On the Complexity of Recognizing Intersection and Touching Graphs of Disks

Heinz Breu* and David G. Kirkpatrick**
Department of Computer Science
University of British Columbia
Vancouver, British Columbia V6T 1Z4
CANADA

Abstract. Disk intersection (respectively, touching) graphs are the intersection graphs of closed disks in the plane whose interiors may (respectively, may not) overlap. In a previous paper [BK93], we showed that the recognition problem for *unit* disk intersection graphs (i.e. intersection graphs of unit disks) is **NP**-hard. That proof is easily modified to apply to unit disk touching graphs as well. In this paper, we show how to generalize our earlier construction to accomodate disks whose size may differ. In particular, we prove that the recognition problems for both bounded-ratio disk intersection graphs and bounded-ratio disk touching graphs are also **NP**-hard. (By bounded-ratio we refer to the natural generalization of the unit constraint in which the radius ratio of the largest to smallest permissible disk is bounded by some fixed constant.) The latter result contrasts with the fact that the disk touching graphs (of unconstrained ratio) are precisely the planar graphs, and are hence polynomial time recognizable. The recognition problem for disk intersection graphs (of unconstrained ratio) has recently been shown to be **NP**-hard as well [Kra95].

1 Introduction

Families of graphs that have realizations as intersection graphs of restricted geometric objects in the plane have attracted the attention of researchers with interests in pure and computational graph theory as well as computational geometry and complexity theory. Issues include the recognition, geometric realization (layout), non-geometric characterization, application (including the modeling of communication and visibility problems) and algorithmic exploitation (for problems that appear to be intractable for general graphs) of such graphs.

In several well-studied cases (eg. interval graphs and permutation graphs [BL76, Spi85]) the recognition problem is solvable in polynomial time. In other situations (such as the intersection graphs of arbitrary curves in the plane [Kra91] or even line segments restricted to two or more slopes [Kra94]) the recognition problem is **NP**-hard. This paper addresses the case in which the objects are all closed disks.

---

* On leave from Hewlett-Packard Laboratories.

** Supported in part by NSERC grant A3583.

A *disk intersection graph* G is the intersection graph of a set of closed disks in the plane. That is, each vertex of G corresponds to a disk in the plane, and two vertices are adjacent in G if and only if the corresponding disks intersect. The set of disks is said to *realize* the graph.

Note that two disks intersect if and only if the distance between their centers is at most the sum of their radii. Therefore, disk graphs can be realized equally well as a set of weighted points in the plane; two vertices are adjacent in the graph exactly when the Euclidean distance between their associated points is at most the sum of the associated weights.

A *disk touching graph* G is the intersection graph of a set of closed disks in the plane whose interiors are constrained to be disjoint. Thus, G is *realized* by a set of interior-disjoint disks, where two disks touch (have a common boundary point) if and only if the associated vertices are adjacent in G.

If all disks have the same size they are said to realize a *unit* disk intersection graph or *unit* disk touching graph. Clearly, not every disk intersection graph or disk touching graph has a unit realization. For example, every star $K_{1,t}$ is a disk touching graph, but only those with $t < 6$ have realizations as disk intersection graphs. (Note that the actual unit of size is not critical, since a set of disks realize the same graph even if the coordinate system is scaled by any convenient amount.)

In addition to their intrinsic interest as geometric graphs, there are several motivations for studying disk intersection and touching graphs. The former provide a natural two-dimensional generalization of interval graphs (cf. [FG65, Rob68]), for which a great deal is known (for example, polynomial time recognition, efficient algorithms for problems that are **NP**-hard in general, and efficient approximation algorithms). In addition, disk intersection graphs (or their unit restriction) have been used to model several physical problems, for example radio frequency assignment [Hal80] and ship-to-ship communications (attributed to Marc Lipman by [Rob91]). They have also been used as test cases for heuristic algorithms designed for arbitrary graphs [JAMS91]. More applications are described by [CCJ90] and [MHR92]. Disk touching graphs (also known as disk packing graphs) play an important role in the construction of high resolution embeddings of planar graphs [MP92]. Unit disk touching graphs can also be seen as a natural generalization of grid graphs [BC87] (in which the disks are constrained to be centred at integer grid points).

