

VOLUME INEQUALITIES FOR FLOW POLYTOPES OF FULL DIRECTED ACYCLIC GRAPHS

Benjamin Braun^{*1} and James Ford McElroy^{†2}

^{1,2}*Department of Mathematics, University of Kentucky, Lexington Kentucky, U.S.A.*
benjamin.braun@uky.edu , jfmcelroy@uky.edu

Submitted: May 29, 2024; Accepted: Mar 7, 2025; Published: Jul 15, 2025

© The authors. Released under the CC BY license (International 4.0).

Abstract. Given a finite directed acyclic graph, the space of non-negative unit flows is a lattice polytope called the flow polytope of the graph. We consider the volumes of flow polytopes for directed acyclic graphs on $n + 1$ vertices with a fixed degree sequence, with a focus on graphs having in- and out-degree two on every internal vertex. When the out-degree of the source is three and the number of vertices is fixed, we prove that there is an interchange operation on the edge set of these graphs that induces a partial order on the graphs isomorphic to a Boolean algebra. Further, we prove that as we move up through this partial order, the volumes of the corresponding flow polytopes weakly decrease. Finally, we show that each such graph is strongly planar and we provide an alternative interpretation of our results in the context of linear extensions for posets that are bipartite non-crossing trees.

Keywords. Flow polytopes, Volumes, Posets, Linear Extensions, Degree Sequence

Mathematics Subject Classifications. 52B20, 05C20, 05C21, 52B05

1. Introduction

Given a finite directed acyclic graph (DAG) G , the space of non-negative flows on G of strength one is a lattice polytope denoted by $\mathcal{F}_1(G)$. While flow polytopes for DAGs can be defined for arbitrary netflow vectors, strength one flows that conserve flow at inner vertices have remained an important and central object of study in geometric and algebraic combinatorics [BV08, BGDH⁺18, BGDH⁺19, DKK12, DHMR23, GDHMY23, KMS21, Lid84, LMSD19, LMM19, MM19, MMS19, MS21, BBB⁺24, vBGDMCY21]. A central problem is to determine the (normalized) volume of $\mathcal{F}_1(G)$, and to relate volumes of flow polytopes to structural qualities of the DAGs in question.

^{*}Partially supported by the US National Science Foundation award DMS-1953785.

[†]Partially supported by the US National Science Foundation award DMS-1953785.

In this work, we consider the following general problem. Consider the family of DAGs on a fixed number of vertices with fixed in- and out-degree sequences. Which DAGs in this family maximize or minimize flow polytope volume? What relationships, if any, are there between the DAGs in this family and their flow polytope volumes? Note that if a DAG has a vertex with either in-degree or out-degree equal to 1, then contracting the single corresponding edge yields a DAG with an integrally equivalent flow polytope. Thus, the first non-trivial case to consider is when the in- and out-degree sequence for inner (i.e., non-sink and non-source) vertices is $(2, \dots, 2)$. When the degree of the source and sink are both 2, this family consists of only one DAG on $n + 1$ vertices, with corresponding flow polytope given by a unit cube.

In this paper, we consider the DAGs having source vertex and sink vertex both of degree 3, where all other in- and out-degrees are 2. We show that DAGs satisfying this condition are in bijection with binary strings. Further, crossing nested edges in such a DAG corresponds to changing 0s and 1s in the associated binary string. The main result of this work is to show that the operation of crossing nested edges weakly decreases the volume of the associated flow polytopes; thus, there is a Boolean algebra structure on the set of DAGs on $n + 1$ vertices with this degree condition where the volumes of flow polytopes are reverse-ordered from the partial order on the Boolean algebra.

Further, the DAGs we consider are planar. Due to Mészáros, Morales, and Striker [MMS19], any result about volumes of flow polytopes arising from DAGs with strongly planar embeddings can be translated to a result about linear extensions of posets arising from the duals of the DAGs. Thus, our results regarding volume inequalities for flow polytopes translate into inequalities for linear extensions of posets.

The remainder of this paper is structured as follows. In Section 2, we provide background regarding flow polytopes and their volumes. We prove in Section 3 results regarding Boolean algebra poset structures on our families of DAGs of interest. In Section 4, we introduce the tool of flow decomposition trees in order to prove Theorem 4.20, which states that the values of the volumes of flow polytopes for DAGs under the Boolean partial order obtained in Corollary 3.20 respect the dual partial order. Finally, in Section 5 we recast our results about flow polytopes in the context of linear extensions for posets.

2. Background

Let $G = (V, E)$ be a directed acyclic graph (DAG) on vertices $V = [n + 1]$, such that every edge (i, j) in G satisfies that $i < j$, and such that G has a single source 1 and a single sink $n + 1$, where we call any vertex with no incoming edges a *source* and any vertex with no outgoing edges a *sink*. We use the notation $\mathbf{a} = (a_1, \dots, a_n, -\sum_i a_i) \in \mathbb{Z}^{n+1}$ for a netflow vector, and we denote by M_G the matrix whose columns are given by the positive type A_n roots $\alpha(i, j) = e_i - e_j$ for $(i, j) \in E$.

Definition 2.1. Given a vertex j in a DAG G , define the *in-degree* of j , denoted ind_j , to be the number of edges of the form (i, j) in G . That is, the number of incoming edges to j . Similarly, define the *out-degree* of j , denoted outd_j , to be the number of edges of the form (j, k) in G . That is, the number of outgoing edges from j .

Definition 2.2. Given a DAG G and netflow vector \mathbf{a} , we define the *flow polytope* of G to be

$$\mathcal{F}_G(\mathbf{a}) := \{f_G \in \mathbb{R}_{\geq 0}^E \mid M_G f_G = \mathbf{a}\}.$$

We think of f_G as a flow on G , that is, an assignment of nonnegative flow values to each edge satisfying the netflow requirements at each vertex given by the netflow vector \mathbf{a} . In particular, this means that

$$\sum_{\text{incoming edges at } i} f_G(i, j) - \sum_{\text{outgoing edges at } i} f_G(h, i) = a_i$$

for all $i \in V$. In this case, we say the flow *respects* the netflow values a_i .

An important choice of netflow vector is $\mathbf{a} = (1, 0, \dots, 0, -1)$, and we use the special symbol

$$\mathcal{F}_1(G) := \mathcal{F}_G(1, 0, \dots, 0, -1)$$

to denote the corresponding flow polytope. Flow polytopes are lattice polytopes, in that their vertices are always elements of the integer lattice \mathbb{Z}^E .

Computing the volume of an arbitrary lattice polytope is a challenging problem. There is a closed formula for flow polytope volumes, using Kostant partition functions, due to Lidskii [Lid84] and Baldoni and Vergne [BV08]. The Kostant partition function K_G evaluated at the vector $\mathbf{a} \in \mathbb{Z}^{n+1}$ is defined as

$$K_G(\mathbf{a}) = \#\left\{ (f(e))_{e \in E(G)} \mid \sum_{e \in E(G)} f(e)\alpha(e) = \mathbf{a}, \text{ and } f(e) \in \mathbb{Z}_{\geq 0} \right\},$$

where $\{\{\alpha(e)\}\}_{e \in E(G)}$ is the multiset of positive type A_n roots corresponding with the multiset $E(G)$. That is, the Kostant partition function counts the number of ways to write the vector \mathbf{a} as a \mathbb{N} -linear combination of vectors $\alpha(e)$ for $e \in E(G)$. The Lidskii formula states that the (normalized) volume of $\mathcal{F}_G(\mathbf{a})$ is

$$\sum \binom{|E| - |V| + 1}{m_1 - 1, \dots, m_n - 1} a_1^{m_1 - 1} \dots a_n^{m_n - 1} K_G(m_1 - \text{outd}_1, \dots, m_n - \text{outd}_n, 0),$$

where we sum over weak compositions $\sum m_i = |E|$ with $(m_1, \dots, m_n) \geq (\text{outd}_1, \dots, \text{outd}_n)$ using dominance order. Note that the Kostant partition function is also difficult to compute in general, and thus the efficacy of this formula depends on the DAG under consideration. In the case of the netflow vector $\mathbf{a} = (1, 0, \dots, 0, -1)$, the Lidskii formula reduces to

$$\text{vol } \mathcal{F}_1(G) = K_G(|E| - |V| + 2 - \text{outd}_1, 1 - \text{outd}_2, \dots, 1 - \text{outd}_n, 0)$$

which is equivalent to the following, via flow reversal [MM19],

$$\text{vol } \mathcal{F}_1(G) = K_G(0, \text{ind}_2 - 1, \dots, \text{ind}_n - 1, |V| - |E| - 2 + \text{ind}_{n+1}). \tag{2.1}$$

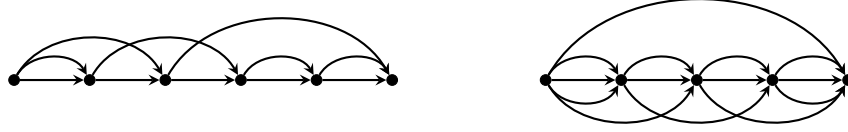
Geometrically, this means that the volume of $\mathcal{F}_1(G)$ is equal to the number of lattice points in the related flow polytope $\mathcal{F}_G(0, \text{ind}_2 - 1, \dots, \text{ind}_n - 1, |V| - |E| - 2 + \text{ind}_{n+1})$ or equivalently, the integer flows on G with netflow vector $(0, \text{ind}_2 - 1, \dots, \text{ind}_n - 1, |V| - |E| - 2 + \text{ind}_{n+1})$. It is worth

noting that an alternative method exists to compute flow polytope volumes for DAGs containing a directed Hamiltonian path, one which relies on combinatorics of G -cyclic orders [GDHMY23, Theorem 4.8].

A family of DAGs of particular interest is that of full DAGs and a generalization which we call r -full DAGs. We are interested in r -full DAGs because von Bell, Braun, Bruegge, Hanely, Peterson, Serhiyenko and Yip proved [BBB⁺24, Proposition 6.15] that the flow polytope $\mathcal{F}_1(G)$ for an r -full DAG G is Gorenstein.

Definition 2.3. A DAG G is r -full if every inner vertex has in- and out-degree r . In the case where $r = 2$, we say that G is full.

Example 2.4. Pictured below is a 2-full (i.e., full) DAG and a 3-full DAG, respectively.



Definition 2.5. Let $\mathcal{F}_{n,k}$ be the set of full DAGs on vertices $[n + 1]$ with $\text{outd}_1 = k$. That is, DAGs with out-degree sequence $(k, 2, 2, \dots, 2, 0)$ and in-degree sequence $(0, 2, 2, \dots, 2, k)$.

Note that every full DAG with a unique source and sink is an element of $\mathcal{F}_{n,k}$ for some k .

Proposition 2.6. A DAG $G \in \mathcal{F}_{n,k}$ has $n + 1$ vertices and $2n + k - 2$ edges.

Proof. The vertex count is true by assumption. The edge count comes from the fact that the number of edges in a DAG is equal to the sum of the out-degrees (or equivalently, the sum of the in-degrees). So

$$|E(G)| = \sum_{i \in V(G)} \text{outd}_i = k + 2(n - 1) = 2n + k - 2 \quad \square$$

We next define the length $n + 1$ vector

$$\mathbf{w}_n := (0, 1, \dots, 1, -(n - 1)),$$

which plays the following special role in computing volumes of flow polytopes for full DAGs.

Proposition 2.7. Suppose $G \in \mathcal{F}_{n,3}$. Then

$$\text{vol } \mathcal{F}_1(G) = K_G(\mathbf{w}_n).$$

In other words, computing the volume of the flow polytope $\mathcal{F}_1(G)$ is equivalent to counting lattice points in the flow polytope $\mathcal{F}_G(\mathbf{w}_n)$.

Proof. This follows directly from applying Definition 2.5 for $k = 3$ and the edge count from Proposition 2.6 to Equation (2.1). \square

Following Karrer and Newman [KN09], we use the following method to construct DAGs of a fixed degree sequence. For each vertex of a DAG G with a fixed degree sequence, we imagine “stubs” attached to the vertex, with as many outgoing stubs as the out-degree and as many incoming stubs as the in-degree. To create a DAG we match outgoing stubs with incoming stubs, where we only create an edge (i, j) if $i < j$. This construction method will be used in Section 3.

3. Poset Structure

In this section, we show first that full DAGs on $n + 1$ vertices with $\text{outd}_1 = 3$ are in bijection with binary strings of length $n - 2$. Then we show that the Boolean partial order induced by this bijection agrees with an edge operation on $\mathcal{F}_{n,3}$.

Lemma 3.1. *Every DAG $G \in \mathcal{F}_{n,3}$ has the following edges:*

$$\{(i, i + 1) \mid i \in [n]\} \cup \{(1, 2), (n, n + 1)\} \subset E(G).$$

Proof. Let $G \in \mathcal{F}_{n,3}$. Note that $\text{ind}_2 = 2$. Since $(i, j) \in E(G)$ implies $i < j$, all edges going in to 2 must start at 1. Thus there must be two copies of the edge $(1, 2)$. A similar argument can be made for two copies of the edge $(n, n + 1)$ using the $\text{outd}_n = 2$.

Now consider a $2 \leq i \leq n - 1$. Since G is full, $\text{ind}_i = 2$. Once again, both edges must begin at a vertex smaller than i . But this is also true for every vertex *before* i . If we look at the available pool of out-degrees at non-consecutive vertices with which to connect the two in-stubs at i , we have

$$\sum_{j=1}^{i-2} \text{outd}_j - \sum_{j=1}^{i-1} \text{ind}_j = (3 + 2(i - 3)) - 2(i - 2) = 1.$$

So at most one of the in-stubs at i can be connected to an out-stub *before* $i - 1$, forcing at least one edge $(i - 1, i)$. □

Definition 3.2. Let G be a DAG with vertices $[n + 1]$. If $(i, i + 1) \in E(G)$ for all $i \in [n]$, we say G has a *spine*. Further, if G has a spine, we call the edge set $\{(i, i + 1) \mid i \in [n]\}$ the spine. Throughout this work, we label the spine edges as $e_i = (i, i + 1)$ to distinguish them from other (multi)edges between consecutively-labeled vertices.

