Tutte Polynomial of Multi-Bridge Graphs

Julian A. Allagan

Abstract

In this paper, using a well-known recursion for computing the Tutte polynomial of any graph, we found explicit formulae for the Tutte polynomials of any multi-bridge graph and some $2-$tree graphs. Further, several recursive formulae for other graphs such as the fan and the wheel graphs are also discussed.

Keywords: Tutte polynomial, multi-bridge graph, 2-tree, fan, wheel.

1 Basic notions

Throughout this paper, we let $G = (V, E)$ be a simple graph, where $V(G)$ and $E(G)$ denote, respectively, the set of *vertices* and the set of unordered pair of vertices called *edges* of G. An edge $e \in E(G)$ with ends $u, v \in V(G)$ is denoted by $\{u, v\}$ or uv; e is said to be *incident* with u and v. An edge $\{u, u\}$ is called a *loop*. An edge $\{u, v\}$ that occurs more than once in E is called a *multiple* (or *parallel*) edge. A graph G is said to be *isomorphic* to a graph H if G can be obtained by relabelling the vertices of H; and we write $G \cong H$.

A graph $G = (V, E)$ is *connected* if there is a path between each pair of its vertices and is *disconnected* otherwise (thus, has more than one *component*). A *separating set* or *vertex cut* of a connected graph G is a set $S \subseteq V$ such that $G - S$ is disconnected. G is said to be $k-$*connected* if $|S| \geq k$. *Blocks* of G are its maximal $2-$connected subgraphs. An edge in a connected graph is a *bridge* if its removal leaves a disconnected graph. For further basic definitions of graphs, we refer the reader to [14].

Let G_1 and G_2 be two graphs. The *join* of G_1 and G_2, denoted by $G_1 \vee G_2$, is the graph H whose vertex set is $V(H) = V(G_1) \cup V(G_2)$, a disjoint union, and whose edge set is $E(G) = E(G_1) \cup E(G_2) \cup \{v_1v_2 \mid v_1 \in V(G_1),\ v_2 \in V(G_2)\}$. For example, $\overline{K}_{n_1} \vee \overline{K}_{n_2} \vee \ldots \vee \overline{K}_{n_k} = K(n_1, n_2, \ldots, n_k)$ is a complete $k-$partite graph with part sizes $n_1, \ldots, n_k$. An $l-$*cycle*, written $C_l := (v_1, v_2, \ldots, v_l)$, consists of l distinct vertices $v_1, v_2, \ldots, v_l$, and l edges $e_j := \{v_j, v_{j+1}\}$, with $1 \leq j \leq l-1$, and $e_l := \{v_l, v_1\}$. When $e_l = \emptyset$, then we have an $(l-1)-$*path* which we denote by P^{l-1}.

The *Tutte polynomial* of a graph $G = (V, E)$ is a bivariate polynomial $T(G) = T(G; x, y)$ which is given by

$$T(G; x, y) = \sum_{S \subseteq E} (x-1)^{c(S)-c(E)} (y-1)^{c(S)+|S|-|V|},$$

where $c(S)$ is the number of components in the spanning subgraph (V, S).

2 Preliminaries

Tutte polynomial (originally known as dichromatic polynomial) has a particular relation with a number of univariate polynomials. A survey of several related (and unrelated) polynomials can be found in [1, 11, 13]. For example, the *reliability polynomial* of G, denoted by $R(G, p)$, is the probability that G remains connected when each edge in G fails with probability p. *The chromatic polynomial* of G, denoted by $P(G, \lambda)$, counts the number of ways the vertices of G can be colored using at most λ colors. The *flow polynomial* of G, denoted by $F(G, k)$, counts the number of nowhere-zero $k-$flows. From the Tutte polynomial of a (loopless) graph, we can recover the chromatic polynomial along $y = 0$ and the flow polynomial along $x = 0$. Thus, for a graph G on n vertices with m edges and c components, the chromatic polynomial, the flow polynomial and the reliability polynomial of G are respectively obtained from the Tutte polynomial by:

$$
\begin{aligned}
P(G,\lambda) &= (-1)^{n-c}\lambda T(G;1-\lambda,0)\\
F(G,k) &= (-1)^{m-n+c}T(G;0,1-k)\\
R(G,p) &= p^{n-c}(1-p)^{m-n+c}T(G;1,\frac{1}{1-p}).
\end{aligned}
$$

This paper does not focus on these previously mentioned polynomials, and yet, their results can be derived from the Tutte polynomial formulae we present using a simple substitution. For instance, knowing $T(C_3;x,y)=x^2+x+y$, easily follows that $P(C_3,\lambda)=\lambda(\lambda-1)(\lambda-2)$, $F(C_3,k)=k-1$, and $R(C_3,p)=p^2(3-2p)$, where C_3 is a cycle on 3 vertices.

