Rose-Hulman Undergraduate Mathematics Journal

Volume 8
Issue 2

Article 1

The Circuit Partition Polynomial with Applications and Relation to the Tutte and Interlace Polynomials

Andrea Austin
Saint Michael's College, aaustin2@smcvt.edu

The Circuit Partition Polynomial with Applications and Relation to the Tutte and Interlace Polynomials

Andrea Austin
Saint Michael's College
Colchester, VT 05439
aaustin2@smcvt.edu

Abstract

This paper examines several polynomials related to the field of graph theory including the circuit partition polynomial, Tutte polynomial, and the interlace polynomial. We begin by explaining terminology and concepts that will be needed to understand the major results of the paper. Next, we focus on the circuit partition polynomial and its equivalent, the Martin polynomial. We examine the results of these polynomials and their application to the reconstruction of DNA sequences. Then we introduce the Tutte polynomial and its relation to the circuit partition polynomial. Finally, we discuss the interlace polynomial and its relationship to the Tutte and circuit partition polynomials.

I. Introduction

The circuit partition polynomial (CPP) is a generating function for the number of Eulerian partitions of an Eulerian graph or digraph *G* into *n* components. It first appeared in [EM98] and was so named in [Bol02]. The CPP is useful for several reasons: it determines the number of Eulerian components of a graph or, in conjunction with other polynomials such as the Tutte and interlace polynomials, makes evident further characteristics of a graph.

The CPP is a shift of the Martin polynomial [Mar77], which was first introduced in 1977 by Pierre Martin in his thesis. Martin defined his polynomial recursively; it encodes information about the families of circuits in 4-regular Eulerian graphs and digraphs [Mar77]. A more general form of the polynomial was developed by Las Vergnas in 1983 in *Le Polynome de Martin d'un Graphe Eulerien* [LV83]. In this paper, he found a closed form of the polynomial and extended its properties to general Eulerian graphs, as we apply it today.

The CPP, along with the Martin polynomial, is applicable to many areas, including non-mathematical fields. These include, although are not limited to, string reconstruction, infrastructure networks, and knot theory.

Perhaps the most interesting and useful current application of Eulerian graphs occurs within the area of DNA sequencing. The CPP enables us to estimate the probability of a correct reconstruction of a DNA sequence through its ability to enumerate Eulerian circuits. Possible reconstructions of a strand of DNA are represented by Eulerian circuits in the de Bruijn graph constructed using fragments of DNA. This de

Bruijn graph is an Eulerian graph, and we are thus able to calculate the CPP and like polynomials to determine its characteristics. One example of this is seen after computing the CPP; when we do this, we determine the number of Eulerian circuits and thus determine the probability that a given reconstruction is the correct one.

This paper will focus on describing the CPP and how it is applied. We begin by reviewing some basic terminology of graph theory and how it pertains to our discussion of the CPP. Then we will review the CPP and its basic applications. Finally, we will discuss the interrelation between both the CPP and the Tutte polynomial and the CPP and the interlace polynomial.

II. Background

There are several key concepts we must know before we can proceed with our work in graph theory and the CPP. Our definitions follow those of Tucker [Tuc07] and Bollobas [Bol98].

Definition 1: A *graph G* consists of finite sets of vertices *V* and edges *E*, which are pairs of vertices.

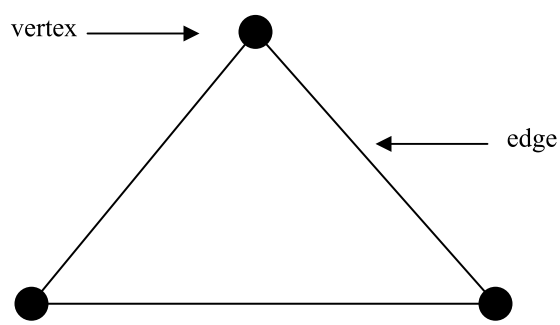


Figure 1: A graph *G*

Vertices will represent elements of an object or system represented by the graph *G*. Edges represent interactions between pairs of elements (vertices). The *degree,* or *valency,* of a vertex is the number of edges incident with it.

Definition 2: A *digraph* is a graph in which the edges are directed, that is, the pair of vertices comprising an edge are ordered.

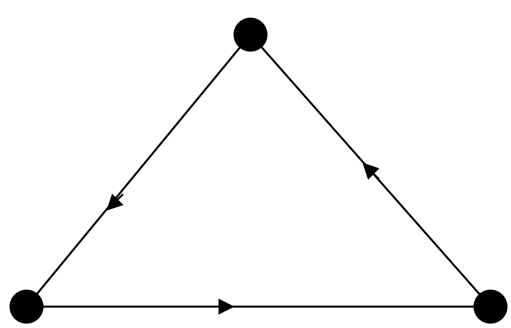


Figure 2: A directed graph

The direction designates an initial vertex and a terminal vertex. Here we will use digraphs to represent a DNA strand reconstruction problem. The DNA reconstruction problem is modeled using Eulerian digraphs, in which the in degree (number of arrows oriented towards a vertex) and the out degree (number of arrows oriented away from a vertex) are equal for each respective vertex. The digraph essentially represents all the possible ways a set of DNA fragments can fit together. Note this means that each vertex is of even degree (although each vertex may be of a different even degree).

The arrows dictate the correct sequence of the nucleotide fragments. In general, the direction applied to the edges in a digraph represents some specific, unilateral, interaction between the two elements corresponding to by the vertices of the graph.

Definition 3: A *multigraph* is a graph that may have multiple edges (2 or more edges between a set of vertices) or loops (a loop is an edge with the same vertex as both endpoints). We count a loop edge twice in determining the degree of a vertex.

