Chromatic and flow polynomials of generalized vertex join graphs and outerplanar graphs

Boris Brimkov and Illya V. Hicks

Computational & Applied Mathematics, Rice University, Houston, TX 77005 USA
E-mail: boris.brimkov@rice.edu, ivhicks@rice.edu

Abstract

A generalized vertex join of a graph is obtained by joining an arbitrary multiset of its vertices to a new vertex. We present a low-order polynomial time algorithm for computing the chromatic polynomials of generalized vertex joins of trees; by duality, this algorithm can also be used to compute the flow polynomials of arbitrary outerplanar graphs. We also present closed formulas for the chromatic polynomials of generalized vertex joins of cliques, and the chromatic and flow polynomials of generalized vertex joins of cycles.

Key words: Chromatic polynomial, Flow polynomial, Vertex join, Outerplanar graph, Tree, Wheel graph

1 Introduction

Graph polynomials contain various information about the structure and properties of graphs; their study is an active area of research with many theoretical consequences and practical applications. Two of the most important single-variable graph polynomials are the chromatic and flow polynomials. Their coefficients, roots, values at specific points, and derivatives have meaningful interpretations related to the chromatic and flow numbers, Hamiltonicity [22], number of acyclic and totally cyclic orientations [19], cycle space [27], and edge-connectivity [12] of the corresponding graphs.

Chromatic and flow polynomials also have connections to other sciences such as statistical physics, combinatorics, and theoretical computer science. The chromatic polynomial is the zero-temperature limit of the anti-ferromagnetic Potts model and is used to model the behavior of crystals and ferromagnets [18]; it is also related to the Stirling and Beraha numbers, which arise in a variety of analytic and combinatorics problems (cf. [13, 1]). The flow polynomial is used in crystallography and statistical mechanics to model the physical properties of ice and other crystals [11]. For more applications of chromatic

Preprint submitted to Elsevier Science 6 October 2015

© 2015. This manuscript version is made available under the Elsevier user license http://www.elsevier.com/open-access/userlicense/1.0/
and flow polynomials, see the comprehensive survey of Ellis-Monagahan and Merino [4] and the bibliography therein.

Unfortunately, computing the chromatic and flow polynomials of a graph are very challenging tasks. These problems are NP-hard for general graphs, and even for bipartite planar graphs and sparse graphs as shown in [16]. In fact, most of the terms of the chromatic and flow polynomials of general graphs cannot even be approximated (see [5, 16]). Thus, a large volume of work in this area is focused on exploiting the structure of specific types of graphs in order to derive closed formulas, algorithms, or heuristics for computing their chromatic and flow polynomials. Such investigations frequently focus on classes of graphs which are generalizations of trees, cliques, and cycles.

In particular, Wakelin et al. [3, 24] considered a class of graphs called polygon trees and computed their chromatic polynomials; they also characterized the chromatic polynomials of biconnected outerplanar graphs and the flow polynomials of their dual graphs. Whitehead [25, 26] characterized the chromatic polynomials of a class of clique-like graphs called $q$-trees. Furthermore, Lazuka [10] obtained explicit formulas for the chromatic polynomials of cactus graphs, Gordon [6] studied Tutte polynomials (a generalization of chromatic and flow polynomials) of rooted trees, and Mphako-Banda [14, 15] derived formulas for the chromatic, flow, and Tutte polynomials of flower graphs.

In this paper, we consider yet another generalization of trees, cliques, and cycles. We define a generalized vertex join of a graph $G$ to be the graph obtained by joining an arbitrary multiset of the vertices of $G$ to a new vertex. We compute the chromatic polynomials of generalized vertex joins of trees, cliques, and cycles, and use the duality of chromatic and flow polynomials to find the flow polynomials of certain other classes of graphs, including outerplanar graphs. Thus, we complement the work of Wakelin et al. [3, 24] on chromatic polynomials of outerplanar graphs and flow polynomials of their duals, by characterizing the flow polynomials of outerplanar graphs and the chromatic polynomials of their duals. Several related results are included as well.

The paper is organized as follows. In the next section, we recall some notions and notations related to graph theory and graph polynomials. In Section 3, we list well-known technical tools used in the computation of chromatic and flow polynomials. In Section 4, we compute the chromatic polynomials of generalized vertex joins of trees; we relate these results to outerplanar graphs in Section 5. In Section 6, we consider generalized vertex joins of cliques and cycles, and related dual results. We conclude with some final remarks and open questions in Section 7.
2 Preliminaries

We assume the reader is familiar with basic graph theoretic notions and operations; refer to [2] for an extensive background on graph theory. In this section, we first recall the definition of a multiset and related terms, followed by select graph theoretic notions used in the paper.

