FROM RECONSTRUCTION CONJECTURE TOWARDS A LEMMA

Subhashis Banerjee[#1], Debajit Sensarma[#1], Krishnendu Basuli[#1], Saptarshi Naskar[#2] and Samar Sen Sarma[#3]

[#1]West Bengal State University, West Bengal, India

{mail.sb88,debajit.sensarma2008,krishnendu.basuli}@gmail.com

[#2] Sarsuna College, West Bengal, India

[2]sapgrin@gmail.com

[#3]University Of Calcutta, West Bengal, India

sssarma2001@yahoo.com

ABSTRACT

The Reconstruction Conjecture has been synthesized under the characterized of matching Polynomial. A Polynomial time algorithm for generating matching polynomial of an undirected graph is given. Algorithms are given for reconstructing a graph from its node-deleted and edge-deleted subgraphs. Also the relation between the isomorphism and reconstruction has been investigated.

KEYWORDS

Multiset, Card, Deck, Matching, Perfect matching, K-matching, Matching polynomial, Tree-Decomposition, Isomorphism

1. INTRODUCTION

The reconstruction conjecture is generally regarded as one of the foremost unsolved problem in graph theory. It was first studied in 1941 by Kelly and Ulam [12]. It got name Reconstruction Conjecture when Harary reformulated it in 1964 [8].

The reconstruction conjecture claims that every graph on at least three vertices is uniquely determined (up to isomorphism) by its collection of vertex deleted subgraphs. Harary[8] formulated the Edge-Reconstruction Conjecture, which states that a finite simple graph with at least four edges can be reconstructed from its collection of one edge deleted sub graphs.

The Reconstruction Conjecture is interesting not only from a mathematical or historical point of view but also due to its applicability in diverse fields. Archaeologists may try to assemble broken fragments of pottery to find the shape and pattern of an ancient vase. Chemists may infer the structure of an organic molecule from knowledge of its decomposition products. In bioinformatics the Multiple Sequence- Alignment problem[2] is to reconstruct a sequence with minimum gap insertion and maximum number of matching symbols, given a list of protein or DNA sequences. In computer networking, a reconstruction problem can appear in the following scenario: given a

DOI : 10.5121/cseij.2012.2106

collection of sketches depicting partial network connection in a city from different locations, reconstruct the network of the entire city.

Reconstruction Conjecture was investigated by several authors and more than 300 papers have appeared so far on this topic. Two approaches were commonly adopted by those authors. One approach is to prove that some classes of graphs are reconstructible in the hope that eventually enough classes will be found to include all graphs. Several classes of graphs are already proved to be reconstructible. This includes disconnected graphs, trees, unicyclic graphs, and some set of separable graphs and some other classes of graphs. A class S of graphs is called reconstructible if each graph in S is reconstructible.

Another approach is to prove some parameters are reconstructible, in the hope that enough parameters to determine the graph completely will be reconstructed. The parameters those have been proved to be reconstructible include degree sequence, number of blocks, number of cut points, connectivity etc. Some graph theorist tried to solve this problem by generating different polynomials such as chromatic polynomial [17], characteristic polynomial [7], rank polynomial [17] etc. But no practical means of reconstruction exists for any of them.

Farrell and Wahid[5] investigated the reconstructibility of matching polynomial and gave a practical method for reconstruction. Although they gave a method for reconstruction of the matching polynomial they did not provide a practical method for generate the matching polynomial for a graph.

In this paper we tried to resolves the problem with reconstruction of matching polynomials m(G)[6]. It is important to know whether or not m(G) is reconstructible. One reason of this is, if m(G) is reconstructible, then any graph that is characterized by m(G) will be reconstructible. It can be easily confirm that the mapping defined by the approach holds. We can say that if the deck of G is given, then the entire graph G can be reconstructed though perhaps not uniquely.

A graph H is called a reconstruction of a graph G if the vertices of G and H can be labeled $v_1, v_2, \ldots, v_n$ and $u_1, u_2, \ldots, u_n$ respectively such that $G - v_i \cong H - u_i$ for every i. A graph G is called reconstructible if every reconstruction of G is isomorphic to G. From the statement we see that it is not the individual graphs G and H which are important, but rather their isomorphism type.

2. PRELIMINARIES

2.1. Some Definitions

Multiset: In mathematics, a Multiset (or bag) is a generalization of a set. While each member of a set has only one membership, a member of a multiset can have more than one membership (meaning that there may be multiple instances of a member in a multiset).

Vertex-Card: A Vertex-Card of G is an unlabelled graph formed by deleting 1 vertex and all edges attached to it.

Edge-Card: A Edge-Card of G is an unlabelled graph formed by deleting 1 edge excluding the end vertices of this edge from G.

