An Exploration of the Tutte Polynomial in Hypergraphs

Authors: Lucas Rusnak, Alice Guo, Andy Jiang, Oliver Kahn, Rich Wang

September 20, 2021

1 Abstract

The Tutte polynomial encodes the structure of the graph and can be used to count various things from spanning trees, to acyclic orientations, to colorings, and even flows. We will examine hypergraphic and sign graphic analogs of the Tutte polynomial to generalize various graph concepts. The Tutte Polynomial of a graph is already well-defined through deletion and contraction and through activities of edges in spanning trees. A graph's bipartite graph will also correspond to, or represent the graph's hypergraph dual. By examining the relationship between a graph and its bipartite through various lenses such as deletion/contraction, Tutte polynomial reclamation, and Dhar's Burning Algorithm, we hope to find a bridge that allows us to generalize such topics for hypergraphs.

2 Background

2.1 Elementary Graph Theory

Let's start by introducing some basic graph theory concepts and establishing a clear notation system.

Definition 2.1.1. An **ordinary graph** $G = (V_G, E_G)$ has a finite vertex set V_G and finite edge set E_G, where an edge $e \in E_G$ connecting two vertices $u, v \in V_G$ can be expressed as (u, v) or (v, u). u and v are the endpoints of edge e.

Consider a graph $G = (V_G, E_G)$. One of the two special types of edges that will be focused on in this paper is the loop.

Definition 2.1.2. An edge $e = (u, v)$ is a **loop** if it has both of its endpoints at the same vertex (i.e. $u = v$).

We will introduce some other terms before defining the isthmus, which is the second special type of edge.

Definition 2.1.3. A **walk** on G is a sequence of n edges connecting a sequence of $n+1$ vertices $v_0v_1...v_n$ such that $v_0, v_1, ...v_n \in V_G$ and $(v_{i-1}, v_i) \in E_G$ for all $i \in \{1...n\}$. A **trail** is a walk with distinct edges and a **path** is a trail with distinct vertices; we say that a path exists between two vertices if they are the first and last vertices in the sequence of vertices of some path in G.

Definition 2.1.4. A **cycle** in G is a non-empty trail in which the first vertex is equal to the last vertex and no other vertices are repeated.

We will choose to let loops be cycles, as a loop $e = (u, u)$ can be represented as a cycle in the form of the finite vertex sequence uu. Now, let us look at the definition of the isthmus.

Definition 2.1.5. An edge $e = (u, v)$ is an **isthmus** if there is exactly one path between u and v, specifically the path consisting of only e itself. In other words, there are no cycles containing both u and v.

Definition 2.1.6. A **subgraph** G' of G is a graph $G' = (V_{G'}, E_{G'})$ where $V_{G'} \subset V_G$ and $E_{G'} \subset E_G$, such that if $(u, v) \in E_{G'}$ then $u, v \in V_{G'}$. G' is a **spanning subgraph** if $V_{G'} = V_G$. G' is an **induced subgraph** if for any two $u, v \in V_{G'}$, $(u, v) \in E_{G'}$ if and only if $(u, v) \in E_G$ (i.e. if $E_{G'}$ is the set of all edges with endpoints in $V_{G'}$).

Definition 2.1.7. An induced subgraph $G' = (V_{G'}, E_{G'})$ of G is a **connected component** if for any two $u, v \in V_{G'}$, there exists a path between u and v, and for any $w \in V_G$ but $w \notin V_{G'}$, there does not exist a path between u and w.

Note that a graph can be partitioned into disjoint connected components, meaning that the number of connected components in a graph can easily be counted. Additionally, we say that two vertices are **connected** if they reside in the same connected component.

Definition 2.1.8. Assume G only has one connected component. A **spanning tree** t of G is a spanning subgraph of G with no cycles. The set of spanning trees of G is denoted as τ_G.

This implies that every edge in t is an isthmus, or that there is exactly one path between each pair of vertices. Recall that since loops are cycles, spanning trees will never contains loops. It also comes naturally that the number of edges in any spanning tree of G will always be $|V_G| - 1$.

Definition 2.1.9. Also known as the circuit rank, cycle rank, or nullity of a graph, the **cyclomatic number** of a graph, ϕ, is the minimum number of edges that must be deleted in order for the graph to have no cycles.

The cyclomatic number can also be thought of as the maximum number of edges that can be deleted such that two connected vertices remain connected. Thus, it is the number of edges that must be removed in order to reach a spanning tree for each connected component, each of whose size we know is fixed. The cyclomatic number of G can therefore be computed using the formula

$$\phi = |E_G| - |V_G| + c,$$

where c is the number of connected components in the graph.

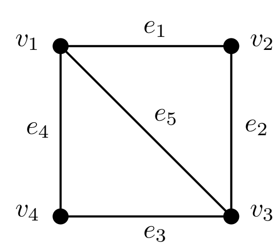
2.2 Deletion and Contraction

Now we introduce two basic graph operations on a graph $G = (V_G, E_G)$: **edge deletion** and **edge contraction**. Let edge $e = (u, v) \in E_G$.

