DOI: https://doi.org/10.53555/nnms.v7i11.945 **Publication URL: http://nnpub.org/index.php/MS/article/view/945**

THE DETERMINATION OF GRAPH ISOMORPHISM USING R SOFTWARE

Chaobing He[1]*
*[1]*School of Mathematics and Statistics, Anyang Normal University, Anyang 455000, China*
Email: chaobing5@163.com

Corresponding Author: -
Email: chaobing5@163.com

Abstract: -
This paper considers the determination of graph isomorphism using R. After the preliminaries about graph theory is introduced systematically, the graph isomorphism determination process are studied. Then computer program for the determination of graph isomorphism is written using R. According to these R codes, the paper determines the isomorphism of three 3-regular graphs on 6 vertices and three 3-regular graphs on 8 vertices respectively, and the output proves that the R codes are very practical and effective.

Keywords: *adjacency matrix; permutation matrix; graph sequence; kregular graph; eigenvalue; isomorphism mapping*
Mathematics Subject Classification: 05C60

Distributed under Creative Commons CC BY-NC 4.0 OPEN ACCESS

1. INTRODUCTION

Graph theory is an important branch of modern mathematics, a bridge between mathematics and computer science and between mathematics and other disciplines, and an important tool for establishing and processing discrete mathematical models. Many states in objective reality can be described by graphs and solved with knowledge of graph theory. Therefore, graph theory is a new instrumental subject with strong practicality.

The isomorphism determination of graphs is one of the basic problems in graph theory. The isomorphism of a graph is simply that two graphs have exactly the same structure. The concept of "isomorphism" seems so simple, but determining whether two graphs are isomorphism is a very difficult task. For a long time, scholars have tried to find a set of invariants to determine the isomorphism of graphs, and some people once thought that the isomorphism of graphs could be determined according to the characteristic polynomial and eigenvalue of the graph's adjacency matrix, but the results all failed. There are a large number of literatures on the graph isomorphism problem including [1–10] among others. In this paper, we write a computer program using R language to determine the isomorphism of graphs by searching permutation matrix according to some relevant conclusions of graph isomorphism such as adjacency matrix.

The rest of this paper is organized as follows. In Section 2, we present a description of preliminaries about graph theory. Section 3 describes the graph isomorphism determination process. In Section 4, computer program for the determination of graph isomorphism is written using R. Finally, we summarize and conclude the paper in Section 5.

2. Preliminaries

Definition 1 *The graph G is an ordered triplet* $G = (V(G), E(G), \phi_G)$*, where* $V(G)$ *is a fixed point non-empty set,* $E(G)$ *is a set of edges, and* ϕ_G *is an association function between edges and a pair of unordered vertices (which can be the same) in G.*

Definition 2 $v(G)$*, v for short, denotes the number of vertices of the graph G;* $\varepsilon(G)$*,* ε *for short, denotes the number of edges of the graph G.*

Definition 3 *If the graph G has neither lifting ring nor multiple edges, then G is called a simple graph.*

Definition 4 *The number of edges associated with vertex u in the graph G is called the degree of vertex u, denoted as* $d_G(u)$ *or* $d(u)$ *for short.* $\delta(G)$ *and* $\Delta(G)$ *respectively denote the minimum and maximum degrees of the vertices in G.*

Theorem 1 $\sum_{v\in V} d(v) = 2\varepsilon.$

Corollary 1 *The number of vertices with odd degrees is even.*

Definition 5 *If* $\delta(G) = \Delta(G)$*, then graph G is called a k-regular graph.*

Definition 6 *The degree sequence of a simple graph is called a graph sequence.*

Theorem 2 *Degree sequence* $d = (d_1, d_2, \cdots, d_n)$ *is a nonincreasing sequence of negative integers, set* $d' = (d_2 - 1, d_3 - 1, \cdots, d_{d1+1} - 1, d_{d1+2}, \cdots, d_n)$*, then d is a graph sequence* $\Longleftrightarrow$ d' *is a graph sequence.*

Definition 7 *If there is a two-sided single-value mapping between the graph* $G = (V(G), E(G), \phi_G)$ *and the graph* $H = (V(H), E(H), \phi_H)$*, which satisfies* $\theta : V(G) \to V(H)$ *and* $\varphi : E(G) \to E(H)$*, and* $\phi_G = uv \Leftrightarrow \phi_H(\varphi(uv)) = \theta(u)\theta(v)$*, we call graphs G and H isomorphic, denote it as* $G \cong H$.