Let \( G = (V, E) \) be a graph. A multiset \( S \) over \( V \) is a collection of vertices of \( V \), each of which may appear more than once in \( S \). The number of times a vertex \( v \) appears in \( S \) is the multiplicity of \( v \). The underlying set of \( S \) is the set \( S' \) which contains the (unique) elements of \( S \). For example, \( S = \{v_1, v_1, v_3, v_4, v_4, v_4\} \) is a multiset over \( V = \{v_1, v_2, v_3, v_4\} \) and the underlying set of \( S \) is \( S' = \{v_1, v_3, v_4\} \). Using this notion, we define the generalized vertex join of \( G \) using \( S \) to be the graph \( G_S = (V \cup \{v^*\}, E \cup \{vv^* : v \in S\}) \). Note that if the multiplicity of \( v \) in \( S \) is \( k \), there are \( k \) parallel edges between \( v \) and \( v^* \) in \( G_S \). See Figure 1 for an example.

![Fig. 1. Left: A graph \( G \). Right: \( G_S \), the generalized vertex join of \( G \) using \( S = \{v_1, v_1, v_3, v_4, v_4\} \).](image)

Given \( G = (V, E) \) and \( S \subset V \), the induced subgraph \( G[S] \) is the subgraph of \( G \) whose vertex set is \( S \) and whose edge set consists of all edges of \( G \) which have both ends in \( S \). Given \( u, v \in V \), the contraction \( G/uv \) is obtained by deleting edge \( uv \) if it exists, and identifying \( u \) and \( v \) into a single vertex. Note that \( G \) does not need to have the edge \( uv \) for \( G/uv \) to be defined. Finally, we say that \( G \) is biconnected if \( G - v \) has exactly one connected component for all \( v \in V \).

Many of the graphs considered in this paper are planar graphs — i.e., they can be drawn in the plane so that their edges do not cross each other. A graph drawn in such a way is called a plane graph. If \( G \) is a plane graph, its dual \( G^* \) is a graph that has a vertex corresponding to each face of \( G \), and an edge joining the vertices corresponding to neighboring faces for each edge of \( G \). Note that if \( G \) is connected, \( G = (G^*)^* \). The weak dual of \( G \) is the subgraph of \( G^* \) whose vertices correspond to the bounded faces of \( G \).

We close this section by introducing the two graph polynomials we will investigate in the sequel. A vertex coloring of \( G \) is an assignment of colors to the vertices of \( G \) so that no edge is incident to vertices of the same color. A \( t \)-coloring of \( G \) is a vertex coloring using at most \( t \) colors. The chromatic polynomial \( P(G; t) \) counts the number of \( t \)-colorings of \( G \); if the dependence
on $t$ is implied in the context, this can be abbreviated as $P(G)$.

A closely related polynomial is the flow polynomial. A nowhere-zero $\mathbb{Z}_t$-flow on $G$ is an assignment of values from \( \{1, 2, \ldots, t-1\} \) to the edges of an arbitrary orientation of $G$ so that the total flow entering each vertex is congruent modulo $t$ to the total flow leaving each vertex. The flow polynomial $F(G; t)$ counts the number of nowhere-zero $\mathbb{Z}_t$-flows on $G$; if the dependence on $t$ is implied, this can be abbreviated as $F(G)$.

# 3 Tools for computing chromatic and flow polynomials

Before we present our main results, we list a number of well-known facts frequently used in the computation of chromatic and flow polynomials. Proofs of these and other related results are given by Tutte [23]. In what follows, let $G = (V, E)$ be a graph, $K_n$ be the complete graph on $n$ vertices and $C_n$ be the cycle on $n$ vertices.

$$P(G) = P(G - e) - P(G/e) \quad \text{for any } e = uv, \text{ where } u, v \in V \quad (1)$$

$$P(G) = P(G + e) + P(G/e) \quad \text{for any } e = uv, \text{ where } u, v \in V$$

If $G = G_1 \cup G_2$ and $G_1 \cap G_2 = K_r$, then $P(G) = \frac{P(G_1)P(G_2)}{P(K_r)} \quad (2)$

If $e \in E(G)$ is a multiple edge, then $P(G) = P(G - e) \quad (3)$

Many of these identities can also be used in the computation of flow polynomials, as the flow polynomial can be obtained from the chromatic polynomial in the following way:

If $G$ is planar, then $F(G) = \frac{1}{t} P(G^*) \quad (4)$

Furthermore, the following identities allow us to express the chromatic and flow polynomials of $G$ in terms of the polynomials of its connected and biconnected components.

If $G = G_1 \cup G_2$ and $|V(G_1 \cap G_2)| = 0$, then $P(G) = P(G_1)P(G_2) \quad (5)$

If $G = G_1 \cup G_2$ and $|V(G_1 \cap G_2)| \leq 1$, then $F(G) = F(G_1)F(G_2)$

Finally, closed formulas for the chromatic polynomials of some specific graphs are known. In particular,
\[ P(K_n) = t(t - 1) \ldots (t - (n - 1)) \]  
\[ P(C_n) = (t - 1)^n + (-1)^n(t - 1) \]  
\[ P(T) = t(t - 1)^{n-1} \] for any tree \( T \) on \( n \) vertices. (8)

**Remark 1** Let \( G = (V, E) \) be a graph, \( S \) be a multiset over \( V \), and \( S' \) be the underlying set of \( S \). By (3), \( P(G_S) = P(G_{S'}) \). Thus, when computing the chromatic polynomial of \( G_S \), we can assume without loss of generality that the multiplicity of every element in \( S \) is 1. The reason we consider multisets instead of sets of vertices is because allowing certain multiple edges in a class of graphs corresponds to a larger class of dual graphs. In turn, this can lead to broader results about flow polynomials and allow us to easily establish the flow-equivalence of certain graphs.

