A Fuzzy Theory Based Topological Distance Measurement for Undirected Multigraphs

1st Jing He
Nanjing University of Finance and Economics
Swinburne University of Technology
Melbourne, Australia
lotusjing@gmail.com

2nd Jinjun Chen
Swinburne University of Technology
Melbourne, Austrialia
jinjun.chen@gmail.com

3rd Guangyan Huang
Deakin University
Melbourne, Austrialia
guangyan.huang@deakin.edu.au

4th Mengjiao Guo
Swinburne University of Technology
Melbourne, Austrialia
mengjiaoguo0702@gmail.com

5th Zhiwang Zhang
Institue of Information Technology, Nanjing University of Finance and Economics
zzwmis@163.com

6th Hui Zheng
Swinburne University of Technology
Melbourne, Australia
hui.zheng@aliyun.com

7th Yunyao Li
Nanjing University of Finance and Economics
Nanjing, China
liyunyao1010@163.com

8th Ruchuan Wang
School of Computer Science, Nanjing University of Posts and Telecommunications
Nanjing, China
wangrc@njupt.edu.cn

9th Weibei Fan
School of Computer Science, Nanjing University of Posts and Telecommunications
Nanjing, China
wbfan@njupt.edu.cn

10nd Chi-Huang Chi
Software and Computational Systems Program, Data 61, CSIRO
Hobart, TAS, Australia
chihung.chi@data61.csiro.au

11th Weiping Ding
School of Information Science and Technology, Nantong University
Nantong, China
dwp9988@163.com

12th Paulo A.de Souza
Griffith University
Nathan, QLD, Australia
paulo.desouza@griffith.edu.au

13th Bin Chen
Ningbo Institue of Materials Technology and Engineering, Chinese Academy of Sciences
Ningbo, China
chenbin@nimte.ac.cn

14th Runwei Li
Ningbo Institue of Materials Technology and Engineering, Chinese Academy of Sciences
Ningbo,
Chinarunweili@nimte.ac.cn

15th Jie Shang
Ningbo Institue of Materials Technology and Engineering, Chinese Academy of Sciences
Ningbo, China
shangjie@nimte.ac.cn

16th André Van Zundert
Department of Anaesthesia, Royal Brisbane & Women's Hospital & University of Queensland
Brisbane, QLD, Australia
vanzundertandre@gmail.com

***Abstract*—The topological distance is to measure the structural difference between two graphs in a metric space. Graphs are ubiquitous, and topological measurements over graphs arise in diverse areas, including, e.g. COVID-19 structural analysis, DNA/RNA alignment, discovering the Isomers, checking the code plagiarism. Unfortunately, popular distance scores used in these applications, that scale over large graphs, are not metrics, and the computation usually becomes NP-hard. While, fuzzy measurement is an uncertain representation to apply for a polynomial-time solution for undirected multigraph isomorphism. But the graph isomorphism problem is to determine two finite graphs that are isomorphic, which is not known with a polynomial-time solution. This paper solves the undirected multigraph isomorphism problem with an algorithmic approach as NP=P and proposes a polynomial-time solution to check if two undirected multigraphs are isomorphic or not. Based on the solution, we define a new fuzzy measurement based on graph isomorphism for topological distance/structural similarity between two graphs. Thus, this paper proposed a fuzzy measure of the topological distance between two undirected multigraphs. If two graphs are isomorphic, the topological distance is 0; if not, we will calculate the Euclidean distance among eight extracted features and provide the fuzzy distance. The fuzzy measurement executes more efficiently and accurately than the current methods.**

***Index Terms*—Fuzzy measurement, topological structure, graph isomorphism, undirected multigraph, polynomial-time solution, permutation theorem, equinumerosity theorem, multiple vertex/edge adjacency matrix**

I. INTRODUCTION

The topological data is one of the most useful information which a graph could carry. Graph similarity and the related problem of graph isomorphism have various applications in big data analytics, data mining, machine learning, pattern recognition and artificial intelligence [1-15]. Measuring the similarity in terms of structure is essential for graph matching,

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

graph searching, and graph mining. Topological distance/similarity distance is defined as follows: given two graphs, their distance/similarity can be recognized as a score quantifying their structural differences in a metric space. The range of topological distance is [0, 1], and it satisfies the triangle inequality. Unfortunately, a measurement of topological distance is often computationally costly. For example, the chemical and CKS [1] are NP-hard while they have important properties. The distance should be zero if and only if the graphs are isomorphic, and they are capturing global structural similarities between two graphs. However, finding an optimal permutation P is notoriously hard [1-15], graph isomorphism, which is equivalent to determine if there exists a permutation P s.t. $AP = PB$, is famously a problem that is neither known to be in P nor shown to be NP-hard [1-15] in the past. The most stringent form of exact graph matching--graph isomorphism, this condition must hold, that is, all the mapping must be a bijection in both directions. Graph isomorphism is challenging and critical in many applications, especially in various scientific areas. To date, there are no bibliography sources of polynomial-time graph isomorphism matching algorithms known for the general case except our recent contributions [16]. Previous isomorphism algorithms always suffer from the enormous computational complexity of analysis methods. In our recent works in [16] based on Permutation Theorem and Equinumerosity Theorem, undirected simple graph isomorphism has been approved to be P, which constructs a good fundamental for this paper.

A. *Undirected Multigraph*

A Multigraph, with the counterpart of a simple group, could be with multiple edges and several loops. For an undirected graph, if there are more than one undirected edge associated with a pair of vertices, these edges are called parallel edges, as the edges 4 and 9 of graph $\boldsymbol{G_1}$ in Fig. 1. If there is one edge of which has the starting node and the ending node are the same node, the edge is called loop, as the edges 1 of graph $\boldsymbol{G_1}$ in Fig. 1. Suppose there are four undirected multigraphs $\boldsymbol{G_1}, \boldsymbol{G_2}, \boldsymbol{G_3}, \boldsymbol{G_4}$ shown in Fig. 1. Multigraphs could have loops that allow an edge that connects a vertex to itself. Graphs with parallel edges and/or loops are called a multigraph in this paper.

B. *Graph Isomorphism*

Graph G_1 is isomorphic to graph G_2 (denoted by $G_1 \cong G_2$) if and only if there exists a bijection $f: V(G_1) \rightarrow V(G_2)$ such that for any two vertices $u, v \in V(G_2), (u, v) \in E(G_1)$, if and only if $(f(u), f(v)) \in E(G_2)$. As shown in Figure 1, the following two multigraphs G_1 and G_2 are isomorphic. One of the vertex correspondences is 1-2, 2-1, 3-6, 4-4, 5-3, 6-5.

C. *Multigraph Isomorphism*

Multigraph isomorphism has opened a wide area of extensive research due to its well-known NP-complete nature and nondeterministic polynomial-complete [17]. In exact graph matching, if there exists a bijective mapping among the vertices and edges on them; Thus, each pair of two isomorphic graphs share a common structure. A multigraph may also contain directed and undirected edges. Multigraphs are more generic than simple graphs. The simple graphs usually are not rich with multi-edge information, while multigraph permits multiple edges/relations between a pair of vertices. And many real-world datasets can be modeled as a network with a set of nodes interconnected with each other with multiple relations. So, the crucial difference is to capture the multi-edge information. This problem appears naturally in various contexts of DNA and molecule structure [18-20]. In this paper, we are addressing a more generic problem (i.e. multigraph with undirected edges and unlabeled vertices).

The contributions of this paper are as follows: 1) we find a solution for multigraph isomorphic problems in conditions of undirected edges and unlabeled vertices. 2) a fuzzy topological distance measurement is also proposed.

II. The Common Setting in This Paper

This section is devoted to reviewing some relevant concepts.

A. *Vertex and Edge Labeling Method for Multigraph*

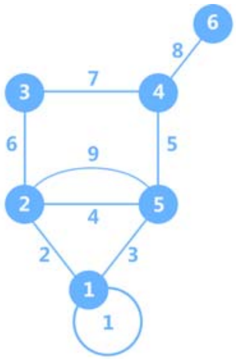
Two multigraphs with unique labels are generally taken values in positive integer range for subscript of vertices $v_1, v_2, \dots, v_n$ where n is the number of the graph vertices and natural number, used only to identify the vertices [16] uniquely, we call $v_1, v_2, \dots, v_n$ the vertex label. Similarly, two multigraphs with unique labels are generally taken values in positive integer range for subscript of edge, $e_1, e_2, \dots, e_n$ where n is the number of the edges of the graph and natural number, used only to identify the edges uniquely, as $e_1, e_2, \dots, e_n$ are the edge labels [16] shown in Figure 1.