Definition 4: A *circuit* is a sequence of linked edges whose starting vertex and ending vertex are the same and in which no edge can appear more than once. Note that a vertex may be visited more than once.

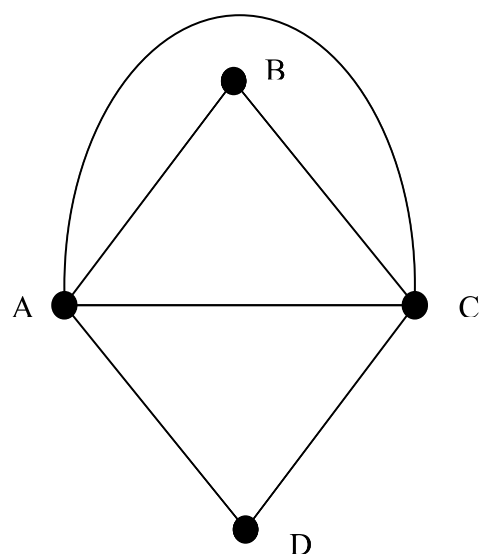


Figure 3: A circuit of *G* is ACBADCA

Note that for an undirected graph, such as figure 3, a circuit exists in both directions from the initial vertex (ADCBA and ABCDA). However, in reference to digraphs, the circuit must follow the orientation of the edges of the graph.

Definition 5: An *even* graph is one where every vertex has even degree (the number of edges adjacent to each vertex is even).

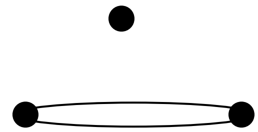


Figure 4: A (disconnected) even graph whose vertices have either degree 0 or 2.

Definition 6: An *Eulerian circuit* is a circuit that visits every edge in the graph exactly once and visits each vertex at least once.

Eulerian circuits can only exist in Eulerian graphs, that is connected (graphs with 1 component), even graphs.

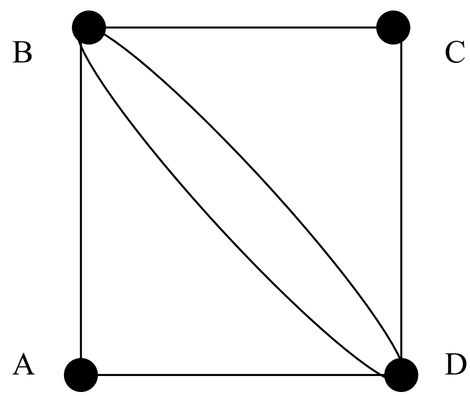


Figure 5: Graph with Eulerian circuit ABDCBDA

Eulerian circuits are particularly important to research involving DNA sequencing. DNA fragments can be represented by an oriented Eulerian graph and thus analyzed using such methods as the CPP.

Definition 7: A *de Bruijn graph* is a directed graph representing overlaps between sequences of symbols in string reconstruction.

Consider the sequence “ababbaa”. We construct a de Bruijn graph by examining the triples present in the sequence. These are aba, bab, abb, bba, and baa; we note that all of the triples of the sequence are unique. The following figure shows the de Bruijn graph corresponding to the sequence, with the dotted edge added to complete the graph to an Eulerian graph.

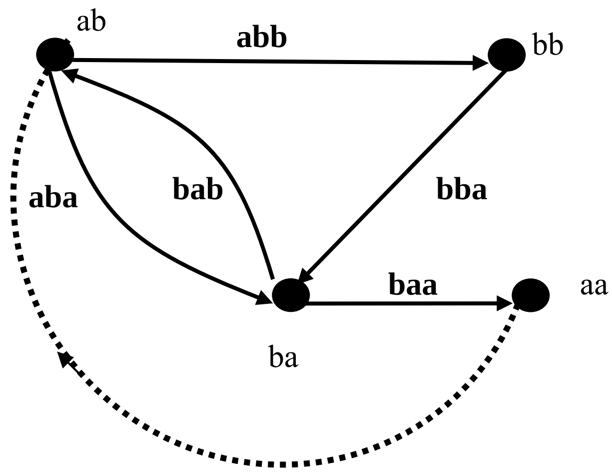


Figure 6: A de Bruijn Graph

The vertices consist of a pair within each of the triples, and the directed edge between two vertices represent the respective triple.

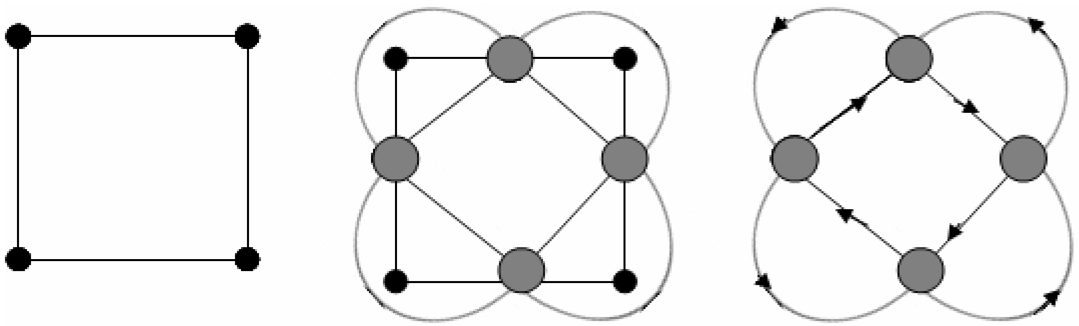
While the de Bruijn graph represents all possible combinations of the symbols, we are merely concerned with only a subset of these which arise from the data. When presented with a string, or in this case, a strand of DNA, we are able to represent it as a graph by constructing its de Bruijn graph. The first step to constructing the graph is to break the strand into fragments. In a real laboratory situation, this is accomplished through the collection of the raw data. Because scientists cannot read entire strands, they read many smaller pieces and then try to figure out what the original looked like. When scientists get the data, they receive the fragments back in a random order, not the order in which they were read/constructed. Using these, we create a digraph. We also insert an edge between the last and first vertex representing the DNA strand. Thus, we have created the de Bruijn (di)graph, and can continue with further calculations about the reconstruction of such a strand of DNA. It is important to note that while every Eulerian circuit in the de Bruijn graph represents one possible reconstruction for the DNA, only one of the circuits, and thus reconstructions, is the correct one. Therefore, it is evident where the next formula mentioned comes from.

