A-020

Efficient algorithms for network localization using cores of underlying graphs

Meng Li Yota Otachi Takeshi Tokuyama

Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan.
{limeng, otachi, tokuyama}@dais.is.tohoku.ac.jp

Abstract

Network localization is important for networks with no prefixed positions of network nodes such as sensor networks. We are given a subset of the set of $\binom{n}{2}$ pairwise distances among n sensors in some Euclidean space. We want to determine the positions of each sensors from the (partial) distance information. The input can be seen as an edge weighted graph. In this paper, we present some efficient algorithms that solve this problem using the structures of input graphs, which we call the *cores* of them. For instance, we present a polynomial-time algorithm solving the network localization problem for graphs with connected dominating sets of bounded size. This algorithm allows us to have an FPT algorithm for some restricted instances such as graphs with connected vertex covers of bounded size.

1 Introduction

Nowadays sensor networks are used for many important practical applications such as monitoring environmental data (see e.g. [8, 25]). Since the nodes in a sensor network do not have physical access to each other, sometimes we should construct it without prefixed positions of the nodes even if it is not a dynamic network; that is, the nodes are not moving. For example, assume that we want to monitor some contaminated environment. It is not possible to put a sensor node manually at a prefixed position since the area is contaminated. Thus we use some flying devices like unmanned helicopters to drop sensor nodes from high altitude. After that we can collect data by crawling the area by the same flying device. Using unmanned aerial vehicles has become a common technique in practical sensor networking [5]. To analyze the contaminated area in detail, it is useful to have spatial data of the nodes. With spatial information, we can decide which area is contaminated and which area is not. The problem to determine the positions of each node in network is the *network localization problem* [2]. Equipping each node with a GPS (Global Positioning System) device might be an answer. However, it would be too expensive and impractical if the number of nodes is large. Instead of equipping GPS devices, we consider the following setting:

- each node can communicate with *some* other nodes;
- if two nodes communicate, then they can measure the distance between them;
- the central device (e.g. a helicopter) collects the distance information with IDs.

The localization problem of this setting is formalized by using graphs as follows.

Problem: Weighted Graph Embeddability in d-space (WGEd)
Instance: A graph G with nonnegative weights $w_e \geq 0$ on each edge $e \in E(G)$.
Question: Is there a mapping $f\colon V(G) \to \mathbb{R}^d$ such that $w_{uv} = \mathrm{dist}(f(u), f(v))$ for each $uv \in E(G)$, where $\mathrm{dist}(f(u), f(v))$ is the Euclidean distance between $f(u)$ and $f(v)$? (We call such a mapping f a *d-embedding* of G.)

Unfortunately, WGEd is known to be strongly NP-hard in general and weakly NP-hard for cycles.

Theorem 1.1 (Saxe [23], Feder and Motwani [11]). *For every positive integer d, WGEd is NP-hard even if every edge has weight one or two. Furthermore, WGE1 is weakly NP-complete even for cycles.*

Theorem 1.1 implies that a partial distance matrix corresponding to a graph is not always helpful to decide the embeddability. Therefore, it is an interesting problem to ask which graphs (and which d) provide a sufficient condition for designing an efficient algorithm for deciding embeddability. This paper gives an initial work for this direction of research. Considering Theorem 1.1, we have the following natural questions: (1) If there is no long cycle without a chord, does the problem remain to be hard? (2) Is the complexity of the problem monotone with respect to the dimension d of the embedded space? (3) If there is a dominating set S for which the embedding can be uniquely determined or the number of possible embeddings is small enough, can we design an efficient algorithm for the reconstruction (this corresponds to the problem in surveying engineering)? We answer to each of these questions. Namely, we give a polynomial-time algorithms to solve WGEd for chordal graphs ($d \geq 1$), for cycles ($d \geq 2$), and for graphs with small connected dominating sets ($d = 1$). Our results with Theorem 1.1 give an evidence of that the complexities of the problem in lower- and higher-dimensions are incomparable in general. We also consider a variant of the problem defined by Feder and Motwani [11], in which two distinct points cannot have the same position.