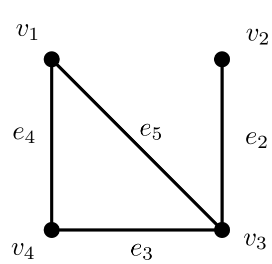
Definition 2.2.1. $G \backslash e$ refers to the graph of G delete e; mathematically, $G \backslash e = (V_G, E_G - \{e\})$.

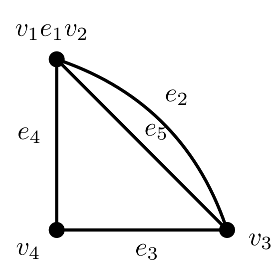
Definition 2.2.2. G/e refers to the graph of G contract e. Vertex u, edge e, and vertex v are merged into one single vertex denoted as uev, and all edges besides e are preserved. Any edge that had an endpoint at u or v now has that endpoint at uev. Loops cannot be contracted.

Example 2.2.3. Consider the following graph, G:



Below are the graphs $G \backslash e_1$ and G/e_1 on the left and right, respectively.





2.3 Edge activities

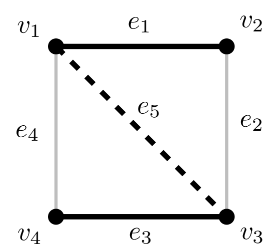
To introduce the idea of edge activities, we now impose an ordering on the edges of a graph. For a graph $G = (V, E)$, we can replaced the unordered edge set $E = \{e_1, e_2, ...e_m\}$ with an ordered edge list $\vec{E} =< e_1, e_2, ...e_m >$, where $\vec{E}$ is ordered from least weighted to most weighted going from left to right. Unless a different edge ordering is specified, e_i has a larger weight than e_j if $i > j$.

Consider a graph $G = (V, \vec{E})$ with only one connected component. Given a spanning tree t of G, we can categorize edges in G as internal or external as well as active or not active. An edge is **internal** if it is in t and **external** otherwise.

Definition 2.3.1. Let e be an internal edge and consider the graph $t \backslash e$. e is *active*, in this case **internally active**, if it is maximal (has the highest weight) among all edges that would form some spanning tree of G if added to $t \backslash e$.

In other words, an edge e is internally active if it is the edge with maximal weight that restores the connectivity of the spanning tree upon e's deletion from the tree.

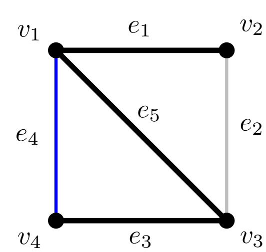
Example 2.3.2. Below is a spanning tree of G that contains e_1, e_3, e_5. Consider the internal edge e_5.



When e_5 is deleted, either e_2, e_4, or e_5 can be added back to the graph to maintain connectivity. Between the three edges e_2, e_4, and e_5, the internal edge e_5 is maximal due to our assumed edge ordering, where $e_i > e_j$ when i > j. Thus, e_5 is internally active.

Definition 2.3.3. Let e be an external edge and consider the unique cycle formed when e is added to the spanning tree t. e is *active*, in this case **externally active**, if it is maximal among all edges in that unique cycle.

Example 2.3.4. Using the spanning tree from the previous example, consider the external edge e_4.



When e_4 is added to t, it forms a cycle that consists of edges e_3, e_4, and e_5. Among these edges, e_4 is not maximal as it has a smaller order than e_5, hence e_4 is not externally active.

2.4 Tutte Polynomial

The focus of our paper is the Tutte polynomial, a versatile graph polynomial that can be computed using several methods.

Definition 2.4.1. Using activities, the **Tutte Polynomial** T_G of a graph $G = (V, \vec{E})$ is defined as

$$T_G(x, y) = \sum_{t \in \tau_G} x^{t_i} y^{t_e}$$

where for every spanning tree t in the set of spanning trees τ_G, t_i is the number of internally active edges in $\vec{E}$ and t_e is the number of externally active edges in $\vec{E}$.

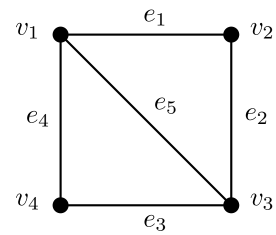
Theorem 2.4.2. A well known theorem is that the Tutte polynomial for a graph $G = (V, E)$ can also be computed using deletion-contraction recursion as follows:

$$T_G(x,y) = \begin{cases} 1, & \text{if } E = \varnothing \\ x^n y^m, & \text{if } G \text{ is comprised of } n \text{ isthmi, } m \text{ loops, and no other edges} \\ T_{G\setminus e}(x,y) + T_{G/e}(x,y) & \text{if edge } e \in E \text{ is neither an isthmus nor a loop} \end{cases}$$

