Lattice Interval Partition of Polynomial Graph Invariants

Alex Rossie

August 13, 2010

The Tutte polynomial introduced by W. Tutte in [9] is a well known bivariate polynomial graph invariant that carries information about finite multigraphs. In [2], B. Bollobas discusses the spanning tree expansion of the Tutte polynomial in the context of Tutte’s original definition of activity. In this paper, the author will give a combinatorial model for the Tutte polynomial using binary trees, and prove the spanning tree expansion of the Tutte polynomial. Lastly, the author will extend the model to the Bollobas-Riordan polynomial, a recently discovered trivariate polynomial graph invariant that generalized the Tutte polynomial for embedded (cyclic) graphs [3][4], using a quasi-tree expansion given by Champanerkar et al. [5]. The goal of this paper is to create a general framework for subgraph lattice partition that can be applied to any deletion-contraction polynomial invariant of graphs.

1 Foundations

Definition. Consider the power set \( \mathcal{P}(E) = \{0, 1\}^E \) where \( E \) is an arbitrary set. For \( T^-, T^+ \in \mathcal{P}(E) \), we introduce a partial order on this set by letting \( T^- \subseteq T^+ \) if \( T^- \subseteq T^+ \). An interval is written in the form \([T^-, T^+]\). For instance, \([\{f\}, \{e, f, g\}] = \{\{f\}, \{e, f\}, \{f, g\}, \{e, f, g\}\}. We define a complex partition \( \bigcup_{L \in \mathcal{L}(E)} [L^-, L^+] \) as a partition of the set \( \mathcal{P}(E) \) into such intervals.

Definition. Fix a set \( E \). We define an \( E \)-tree as a labeled binary tree where each path from the root to a leaf contains the elements of \( E \) exactly once, we refer to the tree as \( \langle E \rangle \), and the set of leaves of this tree \( \mathcal{L}(E) \). We associate each leaf \( L \in \mathcal{L}(E) \) with a leaf order \( \prec_L \), defined as the order of appearance of these \( E \)-elements (i.e. an order on the set \( E \)), with the root always being the greatest. We say that \( \langle E \rangle \) is globally ordered if all such \( \prec_L \) are equal. Otherwise we call it locally ordered. Furthermore, a node with only one child is called a unitary node. Lastly, given a leaf \( L \) in some \( \langle E \rangle \), we speak of interval in \( \mathcal{P}(E) \) associated with \( L \) and denote it with \([L^-, L^+]\).

Theorem 1. For any \( E \)-tree, there exists \( \bigcup_{L \in \mathcal{L}(E)} [L^-, L^+] \) as a unique complex partition of \( \mathcal{P}(E) \)\(^1\).

We introduce an expansion process where we take a unitary node and duplicate its unique subtree into two. We apply this process to \( \langle E \rangle \) starting from the

\(^1\)This theorem is analogous to Prop. 2.3 in [7].
root, and stopping at the leaves. We now have a complete binary tree. We identify each leaf with an element of $\mathcal{P}(E)$ according to inclusion-exclusion and recollapse it back to the original $\langle E \rangle$ to obtain the complex partition. To state this rigorously, we prove the theorem using induction.

**Proof.** The base case where $|E| = 1$ is clear, as $\{\{0\}\}, \{\{1\}\}$ is a complex partition of $\{0,1\}$. We assume the hypothesis for $|E| = n$. For $|E \cup \{e\}| = n + 1$, we consider two cases according to whether the root $e$ is unitary. If the root is binary, we apply the inductive hypothesis and call the left child of the root the complex partition $\bigcup_{\mathcal{L}(E)} [L^-, L^+]$ and the right child the partition $\bigcup_{\mathcal{L}(E)} [L^-, L^+] \cup \{e\}$ (note these two partitions are isomorphic), and we have obtained a complex partition of $\mathcal{P}(E \cup \{e\})$ by taking the union of the two. If the root is unitary, we apply the inductive hypothesis and call its only child the complex partition $\bigcup_{\mathcal{L}(E)} [L^-, L^+]$. It is easy to check that $\bigcup_{\mathcal{L}(E)} [L^-, L^+ \cup \{e\}]$ is a complex partition of $\mathcal{P}(E \cup \{e\})$. This completes the induction. □

![Diagram](image)

Figure 1: A locally ordered binary tree with its associated complex interval partition.