For instance, in the next section, we compute the chromatic polynomials of generalized vertex joins of trees. We show in Section 5 that the duals of these graphs are outerplanar graphs, where the added vertex \( v^* \) is the one corresponding to the outer face. Allowing multiple edges between \( v^* \) and each vertex of the tree means the family of duals includes all biconnected outerplanar graphs, instead of ones for which at most one edge from each bounded face borders the outer face. Thus, we are able to state a broader result about flow polynomials. A similar principle is used in Section 6 with the flow polynomials of generalized vertex joins of cycles.

### 4 Chromatic polynomials of generalized vertex join trees

Let \( T = (V, E) \) be a tree with \(|V| = n\), \( S \) be a multiset over \( V \), and let \( T_S \) be the generalized vertex join of \( T \) using \( S \). For short, we will call \( T_S \) a generalized vertex join tree. In this section, we present an efficient algorithm to find \( P(T_S) \).

First, by Remark 1, we can assume that the multiplicity of every element in \( S \) is 1. Two special cases of \( T_S \) occur when \(|S| = 0\) and when \(|S| = 1\). In the first case, \( T_S \) consists of a tree on \( n \) vertices and an isolated vertex. Thus, by (8) and (5), \( P(T_S) = t^2(t - 1)^{n-1} \). In the second case, \( T_S \) is a tree on \( n + 1 \) vertices, so by (8), \( P(T_S) = t(t - 1)^n \). Thus, from now on, we will assume that \(|S| \geq 2\).

Next, suppose there are \( b \) bridges in \( T_S \), and let \( B \) be the set of vertices in \( T_S \) which are an endpoint of some bridge, but do not belong to a cycle. Note that since \(|S| \geq 2\), there is at least one cycle, so not all edges of \( T_S \) are bridges. Let \( T'_S = T_S - B \) (see Figure 2, middle). Using (2) and (8), it is easy to see that

\[ P(T_S) = P(T'_S) \frac{(t(t - 1))^b}{t^b} = P(T'_S)(t - 1)^b. \] (9)
We now introduce several definitions which are analogous to standard notions in graph theory and are slightly modified to suit our purposes. For simplicity, we will refer to these terms by the names of their standard analogues.

First, select an arbitrary vertex \( r \neq v^* \) in \( T'_S \) called a root. The level of a node in \( T'_S \) is given by the function \( L : V(T'_S) \setminus \{v^*\} \rightarrow \mathbb{N} \cup \{0\} \) with \( L(v) = d(r, v) \), where \( d(r, v) \) is the length of the shortest path between \( r \) and \( v \) in \( T'_S - v^* \). Denote by \( L_i(T'_S) \) the set of nodes at the \( i \)-th level; more precisely, \( L_i(T'_S) = \{v : L(v) = i\} \). Let \( \mathcal{L} \) be the height of \( T'_S \), i.e., \( \mathcal{L} = \max\{L(v) : v \in V(T'_S) \setminus \{v^*\}\} \).

If \( L(v) = i, w \) is a child of \( v \) if \( w \) is adjacent to \( v \) and \( L(w) = i + 1 \). Vertex \( z \) is a descendant of \( v \) if \( z = v^* \) or if there is a path \( v, p_1, \ldots, p_r, z \) such that
\[
L(v) < L(p_1) < \ldots < L(p_r) < L(z).
\]
The set of all descendants of \( v \) is denoted \( D(v) \). See Figure 2 for an illustration.

![Figure 2](image)

**Fig. 2.** Left: Forming \( T'_S \) from a given tree \( T \) and a subset of its nodes \( S \). Middle: Removing the bridges of \( T'_S \) to form \( T'_S' \), and selecting node \( r \). Right: Finding \( L_i(T'_S) \).

Finally, for any \( a \in V(T'_S) \setminus \{v^*\} \) and child \( c \) of \( a \), we define \( T_a = T'_S[a \cup D(a)] \), \( H_a = T_a/av^* \), \( \tilde{T}_c = T'_S[[a] \cup \{c\} \cup D(c)] \), and \( \tilde{H}_c = \tilde{T}_c/av^* \). See Figure 3 for an illustration of these subgraphs.

![Figure 3](image)

**Fig. 3.** From left to right: \( T_{a_1} \); \( \tilde{T}_{a_1} \); \( H_{a_2} \); \( \tilde{H}_{a_2} \), for two vertices \( a_1 \) and \( a_2 \) of the graph \( T'_S \) shown in Figure 2, right.