In a previous paper [BK93] the problem of determining if a given graph is a unit disk intersection graph (equivalently, the problem of determining if the sphericity (cf. [Hav82a, Fis83]) of a graph is less than or equal to two) was shown to be **NP**-hard. This answered an open question mentioned in [CCJ90] and [MHR92]. The reduction can be modified without difficulty to prove that the recognition of unit touching graphs is also **NP**-hard.

In this paper, we show how to generalize our construction presented in [BK93] to accomodate disks whose size may differ. In particular, we prove that the recognition problems for both bounded-ratio disk intersection graphs and bounded-ratio disk touching graphs are also **NP**-hard. More formally, for every $\rho \geq 1$, we define the following:

ρ-BOUNDED DISK INTERSECTION GRAPH RECOGNITION
INSTANCE: Graph $G = (V, E)$.
QUESTION: Does G have a realization as the intersection graph of a set of disks, whose radii fall in the range $[1, \rho]$.

ρ-BOUNDED DISK TOUCHING GRAPH RECOGNITION
INSTANCE: Graph $G = (V, E)$.
QUESTION: Does G have a realization as the touching graph of a set of disks, whose radii fall in the range $[1, \rho]$.

Our main results are the following:

Theorem 1. *ρ-BOUNDED DISK INTERSECTION GRAPH RECOGNITION is* **NP***-hard, for every fixed $\rho \geq 1$.*

Theorem 2. *ρ-BOUNDED DISK TOUCHING GRAPH RECOGNITION is* **NP***-hard, for every fixed $\rho \geq 1$.*

It is interesting to note that the unconstrained DISK TOUCHING GRAPH RECOGNITION problem (equivalently, the ∞-BOUNDED DISK TOUCHING GRAPH RECOGNITION problem) has a familiar polynomial time solution by virtue of the fact[3] that a graph is an (unconstrained) disk touching graph if and only if it is planar. On the other hand, the unconstrained DISK INTERSECTION GRAPH RECOGNITION problem remains **NP**-hard [Kra95].

As indicated previously, our proofs are similar in form to that presented in the unit case. The next section recalls the essential structure of our **NP**-hardness reduction. Section 3 sets out some properties of disk packings (specifically, relating the number of disks in a given packing to the size of its boundary). These properties are used in Section 4 to describe the components of a generic reduction from a variant of SATISFIABILITY to both of the problems described above. Section 5 offers some concluding remarks.

2 Overview of reduction for unit disk graphs

Our proof (cf. [BK93]) of the **NP**-hardness of the UNIT DISK INTERSECTION GRAPH RECOGNITION problem is a reduction from a variant of CNF SATISFIABILITY. Specifically, we show that every conjunctive normal form Boolean formula $\mathcal{F}$, in which every clause contains at most three literals and every variable appears in at most three clauses, can be transformed into a graph $G_{\mathcal{F}}$ with the property that $G_{\mathcal{F}}$ is a unit disk intersection graph if and only if $\mathcal{F}$ is satisfiable.

[3] This result, frequently attributed to W.P. Thurston, was evidently first discovered in 1935 by P. Koebe (cf. [Sac94])

This transformation proceeds in three stages. We begin by defining the bipartite graph $G_{\mathcal{F}}^{SAT}$ determined by the literal-clause incidence relation in $\mathcal{F}$. It is straightforward to see that $\mathcal{F}$ is satisfiable if and only if the edges of $G_{\mathcal{F}}^{SAT}$ can be oriented so as to satisfy certain out- and in-degree constraints at the clause and literal vertices (essentially an edge is oriented from a clause c to a literal u in c if u is "chosen" to satisfy c). Next it is shown that the graph $G_{\mathcal{F}}^{SAT}$ can be embedded on a grid without overlapping edges (here we exploit the restricted form of the input formula; in fact, even edge crossings can be avoided by starting with a more restrictive—yet still **NP**-hard—variant of SATISFIABILITY [Kra95].) Edges of the embedded $G_{\mathcal{F}}^{SAT}$ can be viewed as a sequence of unit-length grid segments. Hence the entire embedded graph $G_{\mathcal{F}}^{SAT}$ can be described as a conglomerate of fixed sized modules (including edge segments and clause and literal junctions). The orientability of $G_{\mathcal{F}}^{SAT}$ is easily recast in terms of orientations of these modules. Finally, we show how to realize each of these modules by a small unit disk intersection graph whose several feasible realizations reflect the several permissible orientations of the corresponding module.