B. *MultiGraph Isomorphism Problem*

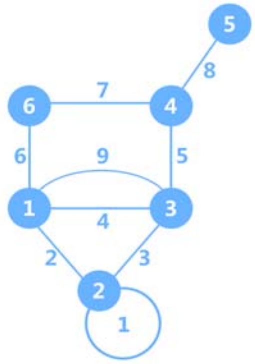
Definition 1 Multigraph Isomorphism based on edge structure: Two isomorphic multigraphs G_1 and G_2 is a bijective mapping f, which exists the vertices of G_1 to the vertices of G_2 correspondingly. That keeps the "edge structure" in the case that there is an edge between vertex u and vertex v in G_1 if and only if there is an edge between $f(u)$ and $f(v)$ in G_2 [3]. In this paper, we use a vertex and edge representation method [16] to label the multigraph. For example, in Figure 1, the array of the vertex for G_1 will be V_1=$\{v_1^1, v_2^1, v_3^1, v_4^1, v_5^1, v_6^1\}$, and the array of the vertex for G_2 will be V_2=$\{v_1^2, v_2^2, v_3^2, v_4^2, v_5^2, v_6^2\}$. The array of an edge for G_1 will be E_1=$\{e_1^1, e_2^1, e_3^1, e_4^1, e_5^1, e_6^1, e_7^1, e_8^1, e_9^1\}$ and the array of an edge for G_2 will be E_2=$\{e_1^2, e_2^2, e_3^2, e_4^2, e_5^2, e_6^2, e_7^2, e_8^2, e_9^2\}$.

C. *Virtual nodes for multigraph*

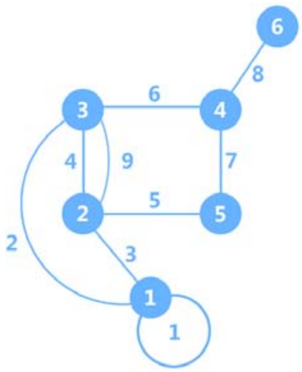
To handle the loop and parallel edges, especially in the multigraph, we build up a virtual node system to identify the different edges. For example, we assign four virtual nodes to node v_1 as v_{11}, v_{12}, v_{13}, v_{14}. Node v_1 has three edges including one loop, and therefore it will have four virtual nodes. Thus, in the vertex adjacency matrix, $a_{V_1}^{11,12} = 1$, $a_{V_1}^{13,21} = 1, a_{V_1}^{14,31} = 1$ shown in Figure 2. For the parallel edges between node 2 and node 5, we build two pairs of virtual nodes as v_{22} and v_{52} for edge 4, and v_{24} and v_{53} for edge 9 in Figure 3.



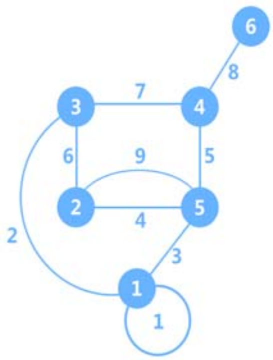
(a). G_1



(b). G_2



(c). G_3



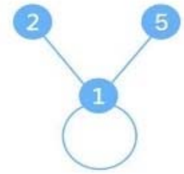
(d). G_4

Figure 1. Four undirected multigraphs G_1, G_2, G_3, G_4.

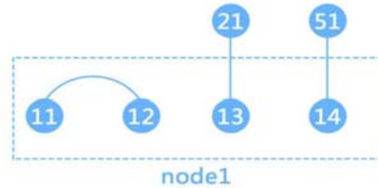
III. Multiple Vertex and Edge Adjacency Matrix Representation Method

A. *Vertex adjacency matrix representation method*

The multiple vertex adjacency matrix is a Boolean square matrix that represents a finite multigraph. Elements (valued 0 and 1) in the matrix denote whether pairs of vertices are connected with each other or not in the graph. For example, in a graph G_1, v_1 is adjacent with v_1, v_2, and v_3. The multiple node adjacency matrix is a Boolean square matrix, which represents a finite multigraph. The elements (valued 0, 1, 2, …, *n*, where *n* is a non-negative integer) in the matrix denote whether pairs of nodes are connected with each other or not in the multigraph. For example, in the graph G_1, v_1 are adjacent with v_{12}, v_2, v_5. Then, in the vertex adjacency matrix, $v_{12}^{11} = 1$, $v_{13}^{21} = 1$, $v_{14}^{51} = 1$. The virtual node representation for loop is shown in Fig. 2. And the virtual node representation for parallel edges is shown in Fig. 3. The vertex adjacency matrix for node 1 in G_1 is shown in Figure 4.

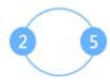


(a). Node 1 in G_1.

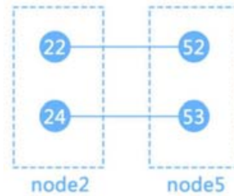


(b). Four virtual nodes in the vertex adjacency matrix.

Figure 2. Virtual node representation for loop.



(a). Parallel edges in G_1.



(b). Two pairs of virtual nodes in the vertex adjacency matrix.

Figure 3. Virtual node representation for parallel edges.

	v_{11}	v_{12}	v_{13}	v_{14}	v_{21}	v_{51}
v_{11}	0	1	0	0	0	0
v_{12}	1	0	0	0	0	0
v_{13}	0	0	0	0	1	0
v_{14}	0	0	0	0	0	1
v_{21}	0	0	1	0	0	0
v_{51}	0	0	0	1	0	0

Figure 4. Vertex adjacency matrix for Node 1 in G_1.

B. *Edge adjacency matrix representation method for multigraph*

The multiple edge adjacency matrix is a Boolean square matrix, which represents a finite multigraph. The elements (valued 0, 1, 2, …, *n,* where *n* is a non-negative integer) in the matrix denote whether pairs of edges are connected with each other or not in the multigraph. For example, in the graph G_2, e_2 are adjacent with e_1, e_3, e_4, e_6, e_9, Then, in the edge adjacency matrix, $b_{E_2}^{13} = 1$, $b_{E_2}^{14} = 1$, $b_{E_2}^{16} = 1$, $b_{E_2}^{19} = 1$. For any parallel edges shown in Figure 5 or loop shown in Figure 6, we need to count twice in the edge adjacency matrix. The edge adjacency matrix for G_1 is shown in Figure 7.

	e_1
e_1	2

Figure 5. Edge adjacency matrix for edge 1 as a loop in G_1

	e_4	e_9
e_4	0	2
e_9	2	0

Figure 6. Edge adjacency matrix for edge 4 and 9 as Parallel edges between Node 2 and Node 5 in G_1.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	Row sum for each edge	Squared sum of row sum of each node
e_1	2	2	2	0	0	0	0	0	0	6	36
e_2	2	0	1	1	0	1	0	0	1	6	36
e_3	2	1	0	1	1	0	0	0	1	6	36
e_4	0	1	1	0	1	1	0	0	2	6	36
e_5	0	0	1	1	0	0	1	1	1	5	25
e_6	0	1	0	1	0	0	1	0	1	4	16
e_7	0	0	0	0	1	1	0	1	0	3	9
e_8	0	0	0	0	1	0	1	0	0	2	4
e_9	0	1	1	2	1	1	0	0	0	6	36
				Sum						44	234

Figure 7. Multiple edge adjacency matrix for Graph G_1.

C. *Triple tuple method*

Triple Tuple for multigraph has been defined as in a multigraph *G* with the number of nodes is *N*, and the number of edges is E. We create a triple tuple for each edge e_E = (*k*, n_s, n_e), where $k = 1, 2, \ldots, E$. Note n_s as a starting node and n_e as an ending node are two nodes for edge e_k. For one node, in order to represent different edges, we create the different virtual nodes as n . The first represents the label of the node, the second represents the order of the virtual node. For example, the triple tuples are produced to represent the finite multigraph G_4 shown in Table 1. The general format of triple tuple for a multigraph is shown in Table 2.

Table 1. The triple tuple of graph G_4.

Edge	Node	Node
1	11	12
2	12	31
3	13	21
4	32	22
5	23	51
6	33	41
7	52	42
8	43	61
9	34	24

Table 2. Triple tuple in general format.

Edge	Node	Node
1	1_s	1_e
2	2_s	2_e
3	3_s	3_e
4	4_s	4_e
…	…	…
m-2	$(m-2)_s$	$(m-2)_e$
m-1	$(m-1)_s$	$(m-1)_e$
M	m_s	m_e

IV. A POLYNOMIAL-TIME UNDIRECTED MULTIGRAPH ISOMORPHISM ALGORITHM

In this section, we present the algorithm execution process and analyze the computational complexity of the proposed isomorphism algorithm. The graph isomorphism algorithm is based on the Permutation Theorem and Equinumerosity Theorem [16]. The spatial complexity of the worst case is *n*^6, where *n* is the number of vertices [16] which is shown in the following pseudocode:

(1) Generate four matrices of vertex adjacency matrix and edge adjacency matrix for two graphs. (The numbers of nodes and edges must be equivalent)

(2) Determine if the row sum of vertex adjacency matrix and edge adjacency matrix of two graphs is a permutation of another or not by the Permutation Theorem. If not, they are not isomorphic.

