Efficient Algorithm for Graph Isomorphism Problem

Rama Murthy Garimella

October 23, 2023

Efficient Algorithm for Graph Isomorphism Problem

Garimella Ramamurthy[1]

Mahindra University, Hyderabad, India
rama.murthy@mahindrauniversity.edu.in

Abstract. In this research paper, it is proved that two arbitrary graphs are isomorphic if and only if the quadratic forms associated with the two adjacency matrices are same(upto reordering the monomials). Based on the proof, a polynomial time algorithm is designed for graph isomorphism problem(i.e. effectively deciding whether two graphs are isomorphic). The algorithm requires $O(N^3)$ comparison operations. It is reasoned that there is no algorithm for testing isomorphism of arbitrary graphs on N vertices whose time complexity is less than $O(N^3)$. Thus the proposed algorithm meets lower bound on time complexityAlso, a polynomial time algorithm for testing whether two graphs are isomorphic is designed under the condition that the associated adjacency matrices are non-singular and are related through a symmetric permutation matrix. The algorithms are essentially based on linear algebraic concepts related to graphs. Also, some new results in spectral graph theory are discussed.

Keywords: Graph Theory · Edge connectivity · Spectral Graph Theory · Symmetric matrix · Isomorphism.

1 Introduction

Directed/undirected, weighted/unweighted graphs naturally arise in various applications. Such graphs are associated with matrices such as weight matrix, incidence matrix, adjacency matrix, Laplacian etc. Such matrices implicitly specify the number of vertices/ edges, adjacency information of vertices (with edge connectivity) and other related information (such as edge weights). In recent years, there is explosive interest in capturing networks arising in applications such as social networks, transportation networks, bio-informatics related networks (e.g. gene regulatory networks) using suitable graphs. Thus, network science led to important problems such as community extraction, frequent sub-graph mining etc. In many applications the problem of deciding whether two given graphs are isomorphic (i.e. the two graphs are essentially same upto relabeling the vertices) naturally arises. This research paper provides one possible solution to such a problem. This research paper is organized in the following manner. In section 2, relevant research literature is briefly reviewed. In section 3, one polynomial time algorithm, to test if two graphs are isomorphic is discussed. In section 4, necessary and sufficient condition for two arbitrary graphs to be isomorphic is proved and a polynomial time algorithm is proposed. The research paper concludes in section 5.

2 RELATED RESEARCH LITERATURE

L. Babai [1, 2] recently claimed quasi-polynomial time algorithm for determining if two graphs are isomorphic [1] . This is the most recent contribution to the graph isomorphism problem. Specifically, research in [2] showed that graph isomorphism problem can be solved in $exp(logn)^{(o(1))}$ time [2]. For the problem, the previous known best bound was $exp(o(\sqrt{(nlogn)}))$ where n is the number of vertices. There are other research efforts which provide approximate solutions to the problem (i.e. approximate algorithms were designed)[3], [4], [5], [6], [8], [11], [12], [13]. Also, the problem of solving Graph Isomorphism has been attempted using the quadratic non-negative matrix factorization problem [14].

3 Polynomial Time Algorithm for Graph Isomorphism Problem(under some Conditions)

We now briefly review relevant results from spectral graph theory.

3.1 Spectral Graph Theory

Spectral graph theory deals with the study of properties of a graph in relationship to the characteristic polynomial, eigenvalues and eigenvectors of matrices associated with the graph, such as its adjacency matrix or Laplacian matrix.

An undirected graph has a symmetric adjacency matrix A and hence all its eigenvalues are real. Furthermore, the eigenvectors are orthonormal.

We have the following definition
Definition: An undirected graph's "spectrum" is the multiset of real eigenvalues of its adjacency matrix, A. Graphs whose spectrum is same are called co-spectral.

Remark 1. It is well known that isomorphic graphs are co-spectral. But co-spectral graphs need not be isomorphic. Thus spectrum being same is only a necessary condition for graphs to be isomorphic (but not sufficient)[12], [13]. Thus, it is clear that the eigenvectors of adjacency matrices of isomorphic graphs must be constrained in a suitable manner (orthonormal basis of eigenvectors of the symmetric adjacency matrices are somehow related for isomorphic graphs) [9].