Definition 8 *Adjacency matrix* $A(G) = [a_{ij}]_{v\times v}$, *where* a_{ij} *is the number of edges connecting vertex A to vertex B.*

Definition 9 *If every row and every column of a square matrix has exactly one number 1, the square matrix is called a permutation matrix.*

Theorem 3 *The adjacency matrices of the graphs G and H is A and B respectively, then* $G \cong H \Longleftrightarrow$ *there is a permutation matrix P, so that* $PAP' = B$.

3. Graph isomorphism determination process

According to the preliminaries in Section 2, we can obtain the graph isomorphism determination process.

Let the degree sequences of vertices in *G* and *H* be d_1 and d_2 in descending order respectively, and the adjacency matrices be *A* and *B* respectively. The steps to determine whether *G* and *H* are isomorphic are as follows:

(1) If $d_1 \neq d_2,$ then *G* and *H* are not isomorphic; otherwise, proceed to the step (2);

(2) If the eigenvalues of *A* are different from the ones of *B*, then *G* and *H* are not isomorphic; otherwise, proceed to the step (3);

(3) If there is a permutation matrix *P* so that $PAP' = B$, then $G \cong H$; otherwise, *G* and *H* are not isomorphic.

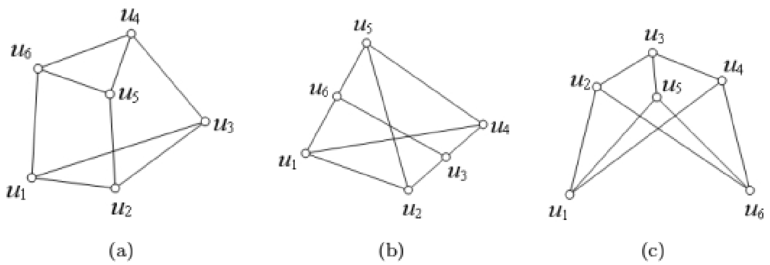

Figure 1Three of 3-regular graphs on 6 vertices

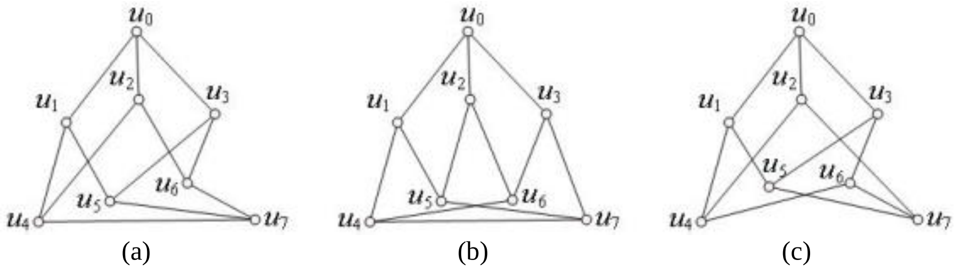

Figure 2Three of 3-regular graphs on 8 vertices

4. R codes for the determination of graph isomorphism

Now we will determine whether the three graphs in Figure 1 are isomorphic and the three graphs in Figure 2 are isomorphic.

First, we determine whether the three graphs in Figure 1 are isomorphic. Since the graph sequences of 3-regular graphs on 6 vertices are the same, we directly determine whether the eigenvalues of their adjacency matrices are equal.

The adjacency matrices of three 3-regular graphs on 6 vertices in Figure 1 is

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 1(a) are as follows.

```
> fig1a=matrix (c (0,1,1,0,0,1,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,
+0,0,0,0,0,1,0,0,0,0,0,0),nrow=6,byrow=TRUE)
> fig1a=fig1a+t(fig1a)
> eigen(fig1a) $values
[1] 3.000000e+00 1.000000e+00 3.108624e-15 1.073775e-16
[5] -2.000000e+00 -2.000000e+00
```

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 1(b) are as follows.

```
> fig1b=matrix (c(0,1,0,1,0,1,0,0,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,0,
+0,0,0,0,0,1,0,0,0,0,0,0),nrow=6,byrow=TRUE)
> fig1b=fig1b+t(fig1b)
> eigen(fig1b)$values
[1] 3.000000e+00 1.776357e-15 0.000000e+00 0.000000e+00
[5] 0.000000e+00 -3.000000e+00
```

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 1(c) are as follows.

```
> fig1c=matrix(c(0,1,0,1,1,0,0,0,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,1,
+0,0,0,0,0,1,0,0,0,0,0,0),nrow=6,byrow=TRUE)
> fig1c=fig1c+t(fig1c)
> eigen(fig1c)$values
[1] 3.000000e+00 1.776357e-15 0.000000e+00 0.000000e+00
[5] 0.000000e+00 -3.000000e+00
```

Let G_1,G_2 and G_3 be the three graphs in Figure 1. Obviously, the eigenvalues of adjacency matrix of G_1 are not the same as the eigenvalues of adjacency matrices of G_2 and G_3, but the eigenvalues of adjacency matrix of G_2 are the same as the eigenvalues of adjacency matrix of G_3. So G_1 is not isomorphic to G_2 and G_3.