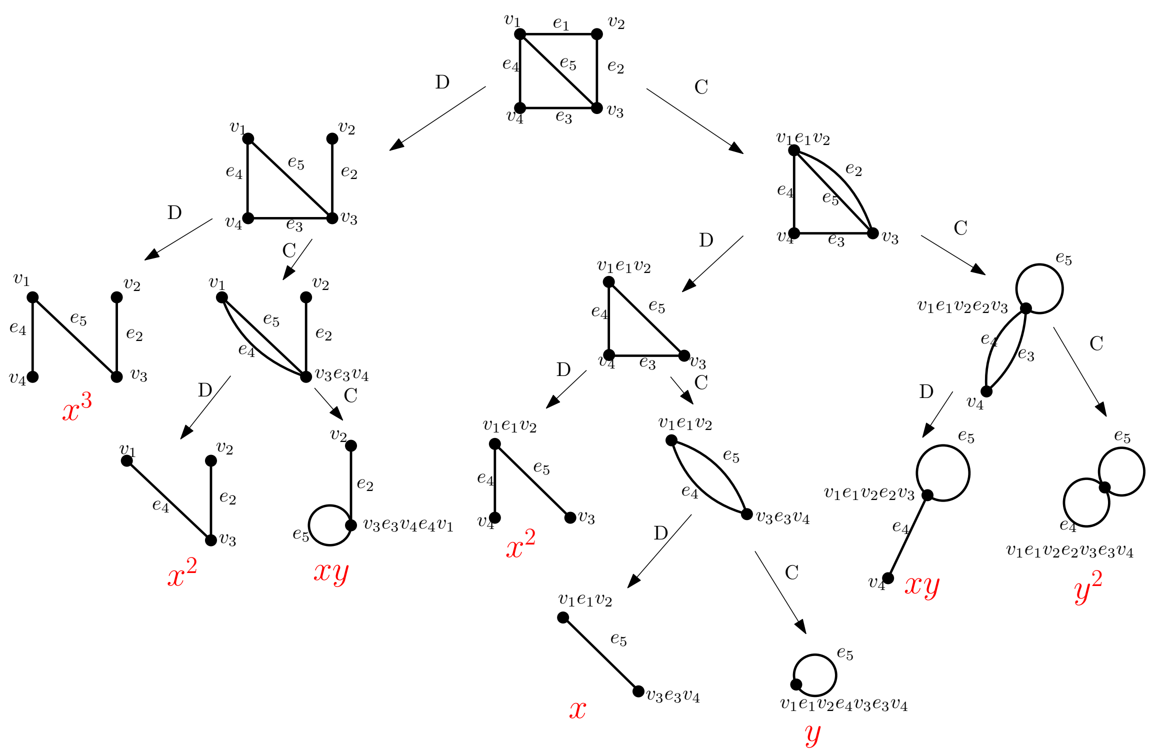
Note that the graph used for this does not need to have an ordered edge set; in other words, the Tutte polynomial that results from this deletion-contraction recursion is the same regardless of what edge is treated first, what edge is treated second, etc.

Definition 2.4.3. A graph comprised of only isthmi and loops is called a **terminal minor**. Thus, the deletion-contraction recursion in Theorem 2.4.2 stops once it reaches a terminal minor.

Example 2.4.4. Consider the following graph, G:



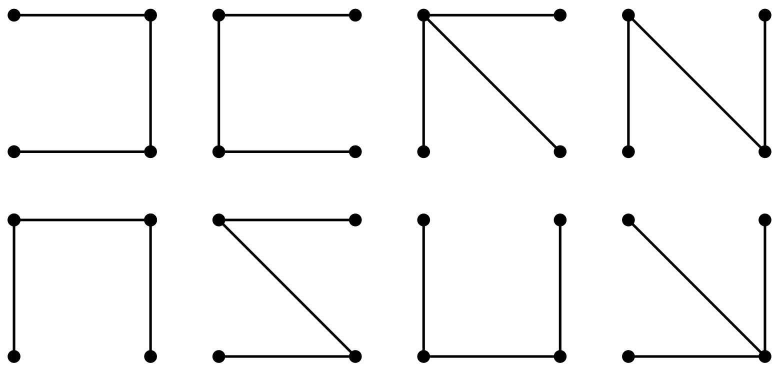
Using deletion and contraction, we can reduce this graph down to its 8 terminal minors, shown below.



Thus, the Tutte polynomial of G can be computed using Definition 2.4.2, resulting in

$$\begin{aligned} T_G(x,y) &= x^3 + x^2 + xy + x^2 + x + y + xy + y^2 \\ &= x^3 + 2x^2 + x + 2xy + y + y^2 \end{aligned}$$

Each of these terminal minors corresponds to a term in the graph's Tutte polynomial and a distinct spanning tree of the graph G, which in this case has 8 distinct spanning trees:



3 Minor-Activity Tutte Polynomial

3.1 Minor-Activity Tutte Polynomial

Having looked at the two variable Tutte polynomial in terms of x and y, we now devise a minor-activity Tutte polynomial in terms of x, y, c, d, given an edge ordering $\vec{E}$; we can see that this new polynomial is non-commutative because it has subscripts below each letter to indicate which edge the letter corresponds to.

