The Loopy Polynomial: A Tutte-Type Graph Invariant

Anatol Kirillov[*] Gleb Nenashev[†] Boris Shapiro[‡]
Arkady Vaintrob[§]

Abstract

We introduce the *loopy polynomial* $\mathscr{L}_G(t, \mathbf{x})$, a new multivariate graph invariant defined by a deletion–contraction rule in which ordinary contraction is replaced by *loopy contraction*: the endpoints of the contracted edge are identified, but the edge itself is retained as a loop at the merged vertex. We show that this recursion is well defined, derive a spanning-forest expansion for $\mathscr{L}_G(t, \mathbf{x})$ in terms of external activities, and prove a 4-term relation for the loopy polynomial. We also establish concrete links with classical graph invariants. In particular, a specialization of $\mathscr{L}_G$ recovers the spanning-forest specialization of the Tutte polynomial, while for trees the loopy polynomial is closely related to connected partition polynomials, the U-polynomial, and Stanley's chromatic symmetric function. These results place the loopy polynomial within the broader landscape of Tutte-type and activity-based graph invariants.

---

[*] Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Huairou District, Beijing, China, kirillov@bimsa.cn

[†] Department of Mathematics and Computer Science, St. Petersburg State University, St. Petersburg, 199178, Russia, glebnen@gmail.com

[‡] Department of Mathematics, Stockholm University, SE-106 91 Stockholm, Sweden and Department of Mathematics with Computer Science, Guangdong Technion-Israel Institute of Technology 241 Daxue road, 515063 Shantou, P. R. China shapiro@math.su.se

[§] Department of Mathematics, University of Oregon, Eugene, OR 97403, USA, vaintrob@uoregon.edu

1 Introduction

Let $G = (V, E)$ be a finite undirected graph. Throughout the paper, graphs are allowed to have loops and multiple edges; in other words, our graphs are multigraphs in the standard terminology.

The Tutte polynomial is one of the central invariants of graph theory and matroid theory. It admits many equivalent definitions and encodes a remarkable amount of combinatorial information; see, for example, [EM]. One of its most important features is the deletion–contraction relation, which recursively reduces a graph by either deleting an edge or contracting it. A second, equally fundamental description is given by Tutte's activity expansion over spanning forests.

The starting point of the present paper is the observation that the usual contraction operation can be modified in a natural way. Instead of deleting a contracted non-loop edge after identifying its endpoints, one may keep that edge as a loop attached to the merged vertex. This leads to a different recursive theory and, as we show below, to a new graph polynomial with a rich combinatorial structure.

More precisely, if e is a non-loop edge of G, we denote by G/e the graph obtained by identifying the endpoints of e and turning e into a loop at the new merged vertex. We call this operation *loopy contraction*. Our main object is the corresponding multivariate invariant $\mathscr{L}_G(t, \mathbf{x})$, defined by a loopy deletion–contraction relation together with multiplicativity and natural initial conditions on one-vertex graphs with loops.

The loopy polynomial shares several formal features with the Tutte polynomial, but it also exhibits genuinely new behavior. In particular, its most useful description is not merely recursive: it admits a spanning-forest expansion in which the contribution of each component depends on its size together with an activity statistic. This makes $\mathscr{L}_G$ an activity-based graph invariant of Tutte type, but with a different mechanism for recording cycle data.

The main results of the paper are as follows.

- We prove that the loopy deletion–contraction procedure is well defined, i.e. independent of the order in which it is applied; see Proposition 2.2.
- We establish a spanning-forest formula for $\mathscr{L}_G(t, \mathbf{x})$; see Theorem 2.3.

This formula plays the same role for the loopy polynomial as Tutte's activity expansion does for the Tutte polynomial.

- We prove that $\mathscr{L}_G$ satisfies a 4-term relation; see Theorem 2.5. This relation is one of the structural features that distinguishes the loopy polynomial from more classical graph invariants.
- We relate $\mathscr{L}_G$ to several known graph polynomials. In particular, a specialization of $\mathscr{L}_G$ recovers the spanning-forest specialization of the Tutte polynomial, while on trees the loopy polynomial specializes to invariants closely related to connected partition polynomials, the U-polynomial, and Stanley's chromatic symmetric function.

Thus, the loopy polynomial should be viewed not simply as another recursively defined graph polynomial, but as a new invariant situated at the intersection of deletion–contraction theory, spanning-forest enumeration, and activity-based graph combinatorics.

Let us briefly recall the corresponding classical picture for the Tutte polynomial.

Definition 1.1. The Tutte polynomial $T_G(x, y)$ is the unique bivariate polynomial with integer coefficients determined by the following properties:

(i) if G_1 and G_2 are disjoint graphs, then

$$T_{G_1 \sqcup G_2} = T_{G_1} T_{G_2};$$

(ii) if $e \in E$ is a loop, then

$$T_G = y T_{G-e};$$

(iii) if $e \in E$ is a bridge, then

$$T_G = x T_{G-e};$$

(iv) if U is the one-vertex graph without edges, then

$$T_U = 1;$$

(v) if $e \in E$ is neither a loop nor a bridge, then

$$T_G = T_{G-e} + T_{G\cdot e}, \tag{1.1}$$

where $G \cdot e$ is the usual contraction of e.

We shall also use Tutte's activity expansion. By a *spanning forest* of G we mean a spanning acyclic subgraph of G. If G is connected, a spanning forest is a spanning tree. As usual, we identify spanning forests with their edge sets.

Definition 1.2. Let F be a spanning forest of G, and let $e \in E \setminus F$. The *fundamental cycle* $Z_F(e)$ is the unique cycle in $F \cup e$. If $e \in F$, the *fundamental cut* $U_F(e)$ is the unique cut determined by deleting e from F.

Fix a linear order $\prec$ on E. For a spanning forest $F \subseteq E$, an edge $e \in F$ is called *internally active* if it is the smallest edge in $U_F(e)$, and an edge $e \in E \setminus F$ is called *externally active* if it is the smallest edge in $Z_F(e)$. We write $I(F)$ and $E(F)$ for the sets of internally and externally active edges, respectively, and denote by $\varepsilon(F) = |E(F)|$ the external activity of F.