Other important evaluations of $T(G;x,y)$ can be found at some specific points of the plane and also along several algebraic curves. For instance, $T(G;2,2)$ gives the number of spanning subgraphs, $T(G;2,1)$ corresponds to the number of acyclic subgraphs, while $T(G;-2,0)$ gives the number of Eulerian orientations. We refer to [6, 7, 8] for details about the combinatorial interpretations of these evaluations and several others.

Two basic graph operations (see Figure 1) define the Tutte polynomial of any graph G.

(1) $G-e$, which means the deletion or the removal of the edge e and

(2) G/e, which means the contraction of the edge e; e is removed and its incident vertices are merged (multiple edges and loops may occur).

These deletion/contraction operations occur naturally in modeling networks which arise from a wide range of problems in optimization, coding theory, statistical physics, biology, engineering, and computer science. For a thorough survey of research and applications of the Tutte polynomial, we refer the reader to [1, 3, 4, 12].

In general, it is difficult to compute the Tutte polynomial of a given graph; such a computation is NP-hard as the recursion grows exponentially in complexity [11]. For this reason, several heuristic algorithms

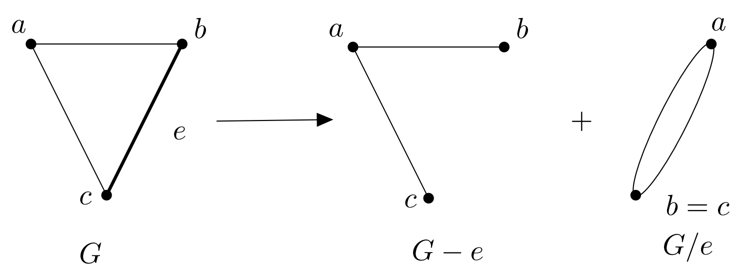


Figure 1. Example of the deletion/contraction of an edge

have been proposed for any given graph with a limited number of vertices and the most efficient algorithm has recently been presented by Haggard et al.[9]. Further, very few recursive, let alone explicit formulae for some graphs are known and the research of finding explicit formulae for different classes of graphs is still active [2, 10, 13]. In this paper, we present some results concerning both recursive and explicit formulae of some graphs.

3 Some recursive formulae

Definition 3.1. *The Tutte polynomial of a graph $G = (V, E)$ is defined by:*

$$T(G; x, y) = \begin{cases} 1 & E(G) = \emptyset \\ yT(G-e; x, y) & e \in E(G) \textit{ and } e \textit{ is a loop} \\ xT(G/e; x, y) & e \in E(G) \textit{ and } e \textit{ is a bridge} \\ T(G-e; x, y) + T(G/e, x, y) & \textit{otherwise.} \end{cases}$$

This definition provides a recursive algorithm also known as deletion/contraction method for computing $T(G; x; y)$, giving the next two corollaries.

Corollary 3.0.1. *Suppose H^l denotes a tree on l edges. Then $T(H^l; x, y) = x^l$.*

Corollary 3.0.2. *The Tutte polynomial of an $n-$cycle is*
$T(C_n; x, y) = \sum_{i=1}^{n-1} x^i + y.$

Remark: When applying the deletion/contraction method, we are free to choose any edge. Therefore, given a graph G, by starting with the edges of G that are either bridges or loops, one can reduce the the computation of its Tutte polynomial to $T(G^\dagger)$, where $G^\dagger$ is obtained from G by removing its initial loops and bridges (multiple components may occur). Thus, the computation of any graph with loops and bridges is simplified by

Proposition 3.1. *Given any graph $G = (V, E)$ with at least l loops and k bridges, then each of the following statements holds:*

(a) $T(G; x, y) = x^k y^l T(G^\dagger; x, y)$.

(b) If G has c connected components $G_1, \ldots, G_c$, then $T(G; x, y) = T(G_1; x, y) \ldots T(G_c; x, y)$.

For this reason, the main results in this section and the rest of this paper concern bridgeless and loopless graphs.

