

# A Novel Algorithm for Isomorphic Graph

Jyotirmayee Rautaray<sup>1</sup>, Raghvendra Kumar<sup>2</sup>, Garima Bajpai<sup>3</sup>

School of Computer Engineering, KIIT University, Odisha, India<sup>1</sup>

School of Computer Engineering, KIIT University, Odisha, India<sup>2</sup>

School of Computer Engineering, KIIT University, Odisha, India<sup>3</sup>

**Abstract:** The graph isomorphism difficulty is very famous; isomorphism problem is still one of the unsolved problems of graph theory. Here in this paper we are dealing with some applications of this problem and major idea behind this is advancement of some accessible graph isomorphism algorithms. In this paper we are also focusing on producing the solution for problems that seems like isomorphic problems but they are not exactly isomorphic.

**Keywords:** Isomorphic property, Graph, Sub graph

## I. INTRODUCTION

When humans analyze or synthesize an engineering system by using the mathematical representation governing its behavior, he or she creates a mathematical model of the engineering system, and then writes the equations using knowledge about them and their relation with the physical reality. In usual engineering practice, one uses a model that is known to be suitable for the system at hand and the aim of the computation. Reformulation of the problem into another

Formally understood mathematical system [1] [2] [3] to the extent that it is done for engineering analysis, usually uses continuum mathematics. This paper shows that representations of graph theory for engineering problems can be used as a base upon which to extend formal theories of reformulation. Research in engineering analysis usually starts with an understanding of the physical system, then the adoption of a suitable mathematical model for the system. In this work reported different approaches are adopted. Rather than starting with the physical system itself or the mathematical representations historically used for the behavior of engineering systems, many other mathematical approaches were investigated to find those which can be useful representations of engineering systems. Representations were sought for which knowledge of the mathematical properties of those representations and the relations between them can be used to provide augmented understanding of the physical engineering system. For instance, if two engineering systems can be represented by dual, then the physical systems are dual, and this in turn leads to new insights regarding analogies between the systems. The Embedded Engineering Knowledge project [4][5][6][7][8][9] was devoted to searching for representations isomorphic with one or more engineering systems, and finding common properties, if they exist, between them. After investigating many mathematical alternatives, attention was focused on graph theory. This paper illustrates the approach using results only from the graph theory representation, augmented by theorems from graph theory. Graph theory is a useful representation because on the one hand the elements of the graph can be defined so as to have a one-to-one correspondence with the elements of many kinds of engineering systems. On the other hand, the theorems and algorithms of graph theory allow one also to represent behavioral properties of the system, such as deformations and forces, or velocities and movements, as properties of the vertices or edges of the graph [9] [10] [11]. This paper illustrates how engineering problems, for example: truss structures, planetary gear systems and dynamic mass spring dashpot systems are mapped into graphs, and then analyzed by using the theorems and algorithms of graph theory; these theorems and algorithms constitute knowledge that is embedded in the graph theory.

The class of regular graphs looks hard for GI but the regular graphs are reconstructible. We can use this fact in GI-algorithms. Instead of finding the isomorphism between some regular graphs  $G$  and  $H$  we can try to find out if their decks  $D(G)$  and  $D(H)$  are isomorphic. We will show on the example of the GI-algorithm from [12] [13] [14] [15] that this method is able to improve some GI-algorithms for regular graphs.

**Algorithm: Isomorphic**

*/\*A main idea of the algorithm is a decomposition of the vertex sets  $V G$  (resp.  $V H$ ) into partitions  $\Pi = \{X_1, \dots, X_k\}$  and  $\Psi = \{Y_1, \dots, Y_l\}$ , where  $X_i \cap X_j = \emptyset, Y_i \cap Y_j = \emptyset$  (for  $i \neq j$ ),  $\cup X_i = V G$  and  $\cup Y_i = V H$ . For every vertex  $x \in G$  ( $y \in H$ ) we denote the number of its neighbors from the set  $X_i$  ( $Y_i$ )  $deg_G(x, X_i)$  ( $deg_H(y, Y_i)$ ).\*/*

**Input:** - Graphs  $G$  and  $H$ ;

Let  $\Pi = \{X_1 = V G\}, \Psi = \{Y_1 = V H\}, k = 1$ ;

I        While the partitions  $\Pi, \Psi$  were refined in the last step do

Begin

A    Assign to every vertex  $x \in G$  (resp.  $y \in H$ ) the sequence  $(deg G(x, X_1) \dots deg G(x, X_k))$

(resp.  $(deg H(y, Y_1) \dots deg H(y, Y_l))$ )

B    Form new partitions  $\Pi = \{X_1, \dots, X_m\}, \Psi = \{Y_1, \dots, Y_m\}$  /\*where  $\forall x, x \in G, x \in X_i \iff (deg G(x, X_1), \dots, deg G(x, X_k)) = (deg G(x, X_1), \dots, deg G(x, X_k))$  and the same holds for  $\forall y, y \in H$ \*/