Vertex-Deck: The Vertex-Deck of G, VD(G), is the collection of all Vertex-Cards of G. Note this is in general a multiset.

Edge-Deck: The Edge-Deck of G, ED(G), is the collection of all Edge-Cards of G. Note this is in general a multiset.

Graph Isomorphism: Let V(G) be the vertex set of a simple graph and E(G) its edge set. Then a graph isomorphism from a simple graph G to a simple graph H is a bijection f: V(G) →V(H) such that u v Є E(G) iff f(u) f(v) Є E(H) (West 2000, p. 7). If there is a graph isomorphism for G to H, then G is said to be isomorphic to H, written G ≈ H.

2.2. Notations

Our alphabet set is Σ= {0, 1}. We use {., ., ., } to denote sets and [:; : : : ; :] to denote multiset. We use U to denote set union as well as multiset union. We consider only finite, undirected graphs with no self-loops. Given a graph G, let V(G) denote its vertex set and let E(G) denote its edge set. For notational convenience, we sometimes represent a graph G by (V; E), where V=V(G) and E=E(G). By the order of a graph G we mean | V(G) |, i.e., the cardinality of its vertex set.

3. MATCHING POLYNOMIAL AND RECONSTRUCTION CONJECTURE

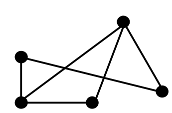
3.1. Matching Polynomial [5, 6]

Matching: A matching cover (or simply a matching) in a graph G is taken to be a subgraph of G consisting of disjoint (independent) edges of G, together with the remaining nodes of G as (isolated) components.

K–matching: A matching is called a K–matching if it contains exactly K edges.

Matching Polynomial: If G contains P nodes, and if a matching contains K edges, then it will have P–2K component nodes. Now assign weights W1 and W2 to each node and edge of G, respectively. Take the weight of a matching to be the product of the weights of all its components. Then the weight of a K–matching will be $W_1^{P-2K} W_2^{K}$. The matching polynomial of G, denoted by m(G), is the sum of the weights of all the matchings in G. The matching polynomial of G has been defined as $m(G)=\Sigma\, a_k W_1^{P-2K} W_2^{K}$. a_k is the number of matchings in G with k edges.

Example.



0 – matching = $1\, W_1^{5}\, W_2^{0}$
1 – matching = $6\, W_1^{3}\, W_2^{1}$
2 – matching = $6\, W_1^{1}\, W_2^{2}$
No 3–matching.

$$m(G) = W_1^{5} + 6\, W_1^{3}\, W_2 + 6\, W_1 W_2$$

Perfect Matching: A perfect matching is a matching of a graph containing n/2 edges, the largest possible. Perfect matchings are therefore only possible on graphs with an even number of vertices. We denote the number of perfect matchings in G by δ(G). Clearly δ(G) is the coefficient of the term independent of W_1 in m(G).

3.2. Matching Polynomial generation using Tree Decomposition

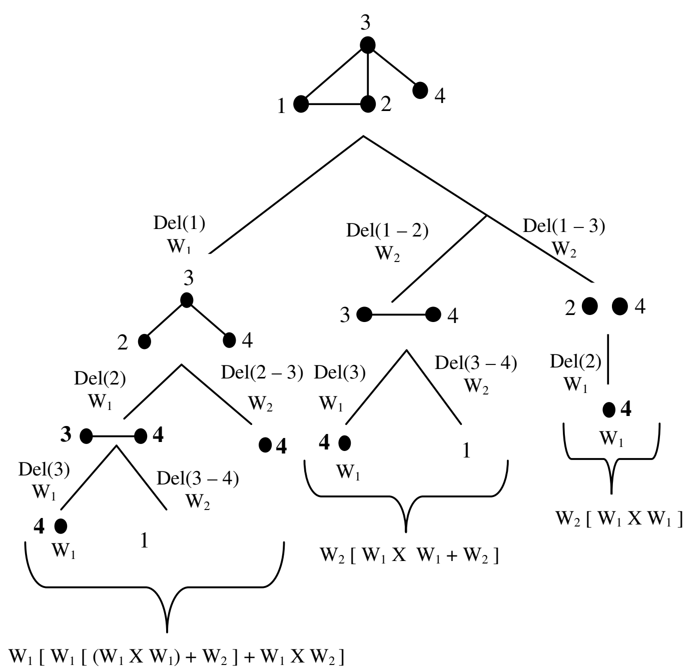
Algorithm: gen_mpoly(Graph G)
Input: A simple connected graph G.
Output: The Matching Polynomial m(G) of this graph.