We assume a computational model used by Saxe [24] in which real numbers are primitive data objects on which exact arithmetic operations (including comparisons and extraction of square roots) can be performed in constant time.

Some proofs are omitted due to space limitations.

2 Preliminaries

All graphs in this paper are finite, undirected, edge-weighted, and without self-loops and parallel edges. We denote the vertex set and the edge set of a graph G by $V(G)$ and $E(G)$, respectively. A graph is *connected* if it has a path between each pair of vertices.

A graph H is a *subgraph* of G, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H of G is *induced* with $V(H)$ if $E(H) = E(G) \cap \binom{V(H)}{2}$. A graph G is *chordal* if every induced cycle of G is of length three [14].

A vertex set $S \subseteq V(G)$ is a *dominating set* of G, if each vertex in $V(G) \setminus S$ has a neighbor in S. A vertex set $S \subseteq V(G)$ is a *p-dominating set* if every vertex in $V(G) \setminus S$ has at least p neighbors in S. For example, in K_3 any two vertices form a 2-dominating set. A vertex set $S \subseteq V(G)$ is a *vertex cover* of G, if every edge of G has an end in S. From the definitions, it is easy to see that if a graph G has no isolated vertex, then any vertex cover of a graph G is a dominating set of G. A dominating set (a vertex cover) S is a *connected dominating set* (a *connected vertex cover*, respectively) if $G[S]$ is connected (see Fig. 1).

Fig. 1 A connected dominating set and a connected vertex cover.

Feder and Motwani [11] studied th problem Graph Turnpike (GT), which is equivalent to the problem WGE1. They also studied the following variant of GT in which two distinct points are not allowed to have the same position.

Problem: Graph Turnpike with Distinctness (GTwD)
Instance: A graph G with nonnegative weights $w_e \geq 0$ on each edge $e \in E(G)$.
Question: Is there a mapping $f\colon V(G) \to \mathbb{R}$ such that $f(u) \neq f(v)$ for $u \neq v$, and $w_{uv} = |f(u) - f(v)|$ for each $uv \in E(G)$?

They showed that this variant is also weakly NP-hard for cycles [11]. Obviously, GTwD can be generalized to higher-dimensions. We call a variant of WGEd, in which two distinct points must have different positions, WGEd with Distinctness (WGEdwD).

3 The length of longest induced cycles and the dimension of spaces

In this section, we present answers to the following questions in Introduction.

1. If there is no long cycle without a chord, does the problem remain to be hard?
2. Is the complexity of the problem monotone with respect to the dimension d of the embedded space?

The first question is natural since no NP-hardness is known for the graphs of bounded length of induced cycles (this can be seen by carefully reading the proofs in [23, 11]). We shall prove that if the length of every induced cycle is no more than three, then the problems WGEd and WGEdwD can be solved in polynomial time for any d. To answer the second question, we consider the problem for cycles in d-space with $d \geq 2$. It turns out that the problem is hard for cycles only if $d = 1$. Thus the case $d = 1$ is somewhat exceptional for them. We first show the easier result on cycles and then prove the tractability for chordal graphs.

3.1 Cycles in higher-dimensional spaces

As we mentioned, it is known that WGE1 is NP-complete on cycles. Here, we shall show that for $d \geq 2$, WGEd can be solved in linear time for cycles.

Theorem 3.1. *If $d \geq 2$, then WGEd is solvable in linear time for cycles*

Proof. Omitted. □

3.2 Polynomial-time algorithm for WGEd on chordal graphs

Next we prove that WGEd on chordal graphs can be solved in polynomial time for every fixed positive integer d. To this end, we need some definitions.

