

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**SciVerse ScienceDirect**

Physics Procedia 25 (2012) 1761 – 1768

Physics

**Procedia**

2012 International Conference on Solid State Devices and Materials Science

## A Partition Method for Graph Isomorphism

Lijun Tian, Chaoqun Liu and Jianquan Xie

*Information Technology Department Hunan University of Finance and Economics No.139, Fenglin 2nd Road, Changsha, 410205, China*

### Abstract

Complexity of the graph isomorphism algorithms mainly depends on matching time which is directly related to efficiency of their partition methods. This paper proposed a partition method by sorted sequences of length-L path numbers, and divided cells of partition into 3 categories: not similar; completely similar; similar but not completely. The method was tested on several types of graphs with different order, each type with the same order 100 graphs. The results indicate that not similar cells can be refined by adding path length to other types or trivial cells if the vertex is not similar with all other vertices. For almost all asymmetric graphs, the path length is a small value, e.g., for 6-regular graphs with 100~1000 vertices the average path length is 4 to get all cells to trivial ones.

© 2012 Published by Elsevier B.V. Selection and/or peer-review under responsibility of Garry Lee

Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Graph isomorphism; Partition method; Similar; Path length; Pattern recognition.

### 1. Introduction

Graph-based methodologies have been proposed as a powerful tool for pattern recognition and computer vision starting from the late 1970s [1,2]. All of these approaches are related to problems called as graph-matching [3].

Graph-matching problems can be divided into two categories: exact and inexact. The graph isomorphism (GI) is the simplest form of exact graph matching, which is still an open question whether it is a NP-complete problem or not [4], while other problems such as subgraph isomorphism problem are proved to be NP-complete.

GI problems can be solved by a type of brute-force backtrack search, while it yield  $O(n!)$  time for an  $n$ -vertex graph in the worst case[5]. This naive approach can be approved by classifying the vertices of the graph into more than one class, i.e., a partition of vertex set. Let  $\pi=(\pi_1, \dots, \pi_r)$  is such a partition, there

need only  $\prod_{i=1}^r |\pi_i|$  time to match.

The elements of a partition,  $\pi_i (1 \leq i \leq r)$ , are usually called its cells. If  $|\pi|=1$ , there is a trivial partition, means that  $\pi$  has only one cell, which complexity yield  $O(n!)$  time. And if  $|\pi|=n$  (the number of vertices), i.e., every cell of  $\pi$  has only one vertex, there need only one time to match, while  $\pi$  is called as a discrete partition.

The complexity of a GI algorithm mainly depends on the efficiency of its partition method, which is lower while the values of cells are smaller. The famous Nauty algorithm [6] took a partition method according to the number of neighbors in each cells (so-called color-class), which known as the one-dimensional Weisfeiler-Lehman method [7], also known as the naive vertex-classification algorithm. Miyazaki extended the naive classification algorithm by defining  $d_i^s$  as being the number of nodes in the  $i$ th color class whose distance to  $v$  is  $\delta$  [5]. Mateus extended Miyazaki's refinement procedure to a generalized function [8]. Zou et al proposed a refinement method by path-numbers used Mateus' function [9].

In this paper, we proposed a partition method according to the value of path-numbers to all other vertices like in [9], but didn't use the Mateus' function. We give short definitions in section II, and describe the partition method in section III, then discuss the results in section IV, finally we make a conclusion.

## 2. Definitions

Here, we consider only undirected graphs without parallel edges and loops, i.e., undirected simple graph, as showed in Fig.1. Definitions are mainly referenced from graph theory [4] and practical graph isomorphism [6].

Definition 1: a graph  $G$  is an ordered pair  $(V(G), E(G))$  consisting of a set  $V(G)$  of vertices and a set  $E(G)$ , together with an incidence function  $\psi: G \rightarrow V \times V$  that associates with each edge of  $G$  an unordered pair of vertices of  $G$ .