Since the Euler circuits in the de Bruijn graph produces all possible reconstructions of a string, we can determine the probability that any given reconstruction is the correct one by the formula:

$$\frac{1}{\#\ \ of\ Eulerian\ circuits}.$$

We assume the de Bruijn graph resulting from a set of DNA fragments is a 2-in 2-out digraph, that is, every vertex has two incoming and two outgoing edges. We would like to determine the number of Eulerian circuits present. While typically the BEST theorem [ST41] can be applied to the de Bruijn graphs, we will calculate the CPP, as its flexibility will allow us for greater variation. The BEST theorem is only applicable to digraphs while the CPP can be applied to unoriented graphs as well. Also, as we will see later, computing the CPP allows us to compute other polynomials, which gives us more information about de Bruijn graphs.

Definition 8: A *medial graph* of a planar graph G (a graph that can be drawn in a plane without edges intersecting) is constructed by putting a vertex on each edge of the graph, and drawing edges around the faces of G. The resulting medial graph G_m of any planar graph G is always a 4-regular graph (every vertex is of degree 4). The *directed medial graph*, $\vec{G}_m$, results from directing the edges so that the arrows point counterclockwise on the edges of the medial graph which enclosed vertices from the original graph G (see figure 7).



A planar graph G | G with G_m superimposed | The directed medial graph $\vec{G}_m$

Figure 7: Constructing medial graphs allows us to examine one class of 4-regular graphs.

We will be using medial graphs later in our analysis of the relations between the CPP and other polynomials.

Definition 9: A *bridge* in a graph G is an edge whose removal results in a graph with more components than G.

Definition 10: A *loop* is an edge than connects a vertex to itself.

Definition 11: A *cut vertex* in a graph G is a vertex whose removal results in a graph with more components than G.

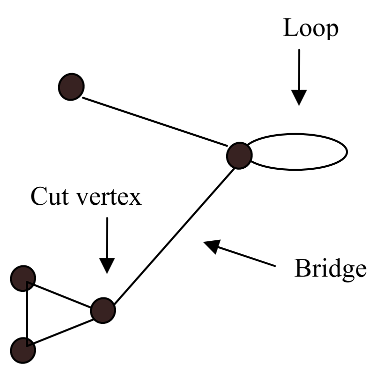


Figure 8

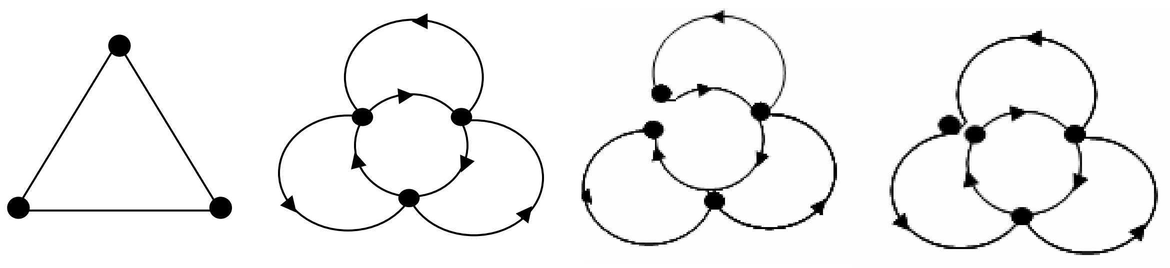
Definition 12: An *Eulerian graph state* of a graph, G, is the result of replacing all $2n$-valent vertices, v, of G, with n 2-valent vertices joining pairs of edges originally adjacent to v (see figures 10-12).

III. The Circuit Partition Polynomial

The circuit partition polynomial is a polynomial that encodes the number of Eulerian graph states in a digraph.

Definition 13: The CPP is defined by: $j(\vec{G};x)=\sum_{k\geq 0} f_k(\vec{G})x^k$ where $f_k(\vec{G})$ is the number of Eulerian graph states of $\vec{G}$ with k components [EM98, Bol02].

To better understand the CPP, we will illustrate it with an example. Consider the following graph G:



G $\qquad$ $\vec{G}_m$ $\qquad$ The two consistent splits at a vertex.

Figure 9

The CPP of the graph $\vec{G}_m$ in figure 9 is:

$$j(\vec{G}_m;x)=x^3+4x^2+3x\,.$$

To make sense of the formula given by the CPP, we can look at the resulting graph states of the medial graph. Consider the x^3 term in the equation above. Such a term, with coefficient 1, means that there exists one Eulerian state of the medial graph with 3 components. This graph state is:

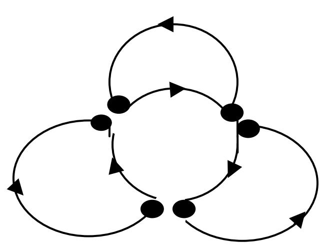
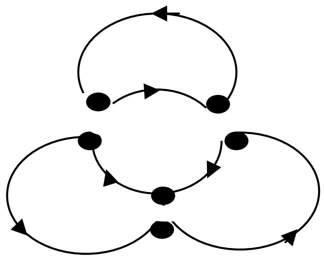


