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 \( \gamma(G_{uv}) = \gamma(G) + 1 \) 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 \( G^* \), \( G^* \), 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 \( P_{\beta}(\lambda) \) of a graph with \( n \) - vertices gives the number of ways of properly coloring the graph, using \( \lambda \) or fewer colors [7].

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 \( \gamma(G_{uv}) = \gamma(G) + 1 \) 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 \( (G-e) \)
2. \( P_n(\lambda) \) of \( G = P_n(\lambda) \) of \( G \) – \( e \) – \( P_n(\lambda) \) of \( (G-e) \)

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 \( G \) can be determined from \( G \).

Since we plan to modify the three operations, we need to analyze how these operations effect graph \( G \). For this purpose, let us consider the graph \( G \) and its complement \( \overrightarrow{G} \) in Fig. 1. Throughout the discussion in all the figures in this section \( G \), \( \overrightarrow{G} \) denotes the graphs in Fig. 2.

Fig. 2

We know that an edge in \( G \) implies no edge in \( \overrightarrow{G} \) 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 \( G_1 : G - \{(1,2)\} \). From Fig.3, we observe that \( G_1 \) is equivalent to \( \overrightarrow{G} \cup \{(1,2)\} \). So we conclude that adding an edge between non adjacent vertices in \( G \) is equivalent to removing the edge between same vertex pair in \( G \).

Revised Manuscript Received on October 05, 2019.

* Correspondence Author

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

ISSN: 2278-3075, Volume-8 Issue-12, October 2019
From Fig. 3, we define this as modified using Edge Deletion and contraction. In Fig. 4, the graph is isomorphic to \( G - \{14\} \). This is because, when \( uv \) is adjacent to \( w \), \( v \) is adjacent to \( w \). When the merged vertex \( uv \) adjacent to \( w \) is in \( G/uv \), vertex \( uv \) not adjacent to \( w \), then \( G/uv \) is isomorphic to \( G - \{14\} \). We define this as modified \( G/uv \).

**Type III Operation (Modified \( G/uv \))**

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

For the graph in Fig. 5, we see that \( G - 26 \) is not isomorphic to \( G \), whereas modified \( G26 \) is isomorphic to \( G \).

### 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 \( H \) 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 \( H \) is a null graph.

#### A. Chromatic Polynomial of \( G \) using Edge Deletion and Edge Contraction

To determine the chromatic polynomial of \( G \), we know that \( P_n(\lambda) \) of \( G = P_n(\lambda) + P_n(\lambda + e) \). In this case, we terminate the procedure when every possible graph is a complete graph. Consider graph \( G \) by applying modified \( G + e \) and \( G + uv \) on \( G \). We generate two graphs \( G_1 \) and \( G_2 \). By the above discussion, we know that \( G \) is equal to \( G + e \) and \( G2 \) is equal to \( G + uv \). 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 \( S_1, S_2, ..., S_k \) such that each \( S_n \), \( i = 1 \) to \( k \) is a null graph implies \( S_k \) is a complete graph for every \( i = 1 \) to \( k \). Chromatic polynomial of \( G \) can be determined by recursive applying Type I, III operations on \( G \), we reach a graph \( H \) such that \( H \) is a null graph.
polynomial of \( S_i \) is \( P_n(\lambda)(S_i) = \prod_{k=0}^{n-1}(\lambda - k) \) if \( S_i \) is a null graph with \( n \) vertices.

\[
P_n(\lambda)(G) = \sum_{i=1}^{k} P_n(\lambda)(S_i) = \sum_{i=1}^{k} \lambda(n_i - 1) \cdots (\lambda - n_i + 1)
\]

where \( n_i \) denotes the number of vertices in graph \( S_i \), \( i = 1, 2, \ldots, k \). This procedure is summarized in Hierarchy – 1.

\[\lambda \]

Hierarchy – 1

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

\[
\begin{align*}
&\quad G + e = G - e \\
&\quad G_{uv} = G_{uv}
\end{align*}
\]

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 newly added edges 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 \( \overline{G}_{uv} \). 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 \( \overline{S_i} = \lambda^k \) if \( S_i \) is a graph with \( n \) vertices. \( P_n(\lambda) \)

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

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.

C. Chromatic Polynomial of \( G_{uv} \) and \( G_{sd} \) of DSNS Graphs

In this section, we provide a method of determining the chromatic polynomial of \( G_{uv} \) and \( G_{sd} \) 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 \( G_1 = G - \{e\} \), \( G_2 = G_{uv} \). Consider \( G_{uv} \). 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 \( G_3 = G - \{e_1\} \) and \( G_4 = G_{uv} \). We observe that \( G_4 \) is isomorphic to \( G \). Now consider graph \( G_5 \). Let us apply edge deletion and edge contraction using \( e_1 \). \( G_5 = G_4 - \{e_1\} - \{w\} \) isomorphic to \( G_1 \). Also \( G_{uv} \) is isomorphic to \( G_1 \).

\[
P_n(\lambda)(G_2) = P_n(\lambda)(G).
\]

\[
P_n(\lambda)(G_3 - \{e_1\} - \{w\}) = P_n(\lambda)(G_{uv})
\]