Definition 2: two graphs  $G$  and  $H$  are isomorphism, if there is a bijection  $\theta: V(G) \rightarrow V(H)$  which preserves adjacency (that is, the vertices  $u$  and  $v$  are adjacent in  $G$  if and only if their images  $\theta(u)$  and  $\theta(v)$  are adjacent in  $H$ ), written  $G \cong H$ , else written  $G \not\cong H$ .

Clearly, in Fig.1 there is  $G \not\cong H$ , but those in Fig.2 are not so easy to estimate.

Definition 3: an automorphism of a graph is an isomorphism of the graph to itself.

Clearly, there is an automorphism of  $H$  in Fig.2 ( $v_1 v_2 v_3 v_4 v_5 v_6 \rightarrow v_2 v_3 v_4 v_5 v_6 v_1$ ).

Definition 4: a partition of graph  $G$  is a set of disjoint non-empty subsets of  $V(G)$ .

There is a partition by degree of graph  $G$  in Fig.1:  $((u_1, u_6), (u_2, u_4), (u_3, u_5))$ . Cell with only one vertex are called as a trivial cell.

Definition 5: vertices  $u$  and  $v$  of a graph  $G$  are similar, if there is an automorphism  $\alpha$  which maps  $u$  to  $v$ , written  $u \sim v$ .

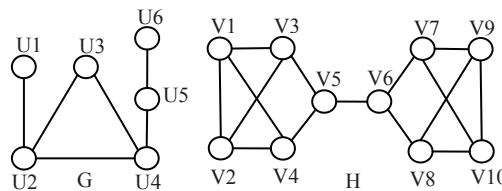


Fig.1 Undirected simple graphs

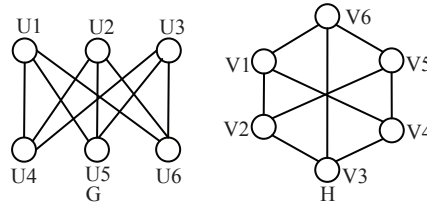


Fig.2 Isomorphism and automorphism

Vertices of graph G in Fig.1 all are not similar, and of H, there are three subnets of similar vertices(called as similar cells):  $(v_5, v_6), (v_1, v_2, v_9, v_{10}), (v_3, v_4, v_7, v_8)$ . All vertices of graph G (or H) in Fig.2 are similar.

Here, we introduce a new concept which called as completely similar.

Definition 6: vertices  $u$  and  $v$  of a graph  $G$  are completely similar, if there is an automorphism  $\alpha$  which maps  $u$  to  $v$ , and all the other vertices map to themselves, written  $u \equiv v$ .

Vertices of completely similar are definitely similar, but those of similar are not always completely.

### 3. Length-L PATH-numbers partition method

#### 3.1 Original sequences

To describe our method, we take the form of adjacency matrix for graphs by a random way, i.e., label the vertices randomly. Consider two matrixes A and B in (1), which is the one of H in Fig.1?

$$\begin{matrix}
 \begin{matrix}
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
 \end{matrix} &
 \begin{matrix}
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
 \end{matrix}
 \end{matrix} \tag{1}$$

A1
B1

Obviously, A1 is the matrix of H, but in fact, B1 is also a matrix of H (labelled vertices in another sequence: 6,8,5,1,9, 2,10,7,4,3 replaced  $v_1, v_2, \dots, v_9, v_{10}$ ).

Compare two matrix, there are two rows (or columns) such as 3,4 and 7,8 are equal to each other respectively in A1, while there are two groups of rows are equal (1,5 and 7,10) in B1, though the labels are different. Except for those are completely equal, there are rows such as 1,2 and 9,10 in A1 (their counterpart are 3,4 and 6,8) are equal to each other which omit themselves.

These properties show that there are vertices equal to each other independent of labelling sequence, which means those vertices don't need to be partitioned.

It's easy to validate those vertices are completely similar. A cell  $\pi_i$  of a partition  $\pi$  is completely similar if  $u \equiv v$  for any two vertices  $u, v \in \pi_i$ .