Definition 3.1.1. We define the minor-activity Tutte polynomial on ordinary graphs as follows:

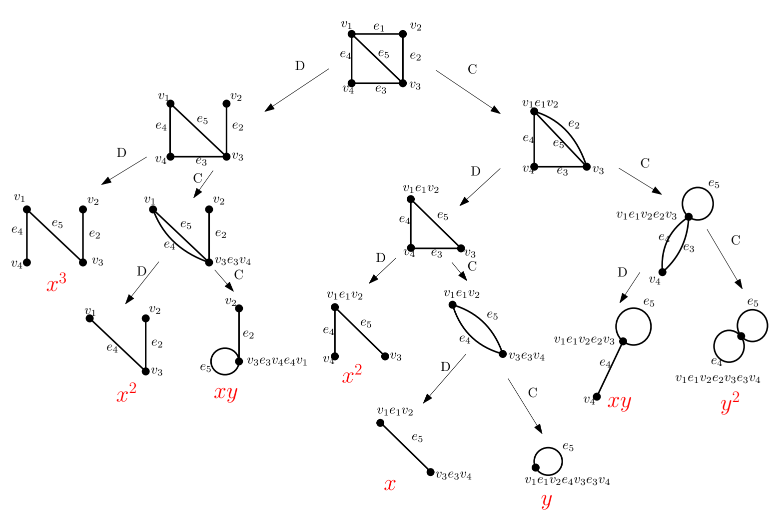
$$T_G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{d};\vec{E}) = \sum_{t\in\tau}\prod_{e\in\vec{E}} A_t(e)$$

where

$$A_t(e) = \begin{cases} x_e, & \text{if } e \text{ is an isthmus} \\ c_e, & \text{if } e \text{ is contracted} \\ y_e, & \text{if } e \text{ is a loop} \\ d_e, & \text{if } e \text{ is deleted} \end{cases}$$

and $\boldsymbol{x} = \{x_1, x_2, ...x_{|E|}\}$ is the vector set of x's, $\boldsymbol{y} = \{y_1, y_2, ...y_{|E|}\}$ is the vector set of y's, etc.

Example 3.1.2.



$$T_G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{d};\vec{E}) = d_{e_1}x_{e_2}d_{e_3}x_{e_4}x_{e_5} + d_{e_1}x_{e_2}c_{e_3}d_{e_4}x_{e_5} + d_{e_1}x_{e_2}c_{e_3}c_{e_4}y_{e_5} + c_{e_1}d_{e_2}d_{e_3}x_{e_4}x_{e_5} + c_{e_1}d_{e_2}c_{e_3}d_{e_4}x_{e_5} + c_{e_1}d_{e_2}c_{e_3}c_{e_4}y_{e_5} + c_{e_1}c_{e_2}d_{e_3}x_{e_4}y_{e_5} + c_{e_1}c_{e_2}c_{e_3}y_{e_4}y_{e_5}$$

From this point onward, we can remove the subscripts of the letters because we ensure that letters appear in the order of the assigned edge ordering $\vec{E}$. Thus, the minor-activity Tutte polynomial can be written more simply as

$$T_G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{d};\vec{E}) = dxdxx + dxcdx + dxccy + cddxx + cdcdx + cdccy + ccdxy + cccyy$$

3.2 Relationship between Deletion/Contraction and Internal/External Activity

Lemma 3.2.1. *Given an edge ordering and a spanning tree t created from a sequence of x's, y's, c's, and d's, an edge $e \in t$ is internally active if and only if it was an isthmus when doing deletion/contraction.*

Proof. Assume e is internally active in t. Consider the graph of t with edge e deleted. The graph will have two disjoint components, and because e is internally active, any non-e edges that reconnect the graph must have order less than that of edge e. These edges are not in spanning tree t and have order less than e, so while doing deletion/contraction, they must be deleted, and edge e will be the only edge left that maintains the connectivity of the graph, hence it is an isthmus.

Conversely, assume that while doing deletion/contraction, edge e is an isthmus. Suppose e is not internally active. Then there exists an edge f, $f > e$ such that edges a, b, f form a spanning tree. But since $f > e$, f has not been dealt with yet, so it still exists in the graph. That means e exists as part of a cycle, which means that e is not an isthmus. ☐

Lemma 3.2.2. *Given an edge ordering and a spanning tree t created from a sequence of x's, y's, c's, and d's, an edge $e \notin t$ is externally active if and only if it was a loop when doing deletion/contraction.*

Proof. Assume e is not externally active. Because e is not externally active, then e must be a part of a cycle C_i along with some other edge d that is maximal to e. However, it is impossible for e to be a loop while doing deletion/contraction, because this means that every other edge in C_i must have been contracted in order for the two endpoints of e to have been pushed together. But because d is maximal to e, this is impossible. Thus, edge e must be deleted, and cannot be a loop, as desired.