Steps:

1. if | V(G) | = 1 then
   return (create_mpoly("W_1"));
2. else if | V(G) | = 0 then
   return (create_mpoly("1"));
3. else
   return (create_mpoly(W_1 + gen_mpoly(G – V_i) + $W_2 X \sum_{e : V_i V_j \in E(G)}$ gen_mpoly(G – e)));
   end if.
4. End

[*create_mpoly():* creates a matching polynomial from the sub matching polynomials of the vertex deleted and edge deleted subgraphs matching polynomial.]

Example.



$$m(G) = W_1[W_1[(W_1 X W_1)+W_2]+W_1 X W_2]+W_2[W_1 X W_1+W_2]+W_2[W_1 X W_1]$$
$$= W_1^4+4W_1^2 W_2+W_2^2$$

3.3. Reconstruction of Matching Polynomial

In order to establish our main result, we will need the following Theorems. Let G be a graph with p nodes and q edges. Then

Theorem 3.3.1 [5] The matching polynomial is node-reconstructible

$$m(G) = \sum_{i=1} \int m(G – V_i) \, dW_1 + \delta(G)$$

Theorem 3.3.2 [5] The matching polynomial is edge-reconstructible

$$\sum^{[p/2]} a_k(q – k) \, W_1^{P – 2k} W_2^{k} = \sum^{q} m(G – e_j).$$

k=0 j=1

By comparing coefficients, a_k could be found for all values of k. Hence m(G) could be found.

4. NODE RECONSTRUCTION OF A GRAPH

4.1. Algorithm

Input: Vertex-Deck $VD=[H_1;H_2;H_3;\cdots;H_k]$ (multiset of subgraphs produced by deleting each vertex of the original graph).
Output: $S=\{G_1,G_2,G_3,\cdots,G_n\}$ where $n\geq 1$ and $G_1\approx G_2\approx G_3\approx\cdot\cdot.\approx G_n$ [≈ : Isomorphic].

Steps:

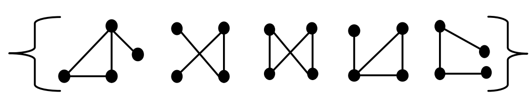
1. Reconstruct the matching polynomial [m1(G)] of the graph from the given Vertex-Deck VD using Theorem 3.31.
2. Select any Vertex-Card [vertex deleted sub graph of the original graph] H_i from VD.
3. Add a new vertex $V_{\mathbf{K}}$ to H_i and connect it with any vertex of H_i by a tentative edge, obtaining a graph $G_{tentative}$ on K vertices.
4. for $i\leftarrow 1$ to $k - 1$ do

   label the tentative edge between i and $V_{\mathbf{K}}$ as X_{iK}.
5. Generate the 2nd matching polynomial[m2(G)] with variable coefficient using the *Tree Decomposition* method [m2(G)= gen_mpoly($G_{tentative}$)].
6. Compare coefficients of m1(G) and m2(G) and generate one or more solutions. Store the solutions in solution vectors $Soln_1$, $Soln_2$, ··········, $Soln_n$. In each solution vector a variable coefficient X_{ij} can have only two values 0 or 1. 0 means the tentative edge will be deleted from $G_{tentative}$, and 1 means the tentative edge will be converted in to a permanent edge. Also calculate the Perfect Matching by substituting the solutions into m2(G).
7. for every solution [$Soln_{1\ to\ n}$] generate a new graph $G_i[i \leq n]$ from the $G_{tentative}$ according to step6 and add the generated graph in to the output set S [$S=S\cup\{G_i\}$].

All of the graphs in the set S are isomorphic. So, select any one of them and this is the reconstructed graph isomorphic to the original graph.

4.2. Example

Input:

Vertex-Deck *VD* = 

Step 1:[Reconstruct the matching polynomial [m1(G)] of the graph from the given Vertex-Deck VD using Theorem 1.]

$m(H_1) = W_1^4 + 4 W_1^2 W_2 + W_2^2$ $\int m(H_1) = W_1^5 / 5 + 4/3 W_1^3 W_2 + W_1 W_2^2$

$m(H_2) = W_1^4 + 3 W_1^2 W_2 + W_2^2$ $\int m(H_2) = W_1^5 / 5 + 3/3 W_1^3 W_2 + W_1 W_2^2$

$m(H_3) = W_1^4 + 4 W_1^2 W_2 + 2W_2^2$ $\int m(H_3) = W_1^5 / 5 + 4/3 W_1^3 W_2 + 2 W_1 W_2^2$

$m(H_4) = W_1^4 + 4 W_1^2 W_2 + W_2^2$ $\int m(H_4) = W_1^5 / 5 + 4/3 W_1^3 W_2 + W_1 W_2^2$

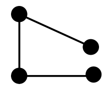
$m(H_5) = W_1^4 + 3 W_1^2 W_2 + W_2^2$ $\int m(H_5) = W_1^5 / 5 + 3/3 W_1^3 W_2 + W_1 W_2^2$

Hence the matching polynomial of G is

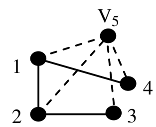
$$m1(G) = W_1^5 + 6 W_1^3 W_2 + 6 W_1 W_2^2 + \delta(G) \quad [\delta(G) : \text{Perfect Matching}]$$

Step 2: [Select any Vertex-Card (vertex deleted sub graph of the original graph) H_i from the deck D.]