Graph H in Fig.1, there are four completely similar cells:  $(v_1, v_2), (v_3, v_4), (v_7, v_8)$  and  $(v_9, v_{10})$ . There are two cells of G in Fig.2:  $(u_1, u_2, u_3), (u_4, u_5, u_6)$ .

**Theorem 1:** a completely similar cell doesn't need to be partitioned.



3 3 3 3 3 3 3 3 3 3	7 7 7 7 7 7 7 7 7 7	(4)
2 2 2 2 2 2 3 3 3 3	7 7 7 7 7 7 7 7 7 7	
2 2 2 2 2 2 1 1 1 1	5 5 5 5 5 5 7 7 7 7	
1 1 1 1 1 1 1 1 1 1	2 2 4 4 4 4 2 2 2 2	
1 1 1 1 1 1 1 1 1 1	2 2 2 2 2 2 2 2 2 2	
0 0 0 0 0 0 0 0 0 0	2 2 2 2 2 2 1 1 1 1	
0 0 0 0 0 0 0 0 0 0	2 2 0 0 0 0 1 1 1 1	
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	
SA2	SA3	

Because the matrix of sorted sequence is independent of labeling order, we need only one matrix for any given length-L. We use SAL to denote the sorted matrix of length-L according to A, i.e., the sorted matrix of A<sup>(L)</sup>.

Compare SA2 and SA3, there are two types of columns in SA2 (3,2,2,1,1,0,0,0,0,0 and 3,3,1,1,1,0,0,0,0,0), and three types of columns in SA3 (7,7,5,2,2,2,2,0,0,0; 7,7,5,4,2,2, 0,0,0,0 and 7,7,7,2,2,1,1,0,0,0). That's mean those vertices cannot be partitioned by length-L, which is called as length-L similar, may be partitioned by length-(L+1). So we can add the length to refine the partition.

**Theorem 2:** not similar cells can be refined to trivial or similar cells by adding path length.

In fact, for lots of graphs, it is easy to partition by a small path length. As in (5), we can detect that partition graph G of Fig.1 into a discrete partition only used length-2, let F be the matrix of G.

0 1 0 0 0 0	1 1 2 2 3 3	(5)
1 0 1 1 0 0	1 1 1 1 1 1	
0 1 0 1 0 0	0 1 1 1 1 1	
0 1 1 0 1 0	0 0 0 1 1 1	
0 0 0 1 0 1	0 0 0 1 0 1	
0 0 0 0 1 0	0 0 0 0 0 0	
F	SF2	

Are any graphs can be partition to discrete ones by adding path length? The answer is no. For those similar vertices, the columns are equal to each other. Compare SA10 in (6) and SA3 in (4), we can find that they have the same categories (the first two columns, the middle four columns and the last four columns are equal).

8619 8619 9915 9915 9915 9915 9933 9933 9933 9933	(6)	
8458 8458 9914 9914 9914 9914 9933 9933 9933 9933		
8458 8458 8458 8458 8458 8458 8421 8421 8421 8421		
5936 5936 8421 8421 8421 8421 8421 8421 8421 8421		
5936 5936 8421 8421 8421 8421 5936 5936 5936 5936		
5669 5669 3488 3488 3488 3488 5669 5669 5669 5669		
5669 5669 3184 3184 3184 3184 3184 3184 3184 3184		
3488 3448 3184 3184 3184 3184 3184 3184 3184 3184		
3488 3448 2032 2032 2032 2032 2184 2184 2184 2184		
3328 3328 2032 2032 2032 2032 2184 2184 2184 2184		
SA10		

**Theorem 3:** Vertex u ~ v if and only if they have same sorted sequence of path number for any length to all other vertices.

Proof: By definition, if u ~ v, they must have same sorted sequence which is independent of labeling order. If u and v have same sorted sequence of path number to all other vertices. By way of contradiction, suppose that u and v are not similar, map u to v by θ, then there must be vertices such as i and j: (i,

$j) = (\theta(i), \theta(j))$ . Let  $x$  be the number of all paths from  $u$  to other vertices through  $(i, j)$ , let  $y$  be the number of all paths from  $u$  to other vertices through  $(\theta(i), \theta(j))$ , clearly,  $x \neq y$ , then  $u$  and  $v$  have difference path number because  $\theta$  maps  $u$  to  $v$ . It's contradiction with the premise, so theorem 3 is true.