We would like to note here that the converse of Theorem 1 is not true. Consider the following complex partition of $\mathcal{P}(E) = \{0,1\}^{\{a,b,c,d\}}$:

$$
\begin{align*}
\{\{a\}, \{a, b\}\}, \\
\{\{c\}, \{b, c, d\}\}, \\
\{\phi, \{b\}, \{d\}, \{b, d\}\}, \\
\{\{c\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}, \\
\{\{a, d\}, \{a, c, d\}, \{a, b, d\}, \{a, b, c, d\}\}.
\end{align*}
$$

2
There is no \( \{a, b, c, d\} \) that corresponds to this partition, as the intervals form a gridlock. This is the smallest counterexample with \( |E| = 4 \).

**Definition.** Fix a subset \( \mathcal{U} \subset E \). A \((\mathcal{U}, E)\)-tree is an \( E \)-tree where the set of leaves \( \mathcal{L}_{\langle E \rangle} \) is injective to the set \( \mathcal{U} \). More specifically, for each \( L \in \mathcal{L}_{\langle E \rangle} \), there exists \( U \in [L^-, L^+] \) such that \( U \in \mathcal{U} \). We shall refer to this tree as \( \langle \mathcal{U}, E \rangle \).

It is worth noting that the difference between a \((\mathcal{U}, E)\)-tree and an \( E \)-tree is that the leaves of a \((\mathcal{U}, E)\)-tree are not only identified with intervals in a complex partition, but also a particular element \( U \in \mathcal{U} \) in each interval. We now adopt the convention to call such complex partition \( \bigcup_{\mathcal{U}} [U] \) when talking about \((\mathcal{U}, E)\)-tree. Finally, we draw distinction between \([U]\) and \([L^-] \cup [L^+]\) as the former refers to the interval associated with a particular element of \( \mathcal{U} \) since \( U \) is both an element of \( \mathcal{U} \) as well as identified with a leaf in the tree \( \langle \mathcal{U}, E \rangle \) whereas the latter simply refers to an interval as \( L \) is a leaf of \( \langle E \rangle \) and the notion of \([L]\) is meaningless otherwise.

## 2 Spanning Tree Expansion of the Tutte Polynomial

It is unfortunate that spanning trees and binary trees are both called “trees.” However, we wish to avoid any possible confusion by stating the difference that a spanning tree is a graph theoretic concept, whereas any binary tree such as \( \langle E \rangle \) or the computation trees we shall encounter later are the familiar set theoretic notions.

**Definition.** Consider the graph \( G = (V, E) \), we shall draw no distinction between a subset of edges \( S \subset E(G) \) and the subgraph spanned by such set. We let \( \mathcal{G} \) be the set of isomorphism classes of all finite multigraphs; the Tutte polynomial is the map \( \mathcal{G} \rightarrow \mathbb{Z}[x, y], G \mapsto T(G; x, y) \), given by

\[
T(G; x, y) = \sum_{S \subseteq E(G)} (x - 1)^{k(S) - k(E)}(y - 1)^{n(S)},
\]

where \( k(S) \) is the number of connected components of \( S \) (likewise for \( E \)) and \( n(S) = |S| - |V(S)| + k(S) \) is the nullity of \( S \).

The Tutte polynomial satisfies the so-called deletion-contraction relation. Consider a graph \( G = (V, E) \) and let \( e = xy \in E \) where \( x, y \in V \) be an arbitrary edge of \( G \). We give the deletion and contraction processes as described below:

**Definition.** To **delete** an edge \( e = xy \), we simply remove \( e \) from \( E \). We call the resulting graph \( G - e \). We do not delete bridges. To **contract** an edge \( e = xy, x \neq y \), we identify \( x \) with \( y \) and remove \( e \) from \( E \). We call the resulting graph \( G/e \). We do not contract loops.
\[ T(G; x, y) = \begin{cases} 
  xT(G/e; x, y) & \text{if } e \text{ is a bridge.} \\
  yT(G - e; x, y) & \text{if } e \text{ is a loop.} \\
  T(G/e; x, y) + T(G - e; x, y) & \text{if } e \text{ is neither a bridge nor a loop.} \\
  1 & \text{if } G \text{ has no edges.} 
\end{cases} \]

It is easy to see that our definition of the Tutte polynomial indeed satisfies this relation\(^2\). In practice, the deletion-contraction relation offers a convenient way to calculate the Tutte polynomial, a process that will play a central part of this paper. Since the Tutte polynomial is well-defined, the order of edges during the computation does not affect the final sum. Moreover, we shall consider only connected graphs from now on as the Tutte polynomial is multiplicative on disjoint graphs, a property that can be readily observed from the spanning subgraph expansion.