Here we select H_5

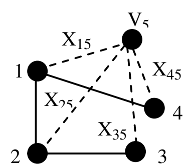


Step 3: [Add a new vertex V_K to H_i and connect it with any vertex of H_i by a tentative edge, obtaining a graph $G_{tentative}$ on K vertices.]

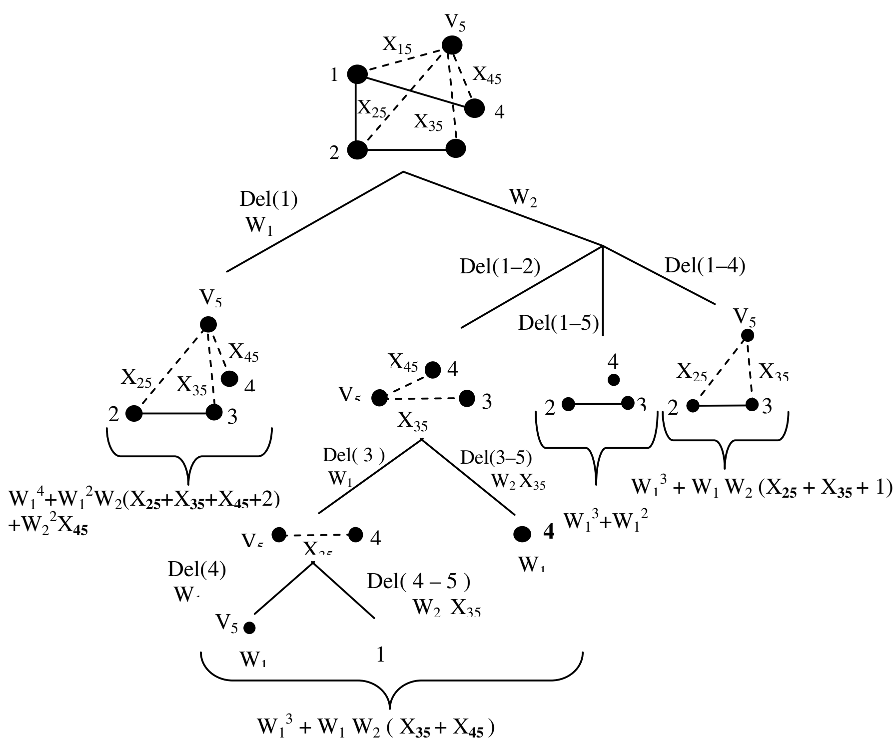


Step 4: [for i ←1 to k – 1 do

label the tentative edge between i and V_K as X_{iK}.]



Step 5: [Generate the 2nd matching polynomial [m2(G)] with variable coefficient using the *Tree Decomposition* method [m2(G) = gen_mpoly($G_{tentative}$)].]



$$m2(G) = W_1^5+W_1^3W_2(X_{15}+X_{25}+X_{35}+X_{45}+3)+W_1W_2^2(X_{15}+X_{25}+2X_{35}+2X_{45}+1)$$

Step 6 : [Compare coefficients of m1(G) and m2(G) and generate one or more solutions. Store the solutions in solution vectors $Soln_1, Soln_2,\cdots,Soln_n$.]

by comparing coefficients of m1(G) and m2(G) we get

$$X_{15} + X_{25} + X_{35} + X_{45} = 3 \qquad X_{15} + X_{25} + 2X_{35} + 2X_{45} = 5$$

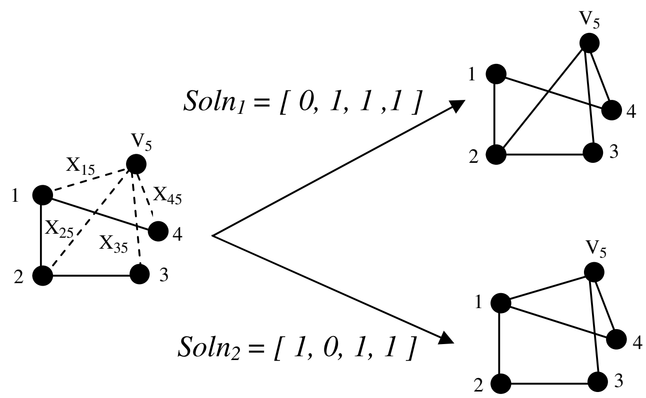
It is clear that the only solutions to these equations are

$Soln_1 = [0, 1, 1 ,1]$ and $Soln_2 = [1, 0, 1, 1]$

by substituting into m2(G) we get in both cases δ(G) = 0.