With this in mind, let \( a \neq v^* \) be a vertex in \( T'_S \) with children \( c_1, \ldots, c_k \), and suppose we know \( P(T_{c_i}) \) and \( P(H_{c_i}) \), \( 1 \leq i \leq k \). Define the function \( f : V(T'_S) \setminus \{v^*\} \rightarrow \{0, 1\} \) by \( f(v) = 1 \) if \( v \in S \), \( f(v) = 0 \) if \( v \notin S \), and let \( I = \{i : f(c_i) = 1\} \) and \( Z = \{i : f(c_i) = 0\} \) be a partition of the children of
a based on whether or not they are connected to \( v^* \). Then, we can compute \( P(H_a) \) as follows.

\[
P(H_a) = \frac{1}{t^{k-1}} \prod_{i=1}^{k} P(\hat{H}_{c_i})
\]

\[
= \frac{1}{t^{k-1}} \prod_{i} P(\hat{H}_{c_i}) \prod_{Z} P(\hat{H}_{c_{i}})
\]

\[
= \frac{1}{t^{k-1}} \prod_{i} P(T_{c_i}) \prod_{Z} \left( P(T_{c_i}) - P(H_{c_i}) \right)
\]

Here, the first equality follows from (2) and the definition of \( \hat{H}_{c_i} \), and the third equality follows from (1).

Next, we compute \( P(T_a) \) by considering two cases: \( a \) is either connected to \( v^* \) or not. Let \( P_1(T_a) = P(T_a) \) where \( f(a) = 1 \), and \( P_0(T_a) = P(T_a) \) where \( f(a) = 0 \). Clearly, \( P(T_a) = f(a)P_1(T_a) + (1 - f(a))P_0(T_a) \). We now find \( P_1(T_a) \) and \( P_0(T_a) \) separately as follows.

\[
P_1(T_a) = \frac{1}{(t(t-1))^{k-1}} \prod_{i=1}^{k} P(\hat{T}_{c_i})
\]

\[
= \frac{1}{(t(t-1))^{k-1}} \prod_{i} P(\hat{T}_{c_i}) \prod_{Z} P(\hat{T}_{c_{i}})
\]

\[
= \frac{\Pi_I(P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_{Z} \left( P(\hat{T}_{c_i} + c_iv^*) + P(\hat{T}_{c_i}/c_iv^*) \right)
\]

\[
= \frac{\Pi_I(P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_{Z} \left( P(T_{c_i} + c_iv^*)(t-2) + P(H_{c_i})(t-1) \right)
\]

\[
= \frac{\Pi_I(P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_{Z} \left( (P(T_{c_i}) - P(H_{c_i}))(t-2) + P(H_{c_i})(t-1) \right)
\]

\[
= \frac{1}{(t(t-1))^{k-1}} \prod_{I} \left( (P(T_{c_i})(t-2)) \prod_{Z} \left( (t-2)P(T_{c_i}) + P(H_{c_i}) \right) \right)
\]

\[
P_0(T_a) = P(T_a + av^*) + P(T_a/av^*) = P_1(T_a) + P(H_a)
\]

In the computation of \( P_1(T_a) \), the first equality follows from (2), the third equality follows from applying (2) in the first product and (1) in the second product, the fourth equality follows from (2) and the definition of \( \hat{T}_{c_i} \), and the fifth equality follows from applying (1) to the edge \( c_iv^* \). In the computation
of $P_0(T_a)$, the first equality follows from (1), and the second equality follows from the definitions of $P_1$ and $H_a$.

Thus, we have shown how to express $P(T_a)$ and $P(H_a)$ in terms of $P(T_c)$ and $P(H_c)$, $1 \leq i \leq k$. Using these identities, we propose the following algorithm for finding the chromatic polynomial of a generalized vertex join tree $T_S$.

**Algorithm 1**

1. Find and remove the bridges of $T_S$ to acquire $T'_S$
2. For $i = L$ to 0
   - Compute $P(T_a)$ and $P(H_a)$ for each $a \in L_i(T'_S)$
3. Compute $P(T_S)$ using (9)

**Theorem 2** Algorithm 1 finds the correct chromatic polynomial of a generalized vertex join tree $T_S$ using $O(n^2 \log n)$ time and $O(n)$ space.

**PROOF.** We showed in (9) and the preceding discussion that $P(T_S)$ can be computed by finding the bridges of $T_S$ and the chromatic polynomial of $T'_S$. Thus, we only need to verify that Step 2 above correctly computes $P(T'_S)$.

We also established that for every $a \in V(T'_S) \setminus \{v^*\}$, $P(T_a)$ and $P(H_a)$ can be expressed in terms of $\{P(T_c), P(H_c) : c$ is a child of $a\}$. Note that this expression is vacuously satisfiable for vertices which have no children, including all vertices in $L_L(T'_S)$.

Now, for $L > i \geq 0$, vertex $a$ in $L_i(T'_S)$ either has no children, or has all of its children in $L_{i+1}(T'_S)$. In either case, $P(T_c)$ and $P(H_c)$ are known for every child $c$ of $a$ — either vacuously or inductively. Thus, $P(T_a)$ and $P(H_a)$ can also be computed using the formulas derived in this section. Since $P(T'_S) = P(T_r)$ and $L_0(T'_S) = \{r\}$, Algorithm 1 indeed finds the correct chromatic polynomial of $T_S$.

To verify the time-complexity, first note that since $|E(T_S)| = O(n)$, we can find all bridges of $T_S$ in $O(n)$ time using the algorithm of Tarjan [21]. Also, the level and list of children of each vertex of $T'_S - v^*$ can be found with $O(n)$ time by a breadth-first scan. Each evaluation of $P(T_a)$ and $P(H_a)$ requires the multiplication of $O(a_k)$ polynomials, where $a_k$ is the number of children of $a$.