Let $P^l := (v_1, e_1, v_2, \ldots, e_l, v_{l+1})$ denote an alternating sequence of distinct vertices v_i and distinct edges e_i. We define an $l-$*fan* by $F^l = P^l \vee \{w\}$, with $w \neq v_i$ for $1 \leq i \leq l+1$. Figure 2(a) is an example. We note that F^0 is an edge of multiplicity 2 (or a $2-$edge) and $F^1 \cong C_3$ which Tutte polynomials are $x + y$ and $x^2 + x + y$ respectively. Thus, it is customary to define a fan graph on $l \geq 2$.

Theorem 3.1. *Suppose F^l is an $l-$fan. Then, $T(F^l; x, y) =$ $xT(F^{l-1}) + \sum_{i=0}^{l-1} y^i T(F^{l-i-1})$ with $T(F^0) = x + y$ and $l \geq 2$.*

Proof. When $l = 2$, let's suppose $F^2 := (v_1, e_1, v_2, e_2, v_3) \vee \{w\}$. We apply the deletion/contraction method on e_2, giving that

$$\begin{aligned} T(F^2) &= T(F^2 - e_2) + T(F^2/e_2) && (1) \\ &= xT(F^1) + T(F^1_*), && (2) \end{aligned}$$

where $F^1_* := (v_1, e_1, v_2) \vee \{w\} \cup \{w, v_2\}$. Further, we apply again the deletion/contraction method on $\{w, v_2\}$ to obtain that

$$T(F^1_*) \quad = \quad T(F^1) + yT(F^0). \tag{3}$$

Thus, from (2) and (3) together, we have

$$T(F^2) \quad = \quad xT(F^1) + T(F^1) + yT(F^0). \tag{4}$$

Hence,

$$\begin{aligned} T(F^2) &= x(x^2 + x + y) + x^2 + x + y + y(x + y) \\ &= x^3 + 2x^2 + 2xy + x + y^2 + y. \end{aligned} \tag{5}$$

Moreover, for all $l \geq 2$, we have

$$\begin{aligned} T(F^l; x, y) &= T(F^l - e_l) + T(F^l/e_l) \\ &= xT(F^{l-1}) + T(F^{l-1}_*), \end{aligned} \tag{6}$$

where $F^{l-1}_* = F^{l-1} \cup \{w, v_l\}$.

Claim 3.1.1. *$T(F^r_*; x, y) = \sum_{i=0}^{r} y^i T(F^{r-i})$ for each $r \geq 1$.*

Proof. By induction on r. For $r = 1$, see (3).

Suppose $F^r_* = F^r \cup \{w, v_{r+1}\}$. Observe that $\{w, v_{r+1}\}$ becomes a 2−edge. So, as one edge is deleted (in deletion), the other becomes a loop (in contraction). Thus, we apply the deletion/contraction method on $\{w, v_{r+1}\}$ to obtain $T(F^r_*; x, y) = T(F^r) + yT(F^{r-1}_*)$. By the inductive hypothesis,

$$\begin{aligned} T(F^r_*; x, y) &= T(F^r) + y\Big(\sum_{i=0}^{r-1} y^i T(F^{r-i-1})\Big) \\ &= T(F^r) + \sum_{i=1}^{r} y^i T(F^{r-i}) \\ &= \sum_{i=0}^{r} y^i T(F^{r-i}). \end{aligned} \tag{7}$$

□

The result follows from (6) and Claim 3.1.1. □

Given $P^l := v_1, e_1, v_2, \ldots, e_l, v_{l+1}$, when $v_1 = v_{l+1}$, then $P^l \cong C_l$ and we define a *wheel* graph by $W^l = C_l \vee \{w\}$ for all $l \geq 2$. C^l is often referred to as the *rim* of the wheel and the edges not in the rim are called *spokes*. We will call a wheel on l rim edges, an $l-$wheel, for short.

Theorem 3.2. *Let W^l be an $l-$wheel. Then, for all $l \geq 4$,*
$T(W^l; x, y) = T(W^{l-1}) + T(F^{l-1}) + \sum_{i=0}^{l-4} y^{i+1}\Big(\sum_{j=0}^{l-i-3} y^j T(F^{l-i-j-3})\Big) +$
$y^{l-2}(x + y + y^2)$, *with* $T(F^0) = x + y$.

Proof. We begin with some initial cases.