Step 7: [for every solution [$Soln_{1\ to\ n}$] generate a new graph $G_i[i \leq n]$ from the $G_{tentative}$ according to step6 and add the generated graph in to the output set S [$S=S\cup\{G_i\}$]].

We can now use $Soln_1$ to constructed a graph G_1 and $Soln_2$ to constructed a graph G_2 from $G_{tentative}$.



It can be easily confirmed that the mapping defined by φ:G1 → G2 such that

φ(1) ≡ 3; φ(2) ≡ 5; φ(3) ≡ 4; φ(4) ≡ 2 and φ(5) ≡ 1 is an isomorphism. Hence GI≈G2.

5. EDGE RECONSTRUCTION OF A GRAPH

5.1. Algorithm

Input: Edge-Deck $ED=[H_1;H_2;H_3;\cdots;H_k]$ (multiset of subgraphs produced by deleting each edge of the original graph).

Output: $S=\{G_1,G_2,G_3,\cdots,G_n\}$ where $n\geq1$ and $G_1\approx G_2\approx G_3\approx\cdots\approx G_n$ [≈ : Isomorphic].

Steps:

1. Reconstruct the matching polynomial [m1(G)] of the graph from the given Edge-Deck ED using Theorem 3.3.2.
2. Select any Edge-Card [edge deleted sub graph of the original graph] H_i from the Edge-Deck ED and use positive integers $1\leq i \leq |V|$ for labelling the vertices of the graph.
3. for each vertex $V_i \in V(G)$ do

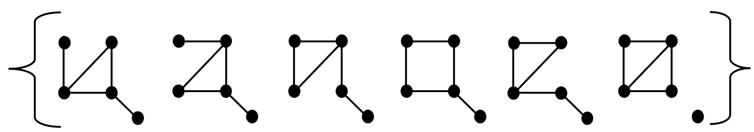
4. for each vertex $V_j \in \{V(G) - V_i\}$ do
5. connect Vi and Vj with a tentative edge and label it $X_{\mathbf{ij}}$ (j > i) and generate a new graph GT_i
6. Generate the [2nd] matching polynomial[m2(G)] with variable coefficient from GT_i using the *Tree Decomposition* method [m2(G)= gen_mpoly(GT_i)].
7. Compare coefficients of m1(G) and m2(G) and generate one or more solutions and Store the solutions in solution vectors $Soln_1$, $Soln_2$, ·········· $Soln_n$.
8. If X_{ij} has a value > 1 or there is a conflict between the values of X_{ij} then go to Step3.
9. Else for every solution [$Soln_{1\ to\ n}$] generate a new graph $Gi_{\mathbf{k}}[k \le n]$ from the GT_i using the same technique as used in the *Node Reconstruction*; and add the generated graph in to the output set S [S=SU{ $Gi_{\mathbf{k}}$ }].

All of the graphs in the set S are isomorphic. So, select any one of them and this is the reconstructed graph isomorphic to the original graph.

5.2. Example

Input:

Edge-Deck *ED* =



Step 1:[Reconstruct the matching polynomial [m1(G)] of the graph from the given Edge-Deck ED by using Theorem 3.3.2]

$m(H_1) = W_1^5 + 5 W_1^3 W_2 + 3 W_1 W_2^2$ $m(H_2) = W_1^5 + 5 W_1^3 W_2 + 3 W_1 W_2^2$

$m(H_3) = W_1^5 + 5 W_1^3 W_2 + 4 W_1 W_2^2$ $m(H_4) = W_1^5 + 5 W_1^3 W_2 + 4 W_1 W_2^2$

$m(H_5) = W_1^5 + 5 W_1^3 W_2 + 4 W_1 W_2^2$ $m(H_6) = W_1^5 + 5 W_1^3 W_2 + 2 W_1 W_2^2$

$$\sum_{k=1}^{6} m(G_i) = 6 W_1^5 + 30 W_1^3 W_2 + 20 W_1 W_2^2$$

From Theorem 3.3.2, we get

$$\sum_{k=0}^{3} a_k (6 - k) W_1^{6-2k} W_2^k = 6 W_1^5 + 30 W_1^3 W_2 + 20 W_1 W_2^2$$

By comparing coefficients in this equation, we get

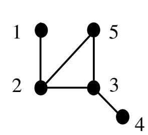
$a_0 = 1,\ \ a_1 = 6,\ \ a_2 = 5$

Hence the matching polynomial of G is

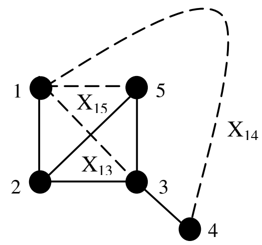
$$m1(G) = W_1^5 + 6 W_1^3 W_2 + 5 W_1 W_2^2$$

Step 2: [Select any Edge-Card [edge deleted sub graph of the original graph] Hi from the Edge-Deck ED and use positive integers 1≤ i ≤ |V| for labelling the vertices of the graph.]