Corollary 3.3. *Every DAG in $\mathcal{F}_{n,3}$ has a spine.*

Theorem 3.4. *The set $\mathcal{F}_{n,3}$ is in bijection with binary strings of length $n - 2$, i.e., the elements of $(\mathbb{Z}/2\mathbb{Z})^{n-2}$.*

Proof. We construct a bijection directly.

First we construct a map $\lambda : (\mathbb{Z}/2\mathbb{Z})^{n-2} \rightarrow \mathcal{F}_{n,3}$. Let $\beta = b_1 b_2 \cdots b_{n-2} \in (\mathbb{Z}/2\mathbb{Z})^{n-2}$. Let J_β be the list of indices in which β contains a 1. Then $J_\beta = \{i \in [n - 2] \mid b_i = 1\} =$

$\{j_1 < j_2 < \dots < j_r\}$, for some $0 \leq r \leq n - 2$. Now define $\lambda(\beta) = G_\beta$, where G_β is a DAG with vertices $V(G_\beta) = [n + 1]$ and edges

$$\begin{aligned} E(G_\beta) = & \{e_i \mid i \in [n]\} \cup \{(1, 2), (n, n + 1)\} \\ & \cup \{(l + 1, l + 2) \mid b_l = 0, \text{ for } 1 \leq l \leq n - 2\} \\ & \cup \{(1, j_1 + 2), (j_1 + 1, j_2 + 2), \dots, (j_{r-1} + 1, j_r + 2), (j_r + 1, n + 1)\}. \end{aligned}$$

These three subsets of $E(G_\beta)$ correspond with the following structures, respectively:

- (i) the edges described in Lemma 3.1;
- (ii) a second edge from vertex $l + 1$ to $l + 2$ for every 0 at index l in β ; and
- (iii) an edge connecting remaining consecutive pairs of out-stubs and in-stubs.

Next we show G_β is full with $\text{outd}_1 = 3$. By construction, G_β is a DAG with $V(G) = [n + 1]$. Our construction also ensures that

$$\text{outd}_1 = 2 + 0 + 1 = 3$$

and

$$\text{ind}_2 = 2 + 0 + 0 = 2.$$

Further, for $3 \leq i \leq n$,

$$\text{ind}_i = \begin{cases} 1 + 1 + 0, & b_{i-2} = 0 \\ 1 + 0 + 1, & b_{i-2} = 1 \end{cases} = 2$$

and, for $2 \leq i \leq n$,

$$\text{outd}_i = \begin{cases} 1 + 1 + 0, & b_{i-1} = 0 \\ 1 + 0 + 1, & b_{i-1} = 1 \end{cases} = 2.$$

Thus G_β has the required degree sequence to be in $\mathcal{F}_{n,3}$.

Now we show λ is injective. Suppose $\beta \neq \beta' \in (\mathbb{Z}/2\mathbb{Z})^{n-2}$. Then there exists an index $l \in [n - 2]$ for which $b_l \neq b'_l$. Without loss of generality we may assume $b_l = 0$ and $b'_l = 1$. Then according to our construction, we have

$$\begin{aligned} \{(l + 1, l + 2), (l + 1, l + 2)\} & \subset E(G_\beta) \text{ and} \\ \{(l + 1, l + 2), (l + 1, k)\} & \subset E(G_{\beta'}), \end{aligned}$$

where

$$l + 2 < k = \begin{cases} n + 1, & b'_l \text{ is the last 1 in } \beta' \\ l + m + 2, & b'_{l+m} = 1 \text{ is the next 1 in } \beta' \text{ for } m \geq 1 \end{cases}.$$

Since both G_β and $G_{\beta'}$ are full and $l + 1$ is an interior vertex, we know there are no other edges out of vertex $l + 1$ in $G_{\beta'}$ or G_β . Thus $G_\beta \neq G_{\beta'}$.

Finally we show that λ is surjective. Since λ is an injection of finite sets, it suffices to show that $|\mathcal{F}_{n,k}| \leq |(\mathbb{Z}/2\mathbb{Z})^{n-2}| = 2^{n-2}$. By Proposition 2.6, we have to decide how to allocate $2n + 1$ directed edges onto $n + 1$ vertices. By Lemma 3.1, $n + 2$ edges are forced by the spine and initial/final vertices. Thus we have

$$2n + 1 - (n + 2) = n - 1$$

edges remaining. By interpreting the remaining edges as connections between unused stubs on vertices [KN09], we can allocate an edge by pairing an out-going stub with an in-going stub, as long as the in-going stub is at a higher vertex.

Observe that once we have placed the edges from Lemma 3.1, and looking at vertex $i = 3$, there are 2 out-going stubs preceding 3 and 1 in-going stub at 3. Thus, there are two choices of incoming edges for vertex 3. Inductively, for $i = 4, \dots, n$, assume that we have made our choice of incoming edges for vertices up to $i - 1$. That means the in-going stub at i has 2 choices of preceding out-going stubs to connect to. Once we reach vertex $n + 1$, there is a unique choice of out-going stub for the in-going stub at $n + 1$ to connect to. In total, this gives 2^{n-2} possible ways to pair in- and out-going stubs to allocate edges. \square

Example 3.5. Bijection construction for $(\mathbb{Z}/2\mathbb{Z})^2$ to $\mathcal{F}_{4,3}$ shown in Figure 3.1. In the first column we begin with the edges described in Lemma 3.1. In the second column, we add a consecutive edge $(l + 1, l + 2)$ for each 0 at index l in β . In the final column, we add the remaining edges corresponding to the 1s in β and complete the missing in- and out-degrees at each vertex.

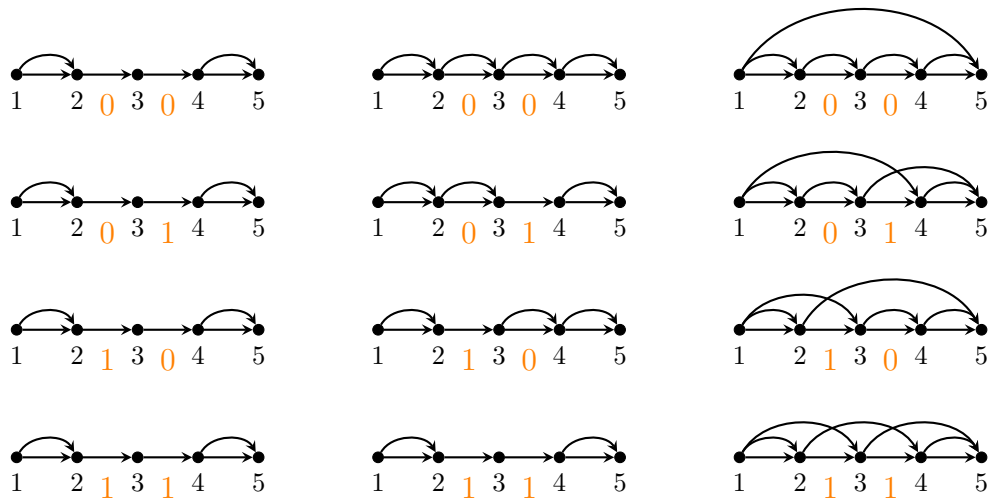


Figure 3.1: Bijection construction for $(\mathbb{Z}/2\mathbb{Z})^2$ to $\mathcal{F}_{4,3}$.

Corollary 3.6. *There are 2^{n-2} full DAGs on $n + 1$ vertices and $\text{out}_1 = 3$.*

Now that we have a bijection between full DAGs and binary strings, we show that the Boolean poset structure of the binary strings is represented in the DAGs by an edge operation.

Definition 3.7. Given a DAG G with a linear order on its vertices, and a pair of edges $(a, d), (b, c)$ satisfying $a < b < c < d$, we say such a pair of edges is *nested*.

Definition 3.8. Given a DAG G with a nested pair of edges $(a, d), (b, c)$, define the *interchange operation* to be the mapping $G \mapsto G'$, where G' is defined by vertices $V(G') = V(G)$ and edges $E(G') = E(G) \setminus \{(a, d), (b, c)\} \cup \{(a, c), (b, d)\}$.

Definition 3.9. Given a DAG G with a linear order on its vertices, and a pair of edges $(a, c), (b, d)$ satisfying $a < b < c < d$, we say such a pair of edges is *crossed* or *twisted* or *interchanged*.

Definition 3.10. Given a DAG G' with a crossed pair of edges $(a, c), (b, d)$, define the *reverse interchange operation* to be the mapping $G' \mapsto G$, where G is defined by vertices $V(G) = V(G')$ and edges $E(G) = E(G') \setminus \{(a, c), (b, d)\} \cup \{(a, d), (b, c)\}$.

Remark 3.11. What we refer to in this paper as the interchange operation is also known as a double edge swap in the contexts of graph sampling [FLNU18]. The specialization to directed graphs continues below.

Example 3.12. Considering Figure 3.1, the right-most DAG labeled by the string $[1, 0]$ has edge $(3, 4)$ nesting in edge $(2, 5)$. Edges $(1, 3)$ and $(2, 5)$ are crossed. Note that the DAGs labeled by strings $[1, 1]$ and $[1, 0]$ are related by the interchanging operations.

Lemma 3.13. $\mathcal{F}_{n,k}$ is closed under the interchange operation.

Proof. Let $G \in \mathcal{F}_{n,k}$. Suppose G has two edges $(a, d), (b, c) \in E(G)$ satisfying $a < b < c < d$. Let G' be the DAG obtained by performing an interchange on G . Then G' still has $n + 1$ vertices, and the edges incident to vertices other than a, b, c, d are unchanged, preserving the in- and out-degrees at these vertices. Thus we only need verify that the in- and out-degrees for a, b, c, d are also unchanged, which follows directly from the definition of the interchange operation. \square

Next, we prove that the interchange operation spans the entirety of $\mathcal{F}_{n,3}$. We first introduce a useful fact about DAGs in $\mathcal{F}_{n,3}$.

Lemma 3.14. Let $G \in \mathcal{F}_{n,3}$ and let i be an interior vertex of G . Then there is exactly one edge in $E(G)$ that passes vertex i . That is, there is a unique edge (a, d) so satisfying $a < i < d$.

Proof: From Theorem 3.4, we know that there exists a length $n - 2$ binary string β so that $G = G_\beta$ with

$$J_\beta = \{i \in [n - 2] \mid b_i = 1\} = \{j_1 < \dots < j_r\}.$$

For edge case convenience, define $j_0 := 0, j_{r+1} := n - 1$ and

$$\widehat{J}_\beta := \{j_0 = 0 < j_1 < \dots < j_r < j_{r+1} = n - 1\}.$$

Note that this ensures \widehat{J}_β is always nonempty, even if J_β is empty. Now let $m = \max\{l \in \widehat{J}_\beta \mid j_l < i - 1\}$. That is, j_m is the largest index in β where there is a 1 left of i .

Then $j_m < i - 1$ by construction, so $j_m + 1 < i$. Our maximum construction also ensures $j_{m+1} > i - 1$, so $j_{m+1} + 2 > i$. Thus the edge $(j_m + 1, j_{m+1} + 2)$ passes over vertex i , and we know $(j_m + 1, j_{m+1} + 2) \in E(G_\beta)$ by the construction in Theorem 3.4.

To prove uniqueness, we once again turn to the construction of $E(G_\beta)$ in the proof of Theorem 3.4. Recall that we have three types of edges:

- (i) Spine edges: $\{(i, i + 1) \mid i \in [n]\} \cup \{(1, 2), (n, n + 1)\}$
- (ii) 0-type edges: $\{(l + 1, l + 2) \mid b_l = 0, \text{ for } 1 \leq l \leq n - 2\}$
- (iii) 1-type edges: $\{(1, j_1 + 2), (j_1 + 1, j_2 + 2), \dots, (j_{r-1} + 1, j_r + 2), (j_r + 1, n + 1)\}$

Since both (i) and (ii) connect consecutive vertices, no edge of those types can pass over an vertex. The only other edges in $E(G_\beta)$ are of the form $(j_l + 1, j_{l+1} + 2)$ for $j_l \in \widehat{J}_\beta$. But if $l \neq m$, then $(j_l + 1, j_{l+1} + 2)$ doesn't pass over i . □

Proposition 3.15. *Let $G \in \mathcal{F}_{n,3}$ and let i be an interior vertex. Then any integral flow f on G with netflow \mathbf{w}_n sends at most $i - 2$ units of flow into vertex i .*

Proof. Suppose f sends all available flow through the spine. Then vertices $2, \dots, i - 1$ each contribute one unit of flow into vertex i . This accounts for all possible outflow before vertex i . □

Proposition 3.16. *Let $G \in \mathcal{F}_{n,3}$ and let i be an interior vertex. Let (a, d) be the unique edge passing over i as in Lemma 3.14. Then any integral flow f on G with netflow \mathbf{w}_n has $i - 2 - f(a, d)$ units of flow entering vertex i .*

Proof. Since (a, d) is the unique edge passing over i , all flow coming out of vertices $2, \dots, i - 1$ not sent on edge (a, d) must pass through vertex i . Since the netflow vector is \mathbf{w}_n , each of these vertices contributes exactly one unit of flow. □

Proposition 3.17. *Let $G \in \mathcal{F}_{n,3}$ with crossed edge pair $(a, c), (b, d)$. Then $c = b + 1$. That is, the middle two vertices of an crossed pair of edges must be consecutive.*

Proof. Assume for the sake of contradiction that $G \in \mathcal{F}_{n,3}$ has interchange pairs $(a, c), (b, d)$ where $c > b + 1$. Then if we look at vertex $b + 1$ (which is interior), we have that (a, c) satisfies $a < b + 1 < c$ and (b, d) satisfies $b < b + 1 < d$. But this means there are two distinct edges passing over vertex $b + 1$, in violation of Lemma 3.14. □

Lemma 3.18. *Let $G_\beta \in \mathcal{F}_{n,3}$ with $\beta = b_1 \cdots b_{n-2}$. Suppose $b_i = 0$. Then $(i + 1, i + 2) \in E(G_\beta)$, and is nested within exactly one other edge in $E(G_\beta)$.*

Proof. Since $b_i = 0$, it follows from the bijective construction in Theorem 3.4 that $(i + 1, i + 2) \in E(G_\beta)$. Then we can apply Lemma 3.14 to vertex $i + 1$ to get a unique edge $(j_l + 1, j_{l+1} + 2)$ satisfying $j_l + 1 < i + 1 < j_{l+1} + 2$.