Definition 1.3 (see [Tu]). If $G = (V, E)$ is equipped with a linear order on E, then

$$T_G(x, y) = \sum_F x^{|I(F)|} y^{|E(F)|},$$

where the sum ranges over all spanning forests F of G.

Tutte showed that this expression is independent of the chosen order on E and agrees with Definition 1.1.

We now introduce the main object of the paper.

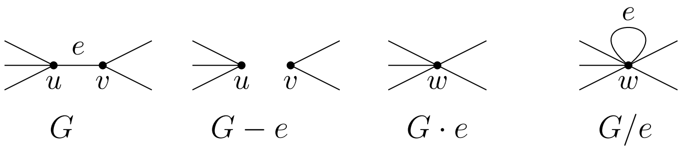


Figure 1: Deletion, ordinary contraction, and loopy contraction

Definition 1.4. The *loopy polynomial* of a graph $G = (V, E)$ is the polynomial

$$\mathscr{L}_G(t, \mathbf{x}) = \mathscr{L}_G(t, x_0, x_1, \ldots) \in \mathbb{Z}[t, x_0, x_1, \ldots]$$

determined by the following properties:

(i) for every non-loop edge $e \in E$,

$$\mathscr{L}_G = \mathscr{L}_{G/e} + t\, \mathscr{L}_{G-e}; \tag{1.2}$$

(ii) if G_1 and G_2 are disjoint graphs, then

$$\mathscr{L}_{G_1 \sqcup G_2} = \mathscr{L}_{G_1} \mathscr{L}_{G_2};$$

(iii) if L_m denotes the graph with one vertex and m loops, then

$$\mathscr{L}_{L_m} = x_m. \tag{1.3}$$

Remark 1.5. The polynomial $\mathscr{L}_G(t, \mathbf{x})$ arose in the course of the work [KNSV], where specializations of $\mathscr{L}_G$ appear as Hilbert series of the so-called bizonotopal algebras of graphs; see Section 2.3.

2 Properties of loopy polynomials

2.1 Correctness and alternative definition

Given a graph G, Properties (i)–(iii) in Definition 1.4 allow us to calculate $\mathscr{L}_G(t, \mathbf{x})$ using induction on the number of edges if we arbitrarily order all edges of G and consecutively apply loopy deletion-contraction (1.2) to the maximal in this order remaining non-loop edge. The next claim shows that this procedure is independent of the arbitrarily chosen order of edges and therefore is well-defined.

Proposition 2.1. *Let e be a bridge of G. Then*

$$\mathscr{L}_G = \mathscr{L}_{G/e} + t\mathscr{L}_{G-e}.$$

Moreover, if $G - e = G_1 \sqcup G_2$, then

$$\mathscr{L}_G = \mathscr{L}_{G/e} + t\mathscr{L}_{G_1}\mathscr{L}_{G_2}.$$

Proof. The first identity is the defining loopy deletion–contraction relation applied to the bridge e. If $G - e = G_1 \sqcup G_2$, then multiplicativity gives

$$\mathscr{L}_{G-e} = \mathscr{L}_{G_1}\mathscr{L}_{G_2},$$

proving the second identity. □

Proposition 2.2. *The loopy polynomial $\mathscr{L}_G$ is well-defined, i.e. independent of the order of edges in the process of loopy deletion-contraction.*

Proof. To settle Proposition 2.2, it suffices to consider the case when two orders of edges differ only by a simple transposition. Assume that two given orders of edges are different by a transposition of $e, f \in E$. Let us consider the step of the loopy deletion-contraction procedure when there are no longer non-loop edges larger than e and f (in both orders) and denote the graph remaining on this step as G'. We need to handle three different cases.

Case 1: If either e, f, or both are loops in G', the result of loopy deletion-contraction is obviously independent of the order of e and f.

Case 2: When e, f are parallel then it is also clear that both orders give the same result for $\mathscr{L}_G(t, \mathbf{x})$.

Case 2: The cases when e, f are neither parallel nor at least one of them is a loop is also trivial, because after the loopy deletion-contraction applied to e, the edge f is still a non-loop and vice versa. □

The next result shows that the loopy polynomial $\mathscr{L}_G(t, \mathbf{x})$ of G can be computed as a sum over the set of spanning forests of G in a way similar to Definition 1.3.

Theorem 2.3. *Fix a linear order on $E(G)$. For a spanning forest F of G and a component $T \in c(F)$, let $\varepsilon(T)$ denote the number of externally active edges assigned to T. Then*

$$\mathscr{L}_G(t, \mathbf{x}) = \sum_{F \in \mathscr{F}(G)} t^{|E(G)| - e(F) - \sum_{T \in c(F)} \varepsilon(T)} \prod_{T \in c(F)} x_{e(T) + \varepsilon(T)}. \tag{2.1}$$

Proof. Let $\Phi_G(t, \mathbf{x})$ denote the right-hand side of (2.1). We prove that Φ_G satisfies the defining loopy deletion–contraction recurrence.

The claim is immediate if all edges are loops: then the only spanning forest is the empty forest, and all loops are externally active, giving

$$\Phi_{L_m} = x_m = \mathscr{L}_{L_m}.$$

Assume now that G has at least one non-loop edge, and let e_0 be the largest non-loop edge in the chosen order. We use the convention fixed above: an edge is externally active if it is the smallest edge in its fundamental cycle. We prove

$$\Phi_G = \Phi_{G/e_0} + t\,\Phi_{G-e_0}.$$

Partition the spanning forests of G into those not containing e_0 and those containing e_0.

First suppose $e_0 \notin F$. Then F is naturally a spanning forest of $G - e_0$. Since e_0 is the largest edge, it is not externally active with respect to F: if $F \cup e_0$ contains a cycle, then e_0 is the largest edge of that cycle, hence not the smallest. All other fundamental cycles and their least elements are unchanged when passing from G to $G-e_0$. Therefore the component weights are the same in G and $G - e_0$, while the exponent of t is larger by one in G. Thus the total contribution of forests not containing e_0 is

$$t\,\Phi_{G-e_0}.$$

Now suppose $e_0 \in F$. Loopy contraction of e_0 identifies its endpoints and turns e_0 into a loop. The set

$$F' := F \setminus \{e_0\}$$

is then a spanning forest of G/e_0, and this construction gives a bijection between spanning forests of G containing e_0 and spanning forests of G/e_0.