Let $l = 2$ and suppose $W^2 := (v_1, e_1, v_2, e_2, v_1) \vee \{w\}$. We apply the deletion/contraction method on e_2 to get that $T(W^2) = T(F^1) + yT(F^0)$, giving that

$$T(W^2) = x^2 + y^2 + xy + x + y. \tag{8}$$

Similarly, when $l = 3$, $T(W^3) = T(F^2) + T(W_*^2)$, where $T(W_*^2) = T(W^2)+yT(F^0)+y^3$, giving that $T(W^3) = T(W^2)+T(F^2)+yT(F^0)+y^3$. Hence, from (5), (8), and the initial condition,

$$T(W^3) = x^3 + y^3 + 3x^2 + 3y^2 + 4xy + 2x + 2y \tag{9}$$

after expansion.

For $l = 4$, it follows from a similar recursion as in the previous case that

$$T(W^4) = T(W^3) + T(F^3) + y\Big(T(F^1) + yT(F^0)\Big) + y^2 T(F_*^0), \tag{10}$$

where $T(F_*^0) = x + y + y^2$. Thus, we satisfy the basis for the recursion.

Claim 3.2.1. *If $G^r = F^r \cup \{w, v_1\} \cup \{w, v_{r+1}\}$, then* $T(G^r; x, y) = \sum_{i=0}^{r} y^i \Big(\sum_{j=0}^{r-i} y^j T(F^{r-j-i})\Big)$ *for all $r \geq 1$.*

Proof. Suppose $G^r_* = F^r \cup \{w, v_1\}$. Then $G^r = G^r_* \cup \{w, v_{r+1}\}$. Following the argument in the proof of Claim 3.1.1, it is clear that $T(G^r; x, y) = \sum_{i=0}^{r} y^i T(G^{r-i}_*)$. Further, since $G^r_* \cong F^r_*$, the result follows. $\square$

Suppose $W^l := (v_1, e_1, v_2, \ldots, v_l, e_l, v_1) \vee \{w\}$ for all $l \geq 4$. We apply the deletion/contraction argument on e_l to get that $T(W^l; x, y) = T(F^{l-1}) + T(W^{l-1}_*)$, where $W^{l-1}_* \cong W^{l-1} \cup \{w, v_1\}$ is a graph that is obtained by identifying the endpoints of e_l by v_1 as a result of the contraction. Further, we consider W^{l-1}_* and apply again the deletion/contraction argument on $\{w, v_1\}$ to obtain that $T(W^{l-1}_*; x, y) = T(W^{l-1}) + yT(F^{l-3}_{**})$, where $F^r_{**} \cong F^r \cup \{w, v_1\} \cup \{w, v_{r+1}\}$. Together, we have that $T(W^l; x, y) = T(W^{l-1}) + T(F^{l-1}) + yT(F^{l-3}_{**})$. From Claim 3.2.1, we obtain that $T(W^l; x, y) = T(W^{l-1}) + T(F^{l-1}) + y\sum_{i=0}^{l-3} y^i \Big(\sum_{j=0}^{l-3-i} y^j T(F^{l-i-j-3}) \Big)$. Further, having $T(F^0_*) = x + y + y^2$ as an initial condition, the result follows for all $l \geq 4$. $\square$

The next result follows from Theorem 3.2 after using $T(W^3)$ as a basis for the recursion.

Corollary 3.2.1. *If W^l is an $l-$wheel then, for all $l \geq 4$,*

$$T(W^l; x, y) = T(W^3) + \sum_{k=3}^{l-1} \left(T(F^k) + \sum_{i=0}^{k-3} y^{i+1} \Big(\sum_{j=0}^{k-i-2} y^j T(F^{k-i-j-2}) \Big) + y^{k-1}(x + y + y^2) \right),$$

with $T(F^0) = x + y$.

We note here that the general recursion formula given in Theorem 3.2 would be quite different, if not impossible to obtain, if the deletion/contraction is applied on the spokes instead.

4 Explicit formula for a 2-tree graph

Suppose $G = K(n,2) \cong \{u_1, \ldots, u_n\} \vee \{v_1, v_2\}$ denotes a complete bipartite graph. For each $e \in E(G)$, we define $e_{i,j} := \{u_i, v_j\}$ with $i = 1, \ldots, n$; $j = 1, 2$ and $n \geq 1$.