3.2 Polynomial Time Algorithm to determine co-spectral Graphs

.

Lemma 1: The problem of determining if two graphs are Co-Spectral is in P (i.e. a polynomial time algorithm exists).

Proof: Since the elements of adjacency matrix are 0's and 1's, the characteristic polynomial of it is a polynomial with integer coefficients. Thus, there exists a polynomial time algorithm (PTA algorithm) [7] to compute the zeroes of such polynomial i.e. spectrum of associated graph. Thus the problem of determining if two graphs are co-spectral is in P (class of polynomial time algorithms) Q.E.D.

Note: By Perron-Frobenius theorem, the spectral radius of an irreducible adjacency matrix (non-negative matrix) is real, positive and simple [10]. Thus, to check for the necessary condition on isomorphic graphs, a first step is to determine if the spectral radii of two graphs are exactly same.

Definition: Two graphs are isomorphic, if the vertices of one graph are obtained by relabeling the vertices of another graph.

3.3 Necessary and Sufficient Conditions: Isomorphism of Graphs

Necessary Conditions: Isomorphism of Graphs The following necessary conditions for isomorphism of graphs with adjacency matrices A, B can be checked before applying the following algorithm. Check if Trace(A) = Trace(B) and if Determinant(A) = Determinant(B). Check if Spectral radius of A, B are same. This can be done using the Jacobi's algorithm for computing the largest zero of a polynomial. Since the coefficients of characteristic polynomial are integers, we expect the computational complexity of this task to be small. If this step fails, all other zeroes need not be computed [9].

We now formulate the problem of determining the isomorphism of graphs in two equivalent ways. Let the symmetric matrices A and B be the adjacency matrices of two graphs.

- Quadratic Non-Negative Matrix Factorization[12],[14]:

The problem of determining isomorphism of two graphs boils down to determining if a Permutation matrix P exists such that that

$$B = PAP^T \tag{1}$$

Such a problem is already being attempted using the approach based on Quadratic Non-Negative Matrix Factorization [14]. The results proposed for such a problem readily apply for determining isomorphism of two graphs.

• Algebraic Riccati Equation: Symmetric Permutation Matrix P: The quadratic matrix equation in (1) (non-linear) has resemblance to the Symmetric Algebraic Riccati Equation of the following form

$$XCX - AX - XA^T + B = 0$$

(with compatible matrices X,C,A,B), where the matrices B and C are symmetric and X is the unknown matrix. As can be readily seen the matrix equation (1) is a structured symmetric Algebraic Riccati equation with P being a symmetric unknown matrix. The known algorithms for solving such a Riccati equation may readily apply for testing isomorphism of two graphs for which P is a symmetric permutation matrix. Specifically, there are efforts to determine the non-negative matrix solutions of Riccati equation [15], [16]. It should be kept in mind that the solution of algebraic Riccati equation that is of interest to us is a structured {0,1} matrix.

Explicit Solution when the Adjacency Matrices of the graphs are non-singular and are related through Symmetric Permutation Matrix:

Lemma 2: Under the above assumptions, two graphs with adjacency matrices {B, C} (whose eigenvalues need NOT be distinct) are isomorphic only if

$$X = [Matrix\ Square\ Root(BC)]C^{-1}$$

is a Permutation matrix.

Proof:

Necessity: Suppose the graphs with adjacency matrices {B, C} are isomorphic and are related through a symmetric permutation matrix, P. Then we have that

$$PCP = B.$$

Hence multiplying on both sides by C, we have that

$$(PC)PC = BC = (PC)^2$$

Thus, using well known result from linear algebra BC is positive definite and hence has a unique positive definite matrix square root. Thus,we have that
PC= Matrix Square root (B C).

Since, C is non-singular, we have that

$$P = [Matrix\ Square\ Root(BC)]C^{-1}$$

which is necessarily a permutation matrix. Thus the above condition is necessary. Thus, if the graphs are isomorphic, the test declares the correct result. Q.E.D.

