International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Chromatic Polynomial of Domination Subdivision Non Stable Graphs

A. Elakkiya, M. Yamuna

***Abstract:** A graph G is said to be domination subdivision non stable (DSNS) if $\gamma (G_{sd}uv) = \gamma (G) + 1$ for all u, $v \in V (G)$, u adjacent to v. In this paper, we provide a method of determining the chromatic polynomial of DSNS graph from G.*

***Keywords:** Chromatic Polynomial, Complement of a Graph, Edge Contraction.*

I. INTRODUCTION

In [1], Shubo Chen has investigated absolute sum of chromatic polynomial coefficients of forest, q - tree, unicyclic graphs and quasi wheel graphs. In [2], Rong-xia Hao et al have proposed a new method to calculate the chromatic polynomial of the complements of a wheel and a fan graph. In [3], Matthias Beck et al. have developed and executed a computer program that efficiently determines the number of proper k-colorings for a given signed graph. In [4], M. Yamuna et al have introduced a new class of graph called non - domination subdivision stable graph (NDSS) and obtained if and only if condition for any graph to be NDSS and also provided a constructive characterization of NDSS trees. In [5], M. Yamuna et al have characterized the planarity and outer – planarity of complement of NDSS graphs. In [6], M. Yamuna et al. have determined the domination number of G^*, $\overline{G^*}$, chromatic polynomial of G^*, spanning tree of G^*, number of spanning trees of G^* from G.

A given graph G of n – vertices can be properly colored in many different ways using a sufficiently large number of colors. This property of a graph is expressed elegantly by means of a polynomial. This polynomial is called the chromatic polynomial of G.

The value of the chromatic polynomial $P_n(\lambda)$ of a graph with n - vertices gives the number of ways of properly coloring the graph, using λ or fewer colors [7].

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
A. Elakkiya, School of Advanced Sciences, Vellore Institute of Technology, Vellore, India, Email : elakkiyaappu@gmail.com
M. Yamuna, School of Advanced Sciences, Vellore Institute of Technology, Vellore, India, Email :myamuna@vit.ac.in

II. CHROMATIC POLYNOMIAL OF A DSNS GRAPHS

Domination subdivision non stable graph (DSNS)
A graph G is said to be non domination subdivision stable if $\gamma (G_{sd}uv) = \gamma (G) + 1$ for all u, $v \in V (G)$, u adjacent to v.
Example of DSNS Graphs

1. Path P_{3n} is DSNS.
2. Cycle C_{3n} is DSNS.
3. Complete graph K_n is DSNS.
4. Star graph S_n is DSNS.
5. The graph G in Fig.1 is DSNS.

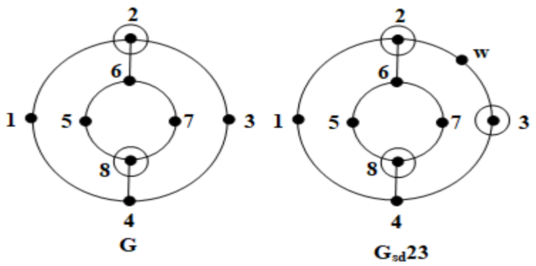


Fig. 1

To determine the chromatic polynomial of G, we use two formulae's

1. $P_n (\lambda)$ of G = $P_n (\lambda)$ of (G + e) + $P_n (\lambda)$ of (G ge)
2. $P_n (\lambda)$ of G = $P_n (\lambda)$ of (G – e) – $P_n (\lambda)$ of (G ge)

In the two formulae's, three graph operations are used namely edge addition, edge deletion and edge contraction. The following discussions and results are for modifying the original method of determining the chromatic polynomial for any graph so that chromatic polynomial of $\overline{G}$ can be determined from G.
Since we plan to modify the three operations, we need to analyze how these operations effect graph $\overline{G}$. For this purpose, let us consider the graph G and its complement $\overline{G}$ in Fig. 1. Throughout the discussion in all the figures in this section G, $\overline{G}$ denotes the graphs in Fig 2.

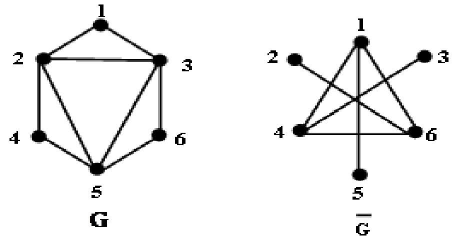


Fig. 2

We know that an edge in G implies no edge in $\overline{G}$ and vice versa. We use this to modify the operations edge addition and edge contraction. Vertices 1 and 2 are adjacent in G. Let us consider the graph $G_1 : \overline{G-\{(12)\}}$. From Fig.3, we observe that G_1 is equivalent to $\overline{G} \cup \{ (1\ 2) \}$. So we conclude that adding an edge between non adjacent vertices in $\overline{G}$ is equivalent to removing the edge between same vertex pair in G.