A *separator* of a graph G is a vertex set $S \subseteq V(G)$ such that there exist two vertices of G that are connected in G but not in $G - S$. A separator of a graph G is a *clique separator* if it induces a complete graph in G. A vertex v of G is *simplicial* if $N_G(v)$ induces a complete graph. Let G be an n-vertex graph and $v_1, v_2, \ldots, v_n$ be an ordering of $V(G)$. For $1 \leq i \leq n$, we define G_i to be the subgraph of G induced by $\{v_i, v_{i+1}, \ldots, v_n\}$. Then the ordering $v_1, v_2, \ldots, v_n$ is a *perfect elimination ordering* if for $1 \leq i \leq n$, the vertex v_i is simplicial in the graph G_i. Note that $N_{G_i}(v_i)$ is a clique separator of G_i.

It is well known that chordal graphs are characterized by perfect elimination ordering, and a perfect elimination ordering of a chordal graph can be found in linear time.

Theorem 3.2 (Fulkerson and Gross [12])**.** *A graph G is chordal if and only if G has a perfect elimination ordering.*

Theorem 3.3 (Rose, Tarjan, and Lueker [22] and Tarjan and Yannakakis [26])**.** *A perfect elimination ordering of a chordal graph can be found in linear time.*

Saxe [24] showed that if a graph is a complete graph, then WGEd on it is easy and its d-embedding, if any, is unique.

Theorem 3.4 (Saxe [24, Appendix II])**.** *For any fixed d, WGEd can be solved in $O(m)$ time for an edge-weighted complete graph G of m edges. Furthermore, if G has a d-embedding f, then it is unique up to rotation and translation and f can be found in $O(m)$ time.*

Using Theorems 3.2, 3.3, and 3.4, we shall prove the polynomial-time solvability of WGEd on chordal graphs. We first prove the following lemma, which is of independent interest.

Lemma 3.5. *Let G be a connected edge-weighted graph, d be a fixed positive integer, and $S \subseteq V(G)$ be a clique separator of G. Let G_1 and G_2 be two induced subgraphs of G such that $V(G_1) \cap V(G_2) = S$ and $V(G_1) \cup V(G_2) = V(G)$. Then, G has a d-embedding if and only if both G_1 and G_2 have d-embeddings.*

Proof. Omitted. □

Now we are ready to prove the theorem.

Theorem 3.6. *For edge-weighted chordal graphs, WGEd can be solved in polynomial time for any fixed positive integer d.*

Proof. Omitted. □

Note that the result in this subsection can be seen as a variant of a result by Laurent [21] who showed that the problem to decide whether a chordal graph can be embedded in d-space, for some d (not given), is solvable in polynomial time.

4 Algorithms for graphs with dominating cores

In geometry and surveying engineering, it is well known that if we have a position of a simplex T and one knows all distances from p to the set of $d+1$ vertices of T, the position of p is uniquely determined. Thus, we can consider $d(d+1)$ variables corresponding to the positions of vertices of T to have a system of equations that seems to be numerically soluble if d is a constant. However, we need to consider a degenerate case. For example, suppose that $d = 3$ and we have m points on a line in space, and the rest of $n - m$ points are located on a plane perpendicular to the line. Then, there remains exponential number of possible locations even if we have the set of all distances corresponding to the bipartite graph, and we currently have no polynomial-time solution for the general case. However, for $d \in \{1, 2\}$, we can solve the problem with some assumption.

4.1 General frameworks for $d \in \{1, 2\}$

For $d \in \{1, 2\}$, we can solve the problem if a graph has a dominating set for which the possible embeddings are efficiently enumerated. Such a dominating set can be seen as a *core* of the sensor network. We first present the following general frameworks.