We compare the weights of F and F'. The edge e_0 contributes one forest edge to the component of F containing it. After loopy contraction, the same edge becomes a loop, hence is externally active and contributes one unit to the activity of the corresponding component of F'. Thus the quantity

$$e(T) + \varepsilon(T)$$

attached to the affected component is preserved.

For any other edge $f \neq e_0$, the fundamental cycle of f is either unchanged, or is obtained from the old fundamental cycle by contracting the largest edge e_0. In either case, the smallest edge of the cycle is unchanged. Hence external activity is preserved for all edges different from e_0. Therefore

$$\sum_{T\in c(F')} \varepsilon_{G/e_0}(T) = 1 + \sum_{T\in c(F)} \varepsilon_G(T).$$

Since $|E(G/e_0)| = |E(G)|$ and $e(F') = e(F) - 1$, the exponent of t is also preserved:

$$|E(G/e_0)| - e(F') - \sum_T \varepsilon_{G/e_0}(T) = |E(G)| - e(F) - \sum_T \varepsilon_G(T).$$

Thus the total contribution of forests containing e_0 is Φ_{G/e_0}.

Combining the two parts gives

$$\Phi_G = \Phi_{G/e_0} + t\,\Phi_{G-e_0}.$$

Together with the initial condition for one-vertex graphs with loops, this proves $\Phi_G = \mathscr{L}_G$. $\square$

Proposition 2.4. *If G is a forest, then*

$$\mathscr{L}_G(t, \mathbf{x}) = \sum_{A \subseteq E(G)} t^{|E(G)|-|A|} \prod_{C \in c(A)} x_{|V(C)|-1}.$$

Proof. Since G is acyclic, no edge outside A is externally active. The result follows directly from Theorem 2.3. $\square$

2.2 4-term recurrence relation and algorithm for calculation of loopy polynomials

2.2.1 4-term recurrence

The next result is of fundamental importance for the study of loopy polynomials.

Theorem 2.5. *Let u, v, w be three distinct vertices of G, and let*

$$e_1 = (uv), \qquad e_2 = (vw), \qquad e_3 = (uw).$$

Then

$$\mathscr{L}_{G+\{e_1,e_3\}} - t\,\mathscr{L}_{G+\{e_1\}} = \mathscr{L}_{G+\{e_2,e_3\}} - t\,\mathscr{L}_{G+\{e_2\}}.$$

Proof. Apply the loopy deletion–contraction relation to the edge e_3 in the graph $G + \{e_1, e_3\}$. Since e_3 is a non-loop edge, we obtain

$$\mathscr{L}_{G+\{e_1,e_3\}} = \mathscr{L}_{(G+\{e_1,e_3\})/e_3} + t\,\mathscr{L}_{G+\{e_1\}}.$$

Hence

$$\mathscr{L}_{G+\{e_1,e_3\}} - t\,\mathscr{L}_{G+\{e_1\}} = \mathscr{L}_{(G+\{e_1,e_3\})/e_3}.$$

Similarly,

$$\mathscr{L}_{G+\{e_2,e_3\}} = \mathscr{L}_{(G+\{e_2,e_3\})/e_3} + t\,\mathscr{L}_{G+\{e_2\}},$$

and therefore

$$\mathscr{L}_{G+\{e_2,e_3\}} - t\,\mathscr{L}_{G+\{e_2\}} = \mathscr{L}_{(G+\{e_2,e_3\})/e_3}.$$

It remains to observe that the two contracted graphs

$$(G+\{e_1,e_3\})/e_3 \qquad \text{and} \qquad (G+\{e_2,e_3\})/e_3$$

are canonically isomorphic. Indeed, contracting $e_3 = (uw)$ identifies u and w into a single vertex. Under this identification, both $e_1 = (uv)$ and $e_2 = (vw)$ become an edge between the merged vertex and v, while e_3 itself becomes a loop at the merged vertex. All other edges of G are affected in the same way in the two contractions.

Thus

$$\mathscr{L}_{(G+\{e_1,e_3\})/e_3} = \mathscr{L}_{(G+\{e_2,e_3\})/e_3},$$

and the desired identity follows. □

2.2.2 Algorithm of inductive construction of $\mathscr{L}_G$

The procedure suggested below does not change the number of loops and vertices except when we are using the multiplicativity property. It is recursive on the number of edges and it substitutes a graph by a simpler one in such a way that the difference of their loopy polynomials is the alternating sum of the loopy polynomials of graphs with fewer edges.

Step 1. Given a graph H with m edges, assume that H is connected and choose any vertex $v \in H$. (Otherwise use the multiplicativity property.)

Step 2. Consider the vertices u, w such that there is a path $v - w - u \in H$. Using the notation $e_1 = (uv), e_2 = (vw), e_3 = (uw)$, observe that edges $e_2, e_3 \in H$ and consider the graph $G - \{e_2, e_3\}$.

Step 3. Using Theorem 2.5 find the relation between $\mathscr{L}_H$, $\mathscr{L}_{G+\{e_1,e_3\}}$ and loopy polynomials for smaller graphs. (The degree of vertex v is greater in $G + \{e_1, e_3\}$ than in H.)

Step 4. Repeat Steps 1–3 several times till we obtain a graph H' with m edges all of them either being loops or being incident to v. Furthermore, $\mathscr{L}_{H'} - \mathscr{L}_H$ is now expressed through loopy polynomials of smaller graphs, i.e. having fewer than m edges.

Step 5. Take a vertex $u' \in H$ such that there are multiple edges (vu'). Then we have

$$L_{H'} - L_{H'-(vu)} = L_{H'-(vu')+(uu')} - L_{H'-(vu')+(uu')-(vu)}$$

and

$$L_{H'-(vu')+(uu')} - L_{H'-(vu')+(uu')-(vu)} = L_{H'-(vu')+(vu)} - L_{H'-(vu')}.$$

Hence, we can get graph $\tilde{H}$ such that all non-loop multiple edges incident to v are (vu) and $\mathscr{L}_{\tilde{H}} - \mathscr{L}_H$ has an expression using loopy polynomials of smaller graphs.