Figure 10

Similarly, the four 2-component states are:



(Each of 3 possibilities
by rotating split to each vertex)

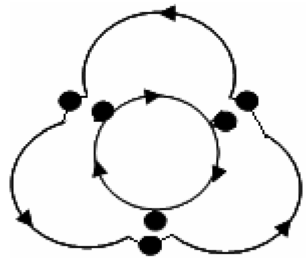
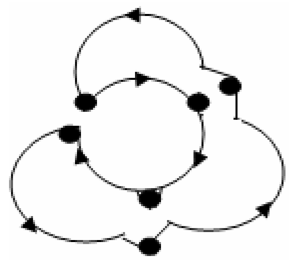


Figure 11

Finally, the three 1-components states are:



(Here again the three 1-components are constructed by rotating the split along the 3 vertices.)

Figure 12

Now that we have defined the CPP and demonstrated its results, the next logical question is how is it applicable? Note that the coefficient of the x^1 term counts the number of Euler circuits in the graph. Thus, an obvious use for the CPP is string reconstruction. The quintessential example of this is the reconstruction of strands of DNA.

Once we have constructed the directed de Bruijn graphs as previously described or have any graph representing a string, we next apply the CPP as described above. Thus we calculated the number of reconstructions of the sequence represented by the graph. The need to accurately reconstruct DNA is necessary in many aspects of scientific research including forensics and medical research.

One final advantage to the CPP is its close relation to several other polynomials. Because of these relationships, its applicability increases even further. We will further describe some of the applications of the CPP when we discuss its relation to other polynomials and how these applications arise because of the interrelation.

IV. The Tutte Polynomial and its Relation to the Circuit Partition Polynomial

In [Mar77, Mar78], Martin found another interesting property of the Martin polynomial. He found that the Martin polynomial of the medial graph $\vec{G}_m$ of a connected planar graph *G*, is equal to the dichromatic, or Tutte, polynomial of the graph *G*.

First, we provide some of the notions needed to understand the definition of the Tutte polynomial. The Tutte polynomial is defined recursively and uses two graph operations. These are the deletion and contraction of an edge *e*. We will denote the deletion of an edge by $G - e$. Similarly, we denote the contraction of an edge with G / e.

Note for each of these operations, e will not be a bridge or a loop. To better understand these graph operations consider:

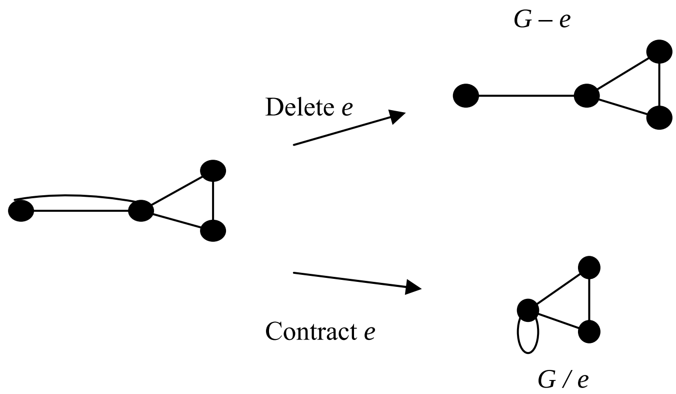


Figure 13

Definition 14: [Bol98] The Tutte polynomial of a graph G is given recursively by:

$$T(G;x,y)=\begin{cases} T(G/e;x,y)+T(G-e;x,y) & \textit{if } e \textit{ is neither a bridge nor loop} \\ xT(G-e;x,y) & \textit{if } e \textit{ is a bridge} \\ yT(G-e;x,y) & \textit{if } e \textit{ is a loop} \\ 1 & \textit{if } G \textit{ has no edges} \end{cases}$$

An important property of the Tutte polynomial is the following:

Proposition 1: $T(G*H)=T(G)T(H)$ if $G*H$ is the disjoint union or one point join of G and H. A one point join of two graphs G and H is formed by identifying vertex u of G and a vertex w of H into a single vertex v of $G*H$, which is necessarily a cut vertex.

Proof: The above statement is true, because we are able to compute the Tutte polynomial of one side of the graph independently of the other, and then by multiplying each of the outcomes together, we achieve the Tutte polynomial for the entire graph $G*H$.

In particular, if $e \in H$ then:

$$(G*H)-e=G*(H-e)$$
$$(G*H)/e=G*(H/e)$$

The equivalent is true if $e \in G$. Therefore, we can see that calculating the Tutte polynomial on the two separate parts of the graph described in proposition 1 and then multiplying the results is equivalent to calculating it for the entire graph.

□

Now that we have defined the Tutte polynomial, we can define its relation to the CPP. We will accomplish this by first proving the Tutte polynomial's relationship to the Martin polynomial. First, we'll need the following two facts about the Martin polynomial.

Proposition 2: Let $m(G;x)$ represent the Martin polynomial of a graph G. The relationship between the CPP and the Martin polynomial of a digraph $\vec{G}_m$ is given by: $j(\vec{G};x) = x \cdot m(\vec{G};x+1)$ [EM98] .

We will be using this relationship in our proof later of the relationship between the CPP and the Tutte polynomial.

Proposition 3: If v is a cut vertex in a 2-in 2-out digraph, then $m(G;x) = x \cdot m(H_1;x)m(H_2;x)$, where H_1 and H_2 are the two component you get by removing v and joining up the 4 resulting half edges into one edge in H_1 and one edge in H_2. If v is not a cut vertex, then $m(G;x) = m(H_1;x) + m(H_2;x)$ where H_1 and H_2 are the two consistent ways of splitting at the vertex as in figure 9[Jae87].