Note: The above condition need not be sufficient, if the matrix square root is not unique.

Note: Unique postive definite Matrix square root of a positive definite matrix can easily be computed in polynomial time using well known results in linear algebra. Duplication of details is avoided for brevity.

•Algorithm:(If graphs are isomorphic, the algorithm declares them correctly). From well known facts from linear algebra, the condition in the above lemma can be checked using a polynomial time algorithm.
Algorithm : Since

$$X = [Matrix \;\; Square \;\; Root(BC)]C^{-1}$$

the testing of necessary condition involves the following steps
i) Multiplication of symmetric matrices B,C.
ii) Computation of Matrix Square root of BC.
iii) Inversion of symmetric matrix 'C'.
iv) Multiplication of [Matrix Square Root(BC)] and C^{-1}. v) Checking if X is a permutation matrix.

Note: The research efforts based on strassen′s algorithm for matrix multiplication readily apply in this case.

From well known complexity of linear algebraic algorithms, the steps (i), (ii), (iii) take O(N^3) operations (multiplication, addition). If 'BC' is positive definite and diagonalizable, its unique matrix square root takes O(N^3) operations.

Also, if BC is a positive definite diagnonalizable matrix i.e.

$$BC = \bar{V}\bar{D}\bar{V}^{-1},$$

where $\bar{D}$ is a diagonal matrix of positive eigenvalues. The unique positive definite matrix square root of BC is given by $\bar{V}\bar{D}^{1/2}\bar{V}^{-1}$, where $\bar{D}^{1/2}$ is the diagonal matrix of positive square roots of eigenvalues of $\bar{B}\bar{C}$.

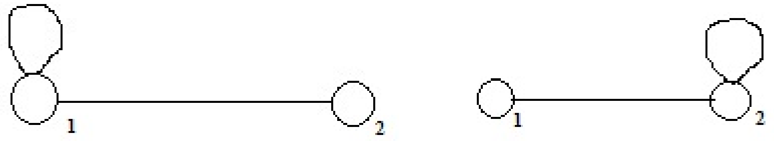
From linear algebra, a polynomial time algorithm is readily available for the above computation of positive definite square root of $\bar{B}\bar{C}$.

Detailed description of the algorithm is avoided for brevity.

Note: In the case of matrix equation, X C X=B, if BC is a positive definite matrix, unique solution which is positive definite can be determined using approach similar to Lemma 2.

Remark 2: In view of the above two equivalent problems, the results available for solution of one problem can be utilized in the solution of other problems.

Example: Consider the following two graphs with two vertices.



Let their symmetric adjacency matrices be given by

$$\bar{B} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Both the matrices$\{\bar{B}, \bar{C}\}$ are nonsingular and are isomorphic. We have that, with

$$\bar{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \bar{P}^T$$

it readily follows that

$$\bar{P}\bar{C}\bar{P}^T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \bar{B} \quad (as \quad expected).$$

We now check the test in Lemma 2. We have that

$$C^{-1} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

We would like to determine the matrix square root of $\bar{B}\bar{C}$

We have that

$$\bar{B}\bar{C} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} Let\ the\ square\ root\ of\ \bar{B}\bar{C}\ be\ the\ matrix\ G = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

We have that

$$G^2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

Solving the equations we have that c = 0, b =1 and a = d = +-1. Thus, we get atleast two solutions for $\bar{G}$ i.e.

$$\bar{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, -\bar{G} = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix}$$

But, the only positive definite square root of $\bar{B}\bar{C}$ (a positive definite matrix with eigenvalues 1, 1) is

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

It follows that

$$[Matrix\ Square\ root(BC)]C^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad a \quad permutation \quad matrix$$

4 Spectral Graph Theory: Polynomial time algorithm for testing isomorphism of arbitrary graphs.

Fact: While the adjacency matrix depends on the vertex labeling, its spectrum is graph invariant i.e. suppose adjacency matrix $\bar{B}$ is obtained by relabeling the vertices of a graph with adjacency matrix $\bar{A}$ (i.e. edge connectivity in $\bar{A}$ is preserved in $\bar{B}$). Then $\bar{A}$,$\bar{B}$ have the same set of eigenvalues.