C    Let  $\Pi = \Pi, \Psi = \Psi$  ( $\Pi = \{X_1, \dots, X_m\}, \Psi = \{Y_1, \dots, Y_m\}$ );

D    if  $m = 1$  then stop  $\rightarrow G, H$  are not isomorphic;

E    if  $k = m = 1$  then  $\Pi, \Psi$  were not changed in this step else let  $k = m$ ;

F    if there are  $x \in X_i, y \in Y_j$  such that  $(deg G(x, X_1), \dots, deg G(x, X_k)) = (deg H(y, Y_1), \dots, deg H(y, Y_l))$  and  $|X_i| = |Y_j|$  then stop  $\rightarrow G, H$  are not isomorphic

End

Order  $\Psi$  in such a way as  $\forall x \in G, \forall y \in H$  and  $\forall i \in \{1, \dots, k\} (x \in X_i) \& (y \in Y_i) \iff (deg G(x, X_1), \dots, deg G(x, X_k)) = (deg H(y, Y_1), \dots, deg H(y, Y_l))$ ;

G    For every bisection  $f: V G \rightarrow V H$  for which  $f(x) = y \iff \exists i (x \in X_i) \& (y \in Y_i)$

Do

II    if  $f$  is an isomorphism then stop  $\rightarrow G, H$  are isomorphic;

Else  $G, H$  isn't isomorphic;

The most important condition to prove isomorphism of a graph is that

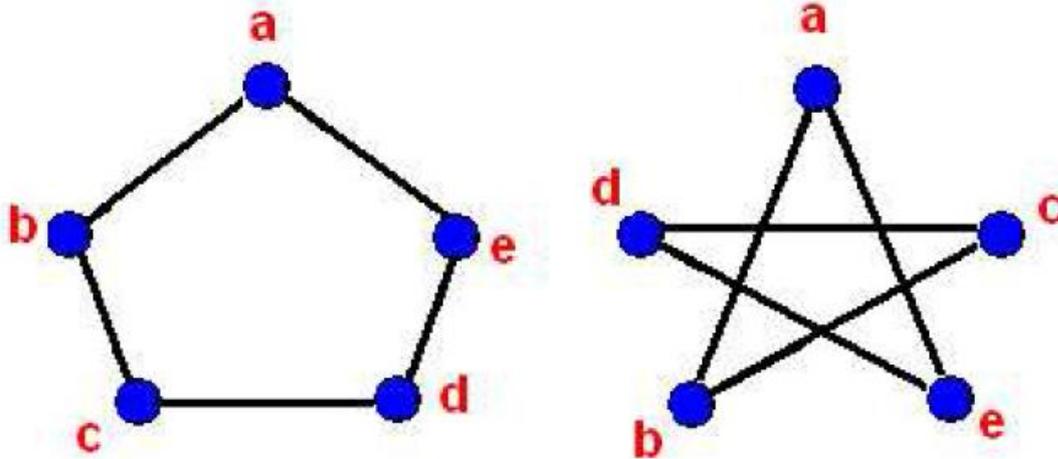
1. The number of vertices is same
2. The number of edge is same

3. The degree of graph is same

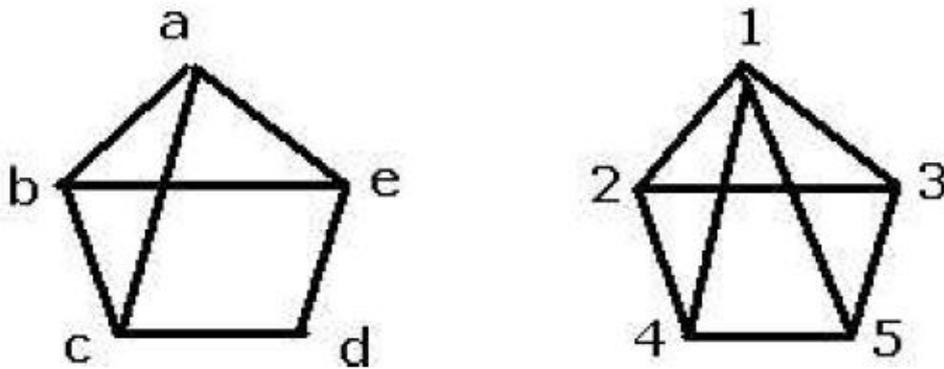
But the most important condition to prove graph is isomorphism of a graph is that

- The sub graph of original graph is also isomorphic

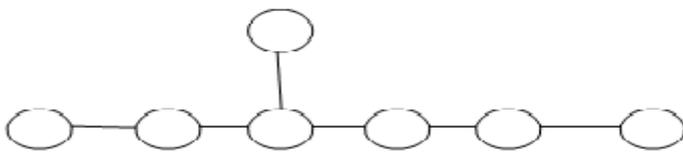
Example1:- Two graphs is isomorphic as they follow all condition