Theorem 4.1. *Let G and S be a given n-vertex graph and its dominating set which is also given. If all possible candidates of 1-embeddings of $G[S]$ can be enumerated in $O(g(|S|))$ time, then we can solve WGE1 and WGE1wD in $O(g(|S|) \cdot n^2)$ time.*

Proof. Omitted. □

Theorem 4.2. *Let G and S be a given n-vertex graph and its 2-dominating set which is also given. If all possible candidates of 1-embeddings of $G[S]$ can be enumerated in $O(g(|S|))$ time, then we can solve WGE2wD in $O(g(|S|) \cdot n^2)$ time.*

Proof. Omitted. □

Note that our proof technique used in the proofs of Theorems 4.1 and 4.2 can not be used directly for the case $d \geq 3$. This is because instead of assuming distinctness, we have to assume the *general position* constraint for $d \geq 3$. However, if we use SAT as in the proofs to check the general position constraint, then we have some clauses with more than two literals. This make the SAT instance intractable since the k-SAT problem is NP-hard for any fixed $k \geq 3$ [13].

4.2 Applications of general frameworks

In this subsection, we present several practical applications of our frameworks; that is Theorems 4.1 and 4.2. The simplest application is for graphs with dominating cliques. Combining Theorems 3.4, 4.1, and 4.1, we have the following corollary.

Corollary 4.3. *For graphs with dominating cliques, WGE1 and WGE1wD can be solved in $O(n^2)$ time. For graphs with 2-dominating cliques, WGE2wD can be solved in $O(n^2)$ time, where n is the number of vertices.*

From the corollary above, one may think that if a graph has a dominating core with a unique embedding, then the localization problem can be solved in polynomial time. In practice, if sensors of a two-dimensional sensor network are densely enough distributed, then it is likely that the sensor network has a large subgraph G that has 2-dominating set S such that $G[S]$ has a unique embedding. If the localization problem can be solved for $G[S]$, then the problem is also solvable for G by Theorems 4.2. If G covers a large part of the sensor network, then we can just ignore the remaining part or may locate the remaining part using the embedding of G. Aspnes et al. [2] studied the localization problem for graphs with unique embeddings using *rigidity theory* (see e.g. [15, 19, 20]), and show that the localization problem for these graphs is NP-hard, unfortunately. However, it is not known whether the problem is NP-hard for rigid graphs with additional conditions; for example, rigid *unit disk graphs* (see [4] for the definition of unit disk graphs).

The next application shows that although its embedding is not unique, a small connected dominating set makes the problem easy.

Theorem 4.4. *Given an edge-weighted graph G with n vertices and its connected dominating set S of size k, WGE1 and WGE1wD can be solved in $O(2^k n^2)$ time.*

Proof. Let T be a spanning tree of $G[S]$. From a 1-embedding of T, we can obtain an orientation of the edges $E(T)$. If f_S is a 1-embedding of $G[S]$, then it is also a 1-embedding of T. Since the number of possible orientations of T is $O(2^k)$, we can conclude that the number of all non-congruent 1-embeddings of $G[S]$ is $O(2^k)$. Now we can apply Theorem 4.1. This completes the proof. □

Note that the above theorem is a generalization of Theorem 3 in [11], which states that WGE1 and WGE1wD can be solved in polynomial time if the input graph has a vertex adjacent to all other vertices.

An example of graphs that have small connected dominating sets is a graph G with a spanning complete bipartite graph K_{n_1,n_2}. Let $v_1, v_2 \in V(G)$ be vertices that have different colors in a proper coloring of its spanning complete bipartite graph. Clearly, $S = \{v_1, v_2\}$ is a connected dominating set of G. Thus we have the following corollary.

Corollary 4.5. *Let G be an edge-weighted n-vertex graph with a spanning complete bipartite graph. Then WGE1 and WGE1wD can be solved in $O(n^2)$ time.*

Graphs with spanning complete bipartite graphs are called *join graphs*, since they are constructed by the *join* operation [17]. The class of join graphs includes very important graphs such as complete graphs, complete bipartite graphs, and complete k-partite graphs. More generally, any connected *cograph* [6] is a join graph. Cographs play important roles in algorithmic graph theory, since they are precisely the graphs of *clique-width* at most two [7, 18].