(3) Singular value decomposition of four matrices.

(4) Determine if the eigenvalues of four matrices are equinumerous or not by Equinumerosity Theorem. If not, they are not isomorphic.

(5) Compute the maximal linearly independent set of left-singular vectors and right-singular vectors according to the multiple eigenvalues for eight matrices and check if they are equinumerous, if not, they are not isomorphic. If yes, they are isomorphic.

We rewrite the definition 1 as follows: G_1 and G_2 are isomorphic if and only if V_1 is a bijective map of V_2, and E_1 is a bijective mapping of E_2. This paper uses our permutation theorem [16] and equinumerosity theorem to develop a polynomial-time algorithm for multigraph isomorphism. The core idea is that we use permutations to act on a structured object (graph) by rearranging their components (vertex and edge). The algorithm can check if two vertex sets based on multiple vertex adjacency matrix and two edge sets based on multiple edge adjacency matrix are respectively bijective. For both vertex and edge arrays of row/column sum based on multiple vertex and edge adjacency matrices, if one array is a permutation of another, the corresponding two multigraphs could be isomorphic. The following four graphs as G_1, G_2, G_3, G_4 are used as an undirected multigraph example in this paper. From the observation, we could see that G_1, and G_2 are isomorphic, while G_3, and G_4 are not isomorphic.

V. FUZZY THEORY BASED DISTANCE MEASUREMENT BETWEEN TWO MULTIGRAPHS

In the problem of graph isomorphism, there are two conditions of distance: 0 and 1, for every two multigraphs. This representation, however, fails to show a real distance between two multigraphs. The real distance could be able to display the grade of differences, whose range should be [0,1] rather than only 0 and 1. This section, therefore, proposes a fuzzy theory-based distance measurement that extends the distance of two multigraphs from 0, 1 to [0, 1]. Due to the multigraphs that could have infinite edges and nodes, to simplify our question, we only focus on the fine measurement between two multigraphs with the equivalent number of edges and nodes. The isomorphism checking for four graphs are shown in Appendix A. The two applications are shown in Appendix B. The more detailed steps of the proposed method are as follows:

(1) Generate the triple tuple sets for two multigraphs G_1 and G_2. If there is an isolated node, remove it. The number of isolated nodes must be the same. If not the same, produce that

for graphs G_1 and G_2, they are not isomorphic. After the removing of the isolated node, the number of nodes and the number of edges must be the same, if not same, produce that for multigraphs G_1 and G_2, they are not isomorphic.

(2) Generate the array of row sum of the multiple vertex adjacency matrix for G_1 and G_2 and produce two sets of an array as a_{V_1} , b_{V_2}. Check if $\sum_{k=1}^{n} a_{V_1}^{k} = \sum_{k=1}^{n} b_{V_2}^{k}$; if so, go to the next step; if not, products that they are not isomorphic.

(3) Check if $\sum_{k=1}^{n} (a_{V_1}^{k})^2 = \sum_{k=1}^{n} (b_{V_2}^{k})^2$, If so, go to the next step; if not, produce that graph G_1 and G_2 are not isomorphic.

(4) Continue to compute until *n* step and check if $\sum_{k=1}^{n} (a_{V_1}^{k})^n = \sum_{k=1}^{n} (b_{V_2}^{k})^n$. If not, produce the results that graph G_1 and G_2 are not isomorphic.

(5) Generate the array of row sum of the multiple edge adjacency matrix and produce two sets of an array as a_{E_1} , b_{E_2}. Compute and check if $\sum_{k=1}^{m} a_{E_1}^{k} = \sum_{k=1}^{m} b_{E_2}^{k}$; if so, go to the next step; if not, produces that graph G_1 and G_2 are not isomorphic.

(6) Compute and check if $\sum_{k=1}^{m} (a_{E_1}^{k})^2 = \sum_{k=1}^{m} (b_{E_2}^{k})^2$. If not, produce the results that multigraph G_1 and G_2 are not isomorphic.

(7) Continue to compute until *m* step and check if $\sum_{k=1}^{m} (a_{E_1}^{k})^m = \sum_{k=1}^{m} (b_{E_2}^{k})^m$. If not, produce the results that graph G_1 and G_2 are not isomorphic. Until the *m*-th step.

(8) Singular value decomposition of four matrices.

(9) Determine if the eigenvalue of four matrices are equinumerous by Equinumerosity Theorem. If not, they are not isomorphic.

(10) Compute the maximal linearly independent set of left-singular vectors and right-singular vectors according to the multiple eigenvalues for four matrices and check if they are equinumerous, if not, they are not isomorphic.

The distance based on Permutation Theorem [16] will count for 0.5 weight (steps 1 to 7). If two graphs are isomorphic according to the Equinumerosity Theorem [16] (steps 8-10), the 0.5 will time 0. Otherwise, the 0.5 will time the Euclidean distance based on Equinumerosity Theorem [20]. The mathematical proof is shown in Appendix C.1 and C.2. Therefore, the range of the topological distance should be [0, 1].

$$\text{Topological distance between two multigraphs} = \frac{\text{Euclidean distance between the original multigraph and another multigraph}}{\text{Euclidean distance two the original multigraph and the zero multigraph}}.$$

The value of the topological distance is the membership value for every pair of two multigraphs. Then, the formula above is the corresponding membership function. We, therefore, have the fuzzy sets for every pair of two multigraphs, as (multigraph G_1, multigraph G_2, membership value/ topological distance). The representation (multigraph G_1, multigraph G_2, membership value/ topological distance) implies how much two multigraphs are isomorphic. It supplies a strict definition of graph isomorphism. In real applications, the definition of graph isomorphism will be flexible and can vary with different conditions of fuzzy sets.

To be more flexible, the membership value can be changed according to the initial graph isomorphism results. Firstly, initial membership value is assigned as the topological distance between two multigraphs; then, we have an initial fuzzy set for every pair of two multigraphs. Secondly, the judgments that whether two multigraphs are isomorphic or not are kept as the initial graph isomorphism results. Thirdly, we transfer current membership value to new membership value, that is,

$$\text{Updated membership value as the follows} = \frac{\text{Euclidean distance between the original multigraph and another multigraph} - e}{\text{Euclidean distance two the original multigraph and the zero multigraph} - e},$$

where *e* is an adjustable factor. The adjustable factor e can be computed by the mean square deviation of the membership value and the value of isomorphic or not, which represents as 0 or 1, respectively. Finally, we can have satisfying membership values with user-defined maximal adjustable factors. The results of four graphs are shown in Appendix A. Furthermore, a real-time application is shown in Appendix B example 1 and a large-scale experiments related to COVID-19 are shown in Appendix B example 2.

VI. Conclusion

This paper proposes a fuzzy measurement of the topological distance between two multigraphs. A polynomial-time settlement to verify if two multigraphs are isomorphic is put forward as well. Three new representation methods of a multigraph as multiple vertex adjacency matrix, and multiple edge adjacency matrix are proposed. The theoretical significance of our algorithm is that it offers the complicated mathematical problem could be addressed in a reasonable polynomial time which has existed for years. Several practical purposes could be achieved. In addition, in computer science, a series of nodes strung together by connections known as edges is a network; in the real world, the set of social network users and their interconnections make up a graph are typically large, with millions of nodes and billions of edges. To recognize social communities or groups effectively and efficiently, which is normally onerous to query matches, so the proposed algorithm can relieve the complexity of computing. The algorithm can be applied for applications as quickly as searching chemical databases, performing fingerprint or facial recognition; is that the same molecule, fingerprint or face.

References

[1] Jose Bento, Stratis Ioannidis, "A family of tractable graph distances." https://arxiv.org/abs/1801.04301

[2] Cordella, Luigi P., et al. "Performance evaluation of the VF graph matching algorithm." Proceedings 10th International Conference on Image Analysis and Processing. IEEE, 1999.

[3] Cordella, Luigi Pietro, et al. "An improved algorithm for matching large graphs." 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition. 2001.