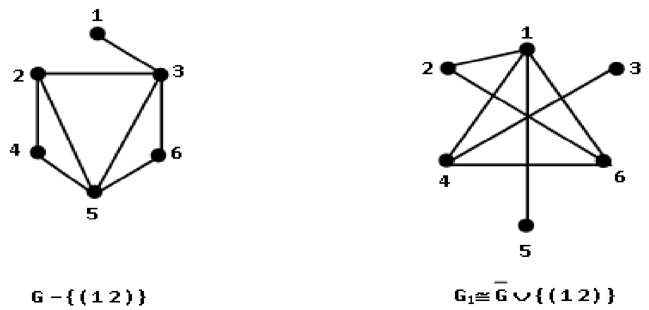


Fig.3

Vertices 1 and 4 are non – adjacent in G. Let us consider the graph $G_2 : \overline{G+\{(14)\}}$. From Fig. 4 we observe that G_2 is isomorphic to $\overline{G} – \{ (1\ 4) \}$.

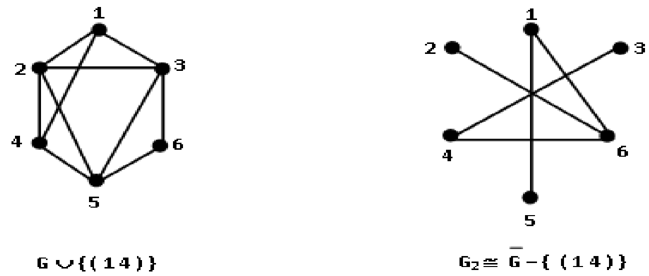


Fig. 4

So we conclude that removing an edge in $\overline{G}$ is equivalent to adding an edge between the same vertices in G. We use these observations for modifying edge addition and edge deletion.

Type – I Operation (Modified G + { e })

Remove an edge e = (u v) in G.

Type – II Operation (Modified G – { e })

Add an edge e = (u v) in G.

One more graph operation used is edge contraction. When two vertices are contracted in $\overline{G}$ a suitable operation should be determined in G. When we merge two vertices u and v adjacency between them does not make any difference in the contraction.

We require $\overline{G \bullet uv}$ to be isomorphic to $\overline{G} \bullet uv$. Let us consider the possible adjacencies in the graphs G, $G \bullet uv$, $\overline{G} \bullet uv$. In graph G, for any two vertices u, v either, u and v are collectively adjacent to a vertex w or both u and v collectively not adjacent to a vertex w. Else u and v are adjacent to different vertices say u adjacent to w_1 and v adjacent to w_2. The possible adjacencies in $\overline{G}$ and $\overline{G} \bullet uv$ is listed in Table – 1.

Table – 1

S.No	G	$G \bullet uv$	$\overline{G \bullet uv}$	$\overline{G} \bullet uv$
1	u, v adjacent to w	uv adjacent to w	uv not adjacent to w	uv not adjacent to w
2	u, v not adjacent to w	uv not adjacent to w	uv adjacent to w	uv adjacent to w
3	u adjacent to w_1, v adjacent to w_2	uv adjacent to w_1w_2	uv not adjacent to w_1w_2	uv adjacent to w_1w_2

From the Table - 1.1 , we observe that when merged vertex uv is adjacent or not adjacent to a vertex w, $\overline{G \bullet uv}$ is isomorphic to $\overline{G} \bullet uv$ .When the merged vertex uv adjacent to w_1, w_2 in $\overline{G \bullet uv}$ is not isomorphic to $\overline{G} \bullet uv$. This is because, when uv is adjacent to w_1, w_2 in $G \bullet uv$ they are not adjacent in $\overline{G \bullet uv}$. In graph $\overline{G}$ u is adjacent to w_2 and v is adjacent to w_1 implies uv adjacent to w_1, w_2. So when u is adjacent to w_1, v is adjacent to w_2 in G, if in $G \bullet uv$, vertex uv not adjacent to w_1, w_2, then $\overline{G \bullet uv}$ is isomorphic to $\overline{G} \bullet uv$. We define this as modified $G \bullet uv$

Type III Operation (Modified $G \bullet uv$)

For a pair of vertices u, v of g denote by modified $G \bullet uv$, the graph obtained from G by deleting the vertices u and v and appending a new vertex denoted by uv that is, adjacent only to those vertices of G – u – v that where originally collectively adjacent to both u and v.

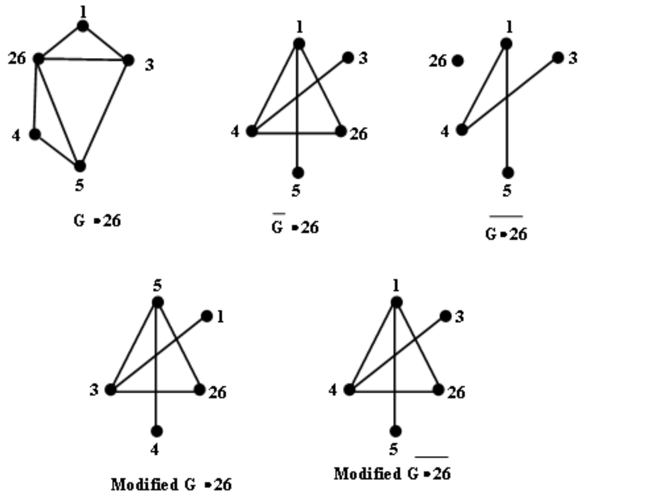


Fig.5

For the graph in Fig. 5, we see that $\overline{G \bullet 26}$ is not isomorphic to $\overline{G} \bullet 26$ whereas modified $\overline{G \bullet 26}$ is isomorphic to $\overline{G} \bullet 26$.