Conversely, assume that while doing deletion/contraction, edge e is deleted. If you can delete e, it must be part of a cycle, as it can't be a loop or isthmus, and thus there must exist other greater edges (following our edge ordering), that will be part of the spanning tree, which makes e not externally active. If those other greater edges aren't part of the spanning tree, they had to have also been deleted and there is another greater edge in the spanning tree. ☐

Theorem 3.2.3. *Given graph G and a total ordering on the edges $\vec{E}$,*

$$T_G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{d};\vec{E}) = \sum_{t\in\mathcal{T}} \prod_{e\in\vec{E}} A_t(e)$$

where

$$A_t(e) = \begin{cases} x_e, & \text{if } e \text{ is an active internal edge in } t \\ c_e, & \text{if } e \text{ is a non-active internal edge in } t \\ y_e, & \text{if } e \text{ is an active external edge in } t \\ d_e, & \text{if } e \text{ is a non-active external edge in } t \end{cases}$$

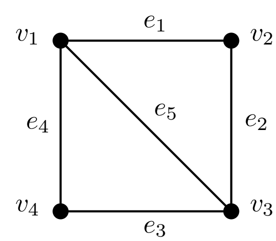
Theorem 3.2.4. *Using deletion-contraction recursion, the non-commutative Tutte polynomial can be defined as follows: given graph $G = (V, E)$ and a total ordering on the edges $\vec{E}$,*

$$T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\vec{E}) = \begin{cases} 1, & \text{if } E = \varnothing \\ x_e \cdot T_{G/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\vec{E} - \{e\}), & \text{if } e \text{ is an isthmus in } G \\ y_e \cdot T_{G\backslash e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\vec{E} - \{e\}), & \text{if } e \text{ is a loop in } G \\ d_e \cdot T_{G\backslash e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\vec{E} - \{e\}) + c_e \cdot T_{G/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\vec{E} - \{e\}), & \text{if } e \text{ is neither an isthmus nor a loop in } G \end{cases}$$

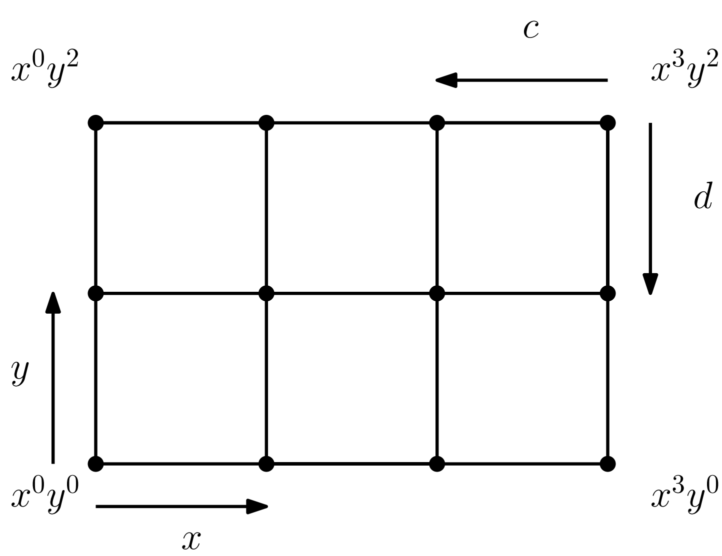
where e is the least weighted edge in $\vec{E}$.

3.3 Grid Walking with the Non-commutative Tutte Polynomial

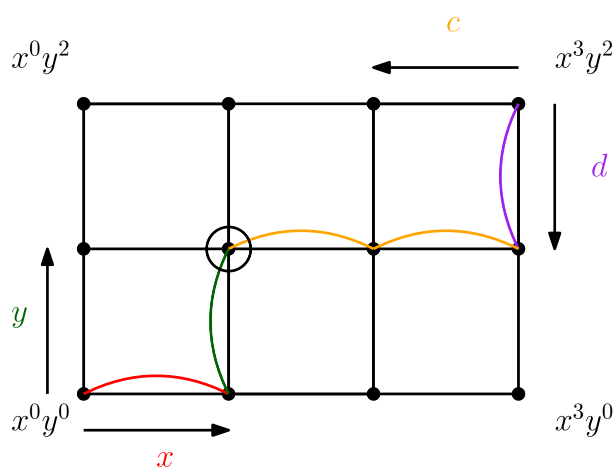
Once again, consider the following graph.