**Definition.** A minor of a graph \( G \) is a graph that can be obtained from \( G \) by a sequence of deletions and contractions. A computation tree of a graph \( G \) is a binary tree where each node is associated with a pair \((G', e)\) where \( G' \) is a minor of \( G \) and \( e \) is an edge of \( G' \). The nodes are related in such a way that the root is always the pair \((G, e)\) with some arbitrary edge \( e \), and the children of a node \((G', e)\) are \((G' - e, f)\) and \((G'/e, f)\), for arbitrary \( e \) and \( f \); in the case where \( e \) is a bridge or a loop, \((G/e, f)\) or \((G - e, f)\) corresponds to a unitary node in the computation tree. A leaf is always \((E_1, \phi)\), where \( E_1 \) is the graph with a single vertex.

**Lemma 1.** A computation tree of \( G \) is isomorphic to a \((T, E)\)-tree where \( T \) is the set of spanning trees of \( G \), and \( E \) is the set of edges of \( G \).

**Proof.** Each node in a computation tree is naturally identified with an edge of \( G \). Since each edge is computed exactly once, we know that the computation tree is an \( E \)-tree. This allows us to associate each leaf of the computation tree with a spanning subgraph interval, which together form a complex partition of the space \( \{0, 1\}^{|E|} \). All that remains to be shown is that the set of leaves in a computation tree is injective to the set of spanning trees of \( G \).

First, we claim that the set of edges contracted for each leaf in a computation tree correspond to a spanning tree. To show this, we note that the contraction process preserves nullity, and therefore the set of edges contracted can never form a cycle, making it a tree. Since the graph is computed until no edge remains, we know that number of edges contracted must be \( V(G) - 1 \), which means it must be a spanning tree. Finally, we claim that no leaf of a computation tree contains two spanning trees. Suppose for contradiction that this were the case, we have \( T \) and some \( T - e \cup f \) (or \( T \cup e - f \)) as the two trees. This suggests a cycle in \( G \) that contains both \( e \) and \( f \) implying that either we have contracted

\(^2\)For a proof, see Thm. X.1 in [2]
a loop (or have deleted a bridge, respectively) or that \( T \) is not a spanning tree, neither of which is possible, by construction. This proves injection, and thus the lemma.

Figure 2: A computation tree for \( K_3 \), and its associated spanning tree expansion of the Tutte polynomial.

In [2], Bollobas gives a spanning tree expansion of the Tutte polynomial (in fact, it was Tutte’s original definition), and proved that it indeed satisfies the deletion-contraction relation. With our groundwork in \( E \)-trees, we will now be able to derive the spanning tree expansion as a collapse of the spanning subgraph expansion.

**Definition.** In a computation tree, for each leaf \( T \) (i.e. a spanning tree), we call the set of unitarily contracted nodes on its root-path *internally active*, denoted \( a(T) \) and the set of unitarily deleted nodes *externally active*, denoted \( b(T) \). For instance, the subgraph interval associated with a spanning tree is \([T \setminus a(T), T \cup b(T)]\).

**Lemma 2.** The definition of activity introduced above is a generalization Tutte’s original definition of activity.

**Proof.** Tutte originally gave his activity by choosing some arbitrary global order \( \prec \) on the edges and calling an edge \( e \) internally active if it is the least edge that joins the two components in \( G - e \), and an edge \( e \) externally active if it is the least edge in the unique cycle in \( G \cup \{ e \} \). To show that our definition is equivalent to his, we choose the leaf order of a tree \( \prec_T \) to be our choice of edge order. Now as the graph is computed along our local order with the greatest edge computed first, an edge will be unitarily deleted if it is the last edge to be computed in a cycle (i.e. a loop); an edge will be unitarily contracted if it is the last edge to
be computed between two given components (i.e. a bridge). This agrees with our definition.

**Lemma 3.** For all \( S \in [T] \), we have \( k(S) - 1 = |\alpha| \) and \( n(S) = |\beta| \), where \( \alpha \) (\( \beta \), respectively) is a subset of \( a(T) \) (\( b(T) \)) and is the set of edges removed from (added to) \( T \) to obtain \( S \).