Complete Graph Identification

By recursive applying Type I, III operations on G, we reach a graph H such that H is a null graph. This means that $\overline{H}$ is a complete graph.

Null graph Identification

By recursive applying Type II, III operations on G, we reach a graph H such that H is a complete graph. This means that $\overline{H}$ is a null graph.

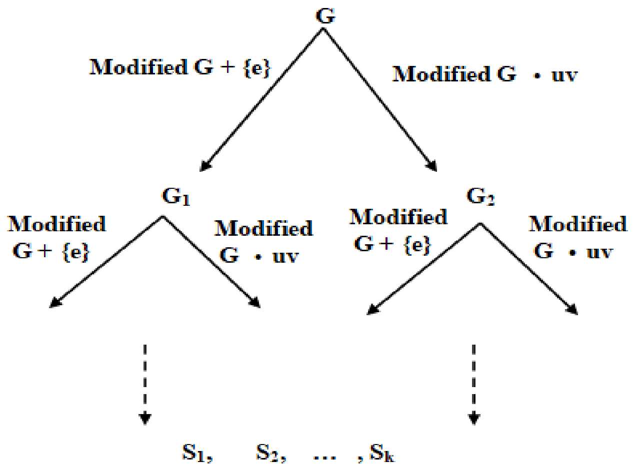
A. Chromatic Polynomial of $\overline{G}$ using Edge Deletion and Edge Contraction

To determine the chromatic polynomial of G, we know that P_n (λ) of G = P_n (λ) (G + e) + P_n (λ) ($G \bullet e$).

In this case, we terminate the procedure when every possible graph is a complete graph. Consider graph G by applying modified $G + e$ and $G \bullet uv$ on G. We generate two graphs G_1 and G_2. By the above discussion, we know that $\overline{G}$ is equal to $\overline{G+e}$ and $\overline{G_2}$ is equal to $\overline{G \bullet uv}$. We continue this procedure recursively on G_1 and G_2 and on the remaining graphs generated until we cannot continue any further. This means that we have a sequence of graphs $S_1, S_2, \ldots, S_k$ such that each S_i, i = 1 to k is a null graph implies $\overline{S_i}$ is a complete graph for every i = 1 to k. Chromatic polynomial of $\overline{S_i}$ =

$$P_n(\lambda)(\overline{S_i}) = \prod_{k=0}^{n-1}(\lambda - k)$$ if S_i is a null graph with n vertices.

$$P_n(\lambda)(\overline{G}) = \sum_{i=1}^{k} P_n(\lambda)(\overline{S_i}) = \sum_{i=1}^{k} \lambda(\lambda-1)\ldots(\lambda-n_i+1)$$ where n_i denotes the number of vertices in graph S_i, i = 1, 2, …, k. This procedure is summarized in Hierarchy – 1.



Hierarchy – 1

From modified $G + e$ and $G \bullet uv$, we understand that for every pair of adjacent vertices (non – adjacent in $\overline{G}$)

- $\overline{G+e} = \overline{G} - e$ and
- $\overline{G \bullet uv} = \overline{G} \bullet uv$

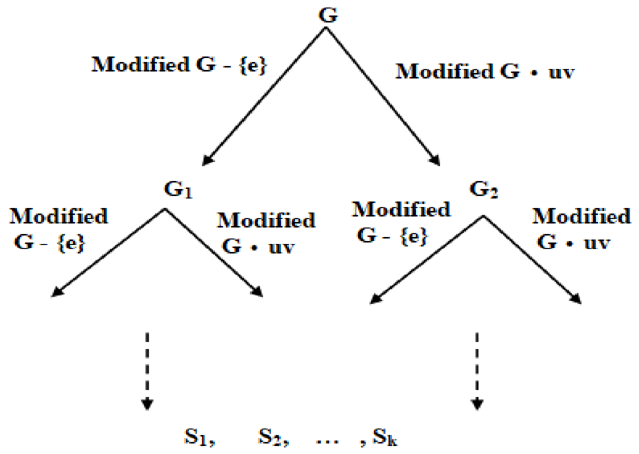
The number of operations involved in determining the chromatic polynomial of $\overline{G}$ from G is equal to the number of operations involved in determining the same from $\overline{G}$, since the operations are repeated for every adjacent pair in G. Fig 5 provides an example of determining the chromatic polynomial of $\overline{G}$ from graph G. In all the iterations blue colour edge indicates the edge chosen for applying the operations.