Since $\beta_i = 0$, both in-stubs for $i + 2$ are filled by consecutive edges (Type (i) and Type (ii)). Thus $j_{l+1} + 2 \neq i + 2$, so we have $j_l + 1 < i + 1 < i + 2 < j_{l+1} + 2$, and the edges $(i + 1, i + 2)$ and $(j_l + 1, j_{l+1} + 2)$ are nested. \square

Lemma 3.19. *Let $\beta \in (\mathbb{Z}/2\mathbb{Z})^{n-2}$ with $b_i = 0$. Define β' to be the binary string obtained by changing β to have a 1 at index i . Define G' to be the DAG obtained by interchanging the edges $(i + 1, i + 2)$ and $(j_l + 1, j_{l+1} + 2)$ in G (as in Lemma 3.18). Then $G_{\beta'} = G'$.*

In short, changing a 0 to a 1 is “the same” as interchanging the edge associated with the 0.

Proof. Since $V(G') = V(G_{\beta'}) = [n + 1]$, we need only show that $E(G') = E(G_{\beta'})$. By Lemma 3.18 and the definition of interchange, we have that

$$E(G') = E(G) / \{(i + 1, i + 2), (j_l + 1, j_{l+1} + 2)\} \cup \{(i + 1, j_{l+1} + 2), (j_l + 1, i + 2)\}.$$

Now consider β' . By assumption, the list of indices where β' has 1s is

$$J_{\beta'} = J_\beta \cup \{i\} = \{\dots j_l < i < j_{l+1} < \dots\}.$$

It follows from the bijective construction that $(i + 1, j_{l+1} + 2), (j_l + 1, i + 2) \in E(G_{\beta'})$. Additionally, since $b'_i \neq 0$, $G_{\beta'}$ does not have the edge $(i + 1, i + 2)$. Finally since j_l and j_{l+1} are not the indices of consecutive 1s in β' , $G_{\beta'}$ does not have the edge $(j_l + 1, j_{l+1} + 2)$. Since no other indices of β were changed to get β' , we have that

$$\begin{aligned} E(G_{\beta'}) &= E(G) / \{(i + 1, i + 2), (j_l + 1, j_{l+1} + 2)\} \cup \{(i + 1, j_{l+1} + 2), (j_l + 1, i + 2)\} \\ &= E(G'). \end{aligned} \quad \square$$

Corollary 3.20. *Every DAG in $\mathcal{F}_{n,3}$ is obtainable by performing interchange operations on the DAG G defined by*

$$\begin{aligned} V(G) &= [n + 1] \\ E(G) &= \{(i, i + 1)^2 \mid i \in [n]\} \cup \{(1, n + 1)\}. \end{aligned}$$

That is, the DAG G in correspondence with the binary string $0 \dots 0$.

Proof. Let $G_\beta \in \mathcal{F}_{n,3}$, where $\beta \in (\mathbb{Z}/2\mathbb{Z})^{n-2}$ is the binary string corresponding to G_β . Then perform interchanges to $G_{0\dots 0}$, as in Lemma 3.19, at every index where β has a 1. Note that we can choose any order to perform the necessary interchanges. This is ensured by the bijection with binary strings and previous lemma. \square

Corollary 3.21. *The interchange operation induces a partial order on $\mathcal{F}_{n,3}$ that agrees with the Boolean partial order on binary strings of length $n - 2$.*

Example 3.22. See Figure 3.2 for the Hasse diagram demonstrating poset equivalence between $\mathcal{F}_{5,3}$ and $(\mathbb{Z}/2\mathbb{Z})^3$. Edge interchange pairs are listed next to the edges they correspond with (assume moving upwards through the Hasse diagram). Note that all 2^3 DAGs in $\mathcal{F}_{5,3}$ are achieved via interchanges of G_{000} .

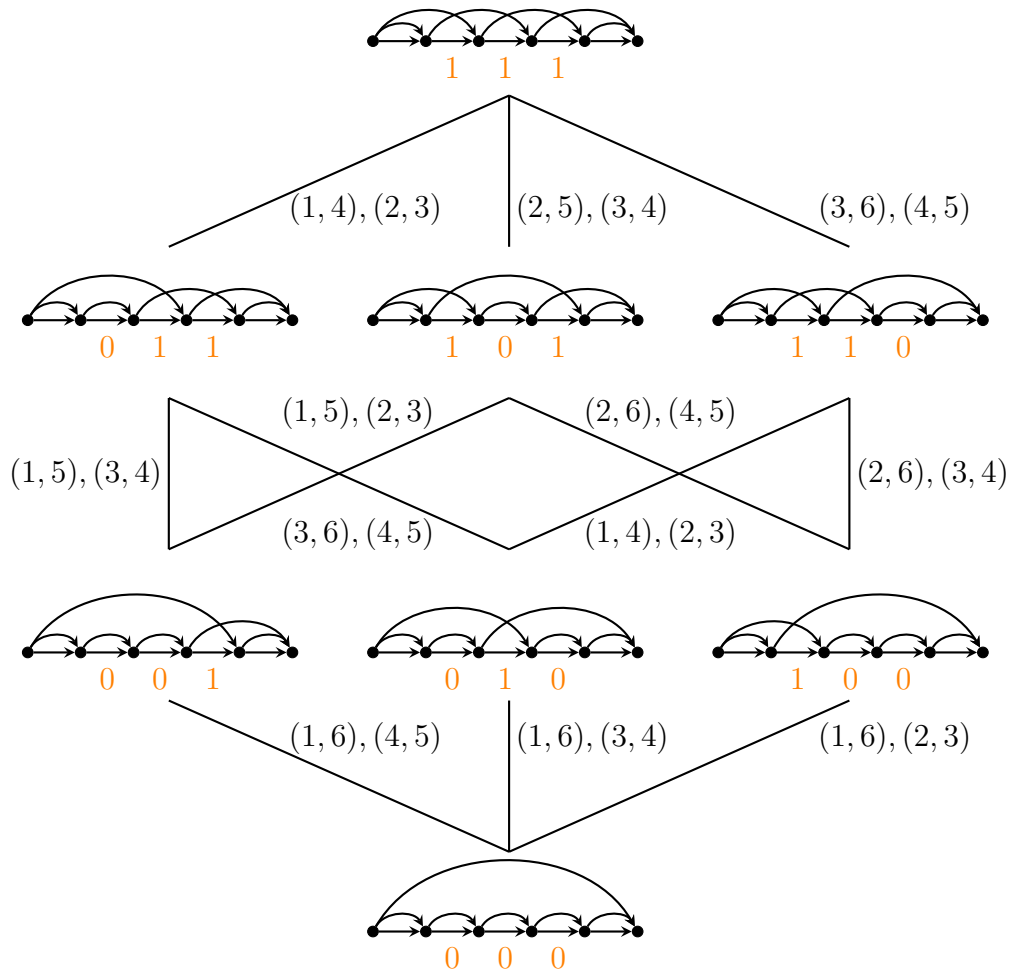


Figure 3.2: $\mathcal{F}_{5,3}$ Hasse diagram with edges labelled by corresponding interchanges. The DAG vertex labels are suppressed for clarity, but each DAG’s vertices are labelled 1 to 6 from left to right.

Remark 3.23. While the interchange operation can be used to define a partial order on other sets of DAGs with fixed degree sequence, the bijection with strings breaks down, and it is not clear that this defines a partial order for every degree sequence. Further, even when considering DAGs with the “next most complicated” degree sequence $(4, 2, \dots, 2, 0)$, which are still full DAGs, we find that the partial order induced by interchange operation is not necessarily a lattice. At this time it seems promising that the techniques in this paper can be used to extend our results to other degree sequences, but it is not clear how to do so. The one case to which this extension is straightforward is the case of r -full DAGs with outdegree $r + 1$ for the source.

4. Volume Inequalities

Our goal in this section is to prove Theorem 4.20, which states that the values of the volumes of flow polytopes for DAGs under the Boolean partial order obtained in Corollary 3.20 respects the dual partial order. To illustrate this, consider Example 4.1, which demonstrates that the function $\text{vol} : \mathcal{F}_{n,3} \rightarrow \mathbb{Z}_{\geq 0}$ is order reversing with respect to our interchange order, as we prove in Theorem 4.20. To prove that this is true in general, we construct a procedure to find the value of $K_G(\mathbf{w}_n)$, and hence find the volume of $\mathcal{F}_1(G)$ (Proposition 2.7). Further, this procedure can be used to compute $K_G(\mathbf{a})$ for any netflow vector \mathbf{a} , and any DAG G (not necessarily full).

Example 4.1. In Figure 4.1, the DAGs in $\mathcal{F}_{5,3}$ are arranged as in Figure 3.2. Additionally, labels are given for each DAG G that provide the volume of its flow polytope $\mathcal{F}_1(G)$.

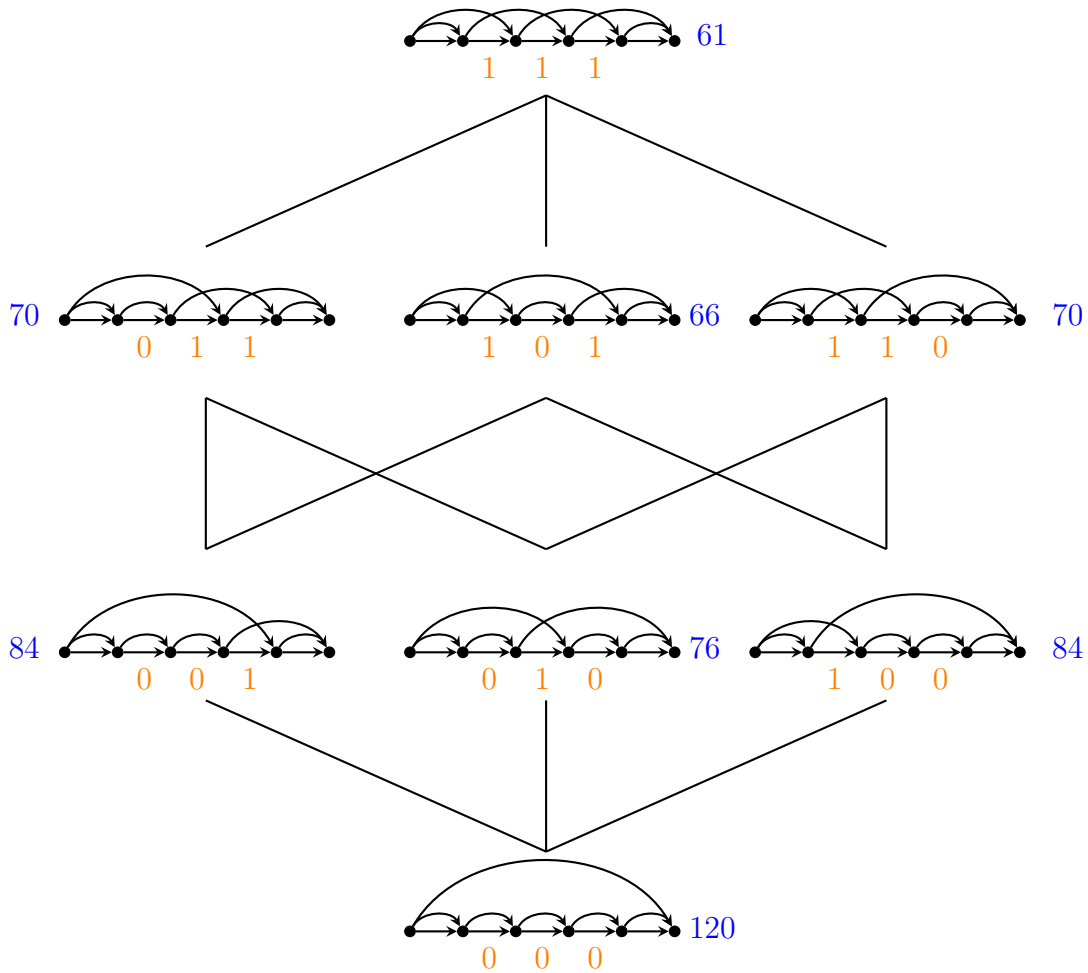


Figure 4.1: $\mathcal{F}_{5,3}$ Hasse diagram with associated flow polytope volumes.

We begin with some notation and definitions regarding partially-determined flow values for a DAG.

Definition 4.2. Let G be a DAG on vertex set $[n + 1]$ with netflow $\mathbf{a} = (a_1, \dots, a_n, -\sum_i a_i)$. Let $L \subseteq E(G)$ and $M \subseteq [n + 1]$. Define a *partial flow* $f_{G,L} : L \rightarrow \mathbb{R}_{\geq 0}$ to be an assignment of flow values to the subset of edges L . We say a partial flow $f_{G,L}$ is *valid on* M if the assigned flow values respect the netflow values a_i for all $i \in M$. That is, the outflow at i minus the inflow at i is equal to a_i for every vertex $i \in M$.