[4] Mendivelso J, Kim S, Elnikety S , et al. "Solving graph isomorphism using parameterized matching [M], String Processing and Information

Retrieval. " Springer International Publishing, 2013.
[5] Schmidt, Douglas C., and Larry E. Druffel. "A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices." Journal of the ACM (JACM) 23.3 (1976): 433-445.
[6] Bonnici, V., Giugno, R., Pulvirenti, A., Shasha, D., and Ferro, A. (2013). "A subgraph isomorphism algorithm and its application to biochemical data. " BMC Bioinformaticss, 14(Suppl 7) :S13.
[7] Ellingham M N, Wormald N C. "Isomorphic factorization of regular graphs and 3-regular multigraphs." Journal of the London Mathematical Society, 1988, 2(1): 14-24.
[8] Luo Shifeng, "Digraph's Isomorphism." Journal of Guangxi University, 1998-04.
[9] Frank O, Shafie T. "Complexity of families of multigraphs. " JSM Proceedings, Section on Statistical Graphics, Alexandria, VA: American Statistical Association, 2012: 2908-2921.
[10] Broersma H, Li X. "Isomorphisms and traversability of directed path graphs." Discussions Mathematica Graph Theory, 2002, 22(2): 215-228.
[11] Xie K, Wu W. "An optimization scheme for determining isomorphism of digraphs or multigraphs. " Journal of Mian yang Normal University, 2013 (5): 18.
[12] Fu J. Linear matching-time algorithm for the directed graph isomorphism problem. In: Staples J., Eades P., Katoh N., Moffat A. (eds) Algorithms and Computations. ISAAC 1995. Lecture Notes in Computer Science, vol 1004. Springer, Berlin, Heidelberg.
[13] Babai, László. "Graph isomorphism in quasipolynomial time [extended abstract]." ACM Sigact Symposium on Theory of Computing ACM, 2016:684-697.
[14] McKay, Brendan D. "Practical graph isomorphism." (1981): 45-87.
[15] McKay, B. D. and Piperno, A. (2014). Practical graph isomorphism, ii. Journal of Symbolic Computation, 60:94–112.
[16] J. He, J. Chen, G. Huang, J. Cao, Z. Zhang, H. Zheng, P. Zhang, R. Zarei, F. Sansoto, R. Wang, Y. Ji, W. Fan, Z. Xie, X. Wang, M. Guo, C. Chi, P. Souza, J. Zhang, Y. Li, X. Chen, Y. Shi, D. Green, T. Kersi, A. Zundert, Polynomial-time algorithm for simple undirected graph isomorphism, Concurrency and Computation Practice and Experience. (DOI: 10.1002/cpe.5484)
[17] Balpande, Vijaya, and Anjali Mahajan. "An approach of graph isomorphism detection based on vertex-invariant." International Journal of Advanced Studies in Computers, Science and Engineering 4.4 (2015): 1.
[18] Guangwu, Liu, et al. "Algorithm of graph isomorphism with three-dimensional DNA graph structures." Progress in Natural Science 15.2 (2005): 181-184.
[19] Jonoska, Natasa, Stephen A. Karl, and Masahico Saito. "Three-dimensional DNA structures in computing." BioSystems 52.1-3 (1999): 143-153.
[20] Jing, He et al. "A polynomial time solution for mixed graph Isomorpherism." submitted to IEEE transaction on intelligent computing, 2020.
[21] Feng Li, Tao Lu, "Graph Isomorphism Determination and Applications, Journal of Fudan University (Natural Science). " 45(4): 480-484, 2006.
[22] Vladimir, Balaz, Vladimir, Kvasnicka and Jiri Pospichal, "Two Metrics in a Graph Theory Modeling of Organic Chemistry. " Discrete Applied Mathematics, 35 (1992): 1-19.
[23] Zang Wei, Li Feng, "Isomorphism determination algorithm for arbitrary graphs: feature vector method, Journal of Computer-Aided Design and Graphics." 19-2, 2007(2)163-167.
[24] https://blog.csdn.net/Banach_I/article/details/51078451, access at 1/26/2020
[25] DNA Sequencing, https://en.wikipedia.org/wiki/DNA_sequencing accessed 4/7/2020
[26] COVID-19 DNA sequencing, https://platform.gisaid.org/ accessed 4/7/2020

APPENDIX A ARRAYS OF ROW SUM FOR FOUR GRAPHS' EDGE ADJACENCY MATRICES

G_1	Row sum for each edge	Squared sum of row sum of each node	G_2	Row sum for each edge	Squared sum of row sum of each node	G_3	Row sum for each edge	Squared sum of row sum of each node	G_4	Row sum for each edge	Squared sum of row sum of each node
	6	**36**		**6**	**36**		**6**	**36**		**6**	**36**
	6	**36**		**6**	**36**		**6**	**36**		**5**	**25**
	6	**36**		**6**	**36**		**6**	**36**		**6**	**36**
	6	**36**		**6**	**36**		**6**	**36**		**5**	**25**
	5	**25**		**5**	**25**		**5**	**25**		**5**	**25**
	4	**16**		**4**	**16**		**4**	**16**		**4**	**16**
	3	**9**		**3**	**9**		**3**	**9**		**4**	**16**
	2	**4**		**2**	**4**		**2**	**4**		**2**	**4**
	6	**36**		**6**	**36**		**6**	**36**		**5**	**25**
Sum	**44**	**234**	**Sum**	**44**	**234**	**Sum**	**44**	**234**	**Sum**	**42**	**208**

The two arrays of row sums of both vertex and edge adjacency matrix of G_1 and G_2 are permutated. The eigenvalue, the maximally independent system of left and right singular vector of the corresponding *P* multiple eigen value of G_1 and G_2 are equinumerous. The distance between G_1 and G_2 is zero, say, they are isomorphic. The two arrays of row sums of edge adjacency matrix of G_3 and G_4 are not permutable, G_3 and G_4 are not isomorphic. The distance between them is $\sqrt{\frac{(44-42)^2+(234-208)^2}{44^2+234^2}} = 0.012$.

APPENDIX B APPLICATIONS

Example 1: The proposed method can be easily extended to the directed/weighted multigraph in [20]. In organic chemistry, the problem of whether the molecular structure of the faction [21][22] is isomers as follows: From Figure 8 to 10, we have three kinds of molecular structure of the faction.

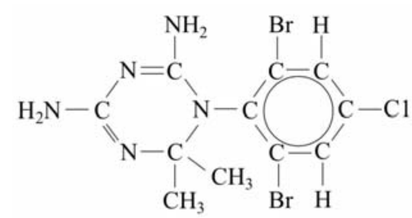


Figure 8. The first molecular structure of the organic matter 1 to be determined [21, 22].

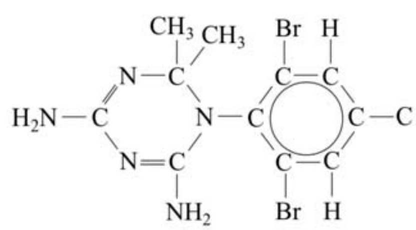


Figure 9. The second molecular structure of the organic matter 2 to be determined [21, 22].

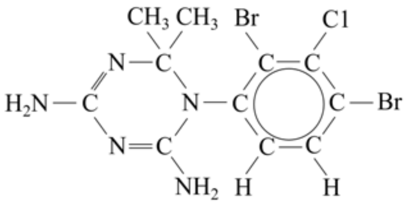


Figure 10. The third molecular structure of the organic matter 3 to be determined [21, 22].

From Figure 8 to 10, we change the molecular structure into the graph format. A single line indicates a covalent bond, a double line indicates two covalent bonds, and a circle in the benzene ring indicates an aromatic bond, which can be 1.5 valence. When comparing, the cesium atom is omitted, and the above organic matter can be established according to the following rules: (1) the covalent bond between C atoms is represented by a one-to-one correspondence of undirected edges, and the weight is 1 (the weight within the benzene ring is 1.5); (2) The covalent bond between a helium atom and the other atoms are represented by a directed edge, the direction is directed to other atoms, and the weight is the number of bits pointed to the atom in the periodic table, i.e., N-7, CL-17, BR-35.

The figure is a multigraph and mixed with weighted edges and directed edges with 19 vertices and 22 edges. Using the method described in this paper, it can be determined that the matter 1, 2 and 3 have the different structures, and mater 1 and 2 are isomorphic, and the three sets of tests take less than 0.001 seconds (CPU 2.21 GHz - Intel(R) Core(TM) i7-6650U, 16 GB of RAM), which shows that the proposed method is effective. In fact, the organic matter of Figures 8, 9 are 2, 6-dibromo-4-chloro-1, 3, 5 triazabenzene, and the organic matter of figure 10 is 4-dibromo-3-chloro-1, 3, 5 triazabenzene. The three molecular formula is $C_{11}H_{12}N_5ClBr_2$, but they have a different molecular structure. The topological distance between the given figures is 0.003383 ($\text{distance} = \frac{\sqrt{(1146-1142)^2}}{\sqrt{60^2+6240^2+180^2+1142^2}}$). The calculated distance opens a new page for organic chemistry in terms of chemical distance and reaction distance [23].