Since we evaluate $P(T_a)$ and $P(H_a)$ for $O(n)$ vertices, and the total number of children in $T'_S$ is $O(n)$, the evaluation of $P(T'_S)$ requires the multiplication of $O(n)$ polynomials. Each of these polynomials has degree at most $O(n)$, since $P(T'_S)$ has degree $O(n)$. The time-complexity of multiplying two polynomials of degree $n$, using a Fast Fourier Transform, is $O(n \log n)$, so the total time complexity of Algorithm 1 is $O(n^2 \log n)$. 

Finally, to verify the space-complexity, note that the total number of vertices in the set of graphs \( \{ T_a, H_a : a \in L(T_S') \} \) is at most \( \mathcal{O}(n) \). The chromatic polynomial of a graph with \( k \) vertices has degree \( k \). Hence, the sum of the degrees of the set of polynomials \{ \( P(T_a), P(H_a) : a \in L(T_S') \) \} is \( \mathcal{O}(n) \). A set of polynomials whose degrees add up to \( n \) can be stored with \( \mathcal{O}(n) \) space. Thus, since we only have to store the polynomials \( P(T_a) \) and \( P(H_a) \) for \( a \) in one level at a time, the total space-complexity of Algorithm 1 is \( \mathcal{O}(n) \). \( \square \)

5 Flow polynomials of outerplanar graphs

Let \( B \) be a biconnected outerplane graph with bounded faces \( F_1, \dotsc, F_s \) and outer face \( F_* \). The weak dual of \( B \) is a tree \( T = (V, E) \), where vertex \( v_i \in T \) corresponds to face \( F_i \in B \) (cf. [20]). Suppose \( F_i \) shares \( f_i \) edges with \( F_* \), and let \( v^* \) be the vertex in \( B^* \) corresponding to \( F_* \). Then, \( B^* \) is the generalized vertex join tree \( T_S \), where \( S \) is the multiset over \( V \) in which \( v_i \) appears \( f_i \) times.

With this in mind, we propose the following procedure to compute the flow polynomial of an arbitrary outerplanar graph \( G \).

**Algorithm 2**

1. Find the biconnected components \( G_1, \dotsc, G_k \) of \( G \)
2. Find the dual graphs \( G_1^*, \dotsc, G_k^* \)
3. Compute \( P(G_1^*), \dotsc, P(G_k^*) \) using Algorithm 1
4. Compute \( F(G) \) by \( F(G) = \frac{1}{n!} \prod_{i=1}^{k} P(G_i^*) \)

**Theorem 3** Algorithm 2 finds the correct flow polynomial of an outerplanar graph \( G \) using \( \mathcal{O}(n^2 \log n) \) time and \( \mathcal{O}(n) \) space.

**PROOF.**

Consider the biconnected components of \( G \) as separate graphs, i.e., \( G_i = G[V_i] \) where \( V_i \) is a maximal subset of \( V(G) \) such that \( G[V_i] \) is biconnected. Then, each \( G_i \) is a biconnected outerplanar graph and by (5) and (4),

\[
F(G) = \prod_{i=1}^{k} F(G_i) = \frac{1}{n!} \prod_{i=1}^{k} P(G_i^*).
\]

Since the dual of a biconnected outerplanar graph is a generalized vertex join tree, Algorithm 1 can be used to compute \( P(G_i^*) \) for \( 1 \leq i \leq k \), so Algorithm 2
indeed finds the correct flow polynomial of $G$.

To verify the time- and space-complexity, let $|V(G)| = n$ and $|V(G_i)| = n_i$; clearly $n_1 + \ldots + n_k = O(n)$. By the algorithms of Hopcroft and Tarjan [7, 8], the biconnected components $G_1^*, \ldots, G_k^*$ of $G$ can be found, embedded in the plane, and have their dual graphs $G_1^*, \ldots, G_k^*$ computed, with $O(n)$ time and space. Finally, note that

$$\sum_{i=1}^{k} \left( n_i \log n_i \right) \leq \left( \sum_{i=1}^{k} n_i \right)^2 \log \left( \sum_{i=1}^{k} n_i \right) = O(n^2 \log n).$$

Hence, Algorithm 1 can be applied to find $P(G_1^*), \ldots, P(G_k^*)$ in $O(n^2 \log n)$ time and $O(n)$ space. Thus, the total time complexity of Algorithm 2 is $O(n^2 \log n)$ and the total space complexity is $O(n)$. \hfill \square

We conclude this section with a characterization of the duality between outerplanar graphs and generalized vertex join trees.

**Proposition 4** Let $G$ be a simple biconnected outerplane graph and $T_S$ be its dual generalized vertex join tree. $G$ is simple if and only if every vertex of $T_S$ has degree at least 3.

**PROOF.** Suppose $G$ is a simple biconnected outerplane graph. $G$ has no parallel edges or loops, so $G$ has no faces of size 1 or 2. Thus, each face of $G$ is incident to at least 3 edges, so each vertex of $T_S$ has degree at least 3.