Note: We use the terminology that graphs are co-spectral/isomorphic if their adjacency matrices $\bar{A}, \bar{B}$ are co-spectral/isomorphic.

We now provide an interesting proof of the above fact. In fact, the corollary 1 of theorem 1 is a much stronger result. We need the following well know theorem. Rayleigh's theorem: The local optima of the quadratic form associated with a symmetric matrix A on the unit Euclidean hypersphere $\{X : X^T X = 1\}$ occur at the eigenvectors with the corresponding value of the quadratic form being the eigenvalue.

- Theorem 1:

i) Eigenvalues of the adjacency matrix of an undirected graph, $\bar{A}$ are invariant(remains same) under relabeling of the vertices i.e. Adjacency matrices $\bar{A}$, $\bar{B}$ are co-spectral (with $\bar{B}$ being the adjacency matrix of graph obtained by relabeling the vertices of graph $\bar{A}$).

ii) Furthermore, graphs with adjacency matrices $\bar{A}$, $\bar{B}$ are isomorphic if and only if (i.e. necessary and sufficient condition) the quadratic forms associated with $\bar{A}$, $\bar{B}$ are same (upto reordering the monomials).

Proof: In Rayleigh's theorem, eigenvalues of $\bar{A}$,$\bar{B}$ are the local optimum of associated quadratic forms evaluated on the unit hyper sphere respectively. Thus, to prove that graphs $\bar{A}$, $\bar{B}$ (i.e. graphs with adjacency matrices $\bar{A}$, $\bar{B}$) are co-spectral, it is sufficient to reason that the quadratic form remains invariant under relabeling of the vertices.

Proof of part(i): We have that

$$X^T A X = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} x_i x_j$$

$$= x_1(x_{i_1} + x_{i_2} + ... + x_{i_N}) + x_2(x_{j_1} + x_{j_2} + ... + x_{j_N}) + ... +$$

$$x_N(x_{N_1} + x_{N_2} + ... + x_{N_N})$$

where, for instance, $i_1, i_2, ...i_N$ are the vertices connected to the vertex 1(one) (and similarly other vertices).

Now, under relabeling of vertices, the monomials are just reordered in the quadratic form expression i.e. $X^T AX$. Hence, from the above expression, it is clear that the quadratic form remains invariant under relabeling of the vertices. Thus by Rayleigh's theorem, eigenvalues of A remain invariant under relabeling

of vertices.
Summarizing, it is clear that the quadratic form remains invariant under relabeling of the vertices. Specifically, relabeling just reorders the expressions. Thus, the eigenvalues of A remain invariant under relabeling of vertices.

Proof of part(ii): From part(i) above, if two graphs with adjacent matrices $\bar{A}$, $\bar{B}$ are isomorphic, it is necessary that the quadratic forms associated with $\bar{A}$, $\bar{B}$ are same(upon reordering o the monomials).

Now, we prove that the condition is also sufficient. To this end, we need to reason that, if the quadratic forms associated with adjacency matrices $\bar{A}$,$\bar{B}$ are same, then $\bar{B} = \bar{P}\bar{A}\bar{P}^T$ where $\bar{B}$ is a permutation matrix.

We have that $X^TAX = X^TBX$. But monomials in X^TBX are reordered from those of X^TAX. For instance, Node 1 based monomials are mapped to node l_1'based monomials.
Thus

$$X^TBX = \sum_{i=1}^{N}\sum_{j=1}^{N} b_{i,j}x_ix_j$$

$$= x_{l_1}(x_{s_1}+x_{s_2}+...+x_{s_k})+x_{l_2}(x_{r_1}+x_{r_2}+...+x_{r_l})+...+x_{l_N}(x_{t_1}+x_{t_2}+...+x_{t_m})$$

. Thus, the monomials in X^TBX are obtained by permuting those in X^TAX. Equivalently, One adjacency matrix $\bar{B}$ is obtained by reordering the elements of other adjacency matrix, $\bar{A}$.