Corollary 1: similar vertices cannot be partitioned by adding path length.

### 4. Results and discussion

#### 4.1 Results

We test length- $L$  path-numbers partition method on lots of graphs, mainly including three types: specific graphs such as regular ring lattice (a graph with  $n$  vertices, each connected to its  $2k$  nearest neighbours by undirected edges);  $k$ -regular graphs; random graphs.

Numbers of some specific graphs for similarity are showed in table I. The column without similar means that there are no similar vertices in a graph (for any two vertices  $u$  and  $v$  are not similar), i.e., all cells can be partitioned to trivial ones by adding path length. And column of similar but not completely means that there are at least two vertices  $u$  and  $v$  are similar, but no vertices are completely similar. Then completely similar column means that there are at least two vertices are completely similar.

Table 1 Similar Cell in Specific Graphs

Name	Graphs		Without similar	Similar not completely	Completely similar
	Vertices	Edges			
Empty	$n$	0	0	0	All
$K_n$	$n$	$n*(n-1)/2$	0	0	All
$K_{n,m}$	$n+m$	$n*m$	0	0	All
Peterson	10	15	0	1	0
Frucht	12	18	1	0	0
Heawood	14	21	0	1	0
Regular ring lattice	$n$	$n*k/2$	0	All	0

The results shows that complete graph, complete bipartite or empty graph all have completely similar vertices. In fact, all vertices of complete and zero graph are completely similar, some specific graphs (such as  $K_{n,m}$ ) have large size completely cell, yet others (e.g. regular ring lattice) have zero. But all vertices of regular ring lattice are similar. For the specific three graphs, vertices of Peterson and Heawood are all similar but not completely, while Frucht don't have any similar vertices.

For random graphs and  $k$ -regular graphs, we test which have 10,20,...,90,100 vertices with degree or average degree of 3,6,9,12, each type 100 graphs. The graphs are denoted a name with type and vertices: such as RG03 means that regular graphs with degree of 3; RD06 means random graphs with average degree of 6, i.e., with  $|V|*6/2$  edges totally.

As showed in Fig.3, graph numbers with completely similar vertices decreased rapidly according to the increasing of degree. For graphs with 9 and 12 degree, they almost don't have any completely graphs except for those with 10 vertices, so they are not included in the figure. Only numbers of random graphs with average degree 3 increased with the increasing of vertex number.

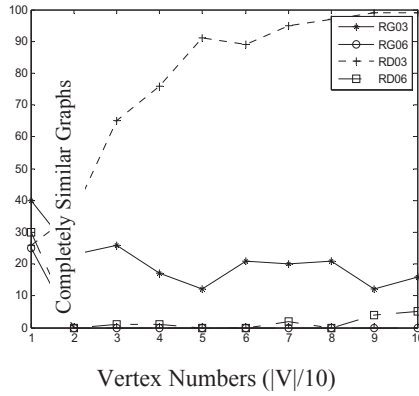


Fig.3 Graphs with Completely Similar Vertices

As showed in Fig.4, graph numbers without any similar vertices (so-called asymmetric graphs). The number of asymmetric graphs increases with the increasing of vertex number or degree except for random graphs with average degree 3.

Then, for asymmetric graphs, how long the path length is needed to get a discrete partition? We test random graphs and k-regular graphs with 100,200,...1000 vertices. We find that all random graphs are easier to partition than those k-regular with the same vertices and degree. All 6-regular graphs with 100-1000 vertices are asymmetric and with a very small value about 4 times to get a discrete partition. The maximal value we find is in 3-regular graphs with 100 vertices, so we test 1000 of this type, and 6-regular graphs, the results are showed in Fig.5.