Theorem 1: The relationship between the Tutte and Martin polynomials, for a planar graph G, is given by $T(G;x,x) = m(\vec{G}_m;x)$.

Next, we can go through the proof of the result. The reason we are using the Martin polynomial as opposed to the CPP is that it is easier to prove; as we have previously seen the equivalence relationship between CPP and Martin, it is not imperative to use the CPP in our proof.

*Proof***:** The proof proceeds by induction on n, the number of edges of a planar graph G. The first case we will consider is when the edge e is neither a bridge nor a loop. The second case will concern what happens when e is a bridge. And finally, the third case will be if e is a loop.

First, we must prove the base case. Let $n = 1$. We will construct all possible graphs with one edge. First, consider the graph G_1:

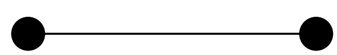
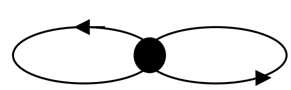


Figure 14: G_1

Then we construct its oriented medial graph, and note its Eulerian graph states:



$(\vec{G})_m$

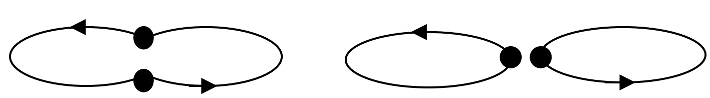


Figure 15: $(\vec{G}_1)_m$ and its Eulerian graph states

Since the original consists of one bridge, by our definition of the Tutte polynomial, we know:

$$T(G_1; x, x) = x .$$

Following this, we compute the Martin polynomial, using definition 13 and proposition 2, on the oriented medial graph to find:

$$j((\vec{G}_1)_m; x) = x^2 + x \quad \text{so,} \quad m((\vec{G}_1)_m; x) = x .$$

Again, let us consider the second graph G_2 that consists of one edge, namely a loop:

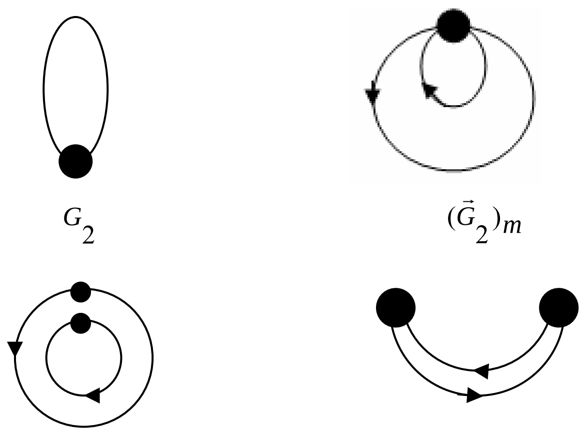


Figure 16: G_2 and $(\vec{G}_2)_m$ and the two Eulerian graph states

When we compute the Tutte and Martin polynomials of *G_2* and $(\vec{G}_2)_m$, respectively, we get:

$$T(G_2; x, x) = x \text{, since } T(G_2; x, y) = y ,$$

and again

$$j((\vec{G}_2)_m; x) = x^2 + x \quad \text{so,} \quad m((\vec{G}_2)_m; x) = x .$$

Thus, we have proven the base case when $n = 1$. Now, we can proceed with the remainder of our argument, assuming that whenever has less than *n* edges, the statement is true.

Case 1: We assume *G* has *n* edges, and *e* is neither a bridge nor a loop.

$$T(G; x, x) = T(G - e; x, x) + T(G / e; x, x)$$

$$= m((\vec{G} - e)_m; x) + m((\vec{G} / e)_m; x).$$

$$= m(\vec{G}_m; x)$$

This follows from proposition 3, and can be illustrated by considering how the deletion and contraction of e in G corresponds to the two consistent ways of splitting at v in $\vec{G}_m$:

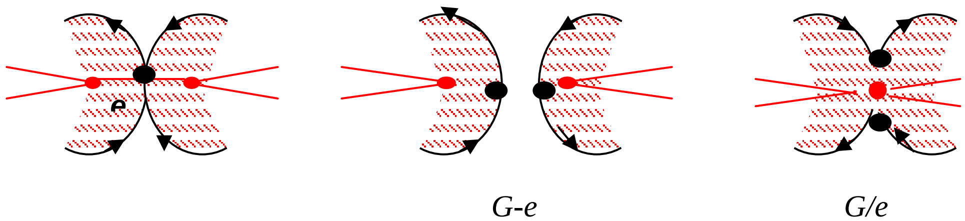


Figure 17: G, pictured in red and $\vec{G}_m$, in black

Case 2: Suppose G has n edges and e is a bridge. Then the vertex in $\vec{G}_m$ corresponding to e is a cut vertex. From definition 14, the induction hypothesis, and proposition 3 we know:

$$\begin{aligned} T(G;x,x) &= xT(G/e;x,x) \\ &= xT(G_1;x,x)T(G_2;x,x) \\ &= xm((\vec{G}_1)_m;x)m((\vec{G}_2)_m;x) \\ &= m(\vec{G}_m;x) \end{aligned}$$

Again, as in case one, we can illustrate to clarify:

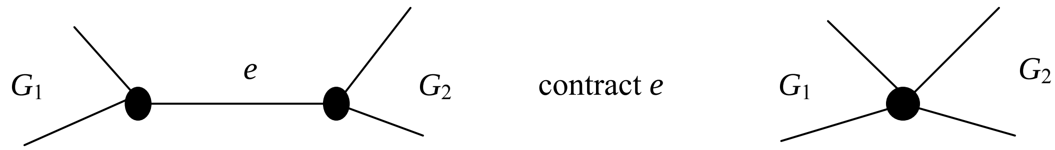


Figure 18

Thus, we are able to multiply the Tutte polynomial for each part of the graph (G_1 and G_2) on either side of the cut vertex, and obtain our correct answer.