The component unit disk intersection graphs are themselves composed of simpler pieces called *cages*. A cage is simply a chordless cycle. Since a cage is realized as a ring of connected disks, every cage has a fixed *capacity* (informally, the maximum number of disjoint disks that can be packed into the interior of some realization). If two cages share a sequence of three of more vertices on their boundary and are realized with neither cage embedded inside the other (as will always be the case), then any connected subgraph attached to one or more of the interior vertices of this sequence must be entirely embedded in one or other of the two cages. (We refer to such a subgraph in the context of its associated cages as a *flipper*.) This is the mechanism used both to express binary "choices" in the modules and, in the event that a cage does not have the residual capacity to host one of its incident flippers due to earlier "choices", to propagate "choices".

Most of the technical details of the proof concern the construction of networks of cages and flippers that permit disk realizations of all and only the desired forms. Since the desired realizations are all achieved with adjacent cage vertices at unit distance (i.e with the corresponding disks merely touching), and the impossibility of undesired realizations is based on capacity arguments, the proof is easily adapted to apply to unit disk touching graph recognition as well. An extension of the proof to the bounded-ratio problems requires a more careful and general treatment of our lowest level building blocks, cages and flippers. This is taken up in the next section.

3 Hexagon packings

Our reduction from CNF-SATISFIABILITY to the recognition problems for bounded-ratio disk intersection and touching graphs is, in fact, slightly simpler than its precurser in that all of the constituent cages are the same size (some sufficiently large multiple of six, dependent in part on the value ρ) and all are realizable (when the given formula is satisfiable) as hexagonal chains of touching disks whose centres lie on a regular hexagon. (We refer to the latter as a *hexagonal realization* of the cage.)

This uniformity implies that our entire construction, at least to the level of detail of cages, respects an underlying hexagonal grid. This substantially simplifies the issues of fabricating and inter-connecting modules. Of course, as before, we must argue that the realizations that we wish to consider are, up to topological equivalence, the only possible realizations. Here we rely on arguments concerning the capacity of cages which, unlike those in our earlier proof, can be arbitrarily large. The *hexagonal capacity* of a cage is the maximum number of interior-disjoint unit disks that can be packed into the interior of a hexagonal realization of the cage. The *unconstrained capacity* of a cage is the maximum number of interior-disjoint disks that can be packed into the interior of *any* realization of the cage. (Note that in both cases the maximum is achieved by a realization of the cage in which adjacent vertices touch only.) Our main tool for arguing about the unrealizability of certain embeddings is the following:

Lemma 3. *As the size of a cage C increases, the ratio of its unconstrained capacity to its hexagonal capacity approaches $2\sqrt{3}/\pi$ (≈ 1.103).*

Proof. The realization of a cage C by a set of maximum-sized touching disks all of whose centres are co-circular clearly has maximum internal area, among all disk intersection realizations of C. It follows that as the size of C increases, the ratio of the unconstrained capacity to the hexagonal capacity of C approaches the ratio of the area of a circle to the area of a regular hexagon with the same circumference.