Now, suppose $T_S$ is a generalized vertex join tree, and that every vertex of $T_S$ has degree at least 3. We will show that $T_S$ is the dual of a simple biconnected outerplanar graph by induction on the number of vertices of $T_S$. If $T_S$ has two vertices $v$ and $v^*$, all the edges in $T_S$ must join $v$ to $v^*$ since by construction, $T_S$ can have no loops. Thus, $T_S$ is the dual of some cycle of size at least 3 (which is simple, biconnected, and outerplanar). Next, let $T_S$ be a generalized vertex join tree on $k+1$ vertices with minimum vertex degree at least 3, and let $v$ be a leaf of $T$. Since $T$ is a tree, $v$ has a unique neighbor $u$ in $T$ with exactly one edge between $u$ and $v$. Moreover, by assumption, $v$ must be connected to $v^*$ by $\ell \geq 2$ edges and $u$ must be incident to at least two edges other than $uv$. Thus, if we delete $v$ from $T_S$ and add an edge from $u$ to $v^*$, we obtain a generalized vertex join tree on $k$ vertices, which by induction is the dual of some simple biconnected outerplanar graph $G$. In this graph, $u$ corresponds to some bounded face $F$ and $v^*$ corresponds to the outer face $F_*$. Since we added an edge $uv^*$, $F$ shares at least one edge $e$ with $F_*$. Now, if we glue a cycle of size $\ell + 1$ to $e$, we obtain a simple biconnected outerplanar graph whose dual is $T_S$. \hfill \square
6 Generalized vertex joins of cliques and cycles

In this section, we compute the chromatic polynomials of generalized vertex joins of complete graphs and cycles, and use duality to compute the flow polynomials of generalized vertex joins of cycles.

6.1 Chromatic polynomials of generalized vertex join cliques

Let $K = (V, E)$ be a complete graph, $S$ be a multiset over $V$ and let $K_S$ be the generalized vertex join of $K$ using $S$. For short, we will call $K_S$ a generalized vertex join clique. Let $|V| = n$, $S'$ be the underlying set of $S$ and $|S'| = s$. Using (3), (2), and (6), we have

$$P(K_S) = P(K_{S'}) = \frac{P(K_n)P(K_{s+1})}{P(K_s)} = (t - s) \prod_{i=0}^{n-1}(t - i).$$

Since in general complete graphs are not planar, graph duality cannot be applied to generalized vertex join cliques to obtain a result about flow polynomials.

**Remark 5** It would be interesting (and likely challenging) to investigate the flow polynomials of generalized vertex join cliques directly. Tutte derived a rather complicated formula for the flow polynomials of cliques; adding a vertex with arbitrary connections to the others will complicate this formula even more.

6.2 Chromatic and flow polynomials of generalized vertex join cycles

Let $C = (V, E)$ be a cycle, $S$ be a multiset over $V$ and let $C_S$ be the generalized vertex join of $C$ using $S$. For short, we will call $C_S$ a generalized vertex join cycle. In the literature, graphs of a similar form have also been called “generalized wheel” graphs, and have been investigated by other approaches and in different contexts (cf. [9, 17]). In the remainder of this section, we will present formulas for $P(C_S)$ and $F(C_S)$ in a unified framework.

Without loss of generality, $C_S$ may be given a “wheel” embedding obtained by placing $v^*$ in the bounded face of a plane drawing of $C$ and drawing edges from the vertices in $S$ to $v^*$ so that the resulting graph remains plane; this embedding of $C_S$ is unique up to topological conjugacy. We label the vertices along the outer face of $C_S$ in clockwise order as $v_1, \ldots, v_n$; see Figure 4. The edges incident to $v^*$ will be called spokes.
If $S = \emptyset$, then by (7), $P(C_S) = tP(C_n) = t((t-1)^n + (-1)^n(t-1))$ and $F(C_S) = t - 1$; thus, suppose hereafter that $S \neq \emptyset$ and consider $S'$, the underlying set of $S$. By (3), $P(C_S) = P(C_{S'})$. Without loss of generality, suppose that $S' = \{v_1, \ldots, v_a\}$ where $1 = a_1 < \ldots < a_s$. Also, let $e_1, \ldots, e_s$ be the spokes of $C_{S'}$, with $e_i = v_a \cdot v^*$, and $F_1, \ldots, F_s$ be the faces of $C_{S'}$, with $F_i$ clockwise of edge $e_i$; see Figure 4 for an illustration.