For convenience, we introduce the following notation. For G a DAG with vertices $[n + 1]$, we set

$$L_k := \{(i, j) \in E(G) \mid i \leq k\}.$$

Definition 4.3. Let G be a DAG with vertices $[n + 1]$ and $\mathbf{a} = (a_1, \dots, a_n, -\sum_i a_i)$ be a netflow vector. We define a tree $T_G(\mathbf{a})$, called the *flow decomposition tree*, as follows:

- The nodes of $T_G(\mathbf{a})$ are of the form (G, i, f_{G,L_i}) , i.e., they consist of the DAG G with a partial flow f_{G,L_i} valid on $[i]$, for each $0 \leq i \leq n$. Note that when $i = 0$, we consider the node $(G, 0, f_{G,L_0})$ to be G with no flow values assigned.
- The edges of $T_G(\mathbf{a})$ are between (G, i, f_{G,L_i}) and $(G, i + 1, g_{G,L_{i+1}})$ whenever

$$f_{G,L_i} = g_{G,L_{i+1}}|_{L_i}.$$

We will use the word “node” when referring to flow decomposition trees and the word “vertex” when referring to DAGs to help distinguish the different objects. Note that the data of a node (G, i, f_{G,L_i}) in $T_G(\mathbf{a})$ includes the DAG G , a vertex i , and a partial flow on L_i valid for $[i]$. However, we can represent the node (G, i, f_{G,L_i}) by just its partial flow f_{G,L_i} , since the DAG is already part of the partial flow’s data, and we choose to make the set of vertices on which the partial flow is valid to match the subscript of L_i . We will use this abbreviation convention unless specifying the full node data is necessary.

Definition 4.4. Let f_{G,L_i} be a partial flow on G valid on $[n]$ that determines a node in $T_G(\mathbf{a})$. Denote the number of branches at the node f_{G,L_i} by $b(f_{G,L_i})$. Similarly, denote the number of leaves at the node f_{G,L_i} by $l(f_{G,L_i})$.

Observe that $T_G(\mathbf{a})$ is a rooted tree whose leaves are (complete) flows on G . Thus

$$K_G(\mathbf{a}) = \# \text{ of leaves in } T_G(\mathbf{a}) = l(G, 0, f_{G,L_0}).$$

Further, we can generate $T_G(\mathbf{a})$ inductively using the following process.

1. Begin with the node f_{G,L_0} , which is the root.
2. Compute all possible partial flows on L_1 valid for vertex 1, and create a node for each. These nodes form the second level of $T_G(\mathbf{a})$.
3. Continue this process until termination. For a node of the form f_{G,L_i} , its children are extensions of f_{G,L_i} valid on L_{i+1} . In other words, given a partial flow on L_i valid on $[i]$, assign flows to edges out of vertex $i + 1$ so that netflow at vertex $i + 1 = a_{i+1}$

4. A node is a leaf if $i = n + 1$ (flows have been assigned to all edges).

We say a node f_{G,L_i} in $T_G(\mathbf{a})$ is *at (or on) level $i + 1$* . We denote the set of level $i + 1$ nodes in $T_G(\mathbf{a})$ by $V_{i+1}(T_G(\mathbf{a}))$, or $V_{i+1}(G)$ if the netflow vector is understood, or just V_{i+1} if both the DAG and netflow vector are clear. The branches of a level i node, which are level $i + 1$ nodes, correspond with the cases of partial flows on L_i valid on $[i]$. Further, looking at all the level $i + 1$ nodes of $T_G(\mathbf{a})$, we see all possible partial flows on L_i valid on $[i]$. Further still, the following are equivalent:

- A partial flow with outflow determined for vertices $[i]$, respecting the netflows on $[i]$;
- A partial flow f_{G,L_i} valid on vertices $[i]$; and
- A level $i + 1$ node in $T_G(\mathbf{a})$.

Remark 4.5. These are not the first trees in the literature related to volumes of flow polytopes. Mészáros and Morales introduce compound reduction trees in [MM19]. Mészáros, Morales, and Striker refine the reduction trees and extend their application to framed DAGs in [MMS19]. However, these trees are fundamentally different than flow decomposition trees. A reduction tree of a DAG G describes subdivisions within the flow polytope $\mathcal{F}_G(\mathbf{a})$, and the flow decomposition tree displays the possible integer flows on G , or equivalently, the lattice points of $\mathcal{F}_G(\mathbf{a})$. Sometimes parts of the trees may coincide. For instance, in the special case where $\mathbf{a} = (1, 0, \dots, 0, -1)$, the leaves of a reduction tree for G will represent a simplicial subdivision (a triangulation) of $\mathcal{F}_G(1, 0, \dots, 0, -1)$. By (2.1), it follows that the number of simplices in the subdivision of $\mathcal{F}_G(1, 0, \dots, 0, -1)$ equals the number lattice points in the flow polytope $\mathcal{F}_G(0, \text{ind}_2 - 1, \dots, \text{ind}_n - 1, |V| - |E| - 2 + \text{ind}_{n+1})$, which is counted precisely by the leaves of the flow decomposition tree of G with netflow vector $(0, \text{ind}_2 - 1, \dots, \text{ind}_n - 1, |V| - |E| - 2 + \text{ind}_{n+1})$. However, this coincidence need only apply to the leaves of the related trees. It can be checked that the graph G of Figure 4 has a flow decomposition tree $T_G(0, 1, 2, -3)$ with eight leaves arranged into two branches of four each, while the canonical compound reduction tree for G also has eight leaves, but arranged into a branch of two leaves and two branches of three leaves.



Figure 4.2: A graph G having a compound reduction tree and an associated flow decomposition tree with the same number of leaves but different tree structure.

Example 4.6. See Figure 4.3 for $T_{K_4}(1, 1, 1, -3)$. Each node of the flow decomposition tree is a copy of K_4 (vertices labelled 1 to 4, but suppressed) with a partial flow. In the partial flows, unassigned and flow 0 edges are dashed, edges with flow 1 are solid, and flows greater than 1 are labelled.

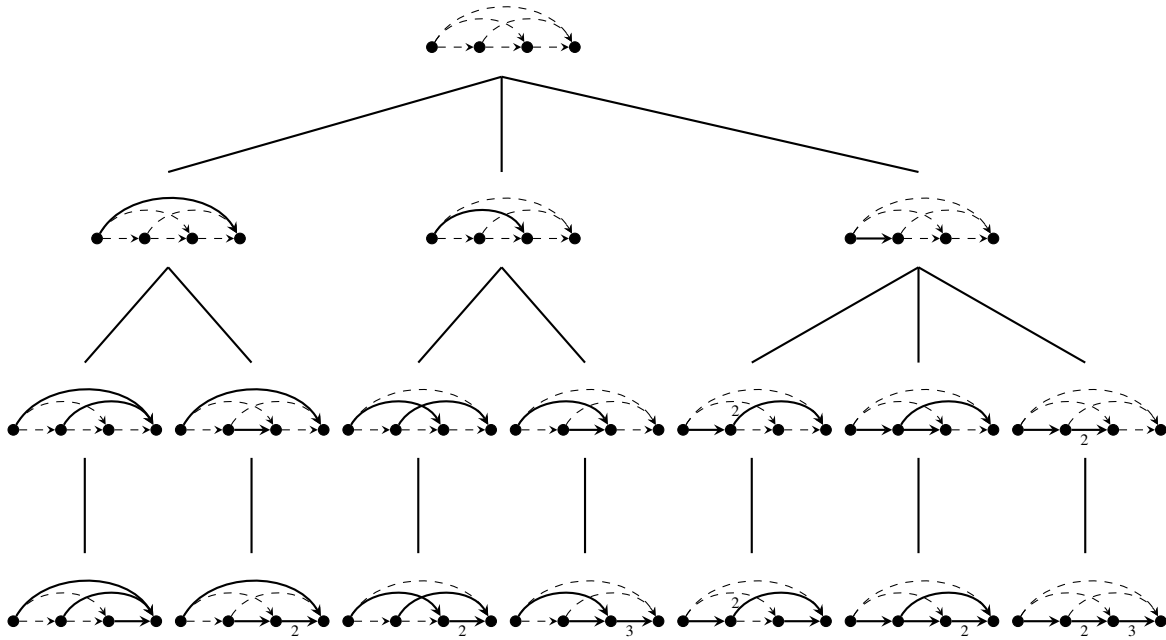


Figure 4.3: The flow decomposition tree $T_{K_4}(1, 1, 1, -3)$.

In the following proposition, we show that the number of branches at a given node in the flow decomposition tree is determined by the partial flow, netflow vector, and out-degree sequence.

Proposition 4.7. *Let $f_{G,L_{i-1}}$ be a level i node in $T_G(\mathbf{a})$ for some $1 \leq i \leq n$. Suppose the partial flow $f_{G,L_{i-1}}$ has t units of inflow entering vertex i . Then the number of branches of the node is*

$$b(f_{G,L_{i-1}}) = \binom{t + a_i + \text{outd}_i - 1}{\text{outd}_i - 1}.$$

Proof. To extend $f_{G,L_{i-1}}$ to a $[i]$ -valid partial flow on L_i , we need to respect the netflow at vertex i . Thus we need the outflow at i to be $t + a_i$ units. This outgoing flow is distributed over the outgoing edges from i . Hence the number of branches is the number of weak compositions of $t + a_i$ into outd_i parts. This quantity is counted by $\binom{t + a_i + \text{outd}_i - 1}{\text{outd}_i - 1}$. \square

For $G \in \mathcal{F}_{n,3}$, we have the following special case.

Corollary 4.8. *Let $G \in \mathcal{F}_{n,3}$ and netflow vector \mathbf{w}_n . Then the number of branches of a node $f_{G,L_{i-1}}$ will be*

$$b(f_{G,L_{i-1}}) = \begin{cases} 1, & i = 1 \\ 2, & i = 2 \\ t + 2, & i \geq 3 \end{cases}.$$

Proof. By assumption $G \in \mathcal{F}_{n,3}$, $\text{out}_d = 3$ and $\text{out}_i = 2$ for interior vertices i . Further, assuming netflow vector to be \mathbf{w}_n gives $a_1 = 0$ and $a_i = 1$ for interior vertices i . Thus

$$b(f_{G,L_{i-1}}) = \begin{cases} \binom{0+0+3-1}{3-1}, & i = 1 \\ \binom{0+1+2-1}{2-1}, & i = 2 \\ \binom{t+1+2-1}{2-1}, & i \geq 3 \end{cases} = \begin{cases} 1, & i = 1 \\ 2, & i = 2 \\ t + 2, & i \geq 3 \end{cases}. \quad \square$$

The following lemma establishes a relationship between level i nodes in flow decomposition trees for DAGs related by an edge interchange.

Lemma 4.9. *Let $G \in \mathcal{F}_{n,3}$. Suppose G' is obtained by interchanging edges (a, d) and $(b, b + 1)$ in G . Then for every $1 \leq i \leq b + 1$, the nodes of level i in $T_G(\mathbf{w}_n)$ are in bijection with the nodes of level i in $T_{G'}(\mathbf{w}_n)$. This bijection is denoted φ .*

Proof. We construct a map φ between the level i nodes of $T_{G'}(\mathbf{w}_n)$ and $T_G(\mathbf{w}_n)$ for $1 \leq i \leq b + 1$. Given a level i node $f_{G',L_{i-1}}$ in $T_{G'}(\mathbf{w}_n)$, we map it to the following node in $T_G(\mathbf{w}_n)$:

$$\varphi(f_{G',L_{i-1}})(j, k) := f_{G,L_{i-1}}(j, k) = \begin{cases} f_{G',L_{i-1}}(a, b + 1), & (j, k) = (a, d) \\ f_{G',L_{i-1}}(b, d), & (j, k) = (b, b + 1) \\ f_{G',L_{i-1}}(j, k), & \text{else} \end{cases}$$

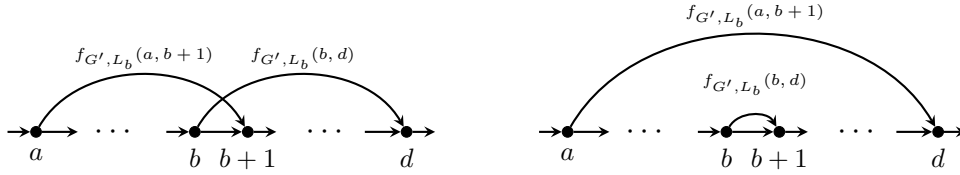


Figure 4.4: A level $b + 1$ node f_{G',L_b} and $\varphi(f_{G',L_b})$, respectively.

This defines a partial flow on L_{i-1} in G . In order for it to correspond to a node in $T_G(\mathbf{w}_n)$, the partial flow must be valid on $[i - 1]$. Since G and G' have edges in common, we need only verify validity for the vertices where edges have changed. The vertices where edges have changed are $a, b, b + 1$ and d . However, since $1 \leq i \leq b + 1$, the only vertices in $[i - 1]$ are a and b . But for both a and b , this map doesn't change the edges going in to a or b , and preserves the outflow from a and b . Thus, netflow at a and b is preserved, so $f_{G,L_{i-1}}$ is valid on a and b .

Given two distinct partial flows in $T_{G'}(\mathbf{w}_n)$, it follows directly from the construction of φ that their images will have distinct flows. Thus φ is injective.