B. Chromatic Polynomial of $\overline{G}$ using Edge Addition and Edge Contraction

We know that chromatic polynomial of a graph can also be determined by edge deletion and edge contraction. In this case, we terminate the procedure when every possible graph is a null graph. Similar to the discussion in above section, $\overline{G_1}$ is equal to $\overline{G} - e$ and $\overline{G_2}$ is equal to $\overline{G} \bullet uv$. As in above section 2.4.1, we continue the procedure recursively on G_1 and G_2 until we generate a sequence of graphs $S_1, S_2, \ldots S_k$ such that each S_i, i = 1 to k is a complete graph implies $\overline{S_i}$ are null graphs for all i = 1 to k. Chromatic polynomial of $\overline{S_i} = \lambda^n$ if S_i is a graph with n vertices. $P_n(\lambda)(\overline{G})$

$$(\overline{G})\sum_{i=1}^{k_1} P_n(\lambda)(\overline{S_i}) - \sum_{j=1}^{k_2} P_n(\lambda)(\overline{S_j}) = \sum_{i=1}^{k_1}\lambda^{n_i} - \sum_{j=1}^{k_2}\lambda^{n_j}, k_1 + k_2 = k$$

where n_i and n_j denotes the number of vertices in graphs S_i and S_j respectively, i = 1 to k_1, j = 1 to k_2. This procedure is summarized in Hierarchy – 2.



Hierarchy – 2

Fig. 7 provides an example of determining the chromatic polynomial of $\overline{G}$ from graph G. In all the iterations blue colour edge indicates the newly added edges in the current operations.

C. Chromatic Polynomial of $G_{sd}uv$ and $\overline{G_{sd}uv}$ of DSNS Graphs

In this section, we provide a method of determining the chromatic polynomial of $G_{sd}uv$ and $\overline{G_{sd}uv}$ from graph G. For this purpose, we use edge deletion and edge contraction. Let G_1 and G_2 be the graphs obtained by removing an edge e = (u v) and merging vertices u , v that is $G_1 = G - \{ e \}$, $G_2 = G \bullet uv$.

Consider $G_{sd}uv$. Let w be the subdivided vertex. Let e_1 = (u w), e_2 = (w v). Let us remove edge e_2 and merge vertices w, v. Let $G_3 = G - \{ e_2 \}$and $G_4 = G \bullet wv$. We observe that G_4 is isomorphic to G. Now consider graph G_3. Let us apply edge deletion and edge contraction using e_1. $G_3 - \{ e_1 \} - \{ w \}$ isomorphic to G_1. Also $G_3 \bullet wv$ is isomorphic to G_1.

$P_n(\lambda)(G_4) = P_n(\lambda)(G)$.

$P_n(\lambda)(G_3) - \{ e_1 \} - \{ w \} = P_n(\lambda)(G_3 \bullet uw) = P_n(\lambda)(G_1)$.

We note that w is always an isolated vertex in $G_3 - \{ e_1 \}$. In this procedure of determining chromatic polynomial using formula 2, we know that, we terminate the procedure when all graphs are null graphs.

Since w is an isolated vertex in $G_3 - \{ e_1 \}$ when we try to determine the chromatic polynomial of $G_3 - \{ e_1 \}$ we observe that this is equivalent to determining the chromatic polynomial of G_1 with vertex w. Since we always terminate at a null graph we can conclude that chromatic polynomial of G_3