Let $f_i$ be the size of face $F_i$, i.e., the number of edges along the boundary of $F_i$. It is easy to see that $f_1 = 2 + a_i + a_i - 1$, for $1 \leq i \leq s - 1$ and $f_s = 2 + (n + 1) - a_s$. For $1 \leq i \leq s$, define $C_i = C_{S'} - \{e_i, \ldots, e_s\}$ and for notational simplicity, $C_{S'}^{+1} = C_{S'}$. Then, applying (1) consecutively on the edges $e_s, \ldots, e_1$ yields the following decomposition; see Figure 5 for an illustration.

\[
P(C_{S'}) = P(C_{S'} - e_s) - P(C_{S'/e_s})
= P(C_{S'}^s) - P(C_{S'}^{s+1}/e_s)
= P(C_{S'}^s) - P(C_{S'/e_{s-1}}) - P(C_{S'}^{s+1}/e_s)
\vdots
= P(C_{S'}^1) - P(C_{S'/e_1}) - \ldots - P(C_{S'/e_{s-1}}) - P(C_{S'}^{s+1}/e_s)
= tP(C_n) - \sum_{i=1}^s P(C_{S'}^{i+1}/e_i).
\]

Thus, $P(C_{S'})$ is decomposed into the chromatic polynomials of the collection of graphs $\{C_{S'}^{i+1}/e_i\}_{i=1}^s$. The faces of $C_{S'}$ can be regarded as cycles of sizes $f_1, \ldots, f_s$. Let $U_i$ be the face of $C_{S'}^{i+1}$ corresponding to the union of $F_i, \ldots, F_s$ after edges $e_{i+1}, \ldots, e_s$ are deleted. Then, the faces of $C_{S'}^{i+1}$ have sizes $f_1, \ldots, f_{i-1}, u_i$, where $u_i$ is the size of $U_i$; more precisely,

\[
u_i = 2 + (f_i - 2) + \ldots + (f_s - 2) = 2(i - s) + \sum_{j=i}^s f_j.
\]

Let $J_i$ be the multiset of sizes of faces of $C_{S'}^{i+1}/e_i$, i.e., $J_1 = \{n\}$ and for
Fig. 5. Decomposing $C_{S'}$ (on far left) into simpler graphs as described in (10). Using (2), the graphs in the top row can be further decomposed into the cycles making up their bounded faces.

$2 \leq i \leq s$,

$$J_i = \{f_1, \ldots, f_{i-2}, f_{i-1} - 1, u_i - 1\}.$$  \hspace{1cm} (12)

Then, starting from a face of $C_{S'}^{i+1}/e_i$ which borders the contracted edge, and using the fact that this face shares just one edge with the rest of the graph, we can successively apply (2) to evaluate $P(C_{S'}^{i+1}/e_i)$. In particular,

$$P(C_{S'}^{i+1}/e_i) = P(K_2) \prod_{j \in J_i} P(C_j).$$

Thus, by (10) we have

$$P(C_S) = P(C_{S'}) = tP(C_n) - \sum_{i=1}^s \frac{\prod_{j \in J_i} P(C_j)}{P(K_2)^{i-1}}$$

$$= t((t-1)^n + (-1)^n(t-1)) - \sum_{i=1}^s \frac{\prod_{j \in J_i} ((t-1)^j + (-1)^j(t-1))}{(t(t-1))^{i-1}}. \hspace{1cm} (13)$$

Note that (13) depends only on the sequence of face-sizes of $C_{S'}$ and hence only on $S$.

Finally, to find the flow polynomial of $C_S$, note that by (4), $F(C_S) = \frac{1}{t} P(C_S^*)$. But $C_S^*$ is again a generalized vertex join cycle. To see why, note that each bounded face of $C_S$ is incident to two spokes — hence the weak dual of $C_S$ is a cycle; in addition, each bounded face of $C_S$ may share any number of edges with the outer face, making the vertex of $C_S^*$ corresponding to the outer face of $C_S$ a generalized vertex join. See Figure 6 for an illustration.

Recall that $s$ denotes the size of $S'$, the underlying set of $S$. Let $\tilde{s} = |S|$ and let $\tilde{C}$ be the weak dual of $C_S$; $\tilde{C}$ is a cycle with $\tilde{s}$ vertices. Let $\tilde{S}$ be the multiset of vertices of $\tilde{C}$ such that $C_{S'}^* = \tilde{C}_{\tilde{S}}$ and let $\tilde{S}'$ be the underlying set of $\tilde{S}$. Then,

$$F(C_S) = \frac{1}{t} P(C_S^*) = \frac{1}{t} P(\tilde{C}_{\tilde{S}}) = \frac{1}{t} P(\tilde{C}_{\tilde{S}'}). \hspace{1cm} (14)$$

It is easy to see that $C_{S'}$ and $\tilde{C}_{\tilde{S}'}$ have the same number of faces. Moreover, if $\tilde{f}_1, \ldots, \tilde{f}_s$ are the sizes of the faces of $\tilde{C}_{\tilde{S}'}$ in clockwise order, then $\tilde{f}_i$ equals
Fig. 6. Left: $C_S$ and its weak dual. Right: $C_S^*$, the dual of $C_S$, is also a generalized vertex join cycle.

the multiplicity of $v_a$ in $S$ plus 2. Thus, to find $F(C_S)$, we simply plug in the sequence of face-sizes of $C_S^*$ into (11), (12), and (13) as follows:

$$F(C_S) = (t - 1)^\tilde{s} + (-1)^\tilde{s}(t - 1) - \frac{1}{t} \sum_{i=1}^{s} \prod_{j \in \tilde{J}_i} ((t - 1)^j + (-1)^j(t - 1)) \left(\frac{1}{t(t - 1)}\right)^{i-1},$$

where $\tilde{J}_1 = \{\tilde{s}\}$ and $\tilde{J}_i = \{\tilde{f}_1, \ldots, \tilde{f}_{i-2}, \tilde{f}_{i-1} - 1, 2(i - s) + \sum_{j=i}^{s} \tilde{f}_j - 1\}$ for $2 \leq i \leq s$. Note that this closed formula again depends only on $S$, since the face-sizes of $C_S^*$ are determined from $S$.