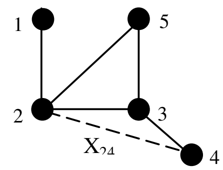
Here we select H_1



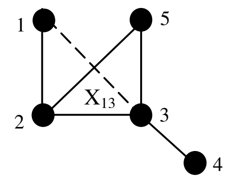
Step 3-5: [for each vertex Vi Є V(G) do
for each vertex Vj Є {V(G) – Vi } do
connect Vi and Vj with a tentative edge and label it Xij (j > i) and generate a new graph GTi]



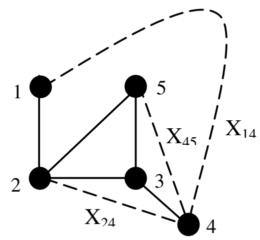
GT_1 Vertex 1 selected



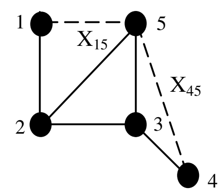
GT_2 Vertex 2 selected



GT_3 Vertex 3 selected



GT_4Vertex 4 selected



GT_5Vertex 5 selected

Step 6: [Generate the 2nd matching polynomial [m2(G)] with variable coefficient using the *Tree Decomposition* method [m2(G) = gen_mpoly(GT_i)].]

$m2(GT_1) = W_1^5 + (5 + X_{13} + X_{14} + X_{15}) W_1^3 W_2 + (3 + X_{13} + 3X_{14} + 2X_{15}) W_1 W_2^2$

$m2(GT_2) = W_1^5 + (5 + X_{13} + X_{14} + X_{15}) W_1^3 W_2 + (3 + X_{13} + 3X_{14} + 2X_{15}) W_1 W_2^2$

$m2(GT_3) = W_1^5 + (5 + X_{13}) W_1^3 W_2 + (3 + X_{13}) W_1 W_2^2$

$m2(GT_4) = W_1^5 + (6 + X_{13} + X_{14} + X_{15}) W_1^3 W_2 + (3 + X_{13} + 3X_{14} + 2X_{15}) W_1 W_2^2$

$m2(GT_5) = W_1^5 + (5 + X_{15} + X_{45}) W_1^3 W_2 + (3 + 2X_{15} + 2X_{45}) W_1 W_2^2$

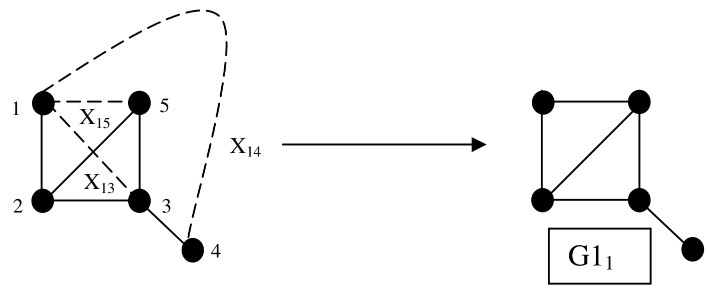
Step 7-9: [Compare coefficients of m1(G) and m2(G) and generate one or more solutions and Store the solutions in solution vectors $Soln_1$, $Soln_2$,···, $Soln_n$.
If X_{ij} has a value > 1 or there is a conflict between the values of X_{ij} then go to Step3.
Else for every solution [Soln1 to n] generate a new graph Gik[k ≤ n] from the $GT_{\mathbf{i}}$;
and add the generated graph in to the output set S [S=SU{ $Gi_{\mathbf{k}}$ }].]

For GT_1 : By comparing coefficients of m1(G) and m2(GT_1), we get

$X_{13} + X_{14} + X_{15} = 1 \qquad X_{13} + 3X_{14} + 2X_{15} = 2$

It is clear that the only solution to these equations is : $Soln_1$ = [0, 0, 1]

We can now use $Soln_1$ to constructed a graph G_1 from GT_1.



For GT_2 : By comparing coefficients of m1(G) and m2(GT_2), we get

$X_{24} = 1$ $X_{24} = 2$

It is clear that the values of X_{24} is conflicting, so, no graph will be produced.