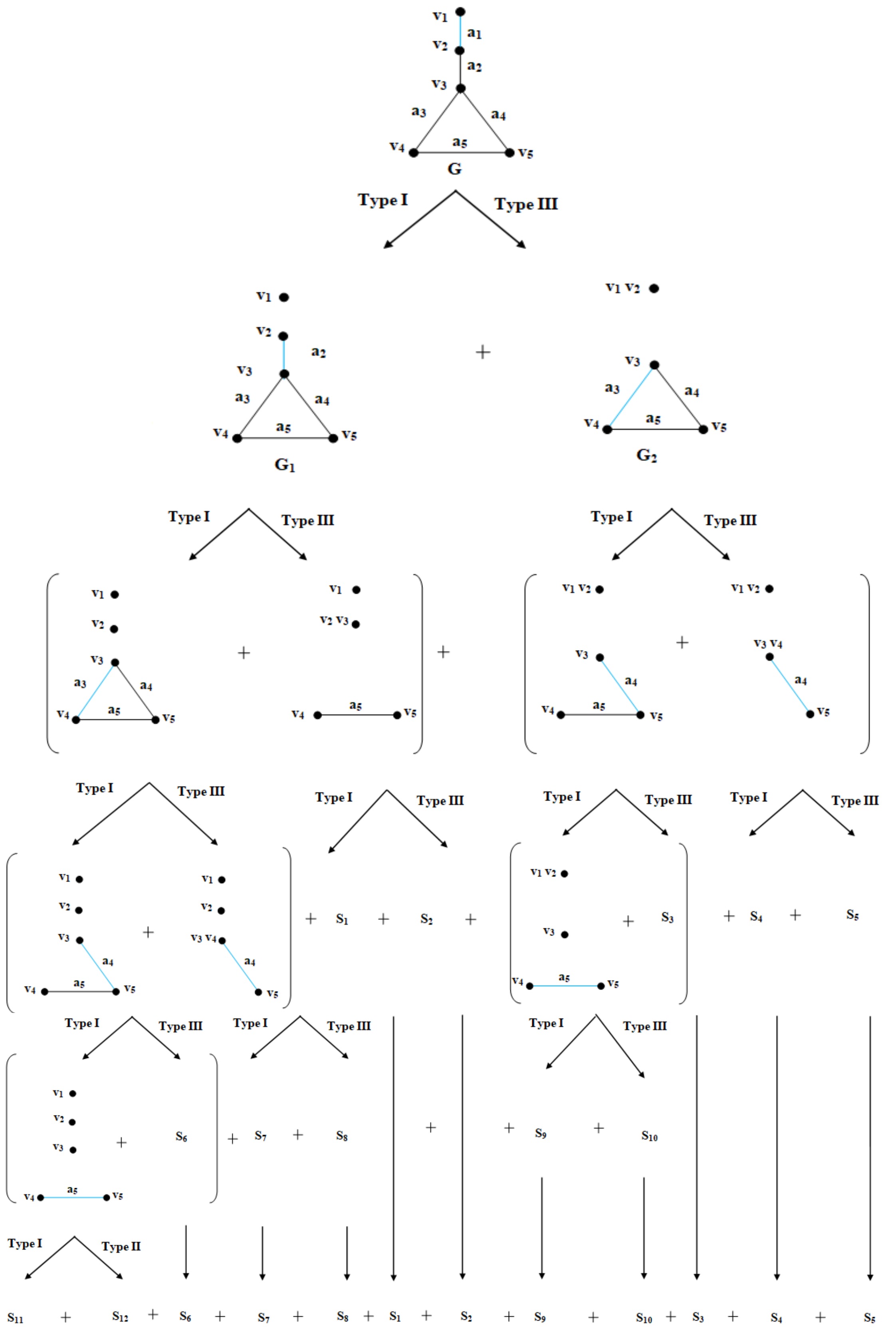


Fig. 6

For the graph in Fig.6 $P_n(\lambda)(\overline{S_i}) = \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3)(\lambda - 4) + \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) + \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) + \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) + \lambda(\lambda - 1)(\lambda - 2) + \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) + \lambda(\lambda - 1)(\lambda - 2) + \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) + \lambda(\lambda - 1)(\lambda - 2) + \lambda(\lambda - 1)(\lambda - 2) + \lambda(\lambda - 1)(\lambda - 2) + \lambda(\lambda - 1)$.

Retrieval Number: K24020981119/2019©BEIESP
DOI: 10.35940/ijitee.K2402.1081219
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