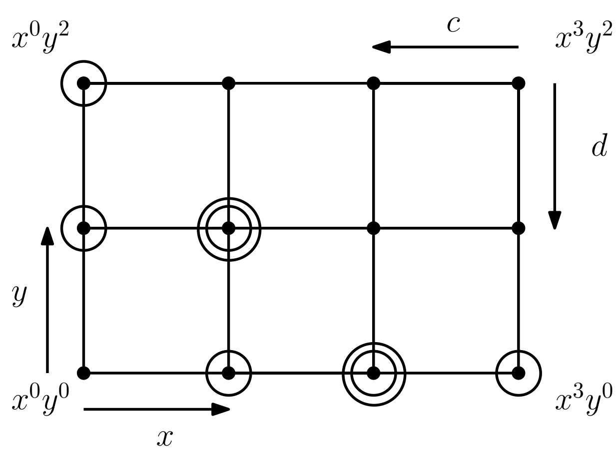
Each term of its non-commutative Tutte polynomial can be plotted through a grid walk, where the height of the grid is the cyclomatic number and the length is the rank of the graph. In other words, let the number of isthmi equal i, number of loops equal ℓ, the size of a spanning tree equal r, and the cyclomatic number equal ϕ. We then visualize the grid from $x^i y^\ell$ to $x^r y^\phi$ as follows:



Each time an x or y appears in the non-commutative Tutte polynomial string, we walk right and up, respectively. We begin starting in the bottom left corner of the grid, which corresponds to x^0y^0.
When a c or d appears, we walk left and down, respectively, starting from the top right corner, which corresponds to x^3y^2



Above is the grid-walk example for the spanning tree that corresponds to the non-commutative Tutte monomial, dxccy.



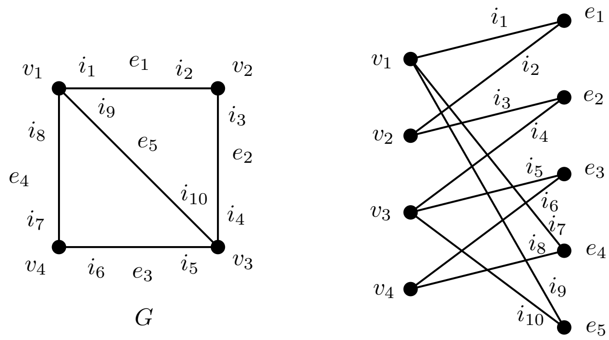
The above grid plots the grid-walking for all monomials in T_G with the respective nodes circled, with multiplicity in some cases.

Observation: Any path from one corner to the other must pass through at least one of these indicated nodes.

4 Bipartite Representation

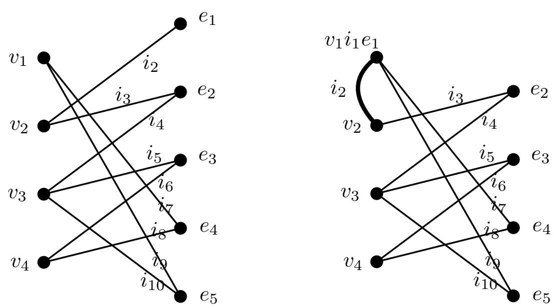
Definition 4.0.1. Given a graph $G = (V, E)$, an **incidence** $i = (v, e)$ is the connection between a vertex $v \in V$ and an edge $e \in E$ that has v has an endpoint. The set of all incidences in G is I.

Definition 4.0.2. The **bipartite representation graph** Γ_G of a graph G is a graph where edges of G are treated as vertices in addition to the original vertices of G, and an original vertex of G is connected to an edge in G by an edge incidence. In more concrete terms, $\Gamma(G) = (V \cup E, I)$.



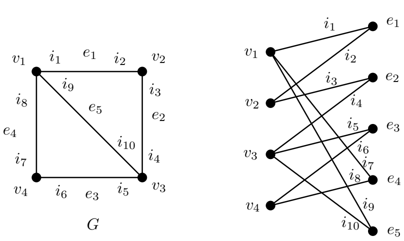
G $\quad$ Γ_G

Since Γ is a graph itself, we can delete and contract the edges of Γ; i.e. the incidences of G.



$\Gamma - i_1$ $\quad$ Γ / i_1

There are significantly more terms, and we can see that a single incidence deletion makes e_1 a 1-edge, while contraction merges a vertex and an edge. The resulting Γ is not bipartite, so single deletion/contraction actions have no meaning in graph G.



G $\quad$ Γ_G

The Tutte polynomial for Γ is:

$$
\begin{aligned}
T_\Gamma(x,y) = {} & x^8 + x^7 + x^6 + x^5 + x^4 + x^3 y \\
& + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 y \\
& + x^6 + x^5 + x^4 + x^3 + x^2 + xy \\
& + x^5 + x^4 + x^3 y + x^2 + x + y \\
& + x^4 + x^3 y + x^2 + xy \\
& + x^3 y + x^2 y + xy + y^2
\end{aligned}
$$

Using the 4-variable Tutte polynomial, we delete and contract the incidences according to the *edge order* from G, which is exemplified by the following lemma.

Lemma 4.0.3. *It is possible to reclaim the Tutte Polynomial of a graph from the Tutte Polynomial of the graph of its bipartite representation.*

- *X in G is equivalent to xx in Γ,*
- *Y in G is equivalent to cy in Γ,*
- *C in G is equivalent to cc in Γ,*
- *D in G is equivalent to cd in Γ*

4.1 Reclaiming the Tutte Polynomial of a graph from the Tutte polynomial of its bipartite graph

Let G be an undirected graph with vertex set V and edge set E and let Γ be the bipartite graph of G with vertex set $V \bigcup E$ and edge set I, where $I \subset V \times E$ is the set of incidences.