Corollary 4. *For all sufficiently large cages C, embeddings of C containing more interior-disjoint disks than $2\sqrt{3}/\pi$ times the hexagonal capacity of C, are unrealizable.*

In our constructions cages are joined at their corners; specifically, any corner vertex and its two adjacent vertices may be shared by two (otherwise disjoint) cages. In such a situation the shared corner vertex cannot respect the hexagonal realization of both cages. In general, this corner vertex is the attachment point for a subgraph (which we refer to as a *flipper*). In any realization the flipper is embedded entirely inside one or other of the two cages. We refer to the number of vertices in the flipper (in the case of disk touching graphs) or the number of independent vertices in the flipper (in the case of disk intersection graphs) expressed as a fraction[4] of the hexagonal capacity of the cage, as its *size*. Figure 1 illustrates[5] a pair of joined cages that share a flipper of size 1. This shared flipper is shown embedded in the right cage.

Typically, a cage may have two or more neighbouring cages, and hence several incident flippers. By Corollary 4, any subset of flippers whose total size exceeds

[4] More precisely, we mean the fraction achieved asymptotically as the size of cages increase.

[5] This and subsequent illustrations apply to disk touching graphs. The modifications for disk intersection graphs are straightforward (cf. [Bre95]).

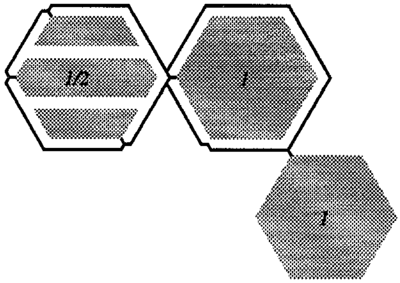

Fig. 1. A pair of joined cages with one shared flipper (and four other flippers).

$2\sqrt{3}/\pi$ cannot be simultaneously embedded inside any realization of their common cage. Thus, for example, it is impossible for the left cage in Figure 1 to house the shared flipper (of size 1) in addition to *any* of its resident flippers (of sizes 1/4 and 1/2). Similarly, the right cage cannot contain both of its incident flippers (of size 1); one of these is shown displaced below the cage.

In general, flippers are constructed from a single connected portion of the hexagonal grid. (They are depicted as such in all of our figures.) The shapes of specific flippers are chosen to permit the simultaneous internal realization (in a hexagonal realization of the cage) of *any* subset of flippers whose total size does not exceed 1. For example, the left cage in Figure 1 has three internally embedded flippers, one of size 1/2 and two of size 1/4.

4 Modules

We now have in place the tools with which we can specify and verify our general construction. We begin by describing the basic building blocks (modules). Each module has up to four designated terminals (a cage corner together with its incident flipper). Modules are connected by identifying some pairs of terminals. We conclude by describing how modules are combined to mimic the bipartite graph orientability problem described in Section 2.

4.1 Clause modules

A clause module is designed to model a clause vertex in the graph $G^{SAT}_{\mathcal{F}}$ with three incident edges at least one of which must be oriented away from the vertex (towards a "chosen" literal). The module, in one of its feasible hexagonal realizations, is illustrated in Figure 2 below. It is easy to verify that all seven embeddings, in which one or more of the flippers incident on vertices T, B or R are embedded externally, have hexagonal realizations. By Corollary 4, the remaining case, in which all flippers are embedded internally, is unrealizable.

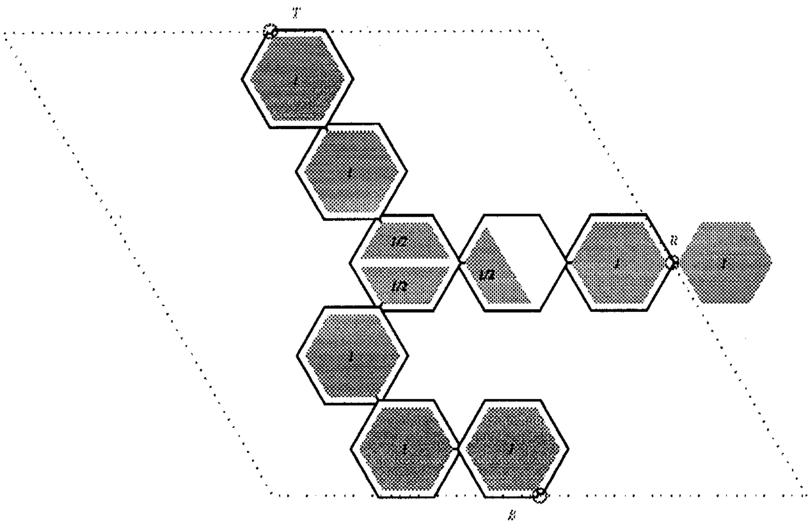

Fig. 2. Clause module with terminals T, B and R.