Given a level i node in $T_G(\mathbf{w}_n)$, call it $f_{G,L_{i-1}}$, consider the following partial flow f on G' defined by

$$f(j, k) := \begin{cases} f_{G,L_{i-1}}(a, d), & (j, k) = (a, b + 1) \\ f_{G,L_{i-1}}(b, b + 1), & (j, k) = (b, d) \\ f_{G,L_{i-1}}(j, k), & \text{else} \end{cases}.$$

Next we verify that f is a level i node in $T_{G'}(\mathbf{w}_n)$. Observe that it is equivalent to check that f is partial flow on L_{i-1} valid on vertices $[i - 1]$ (all in G). If we look at the subgraph of G' induced

by the vertices $[i - 1] \setminus \{a, b\}$, G' agrees with G and f agrees with $f_{G, L_{i-1}}$. Further, f is defined in a way that preserves the netflow at vertices a and b .

Finally, we show that φ maps f to $f_{G, L_{i-1}}$. For every edge $(j, k) \in L_{i-1}$ in G , applying our construction for φ followed by our assumptions about f gives

$$\begin{aligned} \varphi(f)(j, k) &= \begin{cases} f(a, b + 1), & (j, k) = (a, d) \\ f(b, d), & (j, k) = (b, b + 1) \\ f(j, k), & \text{else} \end{cases} \\ &= \begin{cases} f_{G, L_{i-1}}(a, d), & (j, k) = (a, d) \\ f_{G, L_{i-1}}(b, b + 1), & (j, k) = (b, b + 1) \\ f_{G, L_{i-1}}(j, k), & \text{else} \end{cases} \\ &= f_{G, L_{i-1}}(j, k). \end{aligned} \quad \square$$

Example 4.10. In Figure 4.5 and Figure 4.6 we have the flow decomposition trees $T_{G_{11}}(\mathbf{w}_4)$ and $T_{G_{10}}(\mathbf{w}_4)$, respectively. In order to fit the tree on the page, the bottom-level nodes have been stacked vertically below their ancestors. In this example, G_{11} can be obtained by performing an interchange operation on G_{10} with edges $(2, 5)$ and $(3, 4)$. Thus, we have $a = 2$, $b = 3$, and $d = 5$. Observe that for each level up to $b + 1 = 4$, the nodes of $T_{G_{11}}(\mathbf{w}_4)$ and $T_{G_{10}}(\mathbf{w}_4)$ are in bijection. The flow decomposition trees have been arranged in a way that the bijective pairs lie in the same relative position in their respective trees. It is also worth noting that for the netflow vector \mathbf{w}_n , the outflow at vertex 1 is 0, so the root always has a single branch.

Now that we have established basic properties of flow decomposition trees, we seek to organize these DAGs in a way that ensures $T_{G'}(\mathbf{w}_n)$ has fewer leaves than $T_G(\mathbf{w}_n)$, where $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . To do this, we focus on the level $b + 1$ nodes. Throughout the remainder of this section, assume G and G' are related by the edge interchange structure just described.

Lemma 4.11. Assume $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . Let f_{G', L_b} be a level $b + 1$ node in $T_{G'}(\mathbf{w}_n)$, with corresponding level $b + 1$ node $\varphi(f_{G', L_b})$ in $T_G(\mathbf{w}_n)$. Then

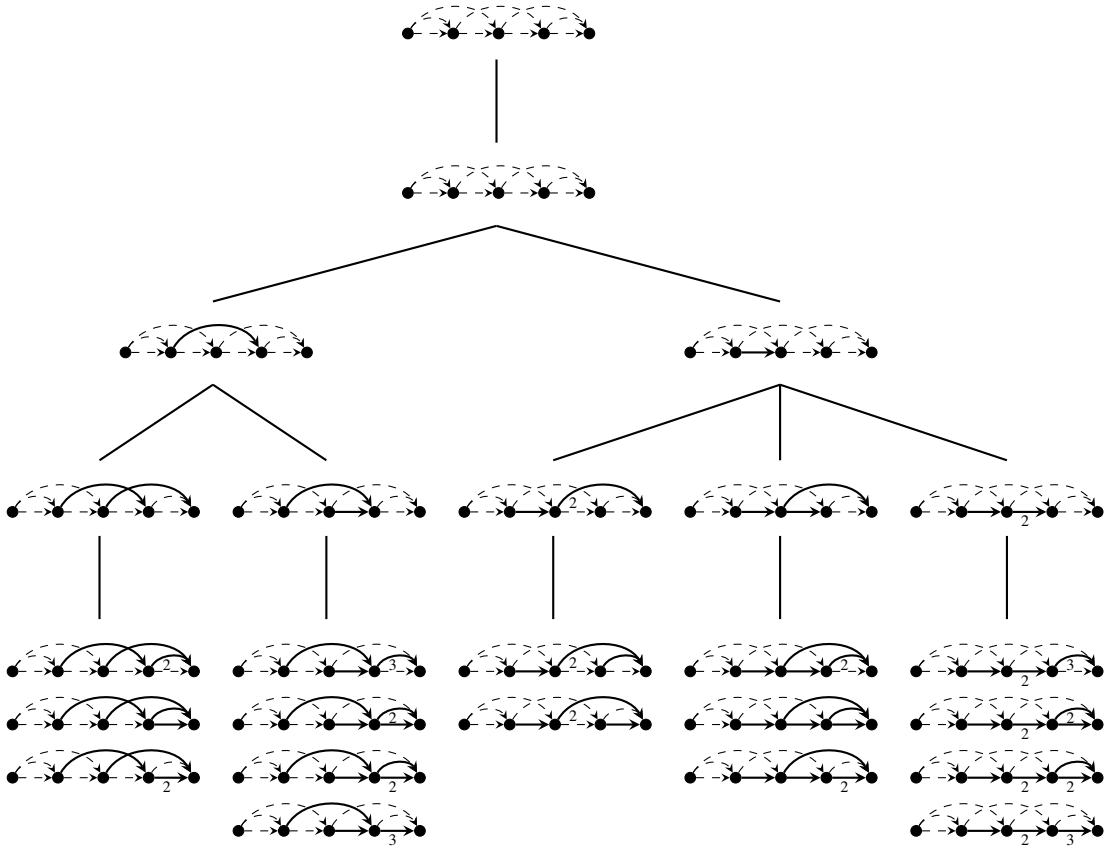
$$l(f_{G', L_b}) > l(\varphi(f_{G', L_b}))$$

if and only if $f_{G', L_b}(a, b + 1) > f_{G', L_b}(b, d)$.

Proof. Let f_{G', L_b} be a level $b + 1$ node in $T_{G'}(\mathbf{w}_n)$. Note that a leaf of f_{G', L_b} is a (complete) flow f on G' that agrees with f_{G', L_b} . Similarly, a leaf in $T_G(\mathbf{w}_n)$ of the level $b + 1$ node $\varphi(f_{G', L_b})$ is a (complete) flow g on G that agrees with $\varphi(f_{G', L_b})$.

For the backward implication of our if and only if, suppose $f_{G', L_b}(a, b + 1) > f_{G', L_b}(b, d)$. We build an injection from the leaves of $\varphi(f_{G', L_b})$ into the leaves of f_{G', L_b} , then show that the injection is not surjective. Given a leaf g of the level $b + 1$ node $\varphi(f_{G', L_b})$ in $T_G(\mathbf{w}_n)$, we map it to the leaf f of $T_{G'}(\mathbf{w}_n)$, where f is defined as

$$f(i, j) = \begin{cases} g(a, d), & (i, j) = (a, b + 1) \\ g(b, b + 1), & (i, j) = (b, d) \\ g(e_i) + g(a, d) - g(b, b + 1), & (i, j) = e_i, b + 1 \leq i \leq d - 1 \\ g(i, j), & \text{else} \end{cases} .$$

Figure 4.5: $T_{G_{11}}(w_4)$.

This map preserves the flows on the non-spine edges leaving a and b , which forces more flow into vertex $b + 1$ in G , which we adjust for by increasing the flow on the spine between vertices $b + 1$ and d .

Next, we verify that f is a valid flow on G' . That is, the flows on $E(G')$ prescribed by f agree with the netflow vector w_n . For vertices $i \neq b + 1, \dots, d$, we have

$$(f \text{ outflow at } i) - (f \text{ inflow at } i) = (g \text{ outflow at } i) - (g \text{ inflow at } i) = w_i.$$

For the vertex $b + 1$,

$$\begin{aligned} (f \text{ outflow at } b + 1) - (f \text{ inflow at } b + 1) &= [f(e_{b+1}) + f(b + 1, b + 2)] - [f(e_b) + f(a, b + 1)] \\ &= [g(e_{b+1}) + g(a, d) - g(b, b + 1) + g(b + 1, b + 2)] \\ &\quad - [g(e_b) + g(a, d)] \\ &= [g(e_{b+1}) + g(b + 1, b + 2)] - [g(e_b) + g(b, b + 1)] \\ &= (g \text{ outflow at } b + 1) - (g \text{ inflow at } b + 1) \\ &= w_{b+1}. \end{aligned}$$

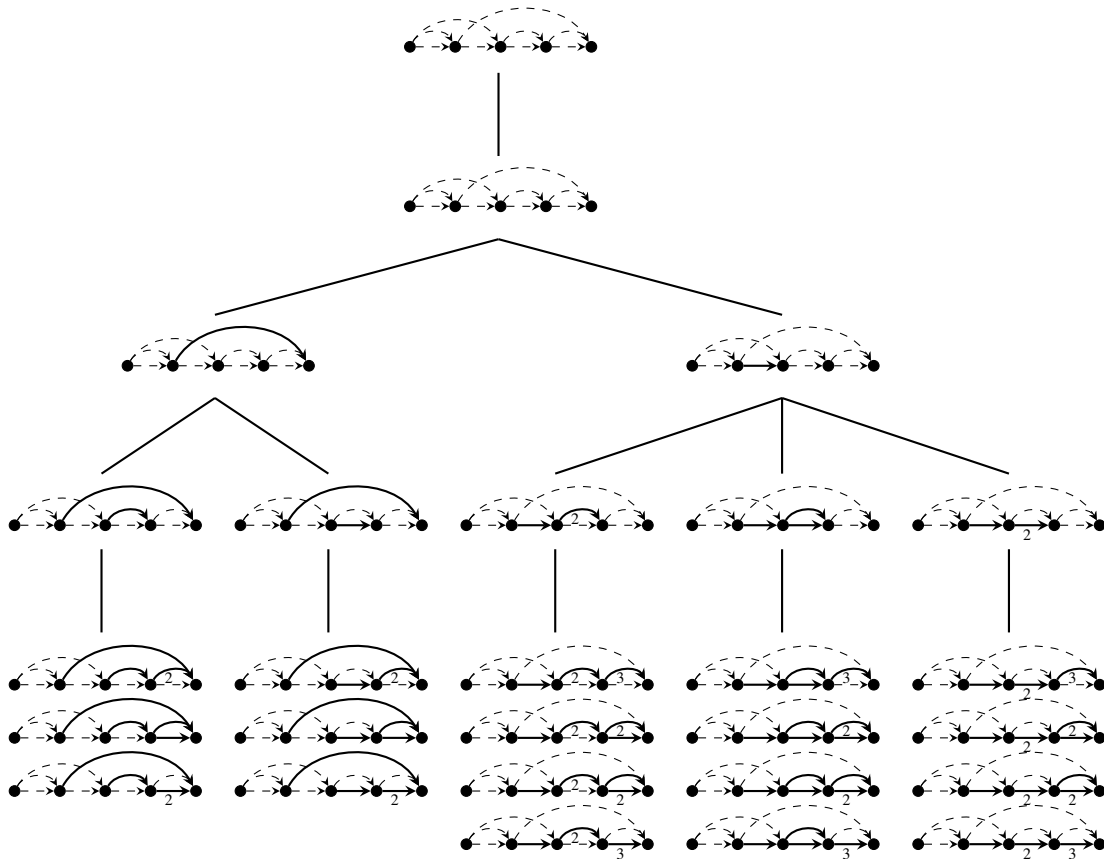


Figure 4.6: $T_{G_{10}}(w_4)$.

If there are any vertices $b + 2 \leq i \leq d - 1$, note that all four incident edges are consecutive edges, else we would violate the unique overpass condition of Lemma 3.14. So we have

$$\begin{aligned}
 (f \text{ outflow at } i) - (f \text{ inflow at } i) &= [f(e_i) + f(i, i + 1)] - [f(e_{i-1}) + f(i - 1, i)] \\
 &= [g(e_i) + g(a, d) - g(b, b + 1) + g(i, i + 1)] \\
 &\quad - [g(e_{i-1}) + g(a, d) - g(b, b + 1) + g(i - 1, i)] \\
 &= [g(e_i) + g(i, i + 1)] - [g(e_{i-1}) + g(i - 1, i)] \\
 &= (g \text{ outflow at } i) - (g \text{ inflow at } i) \\
 &= w_i.
 \end{aligned}$$

The last vertex to check is d , which has two outgoing edges e_d (the spine) and (d, j) for some j .