Step 6. Calculate the loopy polynomial of $\tilde{H}$ using Theorem 2.5. (Observe that it has only $2^{|V(H)|-2} \cdot (\ell - |V(H)| + 2)$ terms, where ℓ is the number of non-loop edges in H.)

2.3 Further properties

Here are some immediate consequences of the above results.

Corollary 2.6. *In the above notation, the loopy polynomial $\mathscr{L}_G$ enjoys the following properties.*

(1) The monomial of the highest degree in x_0 in the specialization $\mathscr{L}_G(1, \mathbf{x})$ equals $x_0^{|V|}$.

(2) The monomial of the second highest degree in x_0 in $\mathscr{L}_G(1, \mathbf{x})$ is

$$|E| x_0^{|V|-2} x_1.$$

(3) $\mathscr{L}_G(t, \mathbf{x})$ is a weighted homogeneous polynomial with weights

$$\deg t = 1, \qquad \deg x_i = i, \quad i = 0, 1, \ldots.$$

Its total weighted degree is $|E|$. In particular, $\mathscr{L}_G(t, \mathbf{x})$ can be reconstructed from its specialization $\mathscr{L}_G(1, \mathbf{x})$.

(4) The specialization $x_0 = \dots x_{|E|} = 1$ gives the Hilbert polynomial of the external zonotopal algebra $\mathscr{A}_G^e$, see [KNSV], i.e.

$$\mathscr{L}_G(t, 1, \dots, 1) = \sum_{n \geq 0} t^n \dim(\mathscr{A}_G^e)^{(n)} .$$

(5) Let

$$\mathscr{L}_G^{(1)}(t, x_0, \dots, x_{|E|}) := \sum_{i=0}^{|E|} c_i t^{|E|-i} x_i$$

be the part of the polynomial $\mathscr{L}_G$ that is linear in the $\mathbf{x}$-variables, where $m = |E|$. Then substituting $x_0 = \dots = x_{|E|} = 1$ we obtain the univariate polynomial

$$h(t) = \sum_{i=0}^{|E|} c_i t^{|E|-i},$$

which coincides with the Hilbert polynomial of the central zonotopal algebra $\mathscr{A}_G^c$, [KNSV].

(6) Denote by $h_G^e(t) := \sum_{k \geq 0} \dim(\mathscr{B}_G^e)^{(k)} \cdot t^k$ the Hilbert series of the external bizonotopal algebra $\mathscr{B}_G^e$, [KNSV]. Then, for a graph G, we have

$$h_G^e(t) = \quad \mathscr{L}_G \left(t, 1, \frac{1-t^2}{1-t}, \frac{1-t^3}{1-t}, \frac{1-t^4}{1-t}, \dots \right).$$

Sketch of proof. Items (1), (2), and (3) follow from the spanning-forest expansion of Theorem 2.3.

Indeed, setting $t = 1$ in (2.1), we obtain

$$\mathscr{L}_G(1, \mathbf{x}) = \sum_{F \in \mathscr{F}(G)} \prod_{T \in c(F)} x_{e(T)+\varepsilon(T)}.$$

The highest possible degree in x_0 comes from the empty forest $F = \varnothing$, whose connected components are the isolated vertices. This gives the monomial $x_0^{|V|}$, proving (1).

The next highest degree in x_0 comes from forests with exactly one non-loop edge and all remaining vertices isolated. There are exactly $|E|$ such forests, and each contributes

$$x_1 x_0^{|V|-2},$$

which proves (2).

For (3), each term has degree

$$(|E| - e(F) - \varepsilon(F)) + (e(F) + \varepsilon(F)) = |E|.$$

Item (4) is the specialization $x_0 = \cdots = x_{|E|} = 1$ of the corresponding result from [KNSV], identifying this specialization with the Hilbert polynomial of the external zonotopal algebra $\mathscr{A}^e_G$.

Item (5) is the corresponding specialization of the linear part of $\mathscr{L}_G$ appearing in [KNSV], where it is identified with the Hilbert polynomial of the central zonotopal algebra $\mathscr{A}^c_G$.

Finally, item (6) follows from the formula for the Hilbert series of the external bizonotopal algebra proved in [KNSV], rewritten in terms of the loopy polynomial via the substitution

$$x_i = \frac{1 - t^{i+1}}{1 - t}, \qquad i \geq 1.$$

□

3 Relations to other graph polynomials

In this section we relate the loopy polynomial $\mathscr{L}_G(t, \mathbf{x})$ to several well-known graph invariants. The key tool is the spanning-forest expansion

$$\mathscr{L}_G(t, \mathbf{x}) = \sum_{F \in \mathscr{F}(G)} t^{|E(G)| - |F| - \varepsilon(F)} \prod_{T \in c(F)} x_{|E(T)| + \varepsilon(T)}.$$

Since each component T of a forest satisfies $|E(T)| = |V(T)| - 1$, the weights can be written as $x_{|V(T)| - 1 + \varepsilon(T)}$.

3.1 Relation to spanning forests

Proposition 3.1.

$$\mathscr{L}_G(1, 1, 1, \dots)$$

equals the number of spanning forests of G.

Proof. At $t = 1$ and $x_i = 1$ for all i, every summand in the spanning-forest expansion of Theorem 2.3 contributes 1. Hence the specialization counts spanning forests. □

3.2 Relation to strong U-type invariants

We now formulate a more precise structural connection between the loopy polynomial and stronger multivariate graph invariants, such as the strong U-polynomial, which record both component sizes and their cycle structure.

Recall that the strong U-polynomial has the form

$$\bar{U}_G(z_{i,j}) = \sum_{A \subseteq E} \prod_{C \in c(A)} z_{|V(C)|,\,|E(C)|-|V(C)|+1},$$

where each connected component C contributes according to its number of vertices and its cyclomatic number.

In contrast, the loopy polynomial $\mathscr{L}_G$ is expressed as a sum over spanning forests, with each tree component T weighted by

$$|V(T)| - 1 + \varepsilon(T),$$

where $\varepsilon(T)$ records the number of externally active edges assigned to T.

This suggests that $\mathscr{L}_G$ is obtained from a refinement of $\bar{U}_G$ by passing to spanning forests and redistributing cycle information via activities.