Hence $\bar{B} = \bar{P}\bar{A}\bar{P}^T$, where $\bar{P}$ is permutation matrix. Thus, the condition that $X^TAX = X^TBX$ is sufficient for the associated graphs to be isomorphic. Q.E.D

Note: Showing that $\bar{X}^TAX = X^TBX$, is much stronger than ensuring that $\bar{A}, \bar{B}$ are co-spectral.

Corollary 1: Since the quadratic form remains invariant under relabeling of the vertices, the local optima of the quadratic form over various constraint sets remain invariant. For instance, the stable values (i.e. local optima of quadratic form associated with a symmetric matrix over the unit hypercube) remain same under relabeling of the vertices of graph.

Corollary 2: Isomorphism of Graphs: From the above proof, it is clear that if two graphs (with associated adjacency matrices A, B) are isomorphic, the associated quadratic form being same is a necessary condition. Also, if the quadratic forms associated with adjacency matrices are same, then it readily follows that one adjacency matrix can be obtained by reordering the elements of other adjacency matrix. Thus, the author reasons that it is also a sufficient condition. Hence, in the following a polynomial time algorithm is designed to check if the quadratic forms associated with two adjacency matrices are same (by matching the second degree monomials in the associated quadratic forms).

Polynomial Time Algorithm : Consider an arbitrary row of $\bar{A}$(e.g row 'i') i.e. consider i^{th} node of the graph. Let the nodes$\{j_1, j_2, ..., j_l\}$ be connected to it i.e. in the quadratic form associated with $\bar{A}$, the monomials based on $(i, j_1), (i, j_2), ..., (i, j_l)$ are present $(1 \leq i \leq N)$.

Step: Matching monomials associated with node 'i' with those at all -other nodes of graph $\bar{B}$.

Consider node 'k'. Check if the non zero tuples associated with node 'k' (k,r): $\{1 \leq r \leq N\}$ for some $(1 \leq k \leq N, k \neq i)$ match with the tuples $(i, j_s) : 1 \leq j_s \leq N\}$.
This procedure takes atmost N^2 matching/comparisons. If i^{th} node based tuples donot match with the tuples associated with any other node of graph $\bar{B}$, the procedure terminates and the graphs are not isomorphic. Else, let 'h' be the node which matches with i. Remove row 'i' from $\bar{A}$ and row 'h' from $\bar{B}$.

Repeat the above step with the remaining nodes of graph(matching tuples associated with the remaining nodes). Thus we have $O(N^3)$ algorithm..

Note: At any stage of iteration, tuples in the matrix 'A' are matched with those in $\bar{B}$

Claim: Since $\bar{A}, \bar{B}$ are symmetric matrices, the number of comparisons required are much smaller. The above algorithm based on part(ii) of theorem 1 is equivalent to the following algorithm.

Step 1: Check if row 1 of $\bar{A}$ matrix is same as any one of the N rows of $\bar{B}$ i.e. for $1 \leq j \leq N$, check if row 1 is exactly equal to atleast one row j (while rows in $\bar{A}$, $\bar{B}$ are not matched). If No, graphs are not isomorphic. If YES, let row 1 equal row 'k'. Remove row 1 from $\bar{A}$ and row 'k' from $\bar{B}$, and proceed to step2.

Step 2: If rows in $\bar{A}$, $\bar{B}$ are larger than or equal to 1, then proceed to step 1 $\implies O(N^3)$ algorithm.

Note: The two algorithms are equivalent.

Under appropriate conditions, we can prove that there is no serial algorithm for testing isomorphism of two arbitrary graphs whose complexity is less than $O(N^3)$.

Note: Since $\bar{A}, \bar{B}$ are symmetric, the number of comparisons are reduced.

Lemma 3: There is no serial algorithm for testing isomorphism of two arbitrary graphs on N vertices, whose time complexity is less than $O(N^3)$.