Case 3: Finally, we can consider the third and final case, when G has n edges and e is a loop. From our definition, we know that the Tutte polynomial for G where e is a loop is:

$$T(G;x,x) = xT(G-e;x,x),$$

illustrated by:

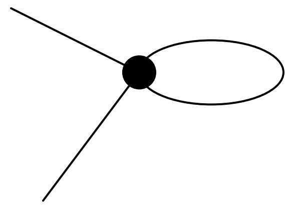


Figure 19

Next, we consider $\vec{G}_m$:

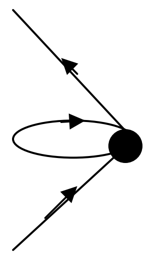


Figure 20

We can calculate the Martin polynomial of the medial graph, which again has a cut vertex corresponding to the edge *e*:

$$m(\vec{G}_m;x) = x \cdot m(\vec{H}_1;x)m(\vec{H}_2;x)$$

where H_1 is all of $\vec{G}_m$ except the loop and H_2 is the loop. In particular $H_1 = \overrightarrow{(G-e)}_m$. Based on definition 13 and proposition 2 , we find that $m(H_2)=1$. Therefore, from proposition 3, we get:

$$m(\vec{G}_m;x) = xm(H_1)$$

Thus, by noting that $m(\vec{H}_1;x) = T(G-e;x,x)$, we see that the equation is true.

We have proven that all the relationship holds for all types of edges. Therefore, we can conclude it is always true.
□

Following from the above proof, we can see the veracity of the following corollary, which follows from proposition 2.

Corollary 1: The relationship between the CPP and the Tutte polynomial is:

$$j(\vec{G}_m;x) = x^{c(G)}T(G;x+1,x+1)$$

where $c(G)$ is the number components of the original graph G.

Finally, we illustrate this with an example. Recall $\vec{G}_m$ (from figures 9 and 21) and its CPP:

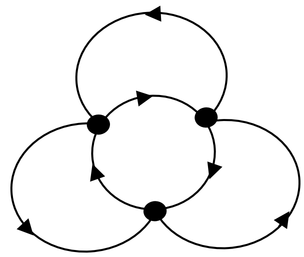


Figure 21

$$j(\vec{G}_m; x) = x^3 + 4x^2 + 3x.$$

Now consider the graph G, and compute its Tutte polynomial:

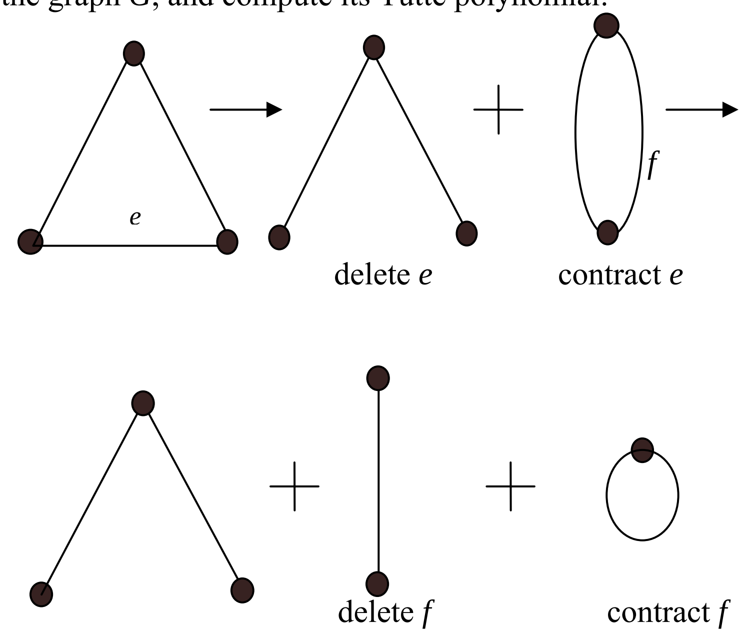


Figure 22

We thus produce the Tutte polynomial:

$$T(G; x, y) = x^2 + x + y, \text{ so}$$

$$T(G; x, x) = x^2 + 2x.$$

From here we can use the relation of corollary 1 to ensure the equation is correct:

$$j(\vec{G}_m; x) = x^{c(G)} T(G; x+1, x+1) = x^1[(x+1)^2 + 2(x+1)] = x^3 + 4x^2 + 3x.$$

Therefore, we have shown that the relationship between the two polynomials is indeed valid for this example.

The Tutte polynomial has many interesting applications, and because of the relationship between it and the CPP, the CPP shares these applications as well. Now we can examine yet another polynomial, the interlace polynomial, to see even more fascinating attributes of the CPP.

V. The Interlace Polynomial and Relation to the Circuit Partition Polynomial

The interlace polynomial was first introduced in [ABS04] by Arratia, Bollobás, and Sorkin in relation to DNA sequencing by hybridization. The interlace polynomial is a technique to encode for any *k*, the number of *k*-component circuit partitions that is, Eulerian graph states with *k* components. Directly from the statement of its applicability, we are able to show that the interlace polynomial is related to the circuit partition polynomial.

Again, before we can properly present the interlace polynomial formula, it is imperative to first understand some of the notation involved in the definition. One graph operator used in defining the interlace polynomial of a graph G, which was also seen in our analysis of the Tutte polynomial, is $G - v$. A new concept is G^{vw}, called pivoting on the edge vw.