4.2 Consistency checking modules

In a satisfying truth assignment for $\mathcal{F}$ a variable or its negation (but not both) can be chosen to satisfy up to three clauses. This is modelled, in the graph $G_{\mathcal{F}}^{SAT}$, by constraining the edges incident with the vertex associated with each variable or its negation to all be oriented away from the vertex. Thus, with each variable v we associate a module formed from two submodules, one for each of the two literals v and $\overline{v}$. The submodule for the positive literal is depicted in one of its feasible realizations in Figure 3. Construct the submodule for the negative literal by rotating Figure 3 by 180 degrees. We can then construct the module for the variable by identifying terminal R with the rotated copy of terminal R. The flipper on terminal R in Figure 3 may be embedded outside the submodule, as illustrated (in which case it must be embedded inside the submodule corresponding to the negative literal). Alternatively, the flipper may be embedded inside the submodule, in which case it easy to see (by Corollary 4) that the flippers on terminals T, B, and L must be embedded outside the submodule.

4.3 Connector modules

The "choices" associated with clause modules need to be propagated to the appropriate consistency checking modules. This propagation can be achieved through the use of connector modules, built by chaining together cages where each successive pair has a shared (unit) flipper. Figure 4 illustrates one realization of a typical connector module. It is easy to see that any embedding of the module, in which at least one of the flippers incident on terminals L and R is

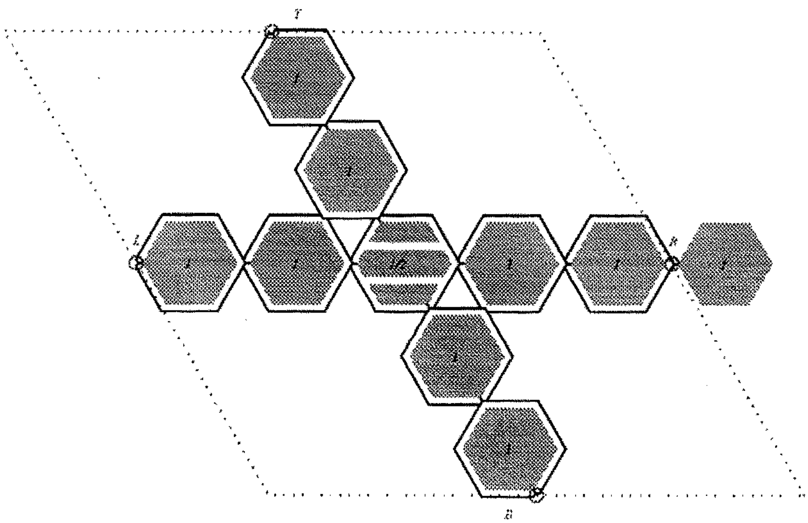

Fig. 3. Positive literal submodule with terminals L, R, T, and B.

embedded externally, has a hexagonal realization. Corollary 4 guarantees that the embedding with both of these flippers internal is unrealizable.

4.4 Crossover modules

As we have described things, connector modules alone are not sufficient to model the interconnection of clause and consistency checking modules. Since the graph $G_{\mathcal{F}}^{SAT}$, under our assumptions about $\mathcal{F}$, is not necessarily planar, the interconnections may be forced to cross. One way to avoid this is to restrict attention to formulas $\mathcal{F}$ for which the associated graph $G_{\mathcal{F}}^{SAT}$ is planar. (SATISFIABILITY restricted to this class of formulas is known to remain **NP**-hard [Kra94].) Alternatively, we can formulate a crossover module. One realization of such a module is described Figure 5. It is a simple exercise to confirm, using Corollary 4, that any embedding of this module, in which the flippers incident on both L and R or those on both T and B are embedded internally, is unrealizable. Similarly, all other embeddings of these flippers have hexagonal realizations.