Proof: Any algorithmic procedure for testing isomorphism of two graphs necessarily requires checking if vertices in the adjacency matrix of A have the same connectivity pattern (i.e. match the rows in $\bar{A}$ with those of graph $\bar{B}$). The above algorithm essentially perform the same steps. Hence it meets the lower bound on time complexity of a serial algorithm. Q. E. D.

Note: Consider a Homogeneous multi-variate polynomial associated with, say, a Fully Symmetric Tensor. The local optima of such a homogeneous form over various constraint sets such as Euclidean Unit Hypersphere, multi-dimensional hypercube remain invariant under relabeling of nodes of a non-planar graph. Effectively relabeling of vertices, reorders the monomials (terms in multivariate polynomial).

5 Conclusion

In this research paper, results in spectral graph theory of structured graphs are discussed. Polynomial time algorithms for testing if two graphs are isomorphic are discussed.

6 Acknowledgment

The author thanks Prof. L. Lovasz and Prof. Sanjeev Arora for providing some references. The author also thanks Prof. Dario Bini for discussion related to structured algebraic Riccati equation. The author dedicates the algorithms to god for enabling their publication in my 60^{th} year (research done in 2015).

References

1. Babai, László. "Graph isomorphism in quasipolynomial time." In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing, pp. 684-697. 2016.
2. Laszlo Babai,Graph Isomorphism in Quasi polynomial Time, Available on Archive, arXiv:1512.03547v2 [cs.DS] 19 Jan 2016.
3. Corneil, Derek G., and Calvin C. Gotlieb. "An efficient algorithm for graph isomorphism." Journal of the ACM (JACM) 17, no. 1 (1970): 51-64.
4. Drago M Cvetkovic, Michael Doob, Horst Sachs, et al. Spectra of graphs, volume 10. Academic Press, New York, 1980.
5. Hopcroft, John E., and Robert Endre Tarjan. "A VlogV algorithm for isomorphism of triconnected planar graphs." Journal of Computer and System Sciences 7, no. 3 (1973): 323-331.
6. Hopcroft, John E., and Jin-Kue Wong. "Linear time algorithm for isomorphism of planar graphs (preliminary report)." In Proceedings of the sixth annual ACM symposium on Theory of computing, pp. 172-184. 1974.
7. A. K. Lenstra, H. W. Lenstra, and L. Lovsz. Factoring polynomials with rational coefficients. Mathematische Annalen, 261(4):515-534, Dec 1982. URL: https://doi.org/10.1007/ BF01457454, doi:10.1007/BF01457454.
8. Luks, Eugene M. "Isomorphism of graphs of bounded valence can be tested in polynomial time." Journal of computer and system sciences 25, no. 1 (1982): 42-65.
9. G. Rama Murthy. Novel Shannon graph entropy, capacity: Spectral graph theory: Polynomial time algorithm for graph isomorphism. Technical Report IIIT/TR/2015/61, 2015.
10. Gilbert Strang,Linear Algebra and its Applications, Thomson-Cole Publishers.
11. Luks, Eugene M. "Computing the composition factors of a permutation group in polynomial time." Combinatorica 7, no. 1 (1987): 87-99.
12. Luks, Eugene M. "Permutation groups and polynomial-time computation." DIMACS Series in Discrete Mathematics and Theoretical Computer Science 11 (1993): 139-175.
13. Luks, Eugene M. "Hypergraph isomorphism and structural equivalence of boolean functions." In Proceedings of the thirty-first annual ACM symposium on Theory of computing, pp. 652-658. 1999.
14. Yang, Zhirong, and Erkki Oja. "Quadratic nonnegative matrix factorization." Pattern Recognition 45, no. 4 (2012): 1500-1510.

15. Guo, Chun-Hua. ”A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation.” Linear algebra and its applications 357, no. 1-3 (2002): 299-302.

16. Bini, Dario A., Bruno Iannazzo, Beatrice Meini, and Federico Poloni. ”Nonsymmetric algebraic Riccati equations associated with an M-matrix: recent advances and algorithms.” In Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.