painless introduction. In: Lecture Notes in Computer Science 159. pp. 65–89. Berlin: Springer-Verlag 1983
16. Fishburn, P.C.: Interval Orders and Interval Graphs. New York: Wiley 1985
17. Flament, C.: Hypergraphes arborés. Discrete Math. **21**, 223–226 (1978)
18. Fournier, J.C.: Hypergraphes de chaines d'arêtes d'un arbre. Discrete Math. **43**, 29–36 (1983)
19. Fulkerson, D.R., Gross, O.A.: Incidence matrices and interval graphs. *Pacific J. Math.* **15**, 835–855 (1965)
20. Gabor, C.P., Hsu, W.-L., Supowit, K.J.: Recognizing circle graphs in polynomial time. Proc. IEEE Symp. on Foundations of Computer Science. (1985)

21. Garey, M.R., Johnson, D.S., Miller, G.L., Papadimitriou, C.H.: The complexity of coloring circular arcs and chords. SIAM J. Algebraic Discrete Methods **1**, 216–227 (1980)
22. Gattass, E.A., Nemhauser, G.L.: An application of vertex packing to data analysis in the evaluation of pavement deterioration. Oper. Res. Letters **1**, 13–17 (1981)
23. Gavril, F.: Algorithms on circular arc graphs. Networks **4**, 357–369 (1974)
24. Gavril, F.: The intersection graphs of subtrees of a tree are exactly the chordal graphs. J. Comb. Theory (B) **16**, 47–56 (1974)
25. Gavril, F.: A recognition algorithm for the intersection graphs of paths in trees. Discrete Math. **23**, 211–227 (1978)
26. Gavril, F., Tamari, R.: Constructing edge-trees from hypergraphs. Networks **13**, 377–388 (1983)
27. Ghosh, S.P., Kambayaski, Y., Lipski, W. Jr.: Data Base File Organization: Theory and Applications of the Consecutive Retrieval Property. New York: Academic Press 1983
28. Gilmore, P.C., Hoffman, A.J.: A characterization of comparability graphs and of interval graphs. Canad. J. Math. **16**, 539–548 (1964)
29. Golumbic, M.C.: Comparability graphs and a new matroid, J. Comb. Theory (B) **22**, 68–90 (1977)
30. Golumbic, M.C.: A note on perfect Gaussian elimination. J. Math. Anal. Appl. **64**, 455–457 (1978)
31. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. New York: Academic Press 1980
32. Golumbic, M.C.: Algorithmic aspects of perfect graphs. Ann. Discrete Math. **21**, 301–323 (1984)
33. Golumbic, M.C.: Interval Graphs and Related Topics, A special issue of Discrete Math. **55**, 113–243 (1985)
34. Golumbic, M.C., Goss, C.F., Dewar, R.B.K.: Macro substitutions in MICRO SPITBOL – a combinatorial analysis (Proc. 11th Southeastern Conf. on Combinatorics, Graph Theory and Computing) Congr. Numerantium 29, 485–495 (1980)
35. Golumbic, M.C., Hammer, P.L.: Stability in circular arc graphs J. Algorithms (to appear 1988).
36. Golumbic, M.C., Jamison, R.E.: The edge intersection graphs of paths in a tree. J. Comb. Theory (B) **38**, 8–22 (1985)
37. Golumbic, M.C., Jamison, R.E.: Edge and vertex intersection of paths in a tree. Discrete Math. **55**, 151–159 (1985)
38. Golumbic, M.C., Monma, C.L.: A generalization of interval graphs with tolerances (Proc. 13th Southeastern Conf. on Combinatorics, Graph Theory and Computing). Congr. Numerantium 35, 321–331 (1982)
39. Golumbic, M.C., Monma, C.L., Trotter, W.T., Jr.: Tolerance graphs. Discrete Appl. Math. **9**, 157–170 (1984)
40. Golumbic, M.C., Rotem, D., Urrutia, J.: Comparability graphs and intersection graphs. Discrete Math. **43**, 37–46 (1983)
41. Golumbic, M.C., Scheinerman, E.R.: Containment graphs, posets and related classes of graphs. (Proc. Third Int'l Conf. on Combinatorial Math., New York, June 1985) Ann. N. Y. Acad. Sci. (to appear 1988).
42. Goodman, N., Shmueli, O.: Syntactic characterization of tree database schemes. J. Assoc. Comput. Mach. **30**, 767–786 (1983)
43. Gupta, U.I., Lee, D.T., Leung, J.Y.-T.: Efficient algorithms for interval graphs and circular arc graphs. Networks **12**, 459–467 (1982)
44. Hanlon, P.: Counting interval graphs. Trans. Amer. Math. Soc. **272**, 383–426 (1982)
45. Hsu, W.-L.: Maximum weight clique algorithm for circular arc graphs and circle graphs. SIAM J. Computing **14**, 224–231 (1985)
46. Jungck, J.R., Dick G., Dick, A.G.: Computer-assisted sequencing, interval graphs, and molecular evolution. Biosystems **15**, 259–273 (1982)
47. Klee, V.: What are the intersection graphs of arcs in a circle? Amer. Math. Mon. **76**, 810–813 (1969)
48. LaPaugh, A.S.: A polynomial time algorithm for optimal routing around a rectangle. In: Proc. IEEE Symp. on Foundations of Computer Science. pp. 282–293 1980