5 Conclusions

We showed, in the preceeding section, how to construct modules suitable for implementing the reduction of the orientability problem for graphs $G_{\mathcal{F}}^{SAT}$ to the realizability problem for bounded ratio disk intersection or touching graphs. (One of the interesting features of this proof is the fact that it depends very little—at least at the level of abstraction that we have been able to describe it here—on the notion of contact.)

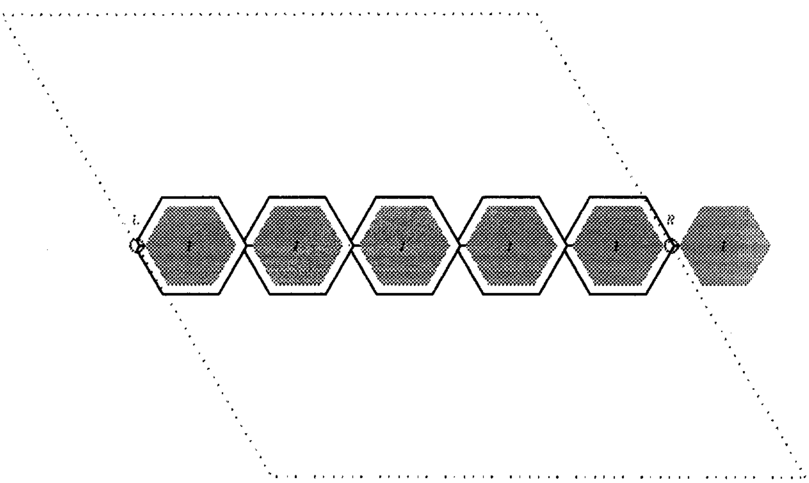

Fig. 4. Connector module with terminals L and R.

Although our proof follows the same general approach as that used in establishing the **NP**-hardness of the unit disk intersection (and touching) graph recognition problems, it differs in several important respects. Two of these, specifically the uniformity of our cages, and the reliance on asymptotic packing capacity bounds rather than properties of particular small configurations of disks, serve to simplify as well as generalize the reduction. This generalization can be exploited to prove analogous results for higher dimensions, for objects others than disks (eg. squares), and for certain grid-constrained versions of our problems [Bre95]. Although grid-constrained versions of our recognition problems (like grid graph recognition [BC87, Grä95]) are in fact **NP**-complete, it is not clear that the unconstrained versions are in **NP**. (Membership in **PSPACE** follows directly from results of Canny [Can88].)

In the case of disk touching graphs, the result in this paper depends critically on the specified bound on disk ratios. Indeed, as noted previously, in the absence of such a bound the recognition problem is straightforward. Our result says not only that the use of arbitrarily large disks is essential to the realization of planar graphs as touching graphs of disks (this is clear from degree considerations alone; in fact it follows from a result of [MP92] that an exponential—in the size of the graph—sized ratio may be required to realize some graphs), but that the need for large disks cannot be determined by efficiently checked conditions (unless **P** = **NP**). Our techniques do not allow us to address the case of non-constant but sub-exponential ratio bounds.