Theorem 4.1. *The Tutte polynomial of the complete bipartite graph $G = K(n,2)$ is*

$$T(G; x, y) = x^2(x+1)^{n-1} + \sum_{i=1}^{n-1} (x+y)^i (x+1)^{n-i-1}, \textit{ for all } n \geq 1.$$

Proof. When $n = 1$, it is clear that $G \cong H^2$, and the result follows (after setting the last term equal to zero)

$$T(K(1,2)) \;=\; x^2, \tag{11}$$

satisfying the basis of the recursion.

For all $n \geq 1$, we apply the algorithm on $e_{n,2}$ (and subsequently on $e_{n,1}$ after $e_{n,2}$ is contracted) to obtain

$$\begin{aligned} T(G) &= T(G - e_{n,2}) + T(G/e_{n,2}) \\ &= T(G - e_{n,2}) + \Big(T\big((G/e_{n,2}) - e_{n,1}\big) + T\big((G/e_{n,2})/e_{n,1}\big) \Big) \\ &= (x+1) T(K(n-1,2)) + T(K^*(n-1,1)). \end{aligned} \tag{12}$$

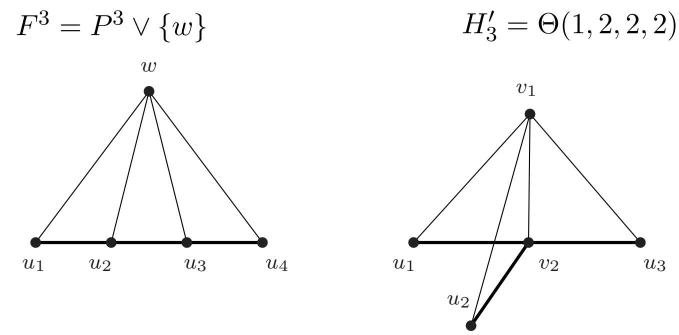
Now, using (11) as the basis for (12) gives

$$\begin{aligned} T(K(n,2)) &= (x+1)^{n-1} T(K(1,2)) + \\ &+ \sum_{i=1}^{n-1} (x+1)^{n-i-1} T(K^*(i,1)), \end{aligned} \tag{13}$$

where $K^*(s,1)$ is $K(s,1)$ with edges, each of multiplicity 2. Therefore, it is clear from Corollary 3.0.1 that

$$T(K^*(s,1)) \;=\; (x+y)^s. \tag{14}$$

The result follows from (11), (13), and (14) for all $n \geq 1$. ∎



(a) Fan graph (b) $4-$bridge graph

Figure 2. Two non-isomorphic $2-$tree graphs

As a generalization of a tree, a $k-$*tree* is a graph which arises from a $k-$clique by 0 or more iterations of adding a new vertex joined to a $k-$clique in the old graph; we shall refer to the initial $k-$clique in any construction as a *base* of the $k-$tree. Thus, we construct any $2-$tree on $n + 2$ vertices from a base edge uv by repeatedly adding n new vertices and making them adjacent to any two ends of an edge in the graph formed so far. This process generates several non-isomorphic $2-$tree graphs. $k-$tree graphs are proven to be useful in constructing reliable network in [5] and Figures 2(a) and 2(b) depict an example of two non-isomorphic $2-$tree graphs. We list here some basic properties of a $2-$tree graph as we recall that a vertex v in a graph G is *simplicial* if its neighborhood in G is a clique.

Proposition 4.1. *Suppose H'_n is a $2-$tree on $n+2$ vertices. Then the following statements are equivalent.*

(a) H'_n is $2-$connected.

(b) H'_n contains exactly n simplicial vertices.

(c) Every edge of H'_n can be used as a base.

(d) H_n' *does not contain any* $4-$*clique.*

(e) $|E(H_n')| = 2(n+2) - 3.$

Proposition 4.2. *Let* $\mathcal{F}$ *denote a class of all non-isomorphic* $2-$*tree graphs. Then* $F^n \in \mathcal{F}$*, where* F^n *is an* $n-$*fan.*

Proof. We denote by G', a graph obtain by the following: starting from an edge (base) $\{u_1, w\}$, we connect each additional vertex u_i to the endpoints of $\{u_{i-1}, w\}$, for $i = 2, \ldots, n+1$. Thus, $G' \cong F^n$, is an $n-$fan (on $n+2$ vertices). Further, from the construction, it is clear that G' is also a $2-$tree. □

For our result, we construct a $2-$tree graph as follows: We connect the endpoints of a base edge $\{v_1, v_2\}$ to the added vertex u_i, for $i = 1, \ldots, n$. We denote the resulting $2-$tree graph on $n+2$ vertices by H_n' and Figure 2(b) is an example.