For GT_3 : By comparing coefficients of m1(G) and m2(GT_3), we get

$X_{13} = 1$ $X_{13} = 2$

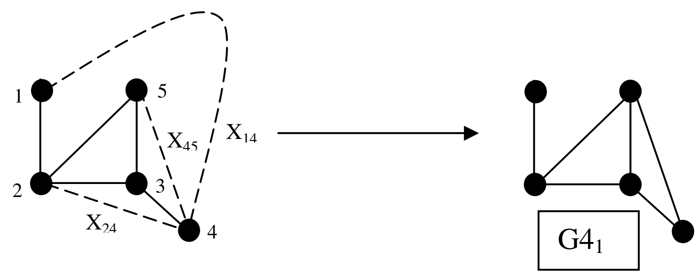
It is clear that the values of X_{24} is conflicting, so, no graph will be produced.

For GT_4 : By comparing coefficients of m1(G) and m2(GT_4), we get

$X_{14} + X_{24} + X_{45} = 1$ $3X_{14} + X_{24} + 2X_{45} = 2$

It is clear that the only solution to these equations is : $Soln_1$ = [0, 0, 1].

We can now use $Soln_1$ to constructed a graph G_4 from GT_4.

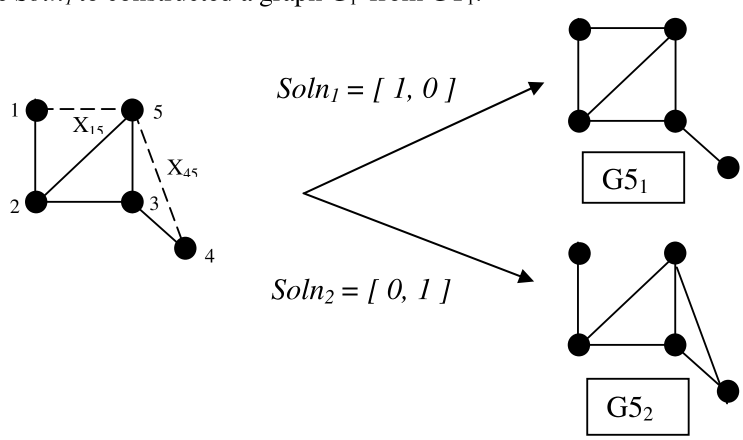


For GT_5 : By comparing coefficients of m1(G) and m2(GT_5), we get

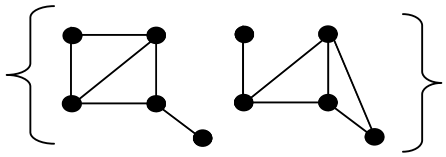
$X_{15} + X_{45} = 1$ $2X_{15} + 2X_{45} = 2$

It is clear that the only solutions to these equations are : $Soln_1$ = [1, 0] and $Soln_2$ = [0, 1]

We can now use $Soln_1$ to constructed a graph G_1 from GT_1.



Output Set S =



It can be easily confirmed that the graphs in the output set are Isomorphic.

6. ANALYSIS

On arbitrary graphs or even planar graphs, computing the matching polynomial is #P–Complete(Jerrum 1987)[11].

But using Tree Decomposition we can compute the matching polynomials for all the subgraphs for the given deck in polynomial time.

Tree Decomposition contains recursive calls to itself, its running time can often be described by a recurrence. The recurrence for Tree Decomposition –

$$T(|V|) = \begin{cases} O(1) & \text{if } |V| \leq 1 \\ T(|V| - V_i) + \sum\limits_{V_iV_j \in E(G)} T(|V| - V_iV_j) & \text{otherwise [for } i \leftarrow 1 \text{ to } n-1] \end{cases}$$

$T(|V|) = O(|V|^2)$

6.1. Analysis of NODE RECONSTRUCTION Algorithm

Step 1 requires $\Theta(|V|^3)$ time (If we use the Tree Decomposition technique to generate the matching polynomials of the subgraphs) . Step 2 & 3 require $\Theta(1)$ time. Step 4 requires $\Theta(|V|)$ times. Step 5 (Tree Decomposition) requires $O(|V|^2)$ time. Step 6 requires exponential time $\Theta(2^{|V|-1})$, and Step 7 requires $\Theta(|V|)$ times.
So, if we can find a polynomial time algorithm for solving this Underdetermined system in Step6 then we can easily reconstruct the original graph from the node deleted subgraphs in polynomial time.

6.2. Analysis of EDGE RECONSTRUCTION Algorithm

This also requires exponential time $O(2^{\Theta(|V|)})$, because to solve the Underdetermined system in step7 requires exponential time. Otherwise all other steps require polynomial time. So, we can easily reconstruct the original graph from the edge deleted subgraphs in polynomial time if we find a polynomial time algorithm for solving this Underdetermined system.