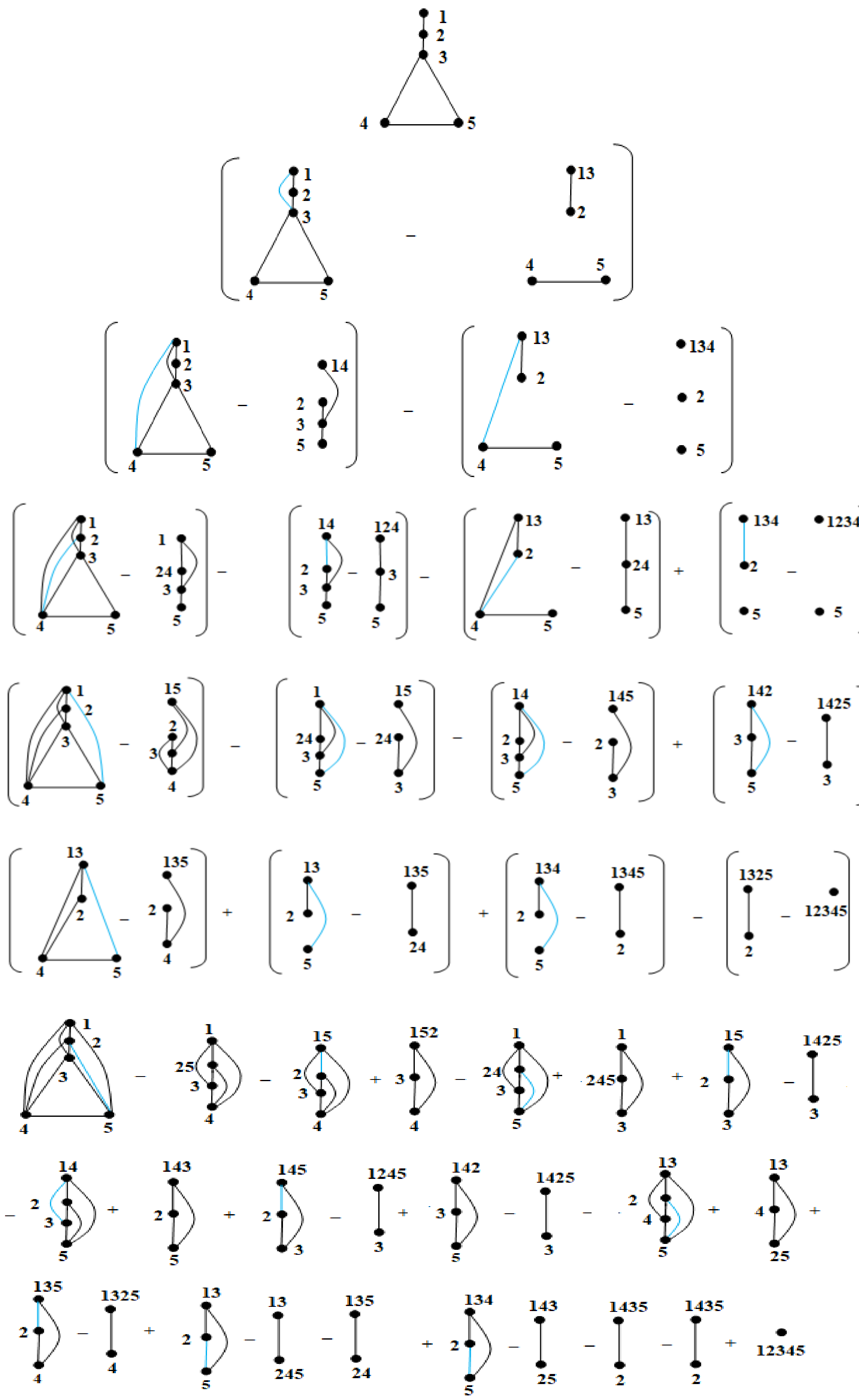


Fig.7

For the graph in Fig. 7 $P_n(\lambda)(\overline{S_i}) = \lambda^5 - \lambda^4 - \lambda^4 + \lambda^3 - \lambda^4 + \lambda^3 + \lambda^3 - \lambda^2 - \lambda^4 + \lambda^3 + \lambda^3 - \lambda^2 + \lambda^3 - \lambda^2 - \lambda^4 + \lambda^3 + \lambda^3 - \lambda^2 + \lambda^3 - \lambda^2 - \lambda^2 + \lambda + \lambda^3 - \lambda^2 - \lambda^2 + \lambda - \lambda^2 + \lambda$.

– { e_1} can be determined from that chromatic polynomial of G_1 with an additional vertex in all the resulting null graphs that is if $N_1, N_2, \ldots, N_k$ are the resulting null graphs obtained to determine the chromatic polynomial of G_1 that is , $P_n(\lambda)(G_1) = P_n(\lambda)(N_1) + P_n(\lambda)(N_2) + \ldots + P_n(\lambda)(N_k)$, then $P_n(\lambda)((G_3) - \{e_1\}) = P_n(\lambda)(M_1) + P_n(\lambda)(M_2) + \ldots + P_n(\lambda)(M_k)$ where each M_i is a null graph such that $|V(M_i)| = |V(N_i)| + 1$ for all i = 1 to k. For this purpose, if $P_n(\lambda)(G)$ denotes the chromatic polynomial of G where all the resulting graphs are null graphs, let us denote the chromatic polynomial of $G \cup \{v\}$ where v is an isolated vertex as $P_n(\lambda)(G \cup \{v\})$. So if $P_n(\lambda)(G) = \sum_{i=1}^{k_1} \lambda^{n_i} - \sum_{j=1}^{k_2} \lambda^{n_j}$, then

$$P_n(\lambda)(G \cup \{v\}) = \sum_{i=1}^{k_1} \lambda^{n_i+1} - \sum_{j=1}^{k_2} \lambda^{n_j+1}$$

where n_i and n_j are the cardinalities of the vertices of the resulting null graphs. So if G is any graph

$$P_n(\lambda)(G_{sd}uv) = P_n(\lambda)(G_1 \cup \{w\}) - P_n(\lambda)(G_1) - P_n(\lambda)(G) \qquad (3)$$

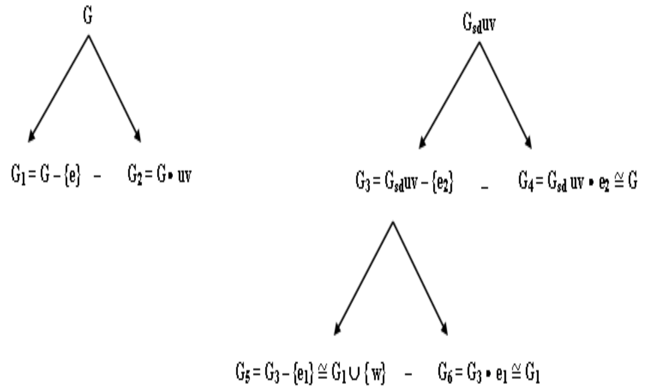
By formula 3 , $P_n(\lambda)(G_{sd}uv)$ involves $P_n(\lambda)(G_1)$ and $P_n(\lambda)(G)$.

$P_n(\lambda)(G) = P_n(\lambda)(G_1) - P_n(\lambda)(G_2)$

implies, $P_n(\lambda)(G_{sd}uv) = P_n(\lambda)(G_1 \cup \{w\}) - P_n(\lambda)(G_1) - P_n(\lambda)(G_1) + P_n(\lambda)(G_2)$.

$$= P_n(\lambda)(G_1 \cup \{w\}) - 2P_n(\lambda)(G_1) + P_n(\lambda)(G_2) \qquad (4)$$

This procedure is summarized in Hierarchy – 3.