In Theorem 4.4, we assumed that a small connected dominating set is given. Therefore, if it is not given, then we should find a small connected dominating set. A naive way is to enumerate all vertex subset of size at most k in $O(n^k)$ time. For each subset, we can check whether it is a connected dominating set in $O(m)$ time. Therefore, we have the following corollary.

Corollary 4.6. *Given a graph with n vertices and m edges, we can solve WGE1 and WGE1wD in $O(n^k m + 2^k n^2)$ time if the graph has a connected dominating set of size at most k.*

Note that since the problem of finding a connected dominating set is W[2]-hard when parameterized by the solution size (see e.g. [10]), it is impossible to improve upon the $O(n^k)$ time complexity for finding a connected dominating set to $O(c^k \mathrm{poly}(n))$ time for any constant c unless W[2] = FPT.[*1] Since every vertex cover of a connected graph is its dominating set an FTP algorithm for connected vertex covers immediately yields an FPT time algorithm for WGE1 and WGE1wD parameterized by the size of the minimum connected vertex cover. It is known that a connected vertex cover of size k can be found, if any, in $O(6^k n + 4^k n^2 + n^2 \log n + mn)$ time [16]. However, we use an $O(mn)$-time 2-approximation algorithm presented by Arkin, Halldórsson, and Hassin [1] to obtain a better running time.

Corollary 4.7. *Given a graph with n vertices and m edges, we can solve WGE1 and WGE1wD in $O(4^k n^2 + mn)$ time if the graph has a connected vertex cover of size at most k.*

Proof. We first find a connected vertex cover C of size at most $2k$ by using the $O(mn)$-time 2-approximation algorithm of Arkin et al. [1]. By Theorem 4.4, we can solve WGE1 and WGE1wD in $O(4^k n^2)$ time since C is also a connected dominating set of size at most $2k$. The combined time complexity is $O(4^k n^2 + mn)$. □

In the rest of this section, we shall discuss the two-dimensional case. We need the notion of *k-trees* which is defined as follows:

- the complete graph of k vertices is a k-tree;
- if G is a k-tree, then the graph obtained from G by adding a simplicial vertex of degree k is also a k-tree.

It is easy to see that a k-tree is a chordal graph. With these terminologies, we can have a two-dimensional generalization of Theorem 4.4 as follows.

Theorem 4.8. *Given an edge-weighted graph G with n vertices and its 2-dominating set S of size k such that $G[S]$ have a spanning 2-tree, WGE2wD can be solved in $O(2^k n^2)$ time.*

Proof. By Theorem 4.2, it suffices to show that all 2-embeddings of $G[S]$ can be enumerated in $O(2^k)$ time. Since the problem has the distinctness constraint, we can assume that each edge in G has positive weight. Let R be a spanning 2-tree of $G[S]$, and let $r_1, \dots, r_k$ be a perfect elimination ordering of R. We shall construct a 2-embedding by embedding vertices in the reverse ordering $r_k, r_{k-1}, \dots, r_1$. We can first embed r_k and r_{k-1} uniquely. Then, when embedding each r_i, $1 \le i \le k-2$, r_i has two neighbors that are already embedded and have different positions. Thus, we have only two possibilities for each r_i. This implies that the number of possible 2-embeddings (up to motion) of S is $O(2^k)$. The enumeration of the candidates can be easily done in $O(2^k)$ time. Thus the theorem holds. □

[*1] If W[2] = FPT, then *Exponential Time Hypothesis* fails [9].

An example of graphs that have small 2-dominating sets with 2-tree spanning trees is a graph G with a spanning complete tripartite graph K_{n_1,n_2,n_3}. Let $v_1, v_2, v_3 \in V(G)$ be vertices that have different colors in a proper coloring of its spanning complete tripartite graph. Then, it is not difficult to see that the set $S = \{v_1, v_2, v_3\}$ is a 2-dominating set of G, and the triangle $G[S]$ itself is a spanning 2-tree of $G[S]$. It is easy to see that such a set S can be found in $O(n^4)$ time by naively examine all $\binom{n}{3}$ vertex triples in $O(n)$ time. Therefore, we have the following corollary.