3.3 Connected partitions and the U-polynomial on trees

For a tree T, every spanning subgraph is a forest and every component is itself a tree. Moreover $\varepsilon(\cdot) = 0$ identically. Hence

$$\mathscr{L}_T(1, \mathbf{x}) = \sum_{F \subseteq E(T)} \prod_{C \in c(F)} x_{|V(C)|-1}.$$

Proposition 3.2. *Let T be a tree. Then*

$$\mathscr{L}_T(1, x, x, \dots) = Q(T, x),$$

where $Q(T, x)$ is the connected partition polynomial.

Proof. Each spanning forest $F \subseteq E(T)$ corresponds uniquely to a partition of $V(T)$ into connected blocks (its components), and every connected partition arises in this way. The specialization $x_i = x$ assigns weight $x^{k(F)}$, which is exactly the definition of $Q(T, x)$. □

More generally, one obtains a weighted version.

Proposition 3.3. *For a tree T,*

$$\mathscr{L}_T(1, x_0, x_1, \dots) = \sum_{\pi} \prod_{B \in \pi} x_{|B|-1},$$

where the sum runs over all connected partitions π of $V(T)$.

We now compare with the U-polynomial of Noble and Welsh. Recall

$$U(G; \bar{x}, y) = \sum_{A \subseteq E} \Big(\prod_{i \geq 1} x_i^{s(i,A)} \Big) y^{|A|-r(A)},$$

where $s(i, A)$ counts components of size i in (V, A).

Proposition 3.4. *For a tree T,*

$$\mathscr{L}_T(1, x_0, x_1, \dots) = U(T; x_1, x_2, \dots; 0).$$

Proof. For a tree, $|A| - r(A) = 0$ for all $A \subseteq E(T)$, so

$$U(T; \bar{x}, 0) = \sum_{A \subseteq E(T)} \prod_{C \in c(A)} x_{|V(C)|}.$$

Comparing with the formula for $\mathscr{L}_T$ gives the result after the index shift $x_i \mapsto x_{i+1}$. □

Thus, on trees, the loopy polynomial coincides with a specialization of the U-polynomial.

3.4 Relation to Stanley's chromatic symmetric function

Let X_G denote Stanley's chromatic symmetric function. It admits the power-sum expansion

$$X_G = \sum_{A \subseteq E(G)} (-1)^{|A|} \prod_{C \in c(A)} p_{|V(C)|}.$$

Equivalently, fixing an ordering of the edges and applying the broken-circuit cancellation, one obtains

$$X_G = \sum_{F \in \mathrm{NBC}(G)} (-1)^{|F|} \prod_{T \in c(F)} p_{|V(T)|},$$

where the sum is over no-broken-circuit spanning forests.

Trees. If G is a tree, then every spanning forest has zero external activity, and $e(T) = |V(T)| - 1$ for each component T. Therefore the loopy polynomial recovers X_G:

$$X_G = (-1)^{|E(G)|} \mathscr{L}_G\big(-1,\ x_i = p_{i+1}\big).$$

General graphs. For graphs with cycles, this direct specialization fails. The loopy polynomial uses the index

$$e(T) + \varepsilon(T) = |V(T)| - 1 + \varepsilon(T),$$

which mixes the component size with the external activity. As a result, $\mathscr{L}_G$ is not equal to X_G under any simple substitution.

An activity refinement. To separate these statistics, define the activity-refined loopy polynomial

$$\widehat{\mathscr{L}_G}(t, \mathbf{y}) = \sum_{F \in \mathscr{F}(G)} t^{|E(G)| - |F| - \varepsilon(F)} \prod_{T \in c(F)} y_{|V(T)|, \varepsilon(T)}.$$

The original loopy polynomial is recovered via

$$y_{a,b} \mapsto x_{a-1+b}.$$

Theorem 3.5. *For every graph G,*

$$X_G = (-1)^{|E(G)|} \widehat{\mathscr{L}_G}\big(-1,\ y_{i,0} = p_i,\ y_{i,j} = 0 \text{ for } j > 0\big).$$

Proof. Under the specialization $y_{i,j} = 0$ for $j > 0$, only spanning forests with $\varepsilon(F) = 0$ contribute. By the broken-circuit characterization, these are precisely the no-broken-circuit forests. The formula then reduces to the NBC expansion of X_G. $\square$

Comparison of invariants. The refined invariant $\widehat{\mathscr{L}_G}$ determines the original loopy polynomial, but not conversely at the level of formal polynomials, since the map

$$y_{a,b} \mapsto x_{a-1+b}$$

identifies different pairs (a, b).

However, computations suggest that this loss of information may not occur on actual graph invariants.

Conjecture 3.6. *For simple graphs G and H,*

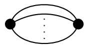
$$\mathscr{L}_G(t,\mathbf{x}) = \mathscr{L}_H(t,\mathbf{x}) \quad \Longleftrightarrow \quad X_G = X_H.$$

Equivalently, the loopy polynomial and the chromatic symmetric function induce the same equivalence relation on simple graphs.

This conjecture has been verified computationally for all simple graphs up to 16 vertices.

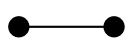
4 Examples

Let us provide examples of computations of the loopy polynomial using both the recursive definition (1.2) and formula (2.1).

(1) For the graph $G_m =$  with two vertices and m parallel edges, using recursion (1.2) we obtain

$$\mathscr{L}_{G_m} = \mathscr{L}_{L_m} + t\mathscr{L}_{G_{m-1}} = \ldots = x_m + tx_{m-1} + \ldots + t^{m-1}x_1 + t^m x_0^2.$$

It is easy to see that the subgraph formula (2.1) gives the same result. Indeed, in this case we have $\phi(G_m) = 1$ and $\mathscr{F}(G_m)$ consists of $m+1$ spanning forests, one with two components and no edges, contributing the term $t^m x_0^2$, and m spanning trees $T_1, \ldots, T_m$, one for each of its edges, with $\varepsilon(T_i) = |E| - i$.

In particular, for the two-vertex path graph $P_2 =$  , this gives

$$\mathscr{L}_{P_2} = x_1 + tx_0^2.$$

(2) More generally, for the sequence $\{P_m\}$ of path graphs, we get the following result.