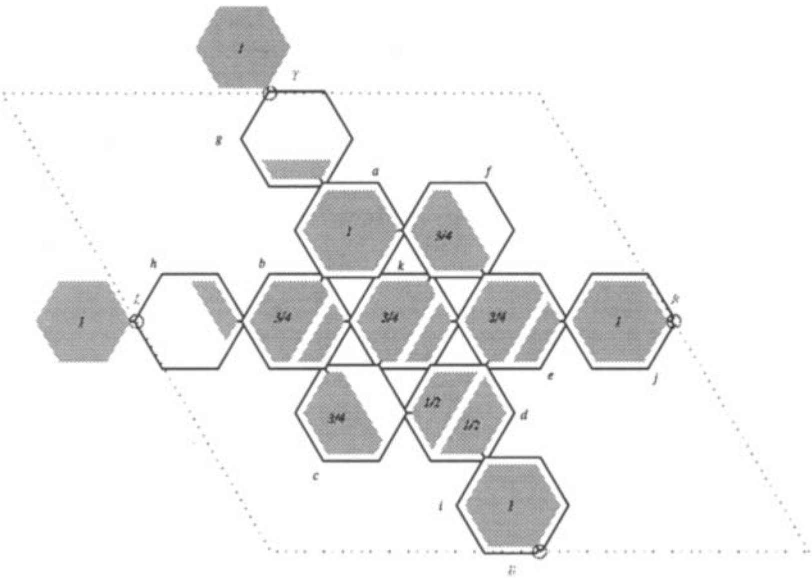

Fig. 5. Crossover module with terminals L, R, T, and B.

References

[Bre95] Heinz Breu. *Algorithmic Aspects of Constrained Unit Disk Graphs.* Ph.D. thesis, University of British Columbia, Vancouver, Canada. In preparation.

[BC87] Sandeep N. Bhatt and Stavros S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Information Processing Letters*, 25(4):263–267, 1987.

[BK93] Heinz Breu and David G. Kirkpatrick. Unit Disk Graph Recognition is NP-Hard. Technical Report 93–27, Department of Computer Science, University of British Columbia, August, 1993. (To appear in *Computational Geometry: Theory and Applications.*)

[BL76] K.S. Booth and G.S. Luecker. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *J. Comput. System Sci.*, 13:335–379, 1976.

[Can88] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the 20th Annual Symposium on the Theory of Computing*, pages 460–467, Chicago, Illinois, 2–4 May 1988.

[CCJ90] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1/3):165–177, 1990.

[FG65] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–355, 1965.

[Fis83] Peter C. Fishburn. On the sphericity and cubicity of graphs. *Journal of Combinatorial Theory, Series B*, 35:309–318, 1983.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: a Guide to the Theory of* **NP***-completeness.* W.H. Freeman and Company, 1979.

[Grä95] Albert Gräf. *Coloring and Recognizing Special Graph Classes.* PhD thesis, Musikwissenschaftliches Institut, Abteilung Musikinformatik, Johannes Gutenberg-Universität Mainz, February 1995. Available as technical report Bericht Nr. 20.

[Hal80] William K. Hale. Frequency assignment: theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980.

[Hav82a] Timothy Franklin Havel. *The Combinatorial Distance Geometry Approach to the Calculation of Molecular Conformation.* PhD thesis, University of California, Berkeley, 1982. Cited by [Fis83].

[JAMS91] David S. Johnson, Cecilia A. Aragon, Lyle.A. McGeogh, and Catherine Schevon. Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partioning. *Operations Research*, 39(3):378–406, May-June 1991.

[Kra91] J. Kratochvíl. String graphs II. Recognizing string graphs is NP-hard. *J. Combin. Theory Ser. B*, 52: 67–78, 1991.

[Kra94] J. Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52: 233–252, 1994.

[Kra95] J. Kratochvíl. Personal communication, February, 1995.

[MHR92] M.V. Marathe, H.B. Hunt III, and S.S. Ravi. Geometry based approximations for intersection graphs. In *Fourth Canadian Conference on Computational Geometry*, pages 244–249, 10–14 August 1992.

[MP92] S. Malitz and A. Papakostas. On the angular resolution of planar graphs. In *Proceedings of the 24th Annual Symposium on the Theory of Computing*, pages 527–538, May 1992.

[Rob68] Fred S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory*, pages 139–146, Academic Press, New York and London, February 1968. Proceedings of the Second Ann Arbor Graph Theory Conference.

[Rob91] Fred S. Roberts. *Quo Vadis, Graph Theory.* Technical Report 91–33, DIMACS, May 1991.

[Sac94] Horst Sachs. Coin graphs, polyhedra, and conformal mapping. *Discrete Mathematics*, 134: 133–138, 1994.

[Spi85] Jeremy Spinrad. On comparability and permutation graphs. *SIAM Journal on Computing*, 14(3):658–670, August 1985.