Our netflow is then

$$\begin{aligned}
(f \text{ outflow at } d) - (f \text{ inflow at } d) &= [f(e_d) + f(d, j)] - [f(e_{d-1}) + f(d-1, d)] \\
&= [g(e_d) + g(d, j)] \\
&\quad - [g(e_{d-1}) + g(a, d) - g(b, b+1) + g(d-1, d)] \\
&= [g(e_d) + g(d, j)] - [g(e_{d-1}) + g(a, d)] \\
&= (g \text{ outflow at } d) - (g \text{ inflow at } d) \\
&= w_d.
\end{aligned}$$

Thus f is indeed a valid flow on G' . Now we show that f agrees with f_{G', L_b} . For every edge $(i, j) \in L_b$, we apply the construction of f , the fact that g agrees with $\varphi(f_{G', L_b})$, and the definition of φ , respectively, to get that

$$\begin{aligned}
f(i, j) &= \begin{cases} g(a, d), & (i, j) = (a, b+1) \\ g(b, b+1), & (i, j) = (b, d) \\ g(i, j), & \text{else} \end{cases} \\
&= \begin{cases} \varphi(f_{G', L_b})(a, d), & (i, j) = (a, b+1) \\ \varphi(f_{G', L_b})(b, b+1), & (i, j) = (b, d) \\ \varphi(f_{G', L_b})(i, j), & \text{else} \end{cases} \\
&= \begin{cases} f_{G', L_b}(a, b+1), & (i, j) = (a, b+1) \\ f_{G', L_b}(b, d), & (i, j) = (b, d) \\ f_{G', L_b}(i, j), & \text{else} \end{cases} \\
&= f_{G', L_b}(i, j).
\end{aligned}$$

At this point, we have only verified that the map $g \mapsto f$ is well-defined. To prove it is an injective map, let g_1 and g_2 be (flows on G that give) leaves in $T_G(\mathbf{w}_n)$. Suppose that both g_1 and g_2 map to a leaf f in $T_{G'}(\mathbf{w}_n)$. Then

$$= \begin{cases} \begin{cases} g_1(a, d), & (i, j) = (a, b+1) \\ g_1(b, b+1), & (i, j) = (b, d) \\ g_1(e_i) + g_1(a, d) - g_1(b, b+1), & (i, j) = e_i, b+1 \leq i \leq d-1 \\ g_1(i, j), & \text{else} \end{cases} \\ \begin{cases} g_2(a, d), & (i, j) = (a, b+1) \\ g_2(b, b+1), & (i, j) = (b, d) \\ g_2(e_i) + g_2(a, d) - g_2(b, b+1), & (i, j) = e_i, b+1 \leq i \leq d-1 \\ g_2(i, j), & \text{else} \end{cases} \end{cases} .$$

We can immediately conclude that $g_1(a, d) = g_2(a, d)$ and $g_1(b, b+1) = g_2(b, b+1)$, and that $g_1(i, j) = g_2(i, j)$ for every edge that isn't a spine edge between $b+1 \leq i \leq d-1$. Then it follows that $g_1(e_i) = g_2(e_i)$ for every spine edge e_i such that $b+1 \leq i \leq d-1$. Hence $g_1 = g_2$, so our map is indeed injective.

To prove that the inequality $l(f_{G',L_b}) \geq l(\varphi(f_{G',L_b}))$ is strict, we describe a leaf of f_{G',L_b} that cannot be in the image of our injection. Consider the flow h on G' , where

$$h(i, j) = \begin{cases} f_{G',L_b}(i, j), & (i, j) \in L_b \\ i - 1 - f_{G',L_b}(b, d), & (i, j) = (i, i + 1), b + 1 \leq i \leq d - 2 \\ d - 1 - f_{G',L_b}(b, d), & (i, j) = e_{d-1} \\ i - 1, & (i, j) = e_i, d \leq i \leq n \\ 0, & \text{else} \end{cases} .$$

To verify that h is a valid flow on G' , we need only check the netflows at vertices $b + 1, \dots, n$ since h agrees with f_{G',L_b} on other vertices. At vertex $b + 1$,

$$\begin{aligned} (h \text{ outflow at } b + 1) - (h \text{ inflow at } b + 1) &= [h(b + 1, b + 2) + h(e_{b+1})] - [h(a, b + 1) + h(e_b)] \\ &= [b - f_{G',L_b}(b, d) + 0] \\ &\quad - [f_{G',L_b}(a, b + 1) + f_{G',L_b}(e_b)] \\ &= (f_{G',L_b} \text{ outflow at } b + 1) - (f_{G',L_b} \text{ inflow at } b + 1) \end{aligned}$$

by Proposition 3.16.

If there are any vertices i for $b + 2 \leq i \leq d - 2$,

$$\begin{aligned} (h \text{ outflow at } i) - (h \text{ inflow at } i) &= [h(i, i + 1) + h(e_i)] - [h(i - 1, i) + h(e_{i-1})] \\ &= [i - 1 - f_{G',L_b}(b, d) + 0] - [i - 2 - f_{G',L_b}(b, d) + 0] \\ &= 1 \\ &= w_i. \end{aligned}$$

At vertex $d - 1$, there is some non-spine edge $(d - 1, j)$. Then our netflow is

$$\begin{aligned} (h \text{ outflow at } d) - (h \text{ inflow at } d) &= [h(d - 1, j) + h(e_{d-1})] - [h(d - 2, d - 1) + h(e_{d-2})] \\ &= [0 + d - 2 - f_{G',L_b}(b, d)] \\ &\quad - [0 + d - 3 - f_{G',L_b}(b, d)] \\ &= 1 \\ &= w_{d-1}. \end{aligned}$$

At vertex d , there is some non-spine edge (d, j) . Then our netflow is

$$\begin{aligned} (h \text{ outflow at } d) - (h \text{ inflow at } d) &= [h(d, j) + h(e_d)] - [h(b, d) + h(e_{d-1})] \\ &= [d - 1 + 0] \\ &\quad - [f_{G',L_b}(b, d) + d - 2 - f_{G',L_b}(b, d)] \\ &= 1 \\ &= w_d. \end{aligned}$$

At a vertex i for $d + 1 \leq i \leq n$, there are incident non-spine edges (k, i) and (i, j) for some $k \geq d - 1$ and $j \geq 1$. Then our netflow is

$$\begin{aligned} (h \text{ outflow at } i) - (h \text{ inflow at } i) &= [h(i, j) + h(e_i)] - [h(k, i) + h(e_{e-1})] \\ &= [0 + i - 1] - [0 + i - 2] \\ &= 1 \\ &= w_i. \end{aligned}$$

At vertex $n + 1$, there are two incoming non-spine edges $(n, n + 1)$ and $(k, n + 1)$ for some $k \geq d - 1$. Then our netflow is

$$\begin{aligned} (h \text{ outflow at } n + 1) - (h \text{ inflow at } n + 1) &= 0 - [h(k, n + 1) + h(n, n + 1) + h(e_n)] \\ &= -(0 + 0 + n - 1) \\ &= -(n - 1) \\ &= w_{n+1}. \end{aligned}$$

We know the leaf associated to h cannot be in the image of the injection since the flow along the spine between vertices $b + 1$ and d is 0, and the injection puts a nonzero amount of flow on the spine through the rule $g(e_i) + g(a, d) - g(b, b + 1)$ for e_i where $b + 1 \leq i \leq d - 1$.

We argue the forward direction of our if and only if by contrapositive. Suppose $f_{G', L_b}(a, b + 1) \leq f_{G', L_b}(b, d)$. We build an injection from the leaves of f_{G', L_b} into the leaves of $\varphi(f_{G', L_b})$. This map and argument is nearly identical to the backward direction. This time, our injection goes from the leaves of f_{G', L_b} into the leaves of $\varphi(f_{G', L_b})$. Given a leaf f of the level $b + 1$ node f_{G', L_b} in $T_{G'}(\mathbf{w}_n)$, we map it to the leaf g of $T_G(\mathbf{w}_n)$, where g is defined as

$$g(i, j) = \begin{cases} f(a, b + 1), & (i, j) = (a, d) \\ f(b, d), & (i, j) = (b, b + 1) \\ f(e_i) + f(b, d) - f(a, b + 1), & (i, j) = e_i, b + 1 \leq i \leq d - 1 \\ f(i, j), & \text{else} \end{cases}.$$

Once again, we verify the netflow equality condition is satisfied, check that g restricts to $\varphi(f_{G', L_b})$, and then verify that the map $f \mapsto g$ is injective. Just like in the backwards direction, the only vertices on which the netflows will change are $b + 1, \dots, d$. At vertex $b + 1$, we know the non-spine outgoing edge is consecutive, as (a, d) is already the unique overpass edge to $b + 2$ (as in Lemma 3.14). So we have

$$\begin{aligned} (g \text{ outflow at } b + 1) - (g \text{ inflow at } b + 1) &= [g(b + 1, b + 2) + g(e_{b+1})] - [g(b, b + 1) + g(e_b)] \\ &= [f(b + 1, b + 2) + f(e_{b+1}) + f(b, d) + f(a, b + 1)] \\ &\quad - [f(b, d) + f(e_b)] \\ &= [f(b + 1, b + 2) + f(e_{b+1})] - [f(a, b + 1) + f(e_b)] \\ &= (f \text{ outflow at } b + 1) - (f \text{ inflow at } b + 1) \\ &= w_{b+1}. \end{aligned}$$

If there are any vertices i such that $b + 2 \leq i \leq d - 1$, note that all four incident edges are consecutive edges, else we would violate the unique overpass condition of Lemma 3.14. So we have

$$\begin{aligned} (g \text{ outflow at } i) - (g \text{ inflow at } i) &= [g(i, i + 1) + g(e_i)] - [g(i - 1, i) + g(e_{i-1})] \\ &= [f(i, i + 1) + f(e_i) + f(b, d) - f(a, b + 1)] \\ &\quad - [f(i - 1, i) + f(e_{i-1}) + f(b, d) - f(a, b + 1)] \\ &= [f(i, i + 1) + f(e_i)] - [f(i - 1, i) + f(e_{i-1})] \\ &= (f \text{ outflow at } i) - (f \text{ inflow at } i) \\ &= w_i. \end{aligned}$$

At vertex d , we have a non-spine outgoing edge (d, j) for some j . Then our netflow is

$$\begin{aligned} (g \text{ outflow at } d) - (g \text{ inflow at } d) &= [g(d, j) + g(e_d)] - [g(a, d) + g(e_{d-1})] \\ &= [f(d, j) + f(e_d)] \\ &\quad - [f(a, b + 1) + f(e_{d-1}) + f(b, d) - f(a, b + 1)] \\ &= [f(d, j) + f(e_d)] - [f(e_{d-1}) + f(b, d)] \\ &= (f \text{ outflow at } d) - (f \text{ inflow at } d) \\ &= w_d. \end{aligned}$$

Thus g is indeed a valid flow on G . Next we verify that g agrees with $\varphi(f_{G', L_b})$ on L_b . Given an edge in G , $(i, j) \in L_b$, since f is a leaf of f_{G', L_b}

$$\begin{aligned} g(i, j) &= \begin{cases} f(a, b + 1), & (i, j) = (a, d) \\ f(b, d), & (i, j) = (b, b + 1) \\ f(i, j), & \text{else} \end{cases} \\ &= \begin{cases} f_{G', L_b}(a, b + 1), & (i, j) = (a, d) \\ f_{G', L_b}(b, d), & (i, j) = (b, b + 1) \\ f_{G', L_b}(i, j), & \text{else} \end{cases} \\ &= \varphi(f_{G', L_b})(i, j). \end{aligned}$$

Finally, we prove that this mapping is injective. Let f_1 and f_2 be (flows on G' that give) leaves in $T'_G(\mathbf{w}_n)$. Suppose that both f_1 and f_2 map to a leaf g in $T_G(\mathbf{w}_n)$. Then

$$\begin{aligned} &\begin{cases} f_1(a, b + 1), & (i, j) = (a, d) \\ f_1(b, d), & (i, j) = (b, b + 1) \\ f_1(e_i) + f_1(b, d) - f_1(b, b + 1), & (i, j) = e_i, b + 1 \leq i \leq d - 1 \\ f_1(i, j), & \text{else} \end{cases} \\ &= \begin{cases} f_2(a, b + 1), & (i, j) = (a, d) \\ f_2(b, d), & (i, j) = (b, b + 1) \\ f_2(e_i) + f_2(b, d) - f_2(a, b + 1), & (i, j) = e_i, b + 1 \leq i \leq d - 1 \\ f_2(i, j), & \text{else} \end{cases} \end{aligned}$$

We can immediately conclude that $f_1(a, b + 1) = f_2(a, b + 1)$ and $f_1(b, d) = f_2(b, d)$, and that $f_1(i, j) = f_2(i, j)$ for every edge that isn't a spine edge between $b + 1 \leq i \leq d - 1$. Then it follows that $f_1(e_i) = f_2(e_i)$ for every spine edge e_i such that $b + 1 \leq i \leq d - 1$. Hence $f_1 = f_2$, so our map is indeed injective. \square

Having established a relationship between the number of leaves below level $b + 1$ nodes in G and G' , our next goal is to relate the total number of leaves in each of the flow decomposition trees.

Definition 4.12. Assume $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . We call a level $b + 1$ node f_{G',L_b} in $T_{G'}(\mathbf{w}_n)$ that satisfies the condition of Lemma 4.11 a *bad* node. We denote the set of bad level $b + 1$ nodes of $T_{G'}(\mathbf{w}_n)$ by $V_{b+1}^{>}(G')$. Otherwise, we say f_{G',L_b} is *good*. We denote the set of good level $b + 1$ nodes of $T_{G'}(\mathbf{w}_n)$ by $V_{b+1}^{\leq}(G')$.

We can use the φ bijection to apply the good and bad classifications to the level $b + 1$ nodes of $T_G(\mathbf{w}_n)$. Using this, we get that a node f_{G,L_b} in $T_G(\mathbf{w}_n)$ is bad if and only if $f_{G,L_b}(a, d) > f_{G,L_b}(b, b + 1)$.

Example 4.13. If we look at the five level $b + 1 = 4$ nodes in Figure 4.5, we see one bad node (the second node from the left). The said node has $f_{G_{11},L_3}(2, 4) = 1 > f_{G_{11},L_3}(3, 5) = 0$. When looking at the corresponding level $b + 1 = 4$ nodes in Figure 4.6, we see that every node has at least as many leaves as its corresponding level $b + 1$ node in $T_{G_{11}}$, except for the second node from the left corresponding to the aforementioned bad level 4 node in Figure 4.5.