Proposition 4.1. *For the path graph P_m with m vertices, we have*

$$\mathscr{L}_{P_m}(t,\mathbf{x}) = \sum_{\mathbf{a}\in\Lambda_m} \binom{|\mathbf{a}|}{\mathbf{a}} t^{|\mathbf{a}|-1}\mathbf{x}^{\mathbf{a}}, \tag{4.1}$$

where

$$\Lambda_m = \{\mathbf{a} = (a_0, a_1, \ldots, a_m) \in \mathbb{Z}_{\geq 0}^{m+1} \, : \, \sum_i a_i(i+1) = m\}$$

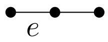
and for a tuple $\mathbf{a}$*, we denote*

$$\mathbf{a} := \sum_i a_i, \quad \mathbf{x}^{\mathbf{a}} := \prod_i x_i^{a_i}, \ \textit{and} \ \binom{|\mathbf{a}|}{\mathbf{a}} := \binom{|\mathbf{a}|}{a_0, \ldots, a_m} = \frac{|\mathbf{a}|!}{\prod_i a_i!}.$$

Proof. Since P_m is a tree, every subset of edges $F \subset E$ gives a spanning forest and the external activity of every subtree $T \subset E$ is 0. Therefore, using (2.1), we have that

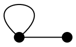
$$\mathscr{L}_{P_m}(t, \mathbf{x}) = \sum_{[k_1,\ldots,k_r]} t^{r-1} x_{k_1-1} \cdots x_{k_r-1},$$

where the sum is taken over all 2^{m-1} compositions of m, i.e. ordered collections $[k_1, \ldots, k_r]$ of positive integers whose sum equals m. Observe that for $i = 1, \ldots, m$, the number of compositions of m containing a_i copies of number i is equal to $\binom{|\mathbf{a}|}{\mathbf{a}}$ which implies formula (4.1). □

In particular, for the path graph with three vertices $P_3 =$ 

, we have

$$\mathscr{L}_{P_3} = \mathscr{L}_{P_3/e} + t x_0 \mathscr{L}_{P_2} = x_2 + t x_1 x_0 + t x_0 (x_1 + t x_0^2) = x_2 + 2 t x_1 x_0 + t^2 x_0^2,$$

where $P_3/e =$  is the graph P_2 with an added loop at one vertex. It is easy to see that the formula (2.1) gives the same result.

(3) Using similar, but slightly more involved argument we can find the loopy polynomials of circle graphs.

Proposition 4.2. *For the cycle graph* C_m *with* m *vertices, we have*

$$\mathscr{L}_{C_m} = x_m + (m-1) t\, x_{m-1} + \sum_{\mathbf{a} \in \Lambda'_m} \gamma_{\mathbf{a}}\, t^{|\mathbf{a}|} \mathbf{x}^{\mathbf{a}}, \tag{4.2}$$

where

$$\Lambda'_m = \Big\{ \mathbf{a} = (a_0, \ldots, a_{m-2}) \in \mathbb{Z}_{\geq 0}^{m-1} \ \Big| \ \sum_{i=0}^{m-2} (i+1) a_i = m, \quad |\mathbf{a}| \geq 2 \Big\},$$

$$|\mathbf{a}| := \sum_{i=0}^{m-2} a_i, \qquad \mathbf{x}^{\mathbf{a}} := \prod_{i=0}^{m-2} x_i^{a_i},$$

and

$$\gamma_{\mathbf{a}} = \frac{m}{|\mathbf{a}|}\binom{|\mathbf{a}|}{a_0, \dots, a_{m-2}}.$$

Equivalently,

$$\gamma_{\mathbf{a}} = \sum_{j\colon a_j \geq 1} (j+1)\binom{|\mathbf{a}|-1}{a_0, \dots, a_j - 1, \dots, a_{m-2}}.$$

Proof. Let the edges of C_m be linearly ordered around the cycle. By Theorem 2.3,

$$\mathscr{L}_{C_m}(t, \mathbf{x}) = \sum_{F \in \mathscr{F}(C_m)} t^{|E(C_m)| - e(F) - \sum_{T \in c(F)} \varepsilon(T)} \prod_{T \in c(F)} x_{e(T)+\varepsilon(T)}.$$

Since $r(C_m) = m-1$, every spanning forest of C_m is obtained by deleting a nonempty set of edges from the cycle.

If exactly one edge is deleted, then F is a spanning tree. There are m such trees. Exactly one of them has external activity 1, namely the tree obtained by deleting the maximal edge in the chosen order; its contribution is x_m. The remaining $m-1$ spanning trees have external activity 0, and each contributes $t\,x_{m-1}$. Hence the total contribution of spanning trees is

$$x_m + (m-1)t\,x_{m-1}.$$

Now suppose that F is obtained by deleting $k \geq 2$ edges. Then F has exactly k connected components, each of which is a path. Moreover, no deleted edge is externally active: indeed, each deleted edge joins two different components of F, so adding it does not create a cycle. Therefore all activity terms vanish, and each component contributes only according to its number of edges.

Write the numbers of vertices in the k path components as

$$b_1, \dots, b_k \geq 1, \qquad b_1 + \cdots + b_k = m.$$

Then the contribution of F is

$$t^k x_{b_1 - 1} \cdots x_{b_k - 1}.$$

Thus forests with $k \geq 2$ components are naturally encoded by cyclic compositions of m.

Fix now a multiplicity vector

$$\mathbf{a} = (a_0, \dots, a_{m-2}) \in \Lambda'_m,$$

where a_i is the number of components with $i+1$ vertices. Then $|\mathbf{a}| = k$ is the number of components, and

$$\sum_{i=0}^{m-2} (i+1)a_i = m.$$

All forests of this type contribute the same monomial

$$t^{|\mathbf{a}|}\mathbf{x}^{\mathbf{a}}.$$

It remains to count how many such forests there are.