7. CONCLUSIONS

As presented in this paper, if G is a simple undirected graph with at least three vertices or at least four edges then we can use the proposed algorithms for reconstruct G from its vertex-deleted or edge-deleted subgraphs. Although the proposed algorithm is not tested for all classes of graph, but we can conclude that suppose that a graph G is characterized by a particular matching polynomial and suppose that the matching polynomial is reconstructible, then G is reconstructible.

ACKNOWLEDGEMENTS

The authors wish to thank the reviewers for their constructive and helpful comments and also the Computer without which no work was possible.

REFERENCES

[1] Banerjee, S., Sensarma, D., Basuli, K., Naskar, S. & Sarma, S. S. (2011) “The Reconstruction Conjecture”, The Second International Conference on Computer Science and Information Technology, Vol. 86, Part III, pp 17-25.

[2] Carrillo, H. & Lipman, D. (1988) “The multiple sequence alignment problem in biology”, SIAM Journal of Applied Mathematics, Vol. 48, No.5, pp 1073 – 1082.

[3] Clark, J. & Holton, D. A. (1995) “A First Look At Graph Theory”. World Scientific Publishing Company.

[4] Cormen, T. H., Leiserson, C. E. & Rivest, R. L., Stein, C. (2010) “Introduction to Algorithms”. 3rd ed. Phi Learning.

[5] Farrell, E.J. &Wahid, S.A. (1987) “On the reconstruction of the matching polynomial and the reconstruction conjecture”. Int. J. Math. Math. Sci. Vol. 10, pp 155–162.

[6] Farrell, E. J. (1979) “Introduction to Matching Polynomials”, Journal of Combinatorial Theory, Vol. 27, pp 75-86.

[7] Gutman, I. & Cvetkovic, D.M. (1975) “The Reconstruction Problem for Characteristic Polynomials of Graphs”, Univ. Beograd. Elektrotehn. Fak. Ser. Mat. Fiz, pp 45-48.

[8] Harary, F.(1964) “On the reconstruction of a graph from a collection of subgraphs”, Theory of Graphs and its applications (M. Fiedler,ed) Prague, pp 47–52.

[9] Harary, F.(1974) “A Survey of the Reconstruction Conjecture in Graphs and Combinatorics”, Lecture Notes in Mathematics, Vol. 406, pp 18-28.

[10] Hemaspaandraa, E., Hemaspaandrab, L. A., Radziszowskia & S. P., Tripathib, R.(2004) “Complexity results in graph reconstruction”, Discrete Appl math, Vol. 155, pp 103-118.

[11] Jerrum, Mark. (1987) “Two-dimensional monomer-Dimer systems are computationally intractable”, Journal of Statistical Physics, Vol.48, No.1, pp 121–134.

[12] Kelly, P. J.(1942) “On isometric transformations”, Ph.D. thesis, University of Wisconsin, USA.

[13] Manvel, B. (1988) "Reconstruction of graphs – Progress and prospects. Congressus Numer, Vol. 63, pp 177–187.

[14] Nash-Williams & C. St. J. A.(1978) “The reconstruction problem, in Selected topics in graph theory”, Academic Press, pp. 205–236.

[15] Ramachandran, S. (2004) “Graph Reconstruction - Some New Developments”. AKCE J. Graphs. Combin., Vol. 1, No. 1, pp 51-61.

[16] Sensarma, D., Banerjee, S., Basuli, K., Naskar, S., & Sarma, S. S. (2011) “Minimizing Boolean Sum of Products Functions Using Binary Decision Diagram”, The Second International Conference on Computer Science and Information Technology , Vol. 86, Part III, pp 36-48.

[17] Tutte, W. T. (1979) “All the King's Horses”, In Graph Theory and Related Topics, Academic Press, pp15-33.

[18] Ulam, S. (1960) “A Collection of Mathematical Problems”. Interscience Publishers, New York, NY, USA.

[19] Yongzhi, Y. (2006) “The reconstruction conjecture is true if all 2-connected graphs are reconstructible”. Journal of Graph Theory, Vol. 12, No. 2, pp 237–243.

Authors

Subhashis Banerjee

M.Sc. Computer Science

Debajit Sensarma

M. Sc. Computer Science

Krishnendu Basuli

Assistant Professor,
Department of Computer Science,
West Bengal State University.

Saptarshi Naskar

Assistant Professor,
Department of Computer Science,
Sarsuna College, West Bengal, India.

Dr. Samar Sen Sarma

Professor, Department of Computer Science and
Engineering, University Of Calcutta, India