Consider a deletion of edge $a \in E$ that connects vertices v_1 and v_2. Within the bipartite graph of G, a is a vertex, and a deletion of a requires us to contract then delete its incidence-edges, $i_a u$ and $i_a v_2$ (where $i_a v_i$ denotes the incidence between edge a and vertex v_i).

Lemma 4.1.1. *$G \backslash a$ is equivalent to $\Gamma / i_a v_1 \backslash i_a v_2$.*

Proof. Because i_a initially had degree two in Γ, with edges $i_a v_1, i_a v_2$, contracting i_a, v_1 and then deleting edge $i_a v_2$, gets rid of both of the initial edges of i_a and pushes i_a into v_1.

This is equivalent to what happens when we delete edge a in the initial graph G, as in essence, the vertex representing edge a in Γ has disappeared. ☐

Lemma 4.1.2. *G/a is equivalent to $\Gamma / i_a v_1 / i_a v_2$.*

Proof. Because i_a initially had degree two in Γ, with edges $i_a v_1, i_a v_2$, contracting i_a, v_1 and then contracting edge $i_a v_2$, gets rid of both of the initial edges of i_a and pushes together the vertices i_a, v_1, v_2.

This is equivalent to what happens when we contract edge a in the initial graph G, as edge a is removed but v_1, v_2 are forced together. ☐

Lemma 4.1.3. *X in G is equivalent to xx in Γ.*

Proof. Because i_a initially had degree two in Γ, with edges $i_a v_1, i_a v_2$, if $i_a v$ is an isthmus, then $i_a v_2$ must be as well. Notice that both $i_a v_1, i_a v_2$ being isthmi implies that edge $a = v_1 v_2$ must also be an isthmus in G.

Similarly, because the adjacencies of edge a to v_1, v_2 are represented by $i_a v_1, i_a v_2$ in Γ, if edge a is an isthmus in G, then $i_a v_1, i_a v_2$ must be isthmi in Γ. ☐

Lemma 4.1.4. *Y in G is equivalent to cy in Γ.*

Proof. Because i_a initially had degree two in Γ, with edges $i_a v_1, i_a v_2$, if contracting $i_a v_1$ gives that $i_a v_2$ is a loop, then we must have $v_1 = v_2$, implying that edge a represent a loop in G.

Similarly, if a is a loop in G, then obviously this implies $i_a v_1$, $i_a v_2$ connect the same endpoints. It is easy to see that contracting then deleting is equivalent to just having a loop in a. ☐

Using our lemma, we can now convert between the Γ monomials and G monomials. This can be shown for the two following examples:

- The Γ monomial $cdxxcdxxxx$ would belong to G as every edge of G has two incidences and the monomial can be partitioned as $cdxxcdxxxx = DXDXX$, which is the first term in the original example.
- The Γ monomial $dcxxcdxxxx$ would NOT belong to G the initial string dc is not allowed.

$$\begin{aligned} T_\Gamma(x,y) = {} & x^8 + x^7 + x^6 + x^5 + x^4 + x^3y \\ & + x^7 + \boldsymbol{x^6} + x^5 + \boldsymbol{x^4} + x^3 + \boldsymbol{x^2y} \\ & + x^6 + x^5 + x^4 + x^3 + x^2 + xy \\ & + x^5 + \boldsymbol{x^4} + x^3y + \boldsymbol{x^2} + x + \boldsymbol{y} \\ & + x^4 + x^3y + x^2 + xy \\ & + x^3y + \boldsymbol{x^2}y + xy + \boldsymbol{y^2} \end{aligned}$$

We can create more generalized statement about connection between the number of spanning trees in a graph G and its bipartite Γ by considering equivalent strings of c, d, x, y in Γ. Consecutive operations may be replaced with others such as $cd \leftrightarrow dx$ and $cy \leftrightarrow dx$. From the Main Theorem, the x and c both build spanning trees, while y and d do not. To build the spanning trees, these are precisely the variable exchanges that do not alter the trees. In other words, the string cd chooses one of two incidences but it is not active, while dx chooses one of two incidences and it is active.

This gives the following formula on terminal minor counts, hence, spanning trees between Γ and G.