Observe from the two previously given constructions that the chromatic polynomial $P(H_n', \lambda) = \lambda(\lambda-1)(\lambda-2)^n = P(F^n, \lambda)$, where n is the number of simplicial vertices of any $2-$tree graph.

Corollary 4.1.1. *The Tutte polynomial of the* $2-$*tree on* n *simplicial vertices,* H_n'*, is*

$$T(H_n'; x, y) = x^2(x+1)^{n-1} + \sum_{i=1}^{n-1}(x+y)^i(x+1)^{n-i-1} + (x+y)^n, n \geq 1.$$

Proof. Observe that $H_n' \cong K(n,2) \cup e$, where $e := \{v_1, v_2\}$ and $K(n,2)$ is the complete bipartite as previously defined. Then, after applying the algorithm on e, follows that for all $n \geq 1$,

$$T(H_n'; x, y) = T(K(n,2)) + T(K^*(n,1)). \tag{15}$$

Hence, the result follows from (15) using (14) and Theorem 4.1. □

5 Explicit formula for multi-bridge graphs

A network is reliable or fault-tolerant if it contains more alternative or disjoint paths between its operatives sites. For this reason, we find it important to discuss this highly connected class of graphs as we recall that two paths a_1 and a_2 are *internally disjoint* (or independent) if they have no common internal vertex.

A *multi-bridge* (or $m-$*bridge*) graph $G = \theta(a_1, \ldots, a_m)$ is the graph obtained by connecting two distinct vertices with $m \geq 2$ internally disjoint paths of lengths $a_1, \ldots, a_m$ respectively, with $a_i \geq 1$. (See Figure 2(b)). For instance, when $m = 2$, $\theta(a_1, a_2) \cong C_{a_1+a_2}$. For our result, we assume $m \geq 2$ and $a_i \geq 1$, though it is customary to define $\theta(a_1, \ldots, a_m)$ for $m \geq 3$ and $a_i \geq 2$. As such, a multi-bridge graph is a generalization of the well-known $\theta-$graph [14].

For the next result, we define

$$\tau_m = \begin{cases} y + \sum_{r=1}^{m-1} x^r & if\ m \geq 2 \\ y & otherwise. \end{cases}$$

As a consequence of this definition, a loop is therefore isomorphic to C_1, a cycle on a single vertex, and the result of Corollary 3.0.2 becomes

$$T(C_m; x, y) = \tau_m, \quad m \geq 1. \tag{16}$$

Theorem 5.1. *If $G = \theta(a_1, \ldots, a_m)$ is a multi-bridge graph, then for each $a_k \geq 1$,*

$$T(G; x, y) = \sum_{i=0}^{m-3} \prod_{j=1}^{i} \Big(\sum_{r=0}^{a_{m-j+1}-1} x^r \Big) \prod_{k=1}^{m-i-1} \tau_{a_k} + \prod_{j=1}^{m-2} \Big(\sum_{r=0}^{a_{m-j+1}-1} x^r \Big) \tau_{a_1+a_2},$$

$m \geq 2$.

Proof. It is easy to verify the special case when $a_k = 1$ for each $1 \leq k \leq m$. Because $\prod_{j=1}^{i} \Big(\sum_{r=0}^{a_{m-j+1}-1} x^r \Big) = 1$ for $i = 0, \ldots, m-3$, we have

$$
\begin{aligned}
T(\theta\underbrace{(1,\ldots,1,1)}_{m}) &= \sum_{i=0}^{m-3}(\tau_1)^{m-i-1}+\tau_2 \\
&= \sum_{i=0}^{m-3} y^{m-i-1}+y+x \\
&= \sum_{i=1}^{m-1} y^{i}+x,
\end{aligned}
$$

which is the Tutte polynomial of an edge of multiplicity $m \geq 2$. When $m = 2$, we set the first term in the original formula equal to zero (and the coefficient of the last term equal to one) to obtain that

$$
T(G;x,y) \;=\; \tau_{a_1+a_2} = T(C_{a_1+a_2}), \tag{17}
$$

satisfying the basis of the recursion.