Choose first a starting deleted edge on the cycle; there are m choices. Then arrange the multiset of component sizes, containing a_i copies of $i+1$, in linear order around the cycle. The number of such linear orders is

$$\binom{|\mathbf{a}|}{a_0, \dots, a_{m-2}}.$$

This gives

$$m\binom{|\mathbf{a}|}{a_0, \dots, a_{m-2}}$$

ordered descriptions. However, each forest is counted exactly $|\mathbf{a}|$ times, since any one of its $|\mathbf{a}|$ deleted edges could have been chosen as the starting deleted edge. Therefore the number of forests of type $\mathbf{a}$ is

$$\gamma_{\mathbf{a}} = \frac{m}{|\mathbf{a}|}\binom{|\mathbf{a}|}{a_0, \dots, a_{m-2}}.$$

Summing over all $\mathbf{a} \in \Lambda'_m$ yields (4.2). The equivalent expression for $\gamma_{\mathbf{a}}$ follows from the identity

$$\frac{m}{|\mathbf{a}|}\binom{|\mathbf{a}|}{a_0, \dots, a_{m-2}} = \sum_{j:\, a_j \geq 1} (j+1)\binom{|\mathbf{a}| - 1}{a_0, \dots, a_j - 1, \dots, a_{m-2}},$$

using $m = \sum_{j=0}^{m-2}(j+1)a_j$. □

For example, for C_3 we have $\Lambda'_3 = \{(1,1)\}$ and

$$\gamma_{(1,1)} = \frac{3}{2}\binom{2}{1,1} = 3.$$

Hence

$$\mathscr{L}_{C_3} = x_3 + 2tx_2 + 3t^2x_0x_1.$$

which matches what we found earlier using deletion-contraction.

For the four cycle C_4 we have

$\Lambda'_4 = \{(a_0, a_1, a_2) \,:\, a_0 \leq 3, a_0{+}2a_1{+}3a_2 = 4)\} = \{(4,0,0), (2,1,0), (1,0,1), (0,2,0)\}.$

and $\gamma_{2,1,0} = \binom{3}{2,1} + 1\cdot\binom{2}{2,0} = 4$, $\gamma_{1,0,1} = \binom{2}{1,1} + 2\cdot\binom{1}{1,0} = 4$, $\gamma_{0,2,0} = \binom{2}{2} + 1\cdot\binom{1}{1,0} = 2$, which gives

$$\mathscr{L}_{C_4} = x_4 + 3tx_3 + t^4x_0^4 + 4t^3x_0^2x_1 + 4t^2x_0x_2 + 2t^2x_1^2.$$

(4) Let us find the loopy polynomial of a star graph S_m with m edges (i.e. the complete bipartite graph $K_{1,m}$). More generally, let $S_{m,\ell}$ be the graph obtained by adding ℓ loops at the center of S_m. Then the deletion-contraction relation (1.2) gives a Pascal-like recursion

$$\mathscr{L}_{S_{m+1,\ell}} = \mathscr{L}_{S_{m,\ell+1}} + tx_0\mathscr{L}_{S_{m,\ell}}$$

with the initial conditions $\mathscr{L}_{S_{0,\ell}} = x_\ell$. Therefore we have

$$\mathscr{L}_{S_{m,\ell}} = \sum_{i=0}^{m}\binom{m}{i}(tx_0)^i x_{\ell+i}\ ,$$

and, in particular,

$$\mathscr{L}_{S_m} = \sum_{i=0}^{m}\binom{m}{i}(tx_0)^i x_i\ .$$

5 Polyhedral and parking-cell interpretation

In this section we explain the polyhedral meaning of the specialization

$$t = q, \qquad x_i = 1 + q + \cdots + q^i.$$

Let

$$P_G = \left\{ a \in \mathbb{R}^V_{\geq 0} : \sum_{v \in S} a_v \leq \kappa_G(S) \text{ for all } S \subseteq V \right\},$$

where $\kappa_G(S)$ is the number of edges of G incident to S. By [KNSV], the lattice points of P_G are the partial score vectors, and therefore the Hilbert series of the external bizonotopal algebra is

$$\mathrm{Hilb}_G(q) = \sum_{a \in P_G \cap \mathbb{Z}^V} q^{|a|}.$$

We prove below that this Hilbert series is obtained from the loopy polynomial by substituting $x_i = 1 + q + \cdots + q^i$ and $t = q$. More precisely, the lattice points of P_G admit a recursive deletion–contraction decomposition whose cells are indexed by spanning forests.

Throughout this section, if $e = \{u, v\}$ is a non-loop edge, then G/e denotes loopy contraction: the endpoints u and v are identified, and the edge e itself becomes a loop at the new vertex.

5.1 The recursive decomposition

Fix an orientation of every edge of G. The orientation is used only to choose which endpoint receives the degree shift in the deletion branch.

Lemma 5.1 (Recursive lattice-point decomposition)**.** *Let $e = \{u, v\}$ be a non-loop edge, oriented from u to v. Then there is a canonical degree-preserving decomposition*

$$P_G \cap \mathbb{Z}^V = A_e \sqcup B_e$$

such that

$$\sum_{a \in A_e} q^{|a|} = \mathrm{Hilb}_{G/e}(q), \qquad \sum_{a \in B_e} q^{|a|} = q \ \mathrm{Hilb}_{G-e}(q).$$

Proof. First define the deletion part. Let

$$\iota : P_{G-e} \cap \mathbb{Z}^V \longrightarrow P_G \cap \mathbb{Z}^V, \qquad \iota(b) = b + \mathbf{e}_u.$$

This map is well defined. Indeed, deleting e lowers $\kappa_G(S)$ by one exactly when S is incident to e, while adding $\mathbf{e}_u$ increases $b(S)$ only when $u \in S$. Thus every inequality defining P_G is satisfied by $b + \mathbf{e}_u$. Set

$$B_e = \iota(P_{G-e} \cap \mathbb{Z}^V).$$

Then

$$\sum_{a \in B_e} q^{|a|} = q \ \mathrm{Hilb}_{G-e}(q).$$

It remains to identify the complement with $P_{G/e}$. Let w be the vertex of G/e obtained by identifying u and v. Define

$$\pi : P_G \cap \mathbb{Z}^V \longrightarrow P_{G/e} \cap \mathbb{Z}^{V/e}$$

by

$$\pi(a)_w = a_u + a_v, \qquad \pi(a)_z = a_z \quad (z \neq w).$$

If $R \subseteq V/e$ and $\widetilde{R} \subseteq V$ is its inverse image, then loopy contraction gives

$$\kappa_{G/e}(R) = \kappa_G(\widetilde{R}).$$

Hence $a \in P_G$ implies $\pi(a) \in P_{G/e}$.