*Proof.* We first note that for \( S \) associated with a spanning tree \( T \), we can create new components by removing edges in \( T \), and likewise add cycles by adding edges to \( T \). These numbers correspond to \( k(S) - 1 \) and \( n(S) \). It now remains to show that \( \alpha \) and \( \beta \) are independent, in that adding or removing an edge can never reconnect two components or beak a previously formed cycle. To show this, let us consider the generalized Tutte activity in the previous lemma. For an edge to be added back to rejoin two components, it must be an externally active edge. This contradicts our construction as the edge previously removed is the least edge that joins the two components. A similar argument can be given to show that no edge can be taken away to break an already formed cycle. This proves the lemma.

**Theorem 2.** The spanning subgraph expansion of the Tutte polynomial is equivalent to the spanning tree expansion

\[
\sum_{S \subseteq E(G)} (x - 1)^{k(S) - 1}(y - 1)^{n(S)} = \sum_{T \in \mathcal{T}} x^{|a(T)|} y^{|b(T)|}.
\]

*Proof.* Lemma 1 allows us to associate each interval in the complex partition of the set of spanning subgraphs with a unique spanning tree. It remains to show that each interval contributes to exactly one monomial in the spanning tree expansion, which we do using Lemma 3:

\[
\sum_{S \in [T]} (x - 1)^{k(S) - 1}(y - 1)^{n(S)} = \sum_{\alpha \subseteq a(T), \beta \subseteq b(T)} (x - 1)^{|\alpha|}(y - 1)^{|\beta|} = \sum_{\alpha \subseteq a(T)} (x - 1)^{|\alpha|} \sum_{\beta \subseteq b(T)} (y - 1)^{|\beta|} = (x - 1 + 1)^{|a(T)|}(y - 1 + 1)^{|b(T)|} = x^{|a(T)|} y^{|b(T)|}.
\]

\[
\square
\]

3 Quasi-Tree Expansion of the Bollobas-Riordan Polynomial

**Definition.** A cyclic graph or embedded graph \( G^\pi \) with \( n \) edges can be described by the pair of permutations \((\sigma, \alpha)\), where \( \sigma \) is a permutation such that each cycle
in the permutation corresponds to the order of half-edges in the counterclockwise
direction around a vertex, and \( \alpha \) is a fixed-point free involution. Note that \( \sigma \)
completely describes \( V(G^\pi) \) and \( \alpha \) completely describes \( E(G^\pi) \), except for the
case where the graph is empty. Lastly, note that faces can be calculated by
\( \phi = \sigma \cdot \alpha \). We multiply on the right.

Figure 3: A cyclic graph with genus 1.

**Definition.** The Bollobas-Riordan polynomial is a generalization of the Tutte
polynomial for cyclic graphs. It is defined according to
\[
C(G^\pi; x, y, z) = \sum_{S \subseteq E(G)} (x - 1)^{k(S) - k(E)} y^{n(S)} z^{g(S)},
\]
where \( g \) denotes the genus of the cyclic graph, calculated according to Euler’s
formula
\[
g = 1 - \frac{1}{2} (z(\sigma) + z(\alpha) + z(\sigma \cdot \alpha)),
\]
where \( z \) counts the number of cycles in a permutation.

In [3], a spanning tree expansion similarly to that of Tutte polynomial is given,
using a definition of activity similar to Tutte’s. We are not, however, interested
in this expansion as it is very similar to the spanning tree expansion of the Tutte
polynomial. In [5], Champanerkar, et al. give an expansion of the Bollobas-
Riordan polynomial using the so-called quasi-trees. Our task here is to place
the quasi-tree expansion in our framework. We shall start by first giving our
definition of its own computation tree.

**Definition.** A loop whose two half-edges partition the vertex they incident into
two disjoint half-edge cycles is called a *true loop*. A loop that is not a true loop
is called a *quasi-loop*. All bridges are “true bridges”. To delete an edge, we
remove the two half edges from \( \sigma \) and \( \alpha \). We do not delete bridges. To contract
an edge, we remove the two half edges from \( \phi = \sigma \cdot \alpha \) and, take \( \phi' = \phi' \cdot \alpha \).
We do not delete true loops.
We can now see that the Bollobas-Riordan polynomial satisfies the deletion-contraction relation given below

\[
C_{G^*} = \begin{cases} 
xC_{G^*/e} & \text{if } e \text{ is a bridge.} \\
(y + 1)C_{G^*-e} & \text{if } e \text{ is a true loop.} \\
C_{G^*/e} + C_{G^*-e} & \text{if } e \text{ is ordinary.} \\
\sum_{S \subseteq E(G^*)} y^{n(S)} z^{g(S)} & \text{if } G^* \text{ has a single vertex.}
\end{cases}
\]

The contraction of quasi-loop gives a more complicated recursion relation, and we shall not treat it in our discourse.