For $m \geq 2$, we apply the deletion/contraction (sequentially) on a_m to get that

$$
T(G;x,y) = \Big(\sum_{r=0}^{a_m-1} x^r\Big) T(\theta(a_1,\ldots,a_{m-1})) + \prod_{j=1}^{m-1} T(C_{a_j}). \tag{18}
$$

Now, we use $T(C_{a_1+a_2})$ (and subsequently expression (16)) as the basis for (18) to get

$$
\begin{aligned}
T(G;x,y) &= \prod_{j=3}^{m}\Big(\sum_{r=0}^{a_j-1} x^r\Big) T(C_{a_1+a_2}) + \sum_{i=1}^{m-2}\prod_{j=3+i}^{m}\Big(\sum_{r=0}^{a_j-1} x^r\Big)\prod_{k=1}^{i+1} T(C_{a_k}) \\
&= \prod_{j=3}^{m}\Big(\sum_{r=0}^{a_j-1} x^r\Big)\tau_{a_1+a_2} + \sum_{i=1}^{m-2}\prod_{j=3+i}^{m}\Big(\sum_{r=0}^{a_j-1} x^r\Big)\prod_{k=1}^{i+1} \tau_{a_k},
\end{aligned}
$$

giving the result for all $a_k \geq 1$ and $m \geq 2$.

$\square$

Corollary 5.1.1. *If $G = \theta(a_1, a_2, a_3)$ is a $\theta-$graph, then for each $a_k \geq 1$, $T(G; x, y) = \Big(\sum_{r=0}^{a_3-1} x^r\Big)\tau_{a_1+a_2} + \tau_{a_1}\tau_{a_2}$.*

Proof. The result follows from Theorem 5.1 when $m = 3$. $\square$

Observe that, because $\theta(\underbrace{1, 2, \ldots, 2}_{m+1}) \cong H'_m$, a $2-$tree on $m+2$ vertices, Theorem 5.1 extends the result of Corollary 4.1.1.

References

[1] B. Bollobas. *Modern Graph Theory*, Graduate Texts in Mathematics, Vol. 184 (1998) Springer, New York NY.

[2] J.E. Bonin, H. Qin. *Tutte polynomials of q-cones*, Discrete Math. **232** (2001), pp. 95–103.

[3] T. Brylawski, J. Oxley. *The Tutte poynomial and its applications*, Encyclopedia Math. Appl., Cambridge Univ. Press (1992), pp. 123–225.

[4] J. Ellis-Monaghan, C. Merino. *Graph polynomials and their applications i: the Tutte Polynomial*, Structural Analysis of Complex Networks (2009).

[5] A.M. Farley. *Networks Immune to Isolated Failures*, Networks **11** (1981), pp. 225–268.

[6] C. Greene, T. Zaslavsky. *On the interpretation of Whitney numbers through arrangements of hyperplanes, zonotopes, non-Radon partitions, and orientations of graphs,* Trans. Amer. Math. Soc. **280** (1983), pp. 97–126.

[7] F. Jaeger, D. Vertigan, D.J.A. Welsh. *On the computational complexity of the Jones and Tutte polynomials*, Mathematical Proceedings of the Cambridge Philosophical society **108** (1990), pp. 35–53.

[8] R. Stanley. *Acyclic orientations of graphs*, Discrete Math. **5** (1973), pp. 171–178.

[9] D. Haggard, D.J. Pearce, G. Royle. *Computing Tutte polynomials*, Trans. Math. Soft. (2011) Vol.**37(3)**, pp. 1–17.

[10] E. Mphako-Banda. *Tutte polynomials of flower graphs*, Bulletin of the Iranian Math. Soc. **35(2)** (2009), pp. 179–190.

[11] J. Oxley, D.J.A. Welsh. *Chromatic, Flow and Reliability Polynomials: The Complexity of their Coefficients*, J. Combinatorics, Probability and Computing **11** (2002), pp. 403–426.

[12] T.W. Tutte. *On dichromatic polynomials*, J. Combin. Theory **2** (1967), pp. 301–320.

[13] D.J.A. Welsh. *Counting, colourings and flows in random graphs*, Combinatorics, Paul Erdös is Eighty **2** (1993), pp. 1–15.

[14] D. West. *Introduction to Graph Theory*, Prentice Hall, 2001.

Julian A. Allagan Received April 19, 2013

Julian A. Allagan,
University of North Georgia, Watkinsville, GA, United States.
Department of Mathematics
E–mail: *julian.allagan@ung.edu*