Corollary 4.9. *Let G be an edge-weighted n-vertex graph with a spanning complete tripartite graph. Then WGE2wD can be solved in $O(n^4)$ time.*

5 Concluding remarks

We have shown that WGEd can be solved in polynomial time for chordal graphs ($d \ge 1$) and for cycles ($d \ge 2$). We have also studied the problems on graphs with small connected dominating set, and have shown that for such graphs WGE1 and WGE1wD can be solved in polynomial time and WGE2wD can be solved in polynomial time if we add a condition. Our results on graphs with small connected dominating set may be considered as that if a sensor network has a small *core*, then the localization problem can be solved efficiently.

Feder and Motwani [11] studied a variant of WGE1, denoted by GTwD, in which two distinct points cannot have the same position. This variant is very natural, and should be investigated more extensively. There is an interesting problem on GTwD.

Problem 5.1. Can GTwD be solved in polynomial time for trees?

We think it might be NP-hard. It is not difficult to see that any tree can be embedded in the line (if we do not care about distinctness) by putting the roof r of a tree at the origin and putting the other points v at the point ℓ, where ℓ is the sum of the weights of edges in a unique r–v path in the tree. Also, by slightly modifying this embedding, we can derive an 2-embedding of the tree in which any two distinct vertices have different positions. Note that embedding trees in $\mathbb{Z}^2$ with distinctness is NP-hard even if every edge has the same weight [3].

References

[1] E. M. Arkin, M. M. Halldórsson, and R. Hassin. Approximating the tree and tour covers of a graph. *Inform. Process. Lett.*, 47:275–282, 1993.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Trans. Mobile Comput.*, 5:1663–1678, 2006.

[3] S. N. Bhatt and S. S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Inform. Process. Lett.*, 25:263–267, 1987.

[4] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Math.*, 86:165–177, 1990.

[5] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation*, 2004.

[6] D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3:163–174, 1981.

[7] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Appl. Math.*, 101:77–114, 2000.

[8] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley, 2010.

[9] R. G. Downey, V. Estivill-Castro, M. R. Fellows, E. Prieto, and F. A. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. *Electron. Notes Theor. Comput. Sci.*, 78:209–222, 2003.

[10] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1998.

[11] T. Feder and R. Motwani. On the graph turnpike problem. *Inform. Process. Lett.*, 109:774–776, 2009.

[12] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[14] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, volume 57 of *Annals of Discrete Mathematics*. North Holland, second edition, 2004.

[15] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. AMS, 1993.

[16] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of vertex cover variants. *Theory Comput. Syst.*, 41:501–520, 2007.

[17] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Massachusetts, 1969.

[18] P. Hliněný, S. Oum, D. Seese, and G. Gottlob. Width parameters beyond tree-width and their applications. *Comput. J.*, 51:326–362, 2008.

[19] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Combin. Theory Ser. B*, 94:1–29, 2005.

[20] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Eng. Math.*, 4:331–340, 2002.

[21] M. Laurent. Polynomial instances of the positive semidefinite and euclidean distance matrix completion problems. *SIAM J. Matrix Anal. Appl.*, 22:874–894, 2000.

[22] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5:266–283, 1976.

[23] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *17th Allerton Conf. Commun. Control Comput.*, pages 480–489, 1979.

[24] J. B. Saxe. Two papers on graph embedding problems. Technical Report CMU-CS-80-102, Department of Computer Science, Carnegie-Mellon University, 1980.

[25] K. Sohraby, D. Minoli, and T. Znati. *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience, 2007.

[26] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13:566–579, 1984.