Example 2: Large data set experiments for DNA sequence with COVID-19

We have conducted the similarity measurement of RNA sequences for COVID-19 carried by 4,489 patients [25, 26]. The related distance is over 0.9.

Appendix C.1 Mathematical Proof for Permutation theorem [16].

To check if two arrays are a permutation of another one such as $array\ \#1 = \{2, 3,\ 3, 2, 2, 3, 3, 2\}$ and $array\ \#2 = \{2, 3,\ 2, 3, 2, 3, 2, 3\}$. We calculate $\Sigma\ (array\ \#1) =\ 2 + 3 + 3 + 2 + 2 + 3 + 3 + 2 = 20$ and $\Sigma\ (array\ \#2) =\ 2 + 3 + 2 + 3 + 2 + 3 + 2 + 3 = 20$, $\sum(array\ \#1)^2 =\ 2^2 + 3^2 + 3^2 + 2^2 + 2^2 + 3^2 + 3^2 + 2^2 = 50$ and $\sum(array\ \#2)^2 =\ 2^2 + 3^2 + 2^2 + 3^2 + 2^2 + 3^2 + 2^2 + 3^2 = 50$, $\sum(array\ \#1)^3 =\ 2^3 + 3^3 + 3^3 + 2^3 + 2^3 + 3^3 + 3^3 + 2^3 = 140$ and $\sum(array\ \#2)^3 =\ 2^3 + 3^3 + 2^3 + 3^3 + 2^3 + 3^3 + 2^3 + 3^3 = 140$, $\sum(array\ \#1)^4 =\ 2^4 + 3^4 + 3^4 + 2^4 + 2^4 + 3^4 + 3^4 + 2^4 = 388$ and $\sum(array\ \#2)^4 = 2^4 + 3^4 + 2^4 + 3^4 + 2^4 + 3^4 + 2^4 + 3^4 = 388$, $\sum(array\ \#1)^5 =\ 2^5 + 3^5 + 3^5 + 2^5 + 2^5 + 3^5 + 3^5 + 2^5 = 1100$ and $\sum(array\ \#2)^5 =\ 2^5 + 3^5 + 2^5 + 3^5 + 2^5 + 3^5 + 2^5 + 3^5 = 1100$, $\sum(array\ \#1)^6 =\ 2^6 + 3^6 + 3^6 + 2^6 + 2^6 + 3^6 + 3^6 + 2^6 = 3172$ and $\sum(array\ \#2)^6 = 2^6 + 3^6 + 2^6 + 3^6 + 2^6 + 3^6 + 2^6 + 3^6 = 3172$, $\sum(array\ \#1)^7 =\ 2^7 + 3^7 + 3^7 + 2^7 + 2^7 + 3^7 + 3^7 + 2^7 = 9260$ and $\sum(array\ \#2)^7 =\ 2^7 + 3^7 + 2^7 + 3^7 + 2^7 + 3^7 + 2^7 + 3^7 = 9260$, $\sum(array\ \#1)^8 =\ 2^8 + 3^8 + 3^8 + 2^8 + 2^8 + 3^8 + 3^8 + 2^8 = 27268$ and $\sum(array\ \#2)^8 = 2^8 + 3^8 + 2^8 + 3^8 + 2^8 + 3^8 + 2^8 + 3^8 = 27268$. Then we check if $\sum array\ \#1 =\ \sum array\ \#2$, if $\sum(array\ \#1)^2 = \sum(array\ \#2)^2$, if $\sum(array\ \#1)^3 = \sum(array\ \#2)^3$, if $\sum(array\ \#1)^4 = \sum(array\ \#2)^4$, if $\sum(array\ \#1)^5 = \sum(array\ \#2)^5$, if $\sum(array\ \#1)^6 = \sum(array\ \#2)^6$, if $\sum(array\ \#1)^7 = \sum(array\ \#2)^7$, if $\sum(array\ \#1)^8 = \sum(array\ \#2)^8$, if and only if they are equal respectively, we could draw a conclusion that array *#1* is a permutation of array #2.

Permutation Theorem. Given two natural number sets of arrays A and B, $A = \{\ a_1, a_2, \dots, a_n\ \}$, $B = \{\ b_1, b_2, \dots, b_n\ \}$, If and only if $\sum_{k=1}^{n} a_n = \sum_{k=1}^{n} b_n$, $\sum_{k=1}^{n} a_n^2 = \sum_{k=1}^{n} b_n^2$, $\sum_{k=1}^{n} a_n^3 = \sum_{k=1}^{n} b_n^3$, …, $\sum_{k=1}^{n} a_n^{n-2} = \sum_{k=1}^{n} b_n^{n-2}$, $\sum_{k=1}^{n} a_n^{n-1} = \sum_{k=1}^{n} b_n^{n-1}$, $\sum_{k=1}^{n} a_n^n = \sum_{k=1}^{n} b_n^n$, A is a permutation of B and vice versa, where $n \geq 1$.

Assertion 4-1: Given two arrays A and B, $A = a_1, a_2, \dots \mathrm{a_n}$, $B = b_1, b_2, \dots\ b_n$, If and only if $\sum_{k=1}^{n} a_n = \sum_{k=1}^{n} b_n$, and $\sum_{k=1}^{n} a_n^2 = \sum_{k=1}^{n} b_n^2$, $\sum_{k=1}^{n} a_n^3 = \sum_{k=1}^{n} b_n^3$, …, $\sum_{k=1}^{n} a_n^{n-2} = \sum_{k=1}^{n} b_n^{n-2}$, $\sum_{k=1}^{n} a_n^{n-1} = \sum_{k=1}^{n} b_n^{n-1}$, $\sum_{k=1}^{n} a_n^n = \sum_{k=1}^{n} b_n^n$, the sequence of the two arrays are bijective and equivalent. (where $n \geq 1$ and is the integer and both a_n and $b_n \geq 1$ and are integers)."

Mathematical proof for Theorem 4. The "only if" of the theorem (necessary condition) is simple because permutation array group is bijective. They always have the n equivalent arrays as $\sum_{k=1}^{n} a_n = \sum_{k=1}^{n} b_n$, and $\sum_{k=1}^{n} a_n^2 = \sum_{k=1}^{n} b_n^2$, $\sum_{k=1}^{n} a_n^3 = \sum_{k=1}^{n} b_n^3$, …, $\sum_{k=1}^{n} a_n^{n-2} = \sum_{k=1}^{n} b_n^{n-2}$, $\sum_{k=1}^{n} a_n^{n-1} = \sum_{k=1}^{n} b_n^{n-1}$, $\sum_{k=1}^{n} a_n^n = \sum_{k=1}^{n} b_n^n$, if two arrays are bijective. The "if" (sufficient condition) requires the following three main lemmas from fundamental theorem of arithmetic for $n = 2$. That is, Given two natural number sets of arrays A and B, $A = \{\ a_1, a_2\}$, $B = \{\ b_1, b_2\ \}$, If and only if $a_1 + a_2 = b_1 + b_2$, and $a_1^2 + a_2^2 = b_1^2 + b_2^2$, A is a permutation of B and vice versa. ($n = 2$ case)

If $a_1 + a_2 = b_1 + b_2$ and $a_1^2 + a_2^2 = b_1^2 + b_2^2$. Then $(a_1 + a_2)^2 = (b_1 + b_2)^2$, $a_1^2 + a_2^2 + 2a_1a_2 = b_1^2 + b_2^2 + 2b_1b_2$, then we have $a_1a_2 = b_1b_2$.

If there is any a_1, a_2, b_1, b_2 equals to 1, n=2 case holds. The proof is as follows. Suppose $a_1 = 1$, we have $1 + a_2 = b_1 + b_2$ and $1 + a_2^2 = b_1^2 + b_2^2$. Thus $a_2 = b_1b_2$, $1 + b_1b_2 = b_1 + b_2$, $1 + b_1(b_2 - 1) = b_2$, $(b_1 - 1)(b_2 - 1) = 0$. Therefore, either $b_1 = 1$ or $b_2 = 1$. When $b_1 = 1$, we have $1 + a_2 = 1 + b_2$ and $1 + a_2^2 = 1 + b_2^2$, $a_2 = b_2$. When

$b_2 = 1$, we have $1 + a_2 = 1 + b_1$ and $1 + a_2^2 = 1 + b_1^2$, $a_2 = b_1$. Therefore, $n = 2$ case holds.