49. Leung, J.Y.-T.: Fast algorithms for generating all maximal independent sets of interval, circular-arc and chordal graphs. J. Algorithms **5**, 22–35 (1984)
50. Lustig, A., Shmueli, O.: Acyclic hypergraph projections and relationships to circular-arc graphs and circular representable hypergraphs. Technion – Israel Institute of Technology, Technical Report, Nov. 1985
51. Maier, D.: The Theory of Relational Databases. Rockville, MD: Computer Science Press 1983
52. Monma, C.L.: Intersection graphs of paths in trees. J. Comb. Theory (B) **41**, 141–181 (1987)
53. Opsut, R.J., Roberts, F.S.: *I*-colorings, *I*-phasings, and *I*-intersection assignments for graphs, and their applications. Networks **13**, 327–345 (1983)
54. Orlin, J., Bonuccelli, M., Bovet, D.: An $O(n^2)$ algorithm for coloring proper circular arc graphs. SIAM J. Algebraic Discrete Methods **2**, 88–93 (1981)
55. Pnueli, A., Lempel, A., Even, S.: Transitive orientation of graphs and identification of permutation graphs. Canad. J. Math. **23**, 160–175 (1971)
56. Preparata, F.P., Shamos, Michael I.: Computational Geometry. New York: Springer-Verlag 1985
57. Quilliot, A.: Circular representation problems on hypergraphs. Discrete Math. **51**, 251–264 (1984)
58. Roberts, F.S.: Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems. Englewood Cliffs, NJ: Prentice-Hall 1976
59. Rose, D.J.: Triangulated graphs and the elimination process. J. Math. Anal. Appl. **32**, 597–609 (1970)
60. Rose, D.J.: Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SIAM J. Comput. **5** 266–283 (1976)
61. Skrien, D.J.: Chronological orderings of interval graphs. Discrete Applied Math. **8**, 69–83 (1984)
62. Spinrad, J.: On comparability and permutation graphs. SIAM J. Computing **14**, 658–670 (1985)
63. Tarjan, R. E., Yannakakis, M.: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM J. Computing **13**, 566–579 (1984)
64. Teng, A., Tucker, A.C.: An $O(qn)$ algorithm to q-color a proper family of circular arcs. Discrete Math. **55**, 233–243 (1985)
65. Trotter, W.T., Jr., Moore, J.I., Jr., Summer, D.P.: The dimension of a comparability graph. Proc. Amer. Math. Soc. **60**, 35–38 (1976)
66. Tucker, A.C.: Characterizing circular arc graphs. Bull. Amer. Math. Soc. **76**, 1257–1260 (1970)
67. Tucker, A.C.: Matrix characterizations of circular arc graphs. Pacific J. Math. **39**, 535–545 (1971)
68. Tucker, A.C.: Structure theorems for some circular arc graphs. Discrete Math. **7**, 167–195 (1974)
69. Tucker, A.C.: Coloring a family of circular arc graphs. SIAM J. Appl. Math. **29**, 493–502 (1975)
70. Tucker, A.C.: An efficient test for circular arc graphs. SIAM J. Computing **9**, 1–24 (1980)
71. Tutte, W.T.: An algorithm for determining whether a given binary matroid is graphic. Proc. Amer. Math. Soc. **11**, 905–917 (1960)
72. Walter, J.R.: Representations of rigid cycle graphs. Doctoral dissertation, Wayne State Univ., 1972.
73. Yannakakis, M.: The complexity of the partial order dimension problem. SIAM J. Algebraic Discrete Methods **3**, 351–358 (1982)

Received: May 20, 1986
Revised: October 2, 1987