Then, determine whether G_2 and G_3 are isomorphic.

The R codes for the output of permutation matrix and isomorphic mapping about G_2 and G_3 are as follows.

```
> library(gtools)
> n=6
> s=diag(n)
> pa=permutations(n,n,1:n)
> pa=t(pa)
> d=s[,pa[,1:ncol(pa)]]
> i=1
> repeat {
+ c=d[,i:(i+n-1)]
+ i=i+n
+ if(all((c%*%fig1b)%*%t(c)== fig1c)) {
+ break
+       }
+ }
> c
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    0    0    0    0
[2,]    0    1    0    0    0    0
[3,]    0    0    1    0    0    0
[4,]    0    0    0    1    0    0
[5,]    0    0    0    0    0    1
[6,]    0    0    0    0    1    0
> apply(c,2, function(x)which(x==1))
[1] 1 2 3 4 6 5
```

The output shows that $G_2 \tilde{=} G_3$. The isomorphism mapping from G_2 to G_3 is

$$u_1 \longleftrightarrow u_1, u_2 \longleftrightarrow u_2, u_3 \longleftrightarrow u_3, u_4 \longleftrightarrow u_4, u_5 \longleftrightarrow u_6, u_6 \longleftrightarrow u_5.$$

Second, we determine whether the three graphs in Figure 2 are isomorphic. Since the graph sequences of 3-regular graphs on 8 vertices are the same, we directly determine whether the eigenvalues of their adjacency matrices are equal.

The adjacency matrices of three 3-regular graphs on 8 vertices in Figure 2 is

$$\begin{pmatrix} 0&1&1&1&0&0&0&0\\ 1&0&0&0&1&1&0&0\\ 1&0&0&0&1&0&1&0\\ 1&0&0&0&0&1&1&0\\ 0&1&1&0&0&0&0&1\\ 0&1&0&1&0&0&0&1\\ 0&0&1&1&0&0&0&1\\ 0&0&0&0&1&1&1&0 \end{pmatrix}, \begin{pmatrix} 0&1&1&1&0&0&0&0\\ 1&0&0&0&1&1&0&0\\ 1&0&0&0&0&1&1&0\\ 1&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&1&1\\ 0&1&1&0&0&0&0&1\\ 0&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&0 \end{pmatrix}$$

And

$$\begin{pmatrix} 0&1&1&1&0&0&0&0\\ 1&0&0&0&1&1&0&0\\ 1&0&0&0&1&0&0&1\\ 1&0&0&0&0&1&1&0\\ 0&1&1&0&0&0&1&0\\ 0&1&0&1&0&0&0&1\\ 0&0&0&1&1&0&0&1\\ 0&0&1&0&0&1&1&0 \end{pmatrix}$$

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 2(a) are as follows.

```
> fig2a=matrix(c(0,1,1,1,0,0,0,0,
+ 0,0,0,0,1,1,0,0,
+ 0,0,0,0,1,0,1,0,
+ 0,0,0,0,0,1,1,0,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,0),nrow=8,byrow=TRUE)
> fig2a=fig2a+t(fig2a)
> eigen(fig2a)$values
[1] 3 1 1 1 -1 -1 -1 -3
```

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 2(b) are as follows.

```
> fig2b=matrix(c(0,1,1,1,0,0,0,0,
+ 0,0,0,0,1,1,0,0,
+ 0,0,0,0,0,1,1,0,
+ 0,0,0,0,0,0,1,1,
+ 0,0,0,0,0,0,1,1,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,0,
+ 0,0,0,0,0,0,0,0),nrow=8,byrow=TRUE)
> fig2b=fig2b+t(fig2b)
> eigen(fig2b)$values
[1] 3.0000000 1.0000000 1.0000000 0.4142136 0.4142136
[6] -1.0000000 -2.4142136 -2.4142136
```