7 Conclusion

We have found low-order polynomial time algorithms for computing the chromatic polynomials of generalized vertex join trees and the flow polynomials of outerplanar graphs. We have also derived closed formulas for the chromatic polynomials of generalized vertex join cliques, and the chromatic and flow polynomials of generalized vertex join cycles. Future work will focus on computing the chromatic polynomials of generalized vertex joins of other families of graphs, and of graphs obtained by a sequence of generalized vertex joins.

8 Acknowledgements

The authors thank the two anonymous referees whose valuable remarks and suggestions helped improve the presentation.

This material is based upon work supported by the National Science Foundation under Grants No. 1450681, CMMI-1300477, and CMMI-1404864.
References

[1] Beraha, S., Kahane, J., and Weiss, N. Limits of chromatic zeros of some families of maps. *Journal of Combinatorial Theory, Series B*, 25(1): 52–65, 1980.

[2] Bondy, A. and Murty, U.S.R. Graph Theory. *Graduate Texts in Mathematics* 244, Springer, 2008.

[3] Chao, C.Y. and Li, N.Z. On trees of polygons. *Archiv der Mathematik*, 45(2): 180–185, 1985.

[4] Ellis-Monaghan, A. and Merino, C. Graph polynomials and their applications I: The Tutte polynomial. *Structural Analysis of Complex Networks* 219–255, 2011.

[5] Goldberg, L.A. and Jerrum, M. Inapproximability of the Tutte polynomial. *Information and Computation*, 206(7): 908–929, 2008.

[6] Gordon, G. Chromatic and Tutte polynomials for graphs, rooted graphs and trees. *Graph Theory Notes of New York*, LIX: 39–50, 2008.

[7] Hopcroft, J. and Tarjan, R. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6): 372–378, 1973.

[8] Hopcroft, J. and Tarjan, R. Efficient planarity testing. *Journal of the ACM*, 21(4): 549–568, 1974.

[9] Koh, K.M. and Teo, C.P. The chromatic uniqueness of certain broken wheels. *Discrete Mathematics*, 96(1): 65–69, 1991.

[10] Lazuka, E. On chromaticity of graphs. *Discussiones Mathematicae Graph Theory*, 15(1): 19–31, 1995.

[11] Lieb, E.H. Residual entropy of square ice. *Phys. Rev.*, 162(1): 162–172, 1967.

[12] Lovász, L., Thomassen, C., Wu, Y., and Zhang, C.-Q. Nowhere-zero 3-flows and modulo k-orientations. *Journal of Combinatorial Theory*, 103(5): 587–598, 2013.

[13] Mohr, A. and Porter, T. Applications of Stirling numbers involving chromatic polynomials. *Journal of Combinatorial Mathematics and Combinatorial Computing* 70: 57–64, 2009.

[14] Mphako-Banda, E. Some polynomials of flower graphs. *International Mathematical Forum* 2(51): 2511–2118, 2007.

[15] Mphako-Banda, E. Tutte polynomials of flower graphs. *Bulletin of the Iranian Mathematical Society* 35(2): 179–190, 2009.

[16] Oxley, J. and Welsh, D. Chromatic, flow and reliability polynomials: the complexity of their coefficients. *Combinatorics, Probability and Computing*, 11(04): 403–426, 2002.

[17] Shahmohamad, H. A survey on flow polynomial. *Utilitas Mathematica Publishing Incorporated*, 62: 13–32, 2002.

[18] Sokal, A. The multivariate Tutte polynomial (alias Potts model) for graphs and matroids. *Surveys in combinatorics*, Cambridge University Press, 173–226, 2005.

[19] Stanley, R.P. Acyclic orientations of graphs. *Classic Papers in Combinatorics*, 453–460, 1987.

[20] Syslo, M. Characterizations of outerplanar graphs. *Discrete Mathematics*, 15
26(1): 47–53, 1979.

[21] Tarjan, E. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6): 160–161, 1974.

[22] Thomassen, C. Chromatic roots and Hamiltonian paths. *Journal of Combinatorial Theory*, 80(2): 218–224, 2000.

[23] Tutte, T.W. Graph Theory. *Encyclopedia of Mathematics and its Applications* 21, Addison-Wesley Publishing Company, 1984.

[24] Wakelin, C.D. and Woodall, D.R. Chromatic polynomials, polygon trees, and outerplanar graphs. *J. Graph Theory*, 16(5): 459–466, 1992.

[25] Whitehead, E. Chromatic polynomials of generalized trees. *Discrete Mathematics*, 72: 391–393, 1988.

[26] Whitehead, E., Chromaticity of two-trees. *Journal of Graph Theory* 9(2): 279–284, 1985.

[27] Zhang, C.-Q. Integer flows and cycle covers of graphs. vol. 205, CRC Press, 1997.