Definition 15: G^{vw} denotes the graph derived from a graph G by "toggling" the edges and non-edges among three sets of vertices: those adjacent to w only, those adjacent to v only, and those adjacent to both w and v. "Toggling" means that any edge between two of these sets is removed, and any non-edge is replaced by an edge. We can also note that if vertices in G are not adjacent to v or w (or both) then we simply leave them, and their adjacent edges, as they were in the original graph.

For example:

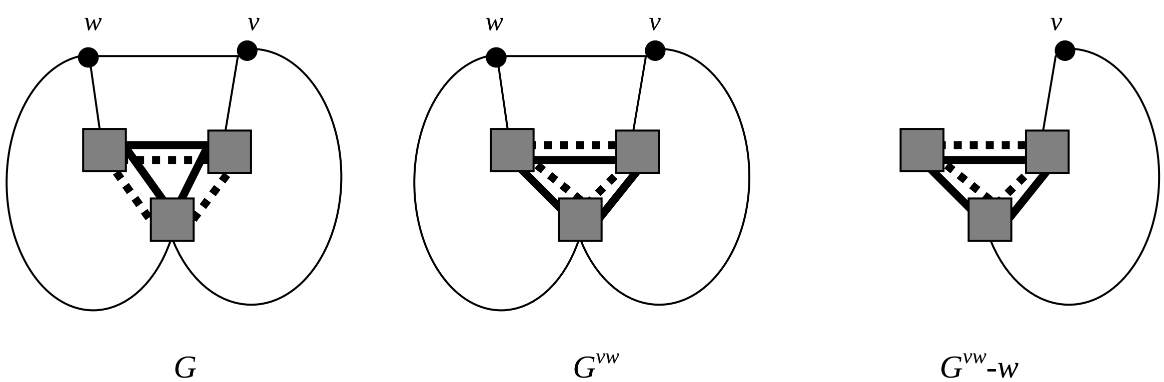


Figure 23: The dotted and heavy lines represent an interchange of edges and non-edges among the set of vertices adjacent to *v* only, *w* only, or both (the gray squares). Vertices adjacent to neither were omitted. The thin solid lines represent all edges present.

Now that we have gone through the notational intricacies of the interlace polynomial, we can present the definition.

Definition 16: [ABS04] The interlace polynomial (much the same as our other polynomials thus far) is defined recursively by:

$$q_G(x) = \begin{cases} x^n & \text{if } G \text{ has } n \text{ vertices and no edges,} \\ q_{G-u}(x) + q_{G^{(uv)}-v}(x) & \text{if } uv \in E(G). \end{cases}$$

To better understand the interlace polynomial, it is beneficial to see its relation to the CPP. We first recall circle graphs and chord diagrams. A chord diagram is a circle with *n* symbols, each appearing twice around the perimeter, with like symbols joined by a chord. In our application, we will construct a chord diagram from an Euler circuit in a 2-in 2-out graph by listing the vertices along the perimeter of the circle in the order they are visited in the Euler circuit. The chord diagram is used to construct a circle graph, making a vertex for each symbol and making adjacent those whose chords intersected in the chord diagram. To better understand the concept, consider the following example.

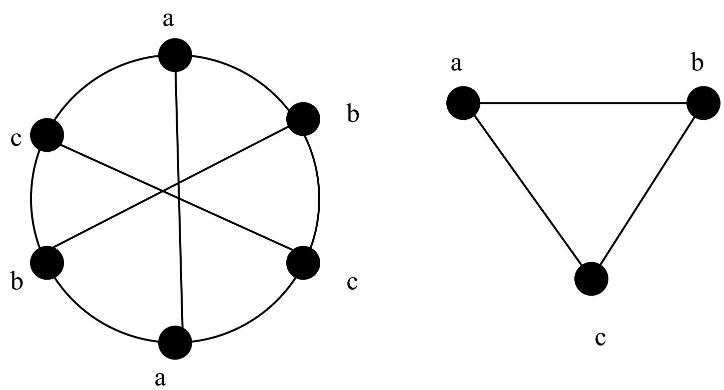


Figure 24: A chord diagram and its corresponding circle graph.

Proposition 4: [Bol02] If $\vec{G}$ is a 2-in 2-out Eulerian digraph, *C* is any Eulerian circuit of $\vec{G}$, and *H* is the circle graph of the chord diagram determined by *C*, then $f(\vec{G};x) = x^{c(G)} q_N(H; x+1)$, where $c(G)$ is defined the same as for the Tutte polynomial.

From definition 16, the relationship between the Tutte and interlace polynomials naturally arises.

Proposition 5: The relationship between the Tutte and interlace polynomials is given by: $T(G;x,x) = q_N(H;x)$ [EMS].

Although we will not prove proposition 4, we will illustrate the relationship among the three polynomials through an example. Consider *G*:

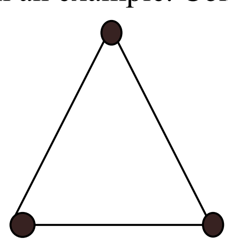


Figure 25

and then $\vec{G}_m$:

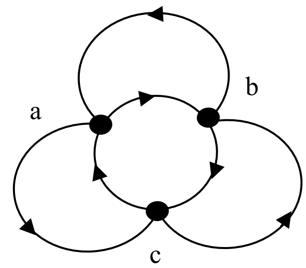


Figure 26

We can compute the Tutte polynomial and the CPP of the respective graphs via the formulas we saw before, thus obtaining the formula previously seen in sections III and IV:

$$T(G;x;x) = x^2 + 2x$$

$$j(\vec{G}_m;x) = x^3 + 4x^2 + 3x\,.$$

Next, we can compare this to the interlace polynomial of the circle graph of the chord diagram of $\vec{G}_m$. First, we must choose one Eulerian circuit in $\vec{G}_m$. For instance:

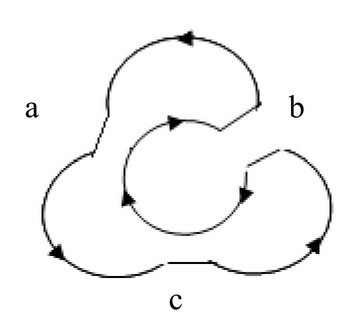


Figure 27

Then we can construct the chord diagram following the circuit above:

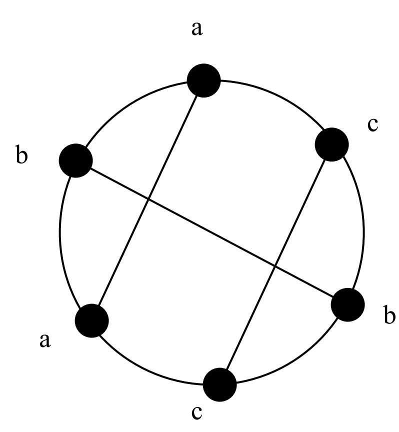


Figure 28

Finally, we make the circle graph H corresponding to the chord diagram:

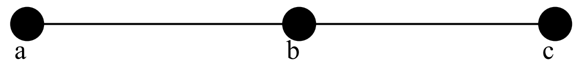


Figure 29

We compute the interlace polynomial of H as $q_N(H;x) = q_N(H-a;x) + q_N(H^{ab}-b;x)$. Note that $q_N(H-a;x) = q_N(H-a-b;x) + q_N((H-a)^{bc}-c;x) = x + x$ since both $H-a$ and $H-a-b$ and $(H-a)^{bc}-c$ are single vertices. Also, $H^{ab} = H$, so $H^{ab}-b$ is just two isolated vertices, and thus $q_N(H^{ab}-b;x) = x^2$. Thus,

$$q_N(H;x) = x^2 + 2x.$$

Accordingly,

$$T(G;x,x) = x^2 + 2x = q_N(H;x)$$

and

$$f(\vec{G}_m;x) = x^3 + 4x^2 + 3x = x^1[(x+1)^2 + (x+1)] = x^{c(G)} q_N(H;x+1),$$

as given by propositions 4 and 5. As we have seen, the interlace polynomial exhibits some interesting properties, including a relationship to the CPP and Tutte polynomial.

VI. Conclusions

Graph polynomials such as the Martin, CPP, Tutte, and interlace are a unique and dynamic aspect of the world of graph theory. Their applicability is tremendous, yet it is widely believed its full potential has not yet been realized. We examined only a small portion of the amount of remarkable research that has been accomplished on the CPP.

Even though leaps and bounds have been made in the area of graph theory especially in regard to the CPP, there are still many open questions. One such question is especially important in analyzing strands of DNA. That is, work is being done to try to classify all graphs with m Eulerian circuits. As was so eloquently stated by Balister et al., "although…a fair amount is proved about the interlace polynomial, it is still a rather mysterious graph invariant" [BBCP02]. While I have highlighted some of the interesting aspects of various polynomials, there is much more available and even yet to be discovered.

Acknowledgements

Support made possible by the Vermont Genetics Network through Grant Number P20 RR16462 from the INBRE Program of the National Center for Research Resources (NCRR), a component of the National Institutes of Health (NIH). Its contents are solely the responsibility of the author and do not necessarily represent the official views of NCRR or NIH. Support for this research was also made possible through the National Security Agency.

Bibliography

[ABS04] Arratia, Richard; Bollobas, Bela; Sorkin, Gregory. The Interlace Polynomial of a Graph. *Journal of Combinatorial Theory Series B* **92** (2004), no. 2, 199-233.

[BBCP02] Balister, P.N.; Bollobas, B; Cutler, J.; Pebody, L. The Interlace Polynomial of Graphs at -1. *Europ. J. Combinatorics* **23** (2002), 761-767.

[Bol02] Bollobas, Bela. Evaluations of the Circuit Partition Polynomial. *Journal of Combinatorial Theory Series B* **85** (2002), no. 2, 261-268.

[Bol98] Bollobas, Bela. *Modern Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag New York, Inc. New York, NY (2002).

[EM98] Ellis-Monaghan, Joanna A. New Results for the Martin Polynomial. *J. Combin. Theory, Series B* **74** (1998) 326-352.

[EMS] Ellis-Monaghan, Joanna A. Sarmiento, Irasema. Distance Hereditary Graphs and the Interlace Polynomial. In press, Combinatorics, Probability and Computing.

[Jae87] Jaeger, Francois. On Transition Polynomials of 4-Regular Graphs. *Laboratoire de Structures Discretes*, 1987.

[LV83] Las Vergnas, M. Le Polynome de Martin d'un Graphe Eulerian, Combinatorial Mathematics (Marseille-Luming, 1981), North-Holland Mathematics Studies, Vol. 75, North Holland, Amsterdam, 1983, pp. 397-411.

[Mar77] Martin, P. Enumerations Euleriennes dans le Multigraphs et Invariants de Tutte-Grothendieck, Thesis, Grenoble, 1977.

[Mar78] Martin P. Remarkable Valuation of the Dichromatic Polynomial of Planar Multigraphs, *Journal of Combinatorial Theory, Series B* **24** (1978) 318-324.

[ST41] Smith, C.A.B., Tutte, W.T. On Unicursal Paths in a Network of Degree 4, *American Mathematics Monthly* **48** (1941) 233-237.

[Tuc07] Tucker, Alan. *Applied Combinatorics*, Fifth Edition, John Wiley and Sons, Inc. Hoboken, NJ. (2007)