If $every\ a_1, a_2, b_1, b_2$ is a positive integer and larger than 1, according to the fundamental theorem of arithmetic, a_1, a_2, b_1, b_2 either is a prime number itself or can be represented as the product of prime numbers; moreover, this representation is unique, up to (except for) the order of the factors. Then
$K = p_1p_2p_3p_4 \cdots p_lp_{l+1}p_{l+2} \cdots p_r =$
$q_1q_2q_3q_4 \cdots q_{l'}q_{l'+1}q_{l'+2} \cdots q_s$, where $a_1 =$
$p_1p_2p_3p_4 \cdots p_l$, $a_2 = p_{l+1}p_{l+2} \cdots p_r$, $b_1 =$
$q_1q_2q_3q_4 \cdots q_{l'}$, $b_2 = q_{l'+1}q_{l'+2} \cdots q_s$,
where p_i and q_i are prime. Assume that m ($m \neq 0$) and n are integers. We say that m could be divided by n if n is a multiple of m, namely, if there exists an integer parameter k such that $n = mk$. If m divides n, it can by represented as $m \mid n$. The order of the factors will not affect the results. We have $p_1 \mid K$, then $p_1 \mid q_1q_2q_3q_4 \cdots q_lq_{l+1}q_{l+2} \cdots q_s$. p_1 divides at least one of q_i; then if we rearrange q_i, we could have $p_1 \mid q_1$. Because q_1 is prime, factors are 1 or q_1; then we have $p_1 = q_1$. Now remove it from both sides of the equation.

$p_2p_3p_4 \cdots p_lp_{l+1}p_{l+2} \cdots p_r =$
$q_2q_3q_4 \cdots q_{l'}q_{l'+1}q_{l'+2} \cdots q_s$.

Repeat the previous proof. p_2 divides at least one of q_i; then if we rearrange q_i, we have $p_2 \mid q_2$. Because q_2 is prime, factors are 1 or q_2; then we have $p_2 = q_2$. Then we remove it from both sides of the equation.

$p_3p_4 \cdots p_lp_{l+1}p_{l+2} \cdots p_r =$
$q_3q_4 \cdots q_{l'}q_{l'+1}q_{l'+2} \cdots q_s$.

Continue this process until all of p_i and q_i are removed. If all of p_i are removed, the left side of the equality is 1, so there is no left q_i. Similarly, if all of q_i are removed, the right side of the equality is 1. The number of p_i is equal to q_i. Then we have proved,

$K =$
$p_1p_2p_3p_4 \cdots p_lp_{l+1}p_{l+2} \cdots p_r =$
$q_1q_2q_3q_4 \cdots q_{l'}q_{l'+1}q_{l'+2} \cdots q_s$, all of p_i and q_i are prime, $r = s,\ l = l'$, rearrange q_i, we have
$p_1 = q_1, p_2 = q_2, p_3 = q_3, \cdots\ p_l = q_{l'} \cdots, p_r = q_s$,
thus $a_1 = a_2$ and $b_1 = b_2$, because a_1 and a_2 are commutative, and b_1 and b_2 are commutative. There could be $a_1 = b_2$ and $a_2 = b_1$, Then the set of a_1 and a_2 is a permutation of the set of $b_1\ and\ b_2$.

Next, we will prove the uniqueness of this condition. That is, there exists a quaternary and quadratic system of equations (1) as

$$\begin{cases} a_1 + a_2 = b_1 + b_2 \\ a_1^2 + a_2^2 = b_1^2 + b_2^2 \end{cases} \quad (1)$$

is a system of two equations involving the four variables a_1, a_2, b_1, b_2, where all variables are natural numbers. A solution to this system of integer equations is an assignment of values to the variables such that all the equations are simultaneously satisfied. Two solutions to the system above are given by:

$$\text{solution set A: } \begin{cases} a_1 = b_1 \\ a_2 = b_2 \end{cases} \quad (2)$$

$$\text{solution set B: } \begin{cases} a_1 = b_1 \\ a_2 = b_2 \end{cases}$$

$A \cup B$ is either solution A or solution B, which is a solution of (1). since it makes all two equations valid. The word "system" indicates that the equations are to be considered collectively, rather than individually87. Because $a_1^2 + a_2^2 = b_1^2 + b_2^2$, $a_1^2 - b_1^2 = b_2^2 - a_2^2$, then we have $(a_1 - b_1)(a_1 + b_1) = (a_2 - b_2)(a_2 + b_2)$. Because $a_1 + a_2 = b_1 + b_2$, then $a_1 - b_1 = a_2 - b_2$.

If (2) is not unique, there must exist another four a_1', a_2', b_1', b_2', holds (1), but belong to $\overline{A \cup B} = \bar{A} \cap \bar{B}$. Suppose k is a integer ($k \neq 0$), because $(a_1 + k) + (a_2 - k) = (b_1 + k) + (b_2 - k)$, then we could construct $a_1' = a_1 + k$, $a_2' = a_2 - k$, $b_1' = b_1 + k$, $b_2' = b_2 - k$, where $a_1' \neq a_1$, $a_2' \neq a_2$, $b_1' \neq b_1$, $b_2' \neq b_2$. $a_1' + a_2' = b_1' + b_2'$ holds. Because $(a_1')^2 + (a_2')^2 = (a_1 + k)^2 + (a_1 - k)^2 = a_1^2 + a_2^2 + 2k^2 + 2a_1k - 2a_2k$ and $(b_1')^2 + (b_2')^2 = (b_1 + k)^2 + (b_1 - k)^2 = b_1^2 + b_2^2 + 2k^2 + 2b_1k - 2b_2k$. To make $(a_1')^2 + (a_2')^2 = (b_1')^2 + (b_2')^2$, we must have $a_1^2 + a_2^2 + 2k^2 + 2a_1k - 2a_2k = b_1^2 + b_2^2 + 2k^2 + 2b_1k - 2b_2k$, and then $a_1 - a_2 = b_1 - b_2$. Because $a_1 - b_1 = a_2 - b_2$, then we have $a_2 = b_2$, and then $a_2 - k = b_2 - k$, $a_2' = b_2'$. a_2' could be equal to b_2', and this result is in a contraction since $\bar{A} \cap \bar{B}$. Therefore, the initial assumption as (2) is not unique must be false. Thus, $n = 2$ case is approved.

Therefore, Theorem 4 where $n = 2$ has been proved. Our mathematically proof with $n \geq 3$ is proved by the following induction-based method.

Questions for $n = 3$, $n = 4$, ..., and $n = N$ for extended permutation theorem.

When $n = 3$, if and only if $a_1 + a_2 + a_3 = b_1 + b_2 + b_3$, ${a_1}^2 + {a_2}^2 + {a_3}^2 = {b_1}^2 + {b_2}^2 + {b_3}^2$, and ${a_1}^3 + {a_2}^3 + {a_3}^3 = {b_1}^3 + {b_2}^3 + {b_3}^3$, the set of a_1, a_2, a_3 is a permutation of the set of b_1, b_2, b_3.

When $n = 4$, if and only if $a_1 + a_2 + a_3 + a_4 = b_1 + b_2 + b_3 + b_4$ and ${a_1}^2 + {a_2}^2 + {a_3}^2 + {a_4}^2 = {b_1}^2 + {b_2}^2 + {b_3}^2 + {b_4}^2$, ${a_1}^3 + {a_2}^3 + {a_3}^3 + {a_4}^3 = {b_1}^3 + {b_2}^3 + {b_3}^3 + {b_4}^3$, and ${a_1}^4 + {a_2}^4 + {a_3}^4 + {a_4}^4 = {b_1}^4 + {b_2}^4 + {b_3}^4 + {b_4}^4$, the set of a_1, a_2, a_3, a_4 is a permutation of the set of b_1, b_2, b_3, b_4.

$n = N$ is established, if and only if $a_1 + a_2 + a_3 + \cdots + a_N = b_1 + b_2 + b_3 + \cdots + b_N$ and ${a_1}^2 + {a_2}^2 + {a_3}^2 + \cdots + {a_{N+1}}^2 = {b_1}^2 + {b_2}^2 + {b_3}^2 + \cdots + {b_N}^2$, ..., ${a_1}^N + {a_2}^N + {a_3}^N + \cdots + {a_N}^N = {b_1}^N + {b_2}^N + {b_3}^N + \cdots + {b_N}^N$ the set of $a_1, a_2, a_3, \cdots a_N$ is a permutation of the set of $b_1, b_2, b_3, \cdots, b_N$.

Mathematical Proof: Let $P(N - 1)$ be the statement of permutation theorem, we give a proof by induction on N.

Base case. The statement holds for $n = 1$ and $n = 2$. $P(1)$ is easily seen to be true, and $P(2)$ is true by the above-mentioned proof when $n = 2$.