We now construct a section of π. Take $c \in P_{G/e} \cap \mathbb{Z}^{V/e}$ and write $N = c_w$. A lift of c to $\mathbb{Z}^V$ is determined by an integer r via

$$a_v = r, \qquad a_u = N - r,$$

with all other coordinates inherited from c. The inequalities defining P_G impose lower and upper bounds on r, so the feasible lifts form an integer interval. This interval is nonempty by the standard polymatroid contraction property applied to the submodular function κ_G.

Choose the lift for which $r = a_v$ is maximal, and denote it by $s(c)$. We claim that $s(c)$ is the unique lift of c which does not belong to B_e. Indeed, a feasible lift a belongs to B_e if and only if $a - \mathbf{e}_u \in P_{G-e}$. For a feasible lift this is equivalent to saying that $a_u > 0$ and that all inequalities involving sets containing v but not u remain valid after increasing a_v by one. Equivalently,

$a_v + 1$ is still a feasible value of the splitting parameter r. Thus a lift lies in B_e if and only if it is not the maximal feasible lift. Therefore the maximal lift $s(c)$ lies in the complement of B_e, and every other lift lies in B_e.

Consequently s gives a bijection

$$P_{G/e} \cap \mathbb{Z}^{V/e} \longrightarrow A_e := (P_G \cap \mathbb{Z}^V) \setminus B_e.$$

The bijection preserves total degree because

$$|s(c)| = |c|.$$

Therefore

$$\sum_{a \in A_e} q^{|a|} = \mathrm{Hilb}_{G/e}(q),$$

and the lemma follows. □

5.2 Canonical parking cells

The preceding lemma can be iterated along the same edge order used in the loopy deletion–contraction definition of $\mathscr{L}_G$. At each non-loop edge, the contraction branch is the maximal lift constructed above, and the deletion branch is the image of P_{G-e} under $b \mapsto b + \mathbf{e}_u$. Thus every lattice point follows a unique path in the deletion–contraction tree.

Theorem 5.2 (Parking-cell decomposition)**.** *Fix an ordering and an orientation of the edges of G. Then*

$$P_G \cap \mathbb{Z}^V = \bigsqcup_{F \in \mathscr{F}(G)} C_F$$

admits a canonical recursive partition indexed by spanning forests. For every spanning forest F, one has

$$\sum_{a \in C_F} q^{|a|} = q^{|E(G)| - e(F) - \sum_{T \in c(F)} \varepsilon(T)} \prod_{T \in c(F)} \left(1 + q + \cdots + q^{e(T) + \varepsilon(T)}\right).$$

Equivalently, C_F is a product of chains as a graded set:

$$C_F \cong \prod_{T \in c(F)} \{0, 1, \ldots, e(T) + \varepsilon(T)\},$$

with total grading shifted by

$$|E(G)| - e(F) - \sum_{T \in c(F)} \varepsilon(T).$$

Proof. Apply Lemma 5.1 to the largest non-loop edge e. The contraction part is identified degree-preservingly with $P_{G/e} \cap \mathbb{Z}^{V/e}$, while the deletion part is identified with $P_{G-e} \cap \mathbb{Z}^V$ with degree shifted by one. Iterating this procedure produces a canonical decomposition of $P_G \cap \mathbb{Z}^V$ into cells indexed by the leaves of the loopy deletion–contraction tree.

The leaves are graphs with only loops. If such a leaf has one vertex and m loops, its polytope is the chain

$$\{0, 1, \dots, m\},$$

with rank-generating function $1 + q + \cdots + q^m$. If the leaf has several vertices, the corresponding polytope is the product of these chains, one for each component.

It remains only to identify the leaf reached by a forest F. The edges in F are precisely those sent to the contraction branch. The edges not in F are sent to the deletion branch unless, at the moment they are processed, they have already become loops inside a contracted component. These are exactly the externally active edges with respect to the chosen edge order. Thus, for a component T of F, the number of loops present at the corresponding leaf is

$$e(T) + \varepsilon(T),$$

namely the contracted forest edges inside T together with the externally active edges assigned to T. The remaining deleted edges contribute the degree shift

$$|E(G)| - e(F) - \sum_{T \in c(F)} \varepsilon(T).$$

Therefore the cell indexed by F has the claimed rank-generating function. □

5.3 Hilbert-series specialization

As an immediate consequence, the loopy polynomial refines the lattice-point enumerator of P_G.

Corollary 5.3. *For every graph G,*

$$\sum_{a \in P_G \cap \mathbb{Z}^V} q^{|a|} = \mathscr{L}_G(q, 1, 1 + q, 1 + q + q^2, \dots).$$

Proof. Summing the identity of Theorem 5.2 over all spanning forests F gives

$$\sum_{a \in P_G \cap \mathbb{Z}^V} q^{|a|} = \sum_{F \in \mathscr{F}(G)} q^{|E(G)| - e(F) - \sum_T \varepsilon(T)} \prod_{T \in c(F)} \left(1 + q + \cdots + q^{e(T)+\varepsilon(T)}\right).$$

By Theorem 2.3, the right-hand side is precisely $\mathscr{L}_G(q, 1, 1+q, 1+q+q^2, \dots)$. ☐

Thus the variables x_i record the chain factors in the parking cells, while the variable t records the number of genuinely deleted non-active edges.

6 Outlook

1. Characterize the information about G encoded by $\mathscr{L}_G$.
2. Find a universality property for $\mathscr{L}_G$.
3. Relate $\mathscr{L}_G$ to strong U-polynomials.
4. Investigate planar duality.
5. Explain the 4-term relation structurally.

Acknowledgements. The third author gratefully acknowledges the hospitality of the Beijing Institute for Mathematical Sciences and Applications in summer 2024, when a substantial part of this project was carried out.

References

[EM] J. Ellis–Monaghan and I. Moffatt (eds.), Handbook of the Tutte Polynomial and Related Topics, Chapman and Hall/CRC, 2022. 2

[KNSV] A. Kirillov, G. Nenashev, B. Shapiro, A. Vaintrob, Bizonotopal graphical algebras, arXiv:2407.19431, Algebraic Combinatorics, to appear. 5, 11, 12, 20

[Tu] W. T. Tutte. A contribution to the theory of chromatic polynomials. Canadian J. Math., 6:80–91, 1954. 4