**Definition.** A quasi-tree \(Q(\sigma, \alpha) \in \mathcal{Q}\) is a spanning subgraph of a cyclic graph such that \(z(\sigma \cdot \alpha) = 1\). A quasi-computation tree for a cyclic graph is analogous to a computation tree with the restriction that each leaf is a minor with no contractable or deletable edges.

We would like to note here that for a (quasi-)computation tree of a graph \(G\), there exists a dual (quasi-)computation tree for \(G^*\) such that deletion and contraction are duals with each other, and (true) bridges are duals of (true) loops. Moreover, in quasi-computation trees, quasi-loops are duals of themselves.

**Figure 4:** A quasi-computation for a cyclic graph and its dual quasi-computation tree; the subgraphs in boxes are respective \(S\) and their \(S'\). Note the duality of contraction and deletion.
Lemma 4. A quasi-computation tree of $G^\pi$ is isomorphic to a $(Q, E)$-tree where $Q$ is the set of quasi-trees of $G^\pi$, and $E$ is the set of edges of $G^\pi(\sigma, \alpha)^3$.

Proof. It should be self-manifest at this point that the quasi-computation tree is an $E$-tree. It remains to show that the set of leaves in $\langle E \rangle$ is injective to $Q$. For a leaf in a quasi-computation tree, call the graph spanned by the set of edges contracted $S$. Since $G^\pi$ is computed to a single-vertex graph, we know $S$ is a connected spanning subgraph of $G^\pi$. Let $S'$ be the subgraph of the graph $(\sigma \cdot \alpha, \alpha)$, the dual of $G^\pi$, such that the $E(S')$ is the complement of $E(S)$ in $E(G^\pi)$. Since deletion and contraction are duals of each other, we know $S'$ is connected as well since we do not contract true loops in the computation steps for $S$, and therefore no bridges are deleted in the computation steps for $S'$. Now the difference between the genus of $S$ and the genus of $G^\pi$ correspond to the number of quasi-loops deleted in the computation steps for $S$. These are the quasi-loops that appear in $S'$, and we have the relation

$$g(S) + g(S') = g(G^\pi).$$

We now add the two Euler’s equations for $S$ and $S'$

$$v(S) - (e(S) + e(S')) + v(S') + f(S) + f(S') = 4 - 2g(S) - 2g(S')$$

$$v(G^\pi) - e(G^\pi) + f(G^\pi) + f(S) + f(S') = 4 - 2(g(S) + g(S'))$$

$$2 - 2g(G^\pi) + f(S) + f(S') = 4 - 2(g(S) + g(S'))$$

$$f(S) + f(S') = 2,$$

This implies that $f(S) = f(S') = 1$. Lastly, to see that there are no two quasi-trees in the same leaf interval, one can check to see that this always correspond to a forbidden calculation step, such as deleting a bridge or contracting a true loop.

Definition. In a quasi-computation tree, the analogues of internally and externally active edges are the internally and externally live edges. An edge that is not live is called dead.

The argument in Lemma 2 in [5] shows exactly that the property of live and dead edges here that we have taken as definition is equivalent to the original definition in [5]. We would like to note, however, even though the construction in [5] was based on some globally ordered $\langle Q, E \rangle$, the argument for the proof in this particular lemma in [5] can be applied to any locally ordered $\langle Q, E \rangle$.

At this point, we arrive at the same quasi-tree expansion of the Bollobas-Riordan polynomial as the one stated in Theorem 1 in [5]. The proof is given there and we shall omit it for brevity. This concludes our extension on the quasi-tree theory.

9

---

3This is analogous to Lemma 1.
4 Summary and Further Generalization

Consider any polynomial graph invariant $P_G$ with some spanning subgraph expansion that satisfies the deletion-contraction relation. We now extend our construction to any arbitrary $(\mathcal{U}, E)$-tree expansion to $P_G$, for $\mathcal{U} \subseteq E(G)$. The steps here involve

1. Characterizing the set $\mathcal{U}$,
2. Describe a similar computation tree procedure for the polynomial in question,
3. Finding an interval sum for the $(\mathcal{U}, E)$-tree partition.

Acknowledgement: The author would like to thank his advisor Olivier Bernardi of the Department of Mathematics at the Massachusetts Institute of Technology for his kind mentorship and consistent guidance throughout the writing of this paper.

5 References