The R codes for the input of adjacency matrix and the output of the eigenvalues about Figure 2(c) are as follows.

```
> fig2c=matrix(c(0,1,1,1,0,0,0,0,
+ 0,0,0,0,1,1,0,0,
+ 0,0,0,0,1,0,0,1,
+ 0,0,0,0,0,1,1,0,
+ 0,0,0,0,0,0,1,0,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,1,
+ 0,0,0,0,0,0,0,0),nrow=8,byrow=TRUE)
> fig2c=fig2c+t(fig2c)
> eigen(fig2c)$values
[1] 3.0000000 1.0000000 1.0000000 0.4142136 0.4142136
[6] -1.0000000 -2.4142136 -2.4142136
```

Let H_1,H_2 and H_3 be the three graphs in Figure 2. Obviously, the eigenvalues of adjacency matrix of H_1 are not the same as the eigenvalues of adjacency matrices of H_2 and H_3, but the eigenvalues of adjacency matrix of H_2 are the same as the eigenvalues of adjacency matrix of H_3. So H_1 is not isomorphic to
H_2 and H_3.
Then, determine whether H_2 and H_3 are isomorphic.

The R codes for the output of permutation matrix and isomorphic mapping about H_2 to H_3 are as follows.

```
> n=8
> s=diag(n)
> pa=permutations(n,n,1:n)
> pa=t(pa)
> d=s[,pa[,1:ncol(pa)]]
> i=1
> repeat {
+ c=d[,i:(i+n-1)]
+ i=i+n
+           if(all((c%*%fig2b)%*%t(c)== fig2c)) {
+       break
+       }
+ }     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
> c
[1,]      1    0    0    0    0    0    0    0
[2,]      0    0    1    0    0    0    0    0
```

```
[3,]    0    1    0    0    0    0    0    0
[4,]    0    0    0    1    0    0    0    0
[5,]    0    0    0    0    0    1    0    0
[6,]    0    0    0    0    0    0    1    0
[7,]    0    0    0    0    0    0    0    1
[8,]    0    0    0    0    1    0    0    0
> apply(c,2, function(x)which(x==1))
[1] 1 3 2 4 8 5 6 7
```

The output shows that $H_2 \cong H_3$. The isomorphism mapping from H_2 to H_3 is

$$u_0 \longleftrightarrow u_1,\ u_1 \longleftrightarrow u_2,\ u_2 \longleftrightarrow u_1,\ u_3 \longleftrightarrow u_3,\ u_4 \longleftrightarrow u_7,\ u_5 \longleftrightarrow u_4,\ u_6 \longleftrightarrow u_5,\ u_7 \longleftrightarrow u_6.$$

5. Conclusions

In this paper, we consider the determination of graph isomorphism using R software. The graph isomorphism determination process is given according to some relevant conclusions of graph isomorphism such as adjacency matrix. Computer program for the determination of graph isomorphism is written using R. The output about the two sets of graphs proves that the R codes written are very practical and effective.

Reference

[1].R. Dondi, G. Mauri, I. Zoppis, Graph isomorphism, Encyclopedia of Bioinformatics & Computational Biology 1 (2019) 933–939.
[2].R. C. Read, D. G. Corneil, The graph isomorphism disease, Journal of Graph Theory 1 (4) (2010) 339–363.
[3].R. Mathon, A note on the graph isomorphism counting problem, Information Processing Letters 8 (3) (1979) 131–136.
[4].H. L. Bodlaender, Bodlaender, h.: Polynomial algorithm for graph isomorphism and chromatic index on partial k-trees. journal of algorithms 11(4), 631-643, Journal of Algorithms 11 (4) (1990) 631–643.
[5].V. Arvind, P. P. Kurur, Graph isomorphism is in spp, Information & Computation 204 (5) (2013) 835–852.
[6].M. Pelillo, Replicator equations, maximal cliques, and graph isomorphism, Neural Computation 11 (8) (1999) 1933.
[7].S. D. Berry, J. B. Wang, Two-particle quantum walks: Entanglement and graph isomorphism testing, Physical Review A 83 (4) (2011) 786–792.
[8].C. Moore, A. Russell, P. ?niady, On the impossibility of a quantum sieve algorithm for graph isomorphism, Siam Journal on Computing 39 (6) (2010) 2377–2396.
[9].Lubiw, Anna, Some np-complete problems similar to graph isomorphism, Siam Journal on Computing 10 (1) (2006) 11–21.
[10]. Y. Aflalo, A. Bronstein, R. Kimmel, On convex relaxation of graph isomorphism, Proceedings of the National Academy of Sciences of the United States of America 112 (10) (2015) 2942–7.