Inductive Step. The following steps will show that for any $N - 1 \geq 0$ that if $P(N - 1)$ holds, then also $P(N)$ holds. This can be done as follows. Assume the induction hypothesis that $P(N - 1)$ is true (for some unspecified value of $N - 1 \geq 2$), that is, if and only if $a_1 + a_2 + a_3 + \cdots +$

$a_{N-1} = b_1 + b_2 + b_3 + \cdots + b_{N-1}$ *and* $a_1{}^2 + a_2{}^2 + a_3{}^2 + \cdots + a_{N-1}{}^2 = b_1{}^2 + b_2{}^2 + b_3{}^2 + \cdots + b_{N-1}{}^2$, ..., $a_1{}^{N-1} + a_2{}^{N-1} + a_3{}^{N-1} + \cdots + a_{N-1}{}^{N-1} = b_1{}^{N-1} + b_2{}^{N-1} + b_3{}^{N-1} + \cdots + b_{N-1}{}^{N-1}$, the set of $a_1, a_2, a_3, \cdots a_{N-1}$ is a permutation of the set of $b_1, b_2, b_3, \cdots, b_{N-1}$.

Using the induction hypothesis, the permutation theorem as $P(N-1)$ can be written to: if and only if $a_1 + a_2 + a_3 + \cdots + a_{N-1} = b_1 + b_2 + b_3 + \cdots + b_{N-1}$ and $a_1{}^2 + a_2{}^2 + a_3{}^2 + \cdots + a_{N-1}{}^2 = b_1{}^2 + b_2{}^2 + b_3{}^2 + \cdots + b_{N-1}{}^2$, ..., $a_1{}^{N-1} + a_2{}^{N-1} + a_3{}^{N-1} + \cdots + a_{N-1}{}^{N-1} = b_1{}^{N-1} + b_2{}^{N-1} + b_3{}^{N-1} + \cdots + b_{N-1}{}^{N-1}$, there must exist a permutation matrix P_π, where

$[a_1 \quad a_2 \quad \cdots \quad a_{N-2} \quad a_{N-1}]\, P_\pi =$
$[b_1 \quad b_2 \quad \ldots \quad b_{N-2} \quad b_{N-1}]$. For
$[a_1 \quad a_2 \quad \cdots \quad a_{N-2} \quad a_{N-1} \; a_N] \begin{bmatrix} P_\pi & 0 \\ 0 & 1 \end{bmatrix} =$
$[[a_1 \quad a_2 \quad \cdots \quad a_{N-2} \quad a_{N-1}]\, P_{\pi_1} \quad a_N \;] =$
$[[b_1 \quad b_2 \quad \ldots \quad b_{N-2} \quad b_{N-1}] \quad a_N \;]$, where $\begin{bmatrix} P_\pi & 0 \\ 0 & 1 \end{bmatrix}$ is a square binary matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere. Because $a_1 + a_2 + a_3 + \cdots + a_{N-1} = b_1 + b_2 + b_3 + \cdots + b_{N-1}$ and $a_1 + a_2 + a_3 + \cdots + a_N = b_1 + b_2 + b_3 + \cdots + b_N$, then we have $a_N = b_N$, thus,
$[a_1 \quad a_2 \quad \cdots \quad a_{N-2} \quad a_{N-1} \; a_N] \begin{bmatrix} P_\pi & 0 \\ 0 & 1 \end{bmatrix} =$
$[b_1 \quad b_2 \quad \ldots \quad b_{N-2} \quad b_{N-1} \quad b_N]$. Then, the set of $a_1, a_2, a_3, \cdots a_N$ is a permutation of the set of $b_1, b_2, b_3, \cdots, b_N$. Therefore, *P(N)* is true, that is, if and only if $a_1 + a_2 + a_3 + \cdots + a_N = b_1 + b_2 + b_3 + \cdots + b_N$ and $a_1{}^2 + a_2{}^2 + a_3{}^2 + \cdots + a_N{}^2 = b_1{}^2 + b_2{}^2 + b_3{}^2 + \cdots + b_N{}^2$, ..., $a_1{}^N + a_2{}^N + a_3{}^N + \cdots + a_N{}^N = b_1{}^N + b_2{}^N + b_3{}^N + \cdots + b_N{}^N$, the set of $a_1, a_2, a_3, \cdots a_N$ is a permutation of the set of $b_1, b_2, b_3, \cdots, b_N$. Thereby it shows that indeed *P(N)* holds. Since both the base case and the inductive step have been performed by mathematical induction, the statement *P(N)* holds for all natural number *N*.

Appendix C.2 Mathematical Proof for Equinumerosity theorem.

1. SINGULAR VALUE DECOMPOSITION OF VERTEX ADJACENCY MATRIX AND EDGE ADJACENCY MATRIX

To check if two arrays are equinumerous, we need to know how many elements are equal in one array. For example, Array 1#= [1, 1, 2, 2, 2, 3, 3, 3, 3] and Array 2#= [4, 4, 5, 5, 5, 6, 6, 6, 6]. There are two elements as [1, 1], [4, 4], three elements [2, 2, 2], [5, 5, 5], and four elements [3, 3, 3, 3], [4, 4, 4, 4] are equal. The corresponding format in term of equinumerosity is then rewritten as: Array 1#= [21, 22, 31, 32, 33, 41, 42, 43, 44] and Array 2#= [21, 22, 31, 32, 33, 41, 42, 43, 44]. They are equinumerous.

Let A be a real symmetric matrix of *n*n*, so there is a singular value decomposition such that $A = U\Sigma V^*$, U is an $n * n$ unitary matrix, Σ is an $n * n$ real diagonal matrix, $V^*$is a conjugate transpose of V, and is also an $n * n$ unitary matrix. The element $\Sigma_{i,i}$ on the diagonal of Σ is the singular value of M. For singular values ranging from large to small, Σ can be uniquely determined by A, of course, U and V cannot be determined. A non-negative real number σ is a singular value of A only if there are k^m unit vectors u and k^n unit vectors v as follows: $Av = \sigma u$ and $A^* u = \sigma v$, where v and u are the left and right singular vectors of σ, respectively. For the above singular value decomposition: $A = U\Sigma V^*$. The elements on the diagonal of the matrix Σ are equal to the singular values of A. The columns of U and V are the left and right singular vectors, respectively. Therefore, the above definition of SVD states:

- A set of orthogonal bases U consisting of the left singular vectors of A can always be found in k^m.
- A set of orthogonal bases V consisting of the right singular vector of A can always be found at k^n.

Without loss of generality, the columns of U and the rows of V^* are defined and used in this paper, which are called left and right singular vectors in this paper.

Definition 2 *P*-multiple eigenvalue. An *n-th* order matrix has n eigenvalues. If there are P eigenvalues are the same, then these P eigenvalues are called P multiple eigenvalues.

Definition 3 A maximally independent vector set. It is defined as: Let S be a vector group, if it satisfies: (1) a_1, a_2, ... ,a_r is linearly independent; (2) if any other vector in the space can be expressed as a linear combination of elements of a maximal set--the basis a_1, a_2, ... , a_r. Then A set of vectors is maximally linearly independent if including any other vector in the vector space would make it linearly dependent.

Definition 4 A maximally independent vector system. Under the linear transformation, the maximally linear independent subset has been transferred to have only one 1 and others is 0. The current format of the original vector set is called the maximally independent vector system.

Property 5 [24] Let A be a real symmetric matrix. There exist an orthogonal symmetric matrix U and a diagonal matrix W such that $U'AU = W$, where the diagonal element of W is the eigenvalue of A and the column vector of U is the eigenvector of A.

Proof: By mathematical induction, it is obviously true for first-dimension square matrices. Suppose the above proposition be true for a square matrix with the dimension of $n-1$. Then for the n dimension square matrix A_n, it obviously has at least one eigenvalue λ, and the eigenvector corresponding to λ is x, and x is extended to a set of the orthogonal basis of R_n, and arranged into a matrix $U_1 = [x\, y_1 \ldots y_{n-1}]$. Then there is $U_1^T A U_1 = \begin{vmatrix} \lambda & 0 \\ 0 & A_{n-1} \end{vmatrix}$ (here $x'Ay = y_k' Ax = 0, x'Ax = \lambda$) . Whereas A_{n-1} is a real symmetric matrix of dimension $n-1$, the orthogonal matrix U_2 is assumed such that $U_2^T A_{n-1} U_2 = \begin{vmatrix} \lambda_2 & & 0 \\ & \ldots & \\ 0 & & \lambda_n \end{vmatrix}$, then $\begin{vmatrix} 1 & 0 \\ 0 & U_2 \end{vmatrix}^T \begin{vmatrix} \lambda & 0 \\ 0 & A_{n-1} \end{vmatrix} \begin{vmatrix} 1 & 0 \\ 0 & U_2 \end{vmatrix} = \begin{vmatrix} \lambda_1 & & 0 \\ & \ldots & \\ 0 & & \lambda_n \end{vmatrix}$. Let U be an orthogonal symmetric matrix, $U = U_1 \begin{vmatrix} 1 & 0 \\ 0 & U_2 \end{vmatrix}$. Since the eigenvalues of $U_1^T A U_1$ and A are the same, the eigenvalues λ_m of A_{n-1} are also the eigenvalues of A. Finally, it is only necessary to prove that the column vector of U is the eigenvector of A. Set $U = [x_1\ x_2\ ...\ x_n]$ and substitute $AU = \begin{vmatrix} \lambda_1 & & 0 \\ & ... & \\ 0 & & \lambda_n \end{vmatrix} U$, then $[Ax_1\ Ax_2\ ...\ Ax_n] = [\lambda_1 x_1\ \ \lambda_2 x_2\ ...\ \lambda_n x_n]$, compared with each other, X_m is the eigenvector of A.