4.2 Discussion

The partition methods used in [6-9] are all take iterating algorithms. Those in [6-8] take the difference metrics, so we don't compare them here. Ref. [9] uses the path length as metrics too, their method is defined a function as in (7):

$$Q_\delta(v) = (\phi(v), \text{SORT}\{(i, d, n, c)\}) \tag{7}$$

Where  $\phi$  is a partition,  $i$  is cell  $i$ ,  $d$  is a shortest distance,  $n$  is the path-number, and  $c$  is the similar vertex number according to this condition.

So, our method is difference from those in [9], though we use a same metrics.

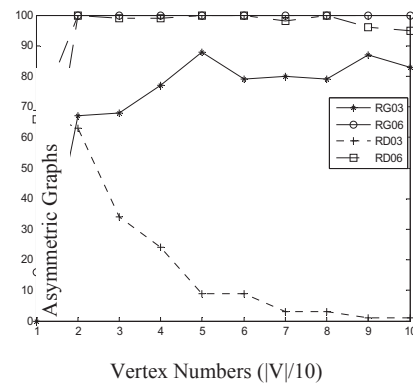


Fig.4 Graphs without Similar Vertices

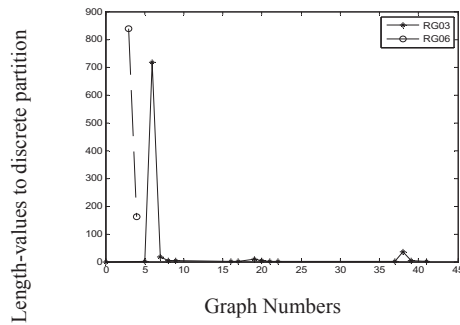


Fig.5 Length-value to discrete partition of 3-regular with 100 vertices

## 5. Conclusion

GI is an unsolved problem which is very important to graph-based pattern recognition and computer vision. In this paper, we proposed a length-L path-number partition method to study the similarity of graph vertices.

Firstly we divided vertices into 3 categories: not similar, completely similar, and similar but not completely. Then we illustrated the method in detail. Finally we tested the method on many graphs.

The conclusion of our research is that completely similar vertices don't need to and cannot be partitioned. For almost asymmetric graphs, only a small value of path-length is needed to partition a graph to discrete ones.

For those of similar but not completely, cannot be partitioned by adding path-length, which need a dynamic way, and this will be our works in the future.

## References

- [1] D. Conte, P. Foggia, C. Sansone, and M. Vento. How and Why Pattern Recognition and Computer Vision Applications Use Graph. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007, doi: 10.1007/978-3-540-68020-8\_4.
- [2] Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3), pp.265–298, doi: 10.1142/S0218001404003228.
- [3] Kandel, Abraham, Bunke, eds. Applied Graph Theory in Computer Vision and Pattern Recognition, Springer, Berlin Heidelberg, New York, 2007. ISBN: 978-3-540-68019-2.
- [4] J.A. Bondy and U.S.R. Murty. Graph theory, Springer, 2008. ISBN: 978-1-84628-969-9, doi: 10.1007/978-1-84628-970-5.
- [5] T. Miyazaki, The complexity of McKay's canonical labeling algorithm, *Groups and Computation II, DIMACS Series Discrete Mathematics Theoretical Computer Science*, 28 (1997), pp. 239-256.
- [6] McKay B.D. "Practical graph isomorphism". *Congressus Numerantium*, 1981, 30(1), pp.45-87.
- [7] J. Cai, M. Furer and N. Immerman, An Optimal Lower Bound on the Number of Variables for Graph Identification, *Combinatorica*, 12 (1992), 4, pp. 389-410.
- [8] Oliveira M, Greve F. A new refinement procedure for graph isomorphism algorithms. In: Feofiloff P, de Figueiredo CMH, Wakabayashi Y, eds. *Electronic Notes in Discrete Mathematics, Proc. of the GRACO*, 2005. Amsterdam: Elsevier North-Holland, 2005, pp.373-379.
- [9] Zou XX, Dai Q. A vertex refinement method for graph isomorphism. *Journal of Software*, 2007, 18(2), pp.213-219. <http://www.jos.org.cn/1000-9825/18/213.htm>.