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 \( G_3 \)
For the graph in Fig. 6 $P_{\lambda}(\lambda) = \lambda (\lambda - 1) (\lambda - 2) (\lambda - 3) (\lambda - 4) + \lambda (\lambda - 1) (\lambda - 2) (\lambda - 3) + \lambda (\lambda - 1) (\lambda - 2) + \lambda (\lambda - 1) (\lambda - 2) (\lambda - 3) + \lambda (\lambda - 1) (\lambda - 2) (\lambda - 3) + \lambda (\lambda - 1) (\lambda - 2) + \lambda (\lambda - 1) (\lambda - 2) + \lambda (\lambda - 1)$. 
For the graph in Fig. 7 $P_n(\lambda)$ ($\rightarrow$ $S_n$) = $\lambda^5 - \lambda^4 - \lambda^3 + \lambda^4 + \lambda^3 + \lambda^3 - \lambda^4 - \lambda^4 + \lambda^3 + \lambda^2 - \lambda^2 + \lambda + \lambda^2 - \lambda^2 - \lambda^2 + \lambda$. 

$$\begin{align*}
\text{Fig. 7}
\end{align*}$$
– [ e₁ ] can be determined from that chromatic polynomial of G₁ with an additional vertex in all the resulting null graphs that is if N₁, N₂, ..., Nₖ are the resulting null graphs obtained to determine the chromatic polynomial of G₁ that is, Pₙ(λ) (G₁) = Pₙ(λ) (N₁) + Pₙ(λ) (N₂) + ... + Pₙ(λ) (Nₖ), then Pₙ

\[ Pₙ(λ) (G₁) - [e₁] = Pₙ(λ) (M₁) + Pₙ(λ) (M₂) + ... + Pₙ(λ) (Mₖ) \]

where each Mᵢ is a null graph such that | V (Mᵢ) | = | V (Nᵢ) | + 1 for all i = 1 to k. For this purpose, if Pₙ(λ) (G) denotes the chromatic polynomial of G where all the resulting graphs are null graphs, let us denote the chromatic polynomial of G ∪ {v} where v is an isolated vertex as Pₙ

\[ Pₙ(λ) (G ∪ {v}) = \sum_{i=1}^{n} λ_{i}^{n_{i}} - \sum_{j=1}^{n_{j}} λ_{j}^{n_{j}} + 1 \]

the cardinalities of the vertices of the resulting null graphs. So if G is any graph

\[ Pₙ(λ) (G_{uv}) = Pₙ(λ) (G₁ ∪ {w}) - Pₙ(λ) (G₁) - Pₙ(λ) (G) \]

implies,

\[ Pₙ (λ) (G_{uv}) = Pₙ (λ) (G₁ ∪ {w}) = Pₙ (λ) (G₁) + Pₙ (λ) (G₂) \]

This procedure is summarized in Hierarchy – 3.

Similarly we can obtain the chromatic polynomial of G₄uv by applying modified G + {e}, G – {e} and G•uv as already discussed.

When we compare G and G₄uv, G₄uv has only 2 additional edges and a vertex extra. When we remove edge e₂, two graphs are generated (labeled as G₁, G₂). G₁ is isomorphic to G with [ e₁ ] and G₂ is isomorphic to G. So to calculate the chromatic polynomial of G₄uv, we should estimate the chromatic polynomial of G₁ and G₂. This means that apart from calculating the chromatic polynomial of G, we should also determine the chromatic polynomial of G₁ also. This requires additional iterations (one complete set of iteration for graph G₁).

The number of iterations to determine Pₙ(λ) (G₁ ∪ {w}) = The number of iterations to determine Pₙ(λ) (G₁) 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 G₁ ∪ {w} can be avoided. It can be generated by following the same iterations as in G₁ along with the pendant vertex w. So these iterations can be avoided if formula 4 is used.

III. NUMBER OF SPANNING TREES OF G₄SDUV

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 τ(G) denotes the number of spanning trees of G, then τ(G) = τ(G – e) + τ(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 G₄uv from graph G. Let G be any graph. We can determine the
number of spanning trees of $G$ using Cayley’s method. Let $u, v \in V(G)$. Let $G_1$ and $G_2$ be the graphs generated by removing edge $(u v)$ and contracting edge $(u v)$ respectively (retaining back self loops and parallel edges). Let $G_{sd} = w$. 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). Let $G_{sd} = G \cup \{ w \} \cup \{ e_1 \}$. By Cayley’s theorem the procedure to determine all possible spanning trees is terminated when all the remaining graphs are trees. Since $G_3 = G_1 \cup \text{a pendant edge}$, the recursive procedure adopted for determining the number of spanning trees for graph $G_1$ can be retained for $G_3$ also implies $\tau(G_{sd}) = \tau(G_1) + \tau(G)$  

(5)

Remark
To determine the spanning trees of $G_{sd}$. We should determine the possible spanning trees $G_{sd}$ and $G - (u v)$. $G_{sd}$ 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 tree of $G$.

\[ \tau(G_1) = 52, \tau(G_2) = 103 \]

Consider $G_{sd}$. As per the formula we know that $\tau(G_{sd}) = \tau(G_1) + \tau(G) = 207$.

Consider $G_{sd}$.\n
\[ Fig.10 \]

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.

REFERENCES

2. Rong-xia Hao, Yan-pei Liu, On chromatic polynomials.