Hierarchy – 3

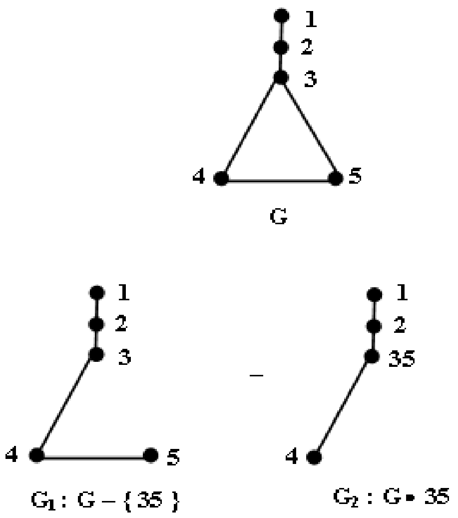


Fig. 8

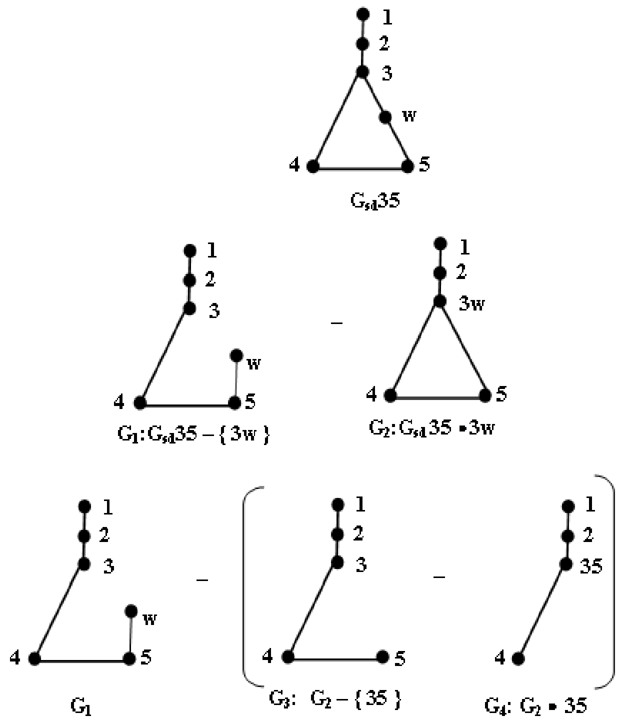


Fig.9

Similarly we can obtain the chromatic polynomial of $\overline{G}_{sd}uv$ by applying modified G + { e } , G – { e } and G guv as already discussed.

When we compare G and $G_{sd}uv$, $G_{sd}uv$ has only 2 additional edges and a vertex extra. When we remove edge e_2, two graphs are generated (labeled as G_3, G_4). G_3 is isomorphic to $G_1 \cup \{e_1\}$ and G_4 is isomorphic to G. So to calculate the chromatic polynomial of $G_{sd}uv$, we should estimate the chromatic polynomial of G_3 and G_4. This means that apart from calculating the chromatic polynomial of G, we should also determine the chromatic polynomial of G_3 also. This requires additional iterations (one complete set of iteration for graph G_3).

The number of iterations to determine $P_n(\lambda)(G_1 \cup \{w\})$ = The number of iterations to determine $P_n(\lambda)(G_1)$. This is true because w is a pendant vertex and the iterations are terminated when all the graphs are null graphs or trees. So determining the chromatic polynomial of $G_1 \cup \{w\}$ can be avoided. It can be generated by following the same iterations as in G_1 along with the pendant vertex w. So these iterations can be avoided if formula 4 is used.

III. NUMBER OF SPANNING TREES OF $G_{SD}UV$

Cayley's formula provides a simple recursive formula for the number of spanning trees in a graph. This procedure involves the operation of edge contraction and edge removal. If $\tau(G)$ denotes the number of spanning trees of G, then $\tau(G) = \tau(G - e) + \tau(G\ g\ e)$. We note that by edge contraction we mean that the graph resulting by merging adjacent vertices. We retain back self loop and parallel edges if any. We use this Cayley's formula to determine the number of spanning trees for $G_{sd}uv$ from graph G. Let G be any graph.

We can determine the number of spanning trees of G using Cayley's method. Let u, v $\in$ E (G). Let G_1 and G_2 be the graphs generated by removing edge (u v) and contracting egde (u v) respectively (retaining back self loops and parallel edges). Let $G_{sd}uv$ = w. Let e_1 = (u w) and e_2 = (w v). Let G_3 and G_4 be the graphs obtained by removing and merging edge e_2 (retaining back self loops and parallel edges) G_4 is isomorphic G implies $\tau(G_4) = \tau(G)$. $G_3 = G \cup \{ w \} \cup \{ e_1 \}$. By Cayley's theorem the procedure to determine all possible spanning trees is terminated when all the remaining graphs are trees. Since G_3 = G_1 union a pendant edge, the recursive procedure adopted for determining the number of spanning trees for graph G_1 can be retained for G_3 also implies

$$\tau(G_{sd}uv) = \tau(G_1) + \tau(G) \quad (5)$$

Remark

To determine the spanning trees of $G_{sd}uv$. We should determine the possible spanning trees G guv and G – (u v). G guv isomorphic to G. So to find the spanning trees of G also. If we us formula (5) we can avoid finding the spanning trees of G – (u v). This method is of more advantage when the number of edges is more in number.