Lemma 4.1.5. *Let G be a graph with bipartite representation Γ, and tree-number τ, then*

$$\tau(\Gamma) = \tau(G)2^{\phi}$$

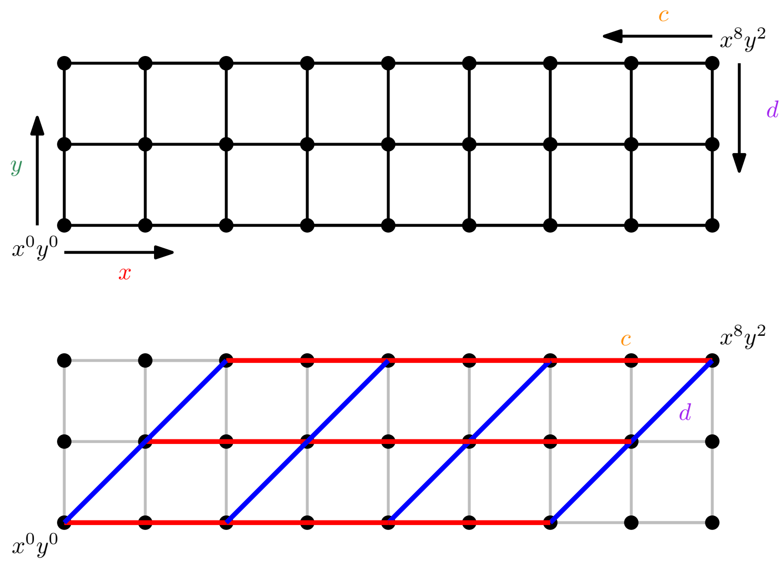
DRAFT:
This lemma can be proven in many ways.

1. Use equations to find X,Y,C,D in terms of x y c d, then plug in
2. Edges outside spanning tree have 2 choices each

New thing: $T_{\Gamma_G}(1,1) = |e| \cdot T_{\Gamma_{G\setminus e}}(1,1) + T_{\Gamma_{G/e}}(1,1)$

4.2 Grid walk of the bipartite and how it relates to G

The previous Lemma and the equivalence class of Γ-monomials via the $cd \leftrightarrow dx$ and $cy \leftrightarrow dx$ replacement can be reinterperted as as grid walks in a larger lattice.



The G grid is contained in the Γ grid via cc and cd steps.

The natural deletion and contraction steps are shown in blue and red, while the replacement steps are not depicted.

Question: Does the grid overlay inform when a $cd \leftrightarrow dx$ and $cy \leftrightarrow dx$ replacement can occur? I.e. you cannot walk outside the sub-grid?

Question: Does this extend to hyperedges? What strings are replaceable? In a 3-edge for example ddc cannot occur, but ddx can as the last incidence in a 3-edge cannot be contracted. This allows boolean lattice of cd strings to be extended to using xy, and the rank 1 elements seem to be the exchangeable steps.

5 Extensions to Hypergraphs

Definition 5.0.1. A **hypergraph** is a graph where edges may connect any positive number of vertices. For hypergraphs, an *n-edge* is an edge connecting n vertices.

Definition 5.0.2. The **incidence dual hypergraph** of a planar graph G is the graph formed by switching the edge and vertex set of G inside of the bipartite representation of the planar graph, Γ.

Definition 5.0.3. We define the hypergraphic Tutte polynomial T_H to be the following, where H is some hypergraph.

$$T_H(x, y, c, d) =$$

5.1 Spanning Tree Relations From the Bipartite to the Hypergraph

Definition 5.1.1. We define a **hypergraphic spanning tree** to be the hypergraph formed by deleting all half-edges from a bipartite spanning tree.

6 Applications

Our research can be applied to analyzing spanning trees, colorings, flows, reliability in graphs, knot theory, statistical physics, and more, which when applied to the real world, can be useful with certain aspects of information networks, data structures, and the transfer of resources from one place to another.

7 Conclusion

In conclusion, we have proven the close relationship between deletion and contraction and internal and external activity and the connections between graphs and their bipartite. We have also observed connections between the grid-walk for the graph and its bipartite.

8 Future Work

The results of our paper suggest interesting relations to chip firing and matroid theory. In particular, many of the implications of Dhar's burning algorithm on bipartite graphs and hypergraphs are still unexplored, and many of the arguments used above seem to be relevant in their application. Various grid-walking arguments within matroids with each step corresponding to an x, y, c, or d in the Tutte Polynomial also can be explored-the most interesting areas of further study include analyzing the number of paths in the grid, and analyzing which paths do or do not appear in certain types of graphs.

9 References

1. *Ellis-Monaghan, Joanna A., and Criel Merino. "Graph Polynomials and Their Applications I: The Tutte Polynomial." Structural Analysis of Complex Networks, 2010, pp. 219–255., doi:10.1007/978-0-8176-4789-69.*

2. *Kálmán, T. (2013). A version of TUTTE'S polynomial FOR HYPERGRAPHS. Advances in Mathematics, 244, 823–873. https://doi.org/10.1016/j.aim.2013.06.001*

3. *Courtiel, J. (2014). A General Notion of Activity for the Tutte Polynomial. ArXiv.* k

x = XX y = CY + DX c = CC d = CD + DX $T_G(x, y, c, d) = T_\Gamma(X, Y, C, D) = T_\Gamma(\sqrt{x}, ..., ..)$

$T_G(1, 1, 1, 1) = T_\Gamma(1, 1/2, 1, 1/2) = (1/2)^{(|y|+|d|)} * T_\Gamma(1, 1, 1, 1)$