At this point, we have partitioned the level $b + 1$ nodes in $T_{G'}(\mathbf{w}_n)$ into nodes that have too few leaves when compared to $T_G(\mathbf{w}_n)$ (bad) and nodes that have the same or more leaves when compared to $T_G(\mathbf{w}_n)$ (good). Next, we control the number of bad nodes relative to the good nodes.

Lemma 4.14. Assume $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . For each bad node f_{G',L_b} in $T_{G'}(\mathbf{w}_n)$, there exists a unique good node g_{G',L_b} such that for every edge $(i, j) \in L_b$,

$$g_{G',L_b}(i, j) = \begin{cases} f_{G',L_b}(b, d), & (i, j) = (a, b + 1) \\ f_{G',L_b}(a, b + 1), & (i, j) = (b, d) \\ f_{G',L_b}(e_i) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d), & (i, j) = e_i, a \leq i \leq b - 1 \\ f_{G',L_b}(i, j), & \text{else} \end{cases} .$$

This yields an injection ψ from the bad nodes of $T_{G'}(\mathbf{w}_n)$ into the good nodes of $T_{G'}(\mathbf{w}_n)$, where $\psi(f_{G',L_b}) := g_{G',L_b}$

Proof. Let f_{G',L_b} be a bad node in $T_{G'}(\mathbf{w}_n)$. Define $\psi(f_{G',L_b}) := g_{G',L_b}$, where

$$g_{G',L_b}(i, j) = \begin{cases} f_{G',L_b}(b, d), & (i, j) = (a, b + 1) \\ f_{G',L_b}(a, b + 1), & (i, j) = (b, d) \\ f_{G',L_b}(e_i) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d), & (i, j) = e_i, a \leq i \leq b - 1 \\ f_{G',L_b}(i, j), & \text{else} \end{cases} .$$

To show g_{G',L_b} is a valid level $b + 1$ node, we need only check the netflow at vertices in $[b]$ where the in- or outflow has changed: a, \dots, b . At vertex a , there is an incoming edge (i, a) for some i . Then our netflow at a is

$$\begin{aligned} & (g_{G',L_b} \text{ outflow at } a) - (g_{G',L_b} \text{ inflow at } a) \\ &= [g_{G',L_b}(a, b + 1) + g_{G',L_b}(e_a)] - [g_{G',L_b}(i, a) + g_{G',L_b}(e_{a-1})] \\ &= [f_{G',L_b}(b, d) + f_{G',L_b}(e_a) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d)] \\ &\quad - [f_{G',L_b}(i, a) + f_{G',L_b}(e_{a-1})] \\ &= [f_{G',L_b}(a, b + 1) + f_{G',L_b}(e_a)] - [f_{G',L_b}(i, a) + f_{G',L_b}(e_{a-1})] \\ &= (f_{G',L_b} \text{ outflow at } a) - (f_{G',L_b} \text{ inflow at } a) \\ &= w_a. \end{aligned}$$

If there are any vertices i such that $a + 1 \leq i \leq b - 1$, all incident edges must be consecutive, else we would violate Lemma 3.14. Then our netflow at i is

$$\begin{aligned} & (g_{G',L_b} \text{ outflow at } i) - (g_{G',L_b} \text{ inflow at } i) \\ &= [g_{G',L_b}(i, i + 1) + g_{G',L_b}(e_i)] - [g_{G',L_b}(i - 1, i) + g_{G',L_b}(e_{i-1})] \\ &= [f_{G',L_b}(i, i + 1) + f_{G',L_b}(e_i) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d)] \\ &\quad - [f_{G',L_b}(i - 1, i) + f_{G',L_b}(e_{i-1}) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d)] \\ &= [f_{G',L_b}(i, i + 1) + f_{G',L_b}(e_i)] - [f_{G',L_b}(i - 1, i) + f_{G',L_b}(e_{i-1})] \\ &= (f_{G',L_b} \text{ outflow at } i) - (f_{G',L_b} \text{ inflow at } i) \\ &= w_i. \end{aligned}$$

At vertex b , we have some incoming edge (i, b) . Our netflow is

$$\begin{aligned} & (g_{G',L_b} \text{ outflow at } b) - (g_{G',L_b} \text{ inflow at } b) \\ &= [g_{G',L_b}(b, d) + g_{G',L_b}(e_b)] - [g_{G',L_b}(i, b) + g_{G',L_b}(e_{b-1})] \\ &= [f_{G',L_b}(a, b + 1) + f_{G',L_b}(e_b)] \\ &\quad - [f_{G',L_b}(i, b) + f_{G',L_b}(e_{b-1}) + f_{G',L_b}(a, b + 1) - f_{G',L_b}(b, d)] \\ &= [f_{G',L_b}(b, d) + f_{G',L_b}(e_b)] - [f_{G',L_b}(i, b) + f_{G',L_b}(e_{b-1})] \\ &= (f_{G',L_b} \text{ outflow at } b) - (f_{G',L_b} \text{ inflow at } b) \\ &= w_b. \end{aligned}$$

Thus g_{G',L_b} is valid on $[b]$.

To verify that g_{G',L_b} is a good node as in Lemma 4.11, observe that

$$g_{G',L_b}(a, b + 1) = f_{G',L_b}(b, d) \leq f_{G',L_b}(a, b + 1) = g_{G',L_b}(b, d).$$

Finally, we check that ψ is injective. Let f_1 and f_2 be two bad level $b + 1$ nodes in $T_{G'}(\mathbf{w}_n)$. Suppose $\psi(f_1) = \psi(f_2)$. Then we can immediately conclude that $f_1(a, b + 1) = f_2(a, b + 1)$, $f_1(b, d) = f_2(b, d)$, and $f_1(i, j) = f_2(i, j)$ for any edge in L_b that isn't $(a, b + 1)$, (b, d) or a spine edge between a and $b - 1$. This in turn implies that $f_1(i, j) = f_2(i, j)$ on the spine edges between a and $b - 1$. Thus $f_1(i, j) = f_2(i, j)$ for all $(i, j) \in L_b$. \square

Example 4.15. Consider the single bad node f_{G_{11},L_3} in Figure 4.5 (the second-from-left level 4 node), pictured again here.



The bad node from f_{G_{11},L_3} .

We see that there is a unique level $b + 1 = 4$ node g_{G_{11},L_3} in $T_{G_{11}}(\mathbf{w}_4)$ with

$$\begin{aligned} g_{G_{11},L_3}(2, 4) &= 0 = f_{G_{11},L_3}(3, 5), \\ g_{G_{11},L_3}(3, 5) &= 1 = f_{G_{11},L_3}(2, 4), \\ g_{G_{11},L_3}(e_2) &= 1 = f_{G_{11},L_3}(e_2) + f_{G_{11},L_3}(3, 5) - f_{G_{11},L_3}(2, 4), \end{aligned}$$

and the same flows on all other edges. This is the second-from-right level 4 node in Figure 4.5, also pictured again here.



The corresponding good node.

Definition 4.16. We use the level $b + 1$ node bijection ψ for $T_{G'}(\mathbf{w}_n)$ to define a version of ψ for the bad nodes of $T_G(\mathbf{w}_n)$. More specifically, define $\psi(f_{G,L_b}) := \varphi(\psi(f_{G',L_b}))$ where $\varphi(f_{G',L_b}) = f_{G,L_b}$.

Lemma 4.17. Assume $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . The construction of ψ for level $b + 1$ nodes in $T_G(\mathbf{w}_n)$ is an injection.

Proof. This follows from the fact that φ is a bijection and ψ as defined for $T_{G'}(\mathbf{w}_n)$ is an injection. \square

Thus, every bad level $b+1$ node in $T_{G'}(\mathbf{w}_n)$ corresponds with a unique good node in $T_G(\mathbf{w}_n)$, and each of these nodes has a bijective corresponding node in $T_{G'}(\mathbf{w}_n)$. We next prove our most crucial intermediate result, pertaining to the number of leaves of these two pairs of nodes.

Lemma 4.18. Assume $G \in \mathcal{F}_{n,3}$ has nested edges that are interchanged to obtain G' . Given a bad level $b + 1$ node f_{G',L_b} in $T_{G'}(\mathbf{w}_n)$,

$$l(f_{G',L_b}) = l(\varphi(\psi(f_{G',L_b}))) \quad \text{and} \quad l(\psi(f_{G',L_b})) = l(\varphi(f_{G',L_b})).$$

Further, the four nodes comprise two pairs of nodes, one pair in $T_{G'}(\mathbf{w}_n)$ and another in $T_G(\mathbf{w}_n)$, and each pair has the same total number of leaves. That is,

$$l(f_{G',L_b}) + l(\psi(f_{G',L_b})) = l(\varphi(f_{G',L_b})) + l(\varphi(\psi(f_{G',L_b}))).$$

Proof. Let f_{G',L_b} be a bad node in $T_{G'}(\mathbf{w}_n)$. Our proof is illuminated by Figures 4.7 and 4.8.

Note that in Figures 4.7 and 4.8 the solid lines represent edges whose flows have been decided by the partial flow and dotted lines represent edges whose flows will be decided later in the flow decomposition tree. (This is different than what dotted edges represented in Figures 4.5 and 4.6.)

In Figure 4.7, we see that the nodes f_{G',L_b} and $\varphi(\psi(f_{G',L_b}))$ will have the same number of leaves because they have the same flows into vertices $b + 1$ (by Proposition 3.16) and d , and identical edges remaining on which to direct those flows. The same argument applies in Figure 4.8 to $\varphi(f_{G',L_b})$ and $\psi(f_{G',L_b})$. \square

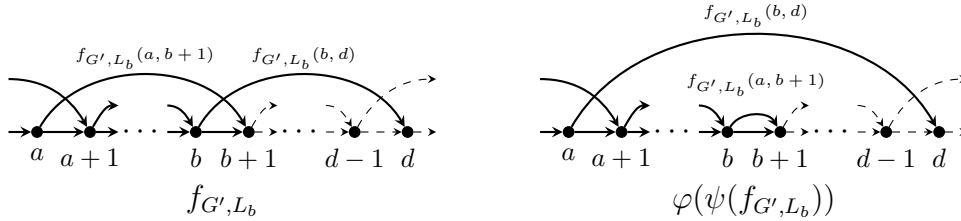


Figure 4.7: Comparing nodes f_{G',L_b} and $\varphi(\psi(f_{G',L_b}))$.

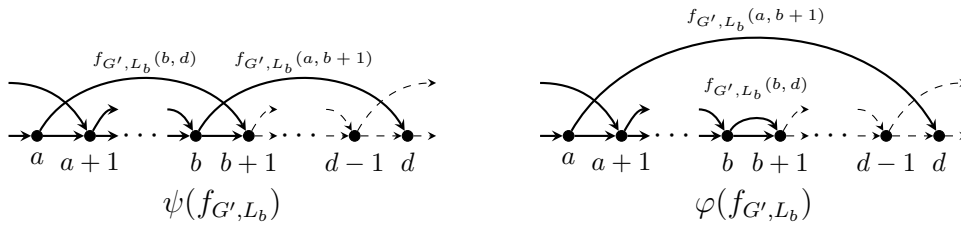


Figure 4.8: Comparing nodes $\psi(f_{G',L_b})$ and $\varphi(f_{G',L_b})$.

Example 4.19. Looking again at the bad node f_{G_{11},L_3} in Figure 4.5, we found its unique good node pair g_{G_{11},L_3} in Example 4.15. If we then find the level $b + 1$ bijective correspondents for these two nodes in $T_{G_{10}}(\mathbf{w}_4)$, f_{G_{10},L_3} and g_{G_{10},L_3} respectively, we see that

$$l(f_{G_{11},L_3}) = 4 = l(g_{G_{10},L_3}) \quad \text{and} \quad l(g_{G_{11},L_3}) = 3 = l(f_{G_{10},L_3}).$$

We are now ready to state our main result of this section.

Theorem 4.20. Let $G \in \mathcal{F}_{n,3}$ and suppose G' is obtained by interchanging a pair of nested edges in G . Then

$$\text{vol } \mathcal{F}_1(G') \leq \text{vol } \mathcal{F}_1(G).$$

Proof. We have established that for any DAG $G \in \mathcal{F}_{n,3}$,

$$\text{vol } \mathcal{F}_1(G) = K_G(\mathbf{w}_n) = \# \text{ of leaves in } T_G(\mathbf{w}_n).$$

Given G and G' as in the theorem statement, we compare the number of leaves in their respective flow decomposition trees by summing over the leaves of the level $b + 1$ nodes. Starting in $T_{G'}(\mathbf{w}_n)$, we have

$$\begin{aligned}
& \# \text{ of leaves in } T_{G'}(\mathbf{w}_n) \\
&= \sum_{f_{G',L_b} \in V_{b+1}^{\leq}(G')} l(f_{G',L_b}) \\
&= \sum_{f_{G',L_b} \in V_{b+1}^{\geq}(G')} [l(f_{G',L_b}) + l(\psi(f_{G',L_b}))] + \sum_{f_{G',L_b} \in V_{b+1}^{\leq}(G') \setminus \psi(V_{b+1}^{\geq}(G'))} l(f_{G',L_b}) \\
&= \sum_{f_{G',L_b} \in V_{b+1}^{\geq}(G')} [l(\varphi(f_{G',L_b})) + l(\varphi(\psi(f_{G',L_b})))] + \sum_{f_{G',L_b} \in V_{b+1}^{\leq}(G') \setminus \psi(V_{b+1}^{\geq}(G'))} l(f_{G',L_b}) \\
&\leq \sum_{f_{G',L_b} \in V_{b+1}^{\geq}(G')} [l(\varphi(f_{G',L_b})) + l(\varphi(\psi(f_{G',L_b})))] + \sum_{f_{G',L_b} \in V_{b+1}^{\leq}(G') \setminus \psi(V_{b+1}^{\geq}(G'))} l(\varphi(f_{G',L_b})) \\
&= \sum_{f_{G,L_b} \in V_{b+1}(G)} l(f_{G,L_b}) \\
&= \# \text{ of leaves in } T_G(\mathbf{w}_n). \quad \square
\end{aligned}$$

Corollary 4.21. *The function $\text{vol} : \mathcal{F}_{n,3} \rightarrow \mathbb{Z}_{\geq 0}$, which maps a DAG to the volume of its flow polytope, is order-reversing with respect to the Boolean (equivalently, interchange) partial order on $\mathcal{F}_{n,3}$.*

5. Linear Extensions

In this section we establish a connection between flow polytope volumes for DAGs in $\mathcal{F}_{n,3}$ and volumes of a related family of order polytopes; order polytopes are lattice polytopes defined by finite posets [Sta86]. Mészáros, Morales, and Striker [MMS19] established a beautiful equivalence between flow polytopes and order polytopes, and we use their equivalences to translate our results from Section 4 about flow polytopes to results about order polytopes. In particular, in Corollary 5.22 we restate Theorem 4.20 as a statement about posets and their linear extensions. We briefly review the background and relevant results of [MMS19].