2. EQUINUMEROSITY THEOREM

Property 6 [23][24] There must be P corresponding linearly independent eigenvectors for the P eigenvalues λ of the vertex adjacency matrix and edge adjacency matrix. (Eigenvalue of the multiplicity of P a real symmetric matrix has exactly P linearly independent eigenvector)

Proof: **Lemma 1**: For the general matrices, there are at most k linearly independent eigenvectors corresponding to k eigenvalues. For a certain eigenvalue λ, there are m linearly independent eigenvectors. The following proves that $m \leq k$. For them, using Schmidt's method to obtain m orthogonal eigenvectors $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m$, $|x| = \begin{cases} A\varepsilon_1 = \lambda\varepsilon_1 \\ \quad ... \\ A\varepsilon_m = \lambda\varepsilon_m \end{cases}$. Extend it to get a set of orthogonal basis $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m\ ,\ \theta_{m+1}, ..., \theta_n$, and $p = [\varepsilon_1, \varepsilon_2, ..., \varepsilon_m\ ,\ \theta_{m+1}, ..., \theta_n]$. Then $P'AP = \begin{vmatrix} \lambda I_m & 0 \\ 0 & A' \end{vmatrix}$ (It can be seen from the form of this matrix that the eigenvalue λ is at least m dimensions). Since $P'AP$ and A have the same eigenvalue, the dimension of the eigenvalue λ of A is $k \geq m$. Thus Lemma 1 is proved. According to the algebra theorem, the total number of algebraic multiples of the nth-dimensional equation roots is n, so the sum of the number of all linearly independent eigenvectors to each eigenvalue $\leq n$, and property 5 proves that there is n independent eigenvector for the real symmetric matrices. That is, in the above inequality, the equal sign holds. The condition that the equal sign is true if there are exactly k linearly independent eigenvectors corresponding to k-multiple eigenvalues.

Combining definitions 2, 3, 4 and property 5, 6, it is proved that the rank of the maximally linearly independent subset a_{n*p}/a_{p*n} of left/right singular vector corresponding to the P-multiple eigenvalue is P.

All of two graphs satisfied the permutation theorem as above must have $A_{n*n}E_{n*1}$=$A'_{n*n}E_{n*1}$, and $Ax = \lambda x$, $A'y = \lambda' y$, then $Ax = \lambda x$, $A'y = \lambda' y$, $y^T A' = y^T \lambda'$, thus $y^T \lambda x = y^T \lambda' x$.

Property 7 Two graphs are isomorphic, if the eigenvalue sequences of the vertex adjacency matrix and edge adjacency matrix are equinumerous.

Lemma 2 Two sequences satisfying the permutation theorem must be equinumerous. Nevertheless, the two equinumerous sequences do not necessarily have the permutation relations.

Property 8 [23][24] The two graphs satisfying the permutation theorem must have the equinumerous eigenvalues.

Equinumerosity Theorem. The two graphs are isomorphic, if and only if the eigenvalues of the two graphs' vertex and edge adjacency matrices are equinumerous, and the maximally linearly independent vector systems of the left and right singular vectors corresponding to the P-multiple eigenvector are equinumerous.

To approve Equinumerosity theorem, the following Lemma is put forward. **Lemma 3** [23][24] a_{p*n} and a'_{p*n} are the right singular vector set for p multiple eigenvalues for graph A and A', respectively. For the elementary row interchange operation matrix exchange matrix P, if there exist M_{p*p}, and $Pa = a'M$, then M is invertible.

Proof: *Rank* $(P^T a')$=*Rank*(a)=*Rank*(a')=p ≥*Rank*(M), according to the matrix theory, p= *Rank*(a)=*Rank*$(P^T a' M) \leq$ *min*(*Rank*$(P^T a')$, *Rank*(M)), so *Rank*$(M) \geq p$, because of $p \geq$ *Rank*(M), then *rank*(M)= p, therefore M is invertible.

Lemma 4 [23][24] There exist the row interchange matrix P, and a_{n*p} and a_{n*p}' satisfies $a = Pa'$ if and only if they are equinumerous. Similarly, there exists the column interchange matrix P, and a_{p*n} and a_{p*n}' satisfies $Pa = a'$ if and only if they are equinumerous.

Proof: The necessary proof is obvious. For sufficient proof. From definition 8, both the vector set a and a' are corresponding to the equinumerous sequence η, then $\eta = \Pi_{i=1}^{p} Q_l a = Qa, \eta = \Pi_{i=1}^{p} Q_l' a' = Q'a'$, so $Qa = Q'a'. Q_l$, Q_l' $(1 \leq l \leq p)$, both Q, and Q' are row interchange operation matrix, then $a = Q^T Q' a'$, then $P = Q^T Q'$, it is still the elementary row interchange matrix. Then the sufficient proof is completed.

Lemma 5 Equinumerosity theorem for graph isomorphism.

If A and A' have the equinumerous eigenvalue, and there exist the elementary row interchange matrix P_{n*n} which satisfies: for every eigenvalue λ (multiplicity of p) in A and A', the corresponding right singular vector set $a_{n*P}^{(\lambda)}$ and $a_{n*P}^{(\lambda)'}$, there exist the square matrix $M_{p*p}^{(\lambda)}$ and $a^{(\lambda)} M^{(\lambda)} = Pa^{(\lambda)'}$, for every eigenvalue λ (multiplicity of p) in A and A', the corresponding left singular vector set $a_{P*n}^{(\lambda)}$ and $a_{P*n}^{(\lambda')}$, there exist the square matrix $M_{p*p}^{(\lambda)}$ and $M^{(\lambda)^T} a^{(\lambda)} = a^{(\lambda')} P^T$. Then two graphs are isomorphic.

Proof: Both A and A' are similar to the diagonal matrix $\wedge$ (the element at the diagonal is the eigenvalue sequence). Suppose there are t different eigenvalue. Then,

$$A = (a^{(\lambda_1)}, ..., a^{(\lambda_t)}) \cdot\wedge\cdot (a^{(\lambda_1)}, ..., a^{(\lambda_t)})^{-1} \ ...\ (3)$$

$$A' = \left(a^{(\lambda_1)'}, ..., a^{(\lambda_t)'}\right) \cdot\wedge\cdot \left(a^{(\lambda_1)'}, ..., a^{(\lambda_1)'}\right)^{-1} = P^T (a^{(\lambda_1)} M^{(\lambda_1)}, ..., a^{(\lambda_t)} M^{(\lambda_t)}) \cdot\wedge\cdot (a^{(\lambda_1)} M^{(\lambda_1)}, ..., a^{(\lambda_t)} M^{(\lambda_t)})^{-1} (P^T)^{-1} ...\ (4)$$

From Lemma 3, $M^{(\lambda)}$ is invertible; then $a^{(\lambda)} M^{(\lambda)}$ could be the p linearly independent eigenvector of A, substitute $a^{(\lambda)}$ in (1) by $a^{(\lambda)} M^{(\lambda)}$, then: $A = (a^{(\lambda_1)} M^{(\lambda_1)}, ..., a^{(\lambda_t)} M^{(\lambda_t)}) \cdot\wedge\cdot (a^{(\lambda_1)} M^{(\lambda_1)}, ..., a^{(\lambda_t)} M^{(\lambda_t)})^{-1} \ ...\ (5)$

Put (5) into (4), and $P^T = P^{-1}$. If left singular vector set also applies, then A=$PA'P^T$. Under the same principle, for edge adjacency matrix $B = P_B B' P_B{}^T$, therefore two graphs are isomorphic. Both Permutation and Equinumerosity Theorem have laid the foundations for the algorithm in this paper.