Example

Consider the graph in Fig.10. Let us determine the number of spanning tress of G.

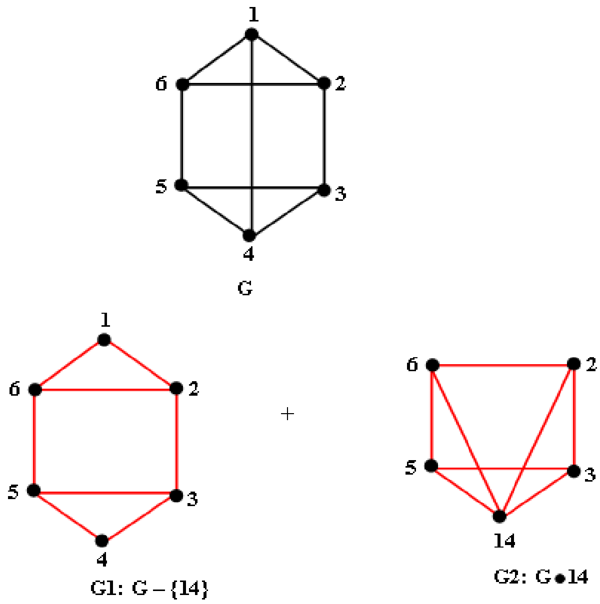


Fig.10

$\tau(G_1) = 52$, $\tau(G_2) = 103$

Consider $G_{sd}14$. As per the formula we know that

$\tau(G_{sd}uv) = \tau(G_1) + \tau(G) = 207$

Consider $G_{sd}14$.

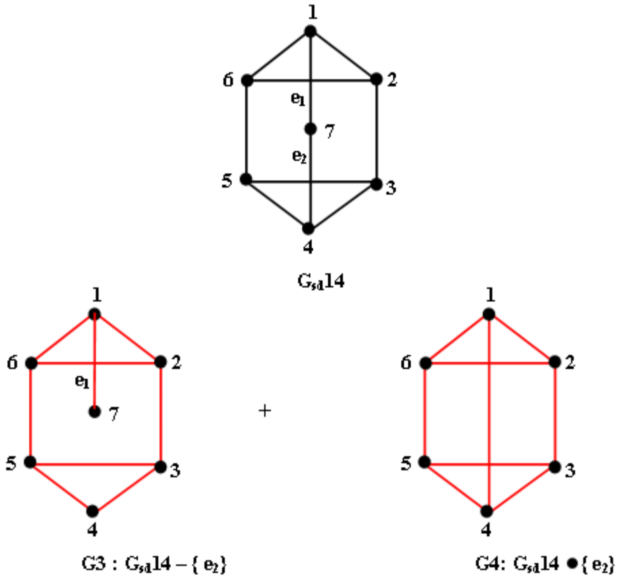


Fig. 11

If we use formula 5 then the additional calculation required to determine the spanning trees of G_3 can be avoided. Formula 5 is a relatively easier one as it reduces the number of calculation. This is of more use if G has more number of edges.

For the graph in Fig. 10 $\tau(G_1) = 51$, $\tau(G_2) = 103$ implies $\tau(G_{sd}12) = 51 + 51 + 103 = 205$ implies estimating 51 graphs can be reduced.

To calculate the number of spanning trees of G_{sd} uv, we should calculate the number of spanning trees of G_1 and G_2 independently. Then τ (G_{sd} uv) can be determined using formula 2. Here we note that G_1 and G_2 are the graphs obtained by deleting and contracting edge uv.

IV. CONCLUSION

In this paper, we have devised a technique of determining the chromatic polynomial of DSNS graph without actual construction of DSNS graph. This result paves way for new method of approaching graph problems.

REFRENCES

1. Shubo Chen, On the absolute sum of chromatic polynomial coefficient of graphs, *International Journal of Mathematics and Mathematical Sciences*, Volume 2011.
2. Rong-xia Hao, Yan-pei Liu, On chromatic polynomials
3. of some kinds of graphs, *Acta Mathematicae Applicatae Sinica, English Series* Vol. 20, No. 2, 2004, 239–246
4. https://arxiv.org/pdf/1311.1760.pdf.
5. Yamuna, M., Elakkiya, A., "Non domination subdivision stable graphs", *IOP Conf. Series: Materials Science and Engineering*. Vol 263, 2017 .
6. Yamuna, M., Elakkiya, A, "Planar graph characterization of NDSS graphs", *IOP Conf. Series: Materials Science and Engineering* ,Vol 263 , 2017.
7. M.Yamuna, K.Karthika, "G^* properties – without G^* construction", *WSEAS Transaction on Mathematics*, Vol 15, 2016, pp 312 – 314.
8. Narsing Deo, Graph Theory with Application to Engineering and Computer Science, Prentice Hall India, 2010.