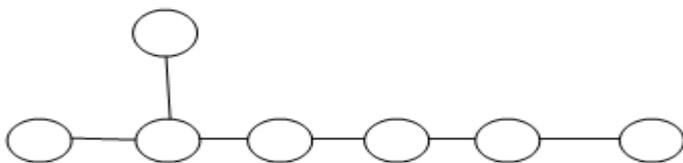
Example2:- Two graphs is non isomorphic



Example3:-Two graphs are non isomorphic but they have same property



Graph a



Graph b

## II. CONCLUSION

Using the isomorphism algorithm, we have shown that graph is isomorphic only on when the sub graph is also isomorphic that have same number of vertices and edge and also have the same number of degree. In future we will try to design and develop a new technique that is easy to compare that two graph is isomorphic or not.

## REFERENCES

- [1] Arvind V., Kurur P.P., Graph Isomorphism is in SP P, Electronic Colloquium on Computational Complexity, Report No.37, 2002
- [2] Babia L., Erdos P., Selkows S., Random Graph Isomorphism, SIAM Journal of Computing, pp.628-635, 1980
- [3] Boppana R., Hastad J., Zachos S., Does  $co-NP$  Have Short Interactive Proofs, Information Processing Letters, pp.27-32, 1987
- [4] Bosak J., Decompositions of Graphs (in Slovak), VEDA Bratislava, 1986
- [5] Chen W.Ch., Hierarchy of Graph Isomorphism Testing, Technical Report, Department of Computer Science, California Institute of Technology, 1984
- [6] Clarke E.M., Enders R., Filkorn T., Jha S., Exploiting Symmetries in Temporal Modal Logic Model Checking, Formal Methods in System Design, 9, 1996
- [7] Cordella L.P., Foggia P., Sansone C., Vento M., Performance Evaluation of the VF Graph Matching Algorithm, Proc. of the 10th ICIAP, IEEE Computer Society Press, pp.1172-1177, 1999
- [8] Czimmermann P., Pesko S., The Regular Permutation Scheduling on Graphs, Journal of Information, Control and Management Systems, vol.1, 2003
- [9] Fortin S., The Graph Isomorphism Problem, Technical Report TR 96-20, Department of Computing Science, The University of Alberta, Canada, 1996
- [10] Garey M.R., Johnson D.S., Computer and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, 1979
- [11] Harary F., Graph Theory, Reading(Mass.), Addison-Wesley, 1969
- [12] Kucera L., Kombinatoricke algoritmy, Praha, SNTL, 1982
- [13] Ladner R.E., On the Structure of the Polynomial Time Reducibility, Journal of the ACM, pp. 155-171, 1975
- [14] Lauri J., Vertex-deleted and Edge-deleted Subgraphs, Collected Papers, The University of Malta, 1992
- [15] Mathon R., A Note on the Graph Isomorphism Counting Problem, Information Processing Letters, pp.131-132, 1979

## BIOGRAPHY



**Jyotirmayee Rautaray** has received B. Tech. (Bachelor of Technology) degree in Computer Science and Engineering from Raajdhani Engineering College “BPUT University, Bhubaneswar (Odisha), India in 2011. She is Pursuing her M. Tech. (Master of Technology) in Computer Science from KIIT University, Bhubaneswar (Odisha), India in 2013. Her subjects of interest include Computer Networking, Theory of Computer Science, Data Mining, NLP and Analysis & Design of Algorithms. She has published 10 research papers in international journal. Her researches areas are Computer Networks, Data Mining, cloud computing, NLP and Secure Multiparty Computations.



**Raghvendra Kumar** has received B.Tech. (Bachelor of Technology) degree in Computer Science and Engineering from SRM University “Chennai” (Tamil Nadu), India in 2011. He is Pursuing his M.Tech. (Master of Technology) in Computer Science from KIIT University, Bhubaneswar (Odisha), India in 2013. His subjects of interest include Computer Networking, Theory of Computer Science, Data Mining and Analysis & Design of Algorithms. He has published 13 research papers in international journal and 5 papers in IEEE. His researches areas are Computer Networks, Data Mining, cloud computing and Secure Multiparty Computations.



**Garima Bajpai** has received B. Tech. (Bachelor of Technology) degree in Computer Science and Engineering from Shri Rammurti Smarak college of Engineering and Technology “UPTU University, Barielly (Uttar Pradesh), India in 2009. She is Pursuing her M. Tech. (Master of Technology) in Computer Science from KIIT University, Bhubaneswar (Odisha), India in 2013. Her subjects of interest include Computer Networking, Theory of Computer Science, Data Mining, NLP and Analysis & Design of Algorithms. She has published research papers in international journal. Her researches areas are Computer Networks, Data Mining, cloud computing, NLP and Secure Multiparty Computations.