Definition 5.1. Given a finite poset P , define the *order polytope* to be

$$\mathcal{O}(P) := \{x \in [0, 1]^{|P|} : x_i \leq x_j \text{ if } i \leq_P j\}.$$

Definition 5.2. A DAG G is *strongly planar* if G is a planar graph with a planar realization such that the following two conditions hold. First, for every directed edge (i, j) of G , the x -coordinate of i is strictly less than the x -coordinate of j . Second, each edge (i, j) is embedded in the plane as the graph of a piecewise differentiable function of x .

Definition 5.3. A poset P is *strongly planar* if the Hasse diagram of P is a planar graph with a planar realization such that the following two conditions hold. First, for every directed edge (i, j)

in the Hasse diagram, the y -coordinate of i is strictly less than the y -coordinate of j . Second, each edge (i, j) is embedded in the plane as the graph of a piecewise differentiable function of y .

Definition 5.4. Given a strongly planar DAG G , we define the *truncated dual graph* G^* to be a strongly planar realization of the dual graph of G with the vertex corresponding to the infinite region deleted. To define an orientation on the edges of G^* we leverage the fact that each edge can be drawn intersecting transversely to the corresponding edge of G . Specifically, the edges of G^* shall be directed a quarter turn counterclockwise from the edge orientation in G . When G is strongly planar, the directed graph G^* can be read as a Hasse diagram of a strongly planar poset where directed edge (a, b) in G^* is interpreted as a covering relation $a \leq b$. We denote the resulting poset by P_G . Similarly, given a strongly planar poset P , let H be the Hasse diagram of $P \cup \{\hat{0}, \hat{1}\}$. Reading H as a graph directed from $\hat{0}$ to $\hat{1}$, we define G_P to be the strongly planar DAG obtained as the strongly planar dual of H , with the infinite region divided into a left and right side. We obtain an orientation on the edges of G by rotating the orientation the edges of H a quarter turn clockwise.

Example 5.5. Refer to Figure 5.1 for the truncated duals of the DAGs in $\mathcal{F}_{4,3}$.

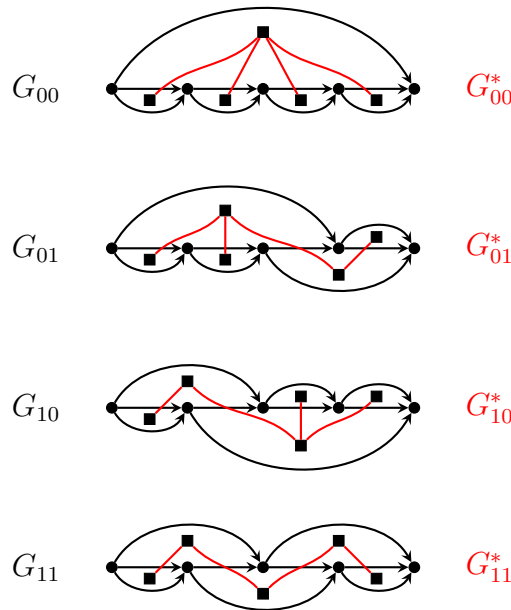


Figure 5.1: Truncated duals G^* for $G \in \mathcal{F}_{4,3}$.

Remark 5.6. A few conditions in the definitions of Mészáros, Morales, and Striker [MMS19, Section 3] have been strengthened here in order for their proofs to hold in general.

- (i) The definition of orientation of edges in the dual graphs at the beginning of [MMS19, Section 3.2] has been strengthened in Definition 5.4;

- (ii) The restriction to ‘planar’ DAGs at the beginning of [MMS19, Section 3.2] has been strengthened in Definition 5.2, and the terminology has been changed to *strongly planar* to reflect the fact that it is a stronger condition than planarity, as well as to be consistent with the analogous definition for posets;
- (iii) The restriction to strongly planar posets at the beginning of [MMS19, Section 3.3] has been strengthened in Definition 5.3.

In the form originally given in [MMS19], the strongly planar definitions are not sufficient to support the proofs they present due to issues of duality not being well-defined. The strengthening of the dual graph edge orientation definition is necessary because [MMS19] defines this orientation using y -coordinates of the dual vertices, which results in ambiguous dual edge orientations in some cases.

To see why the strengthened Definition 5.2 is necessary, consider the graph G in Figure 5.2. It is possible to find a planar embedding that satisfies the requirements of [MMS19], and so we compute the truncated dual G^* using Definition 5.4, as seen in Figure 5.3. However, this results in a truncated dual containing a directed cycle, and thus there is no way to interpret G^* as a valid Hasse diagram. To see that all planar embeddings of G result in truncated duals with directed cycles, note that the condition that vertex x -coordinates must increase with their labels forces two left-to-right paths from vertex 1 to 6: $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ and $1 \rightarrow 3 \rightarrow 5 \rightarrow 6$. Attaching these two paths together at vertices 1 and 6 creates an undirected cycle enclosing a face $1, 3, 5, 6, 4, 2, 1$. Drawing the duplicate edges out of 1 and into 6 is not a problem, but one can check that all non-crossing configurations $(2, 5)$ and $(3, 4)$ result in truncated duals with cycles. Such a graph is precluded by Definition 5.2 as it has no planar embeddings where all the edges are functions of x .

A similar argument can be used to demonstrate the necessity of Definition 5.3 with the poset P having Hasse diagram obtained by taking the DAG in Figure 5.2, rotating it a quarter-turn counter-clockwise, and removing one of each multiple edge. It can be verified in a fashion similar to the above DAG case that there is no strongly planar embedding of $P \cup \{\hat{0}, \hat{1}\}$, and every planar embedding of $P \cup \{\hat{0}, \hat{1}\}$ results in a dual with directed cycles.

With these strengthened definitions in place, the necessary dual graphs are well-defined and the proofs in [MMS19] go through, for example for the following theorem.

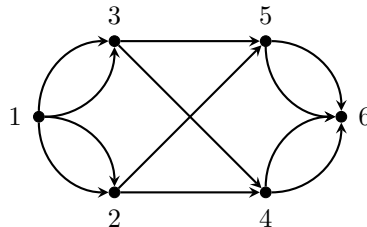


Figure 5.2: The graph G ; a directed $K_{2,2}$ with a unique source and sink added, free of idle edges.

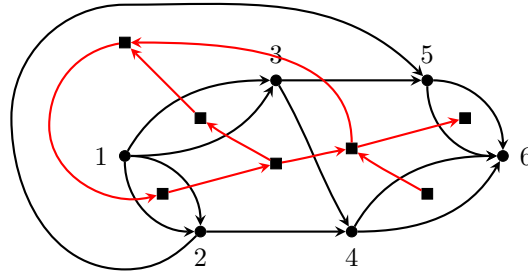


Figure 5.3: Drawing the truncated dual G^* .

Theorem 5.7 (Mészáros, Morales, and Striker [MMS19]). *Given a strongly planar DAG G , the flow polytope $\mathcal{F}_1(G)$ is integrally equivalent to the order polytope of the strongly planar poset dual to G . Conversely, given a strongly planar poset P , the order polytope $\mathcal{O}(P)$ is integrally equivalent to the flow polytope of the strongly planar DAG G_P .*

As a corollary, we see that the volume of a flow polytope for a strongly planar DAG is equal to the volume of the order polytope for the strongly planar dual, and vice versa. Further, we have the following result due to Stanley [Sta86].

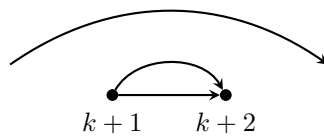
Theorem 5.8 (Stanley [Sta86]). *For a poset P , the number of linear extensions of P is counted by the volume of the order polytope $\mathcal{O}(P)$. That is*

$$e(P) = \text{vol } \mathcal{O}(P).$$

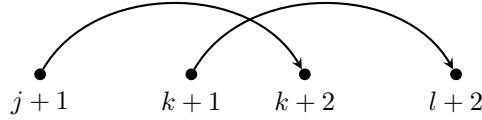
We show below that DAGs in $\mathcal{F}_{n,3}$ are strongly planar, so we can apply the correspondence given by Theorem 5.7 and Theorem 5.8 to our volume results from Section 4. Our flow polytope volume inequalities from Theorem 4.20 thus extend to linear extension inequalities for the posets arising as truncated duals. We begin by proving an observation about drawing DAGs in $\mathcal{F}_{n,3}$ with non-planar embeddings.

Proposition 5.9. *Let $G \in \mathcal{F}_{n,3}$. If we draw the vertices in a row in so that their labels increase from left to right and draw non-spine edges as upward-facing semicircles (directed from smaller vertex labels to larger), G has an edge crossing between vertices $k + 1$ and $k + 2$ if and only if its corresponding binary string $\beta = b_1 \cdots b_{n-2}$ has a 1 in position k .*

Proof. Suppose $b_k = 0$. Then by the bijection λ in Theorem 3.4, the DAG G has edges e_{k+1} and $(k + 1, k + 2)$. Further, Lemma 3.14 ensures that only one other edge can pass over vertices $k + 1$ and $k + 2$. But since the consecutive edges e_{k+1} and $(k + 1, k + 2)$ fill all out-going stubs of $k + 1$ and all in-going stubs of $k + 2$, the unique edges passing over $k + 1$ and $k + 2$ are one in the same. Drawing these three edges as upper semicircular arcs does not create any crossings.



Conversely, suppose $b_k = 1$. Denote the indices of the adjacent 1s of β with j and l , where $j := 0$ if k is the first 1 in β and $l := n - 1$ if k is the last 1 in β . So we have $j < k < l$. Then the bijection λ in Theorem 3.4 gives edges $(j + 1, k + 2)$, $(k + 1, l + 2)$. Note that $j < k < l$ ensures that neither $(j + 1, k + 2)$ nor $(k + 1, l + 2)$ can be a consecutive edge. These two edges cross between $k + 1$ and $k + 2$ if drawn as upper semicircular arcs.



□

Proposition 5.10. *Let $G \in \mathcal{F}_{n,3}$. Then G is strongly planar.*

Proof. Let $G \in \mathcal{F}_{n,3}$. Then G can be drawn with the vertices in a horizontal row with increasing labels, and the edges so that each spine edge e_i is realized as a horizontal line from i to $i + 1$ and the non-spine edges are realized as upward-facing semi-circular arcs. This drawing convention ensures that every directed edge (i, j) of G has the x -coordinate of i strictly less than the x -coordinate of j , and that each edge is embedded in the plane as the graph of a piecewise differentiable function of x . Now we seek to transform some of the edges in a way that preserves these properties, but gives a planar embedding of G , in order to satisfy Definition 5.2.

We claim we can decompose $E(G)$ into three paths from vertex 1 to vertex $n + 1$. This is useful because a path in DAG embedded so that every edge (i, j) has the x -coordinate of i strictly less than the x -coordinate of j cannot intersect itself. The first path is the spine: (e_1, \dots, e_n) . The next path begins with the edge $(1, 2)$, and the final path begins with the other edge leaving vertex 1, $(1, j)$ say. For convenience, we label the non-spine path beginning with $(1, 2)$ as A and the non-spine path beginning with $(1, j)$ as B . To prove that paths A and B do indeed end at vertex $n + 1$, consider the subgraph of G obtained by deleting the spine. Every interior vertex in the subgraph has in- and out-degree 1. Hence, each interior vertex i can be in at most one path, and the path can continue past vertex i .

Now, if we reflect each edge in path A below the spine, path A does not intersect the spine or path B , since path B lies entirely above the spine. For the same reason, path B does not intersect the spine. Hence we have a planar embedding of G . Further, the reflection of the edges in path A preserves the properties necessary for this embedding to be strongly planar. □

Example 5.11. Figure 5.1 provides a strongly planar embedding for each $G \in \mathcal{F}_{4,3}$.

Example 5.12. Figure 5.4 provides a strongly planar embedding of G_{1011} .

Our next goal is to identify the posets that are realizable as truncated duals of DAGs in $\mathcal{F}_{n,3}$.

Proposition 5.13. *Let $G \in \mathcal{F}_{n,3}$. Then the truncated dual G^* is a bipartite non-crossing tree on $n + 1$ vertices.*

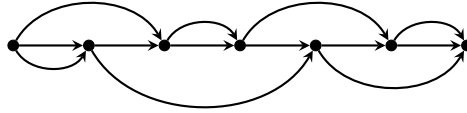


Figure 5.4: A strongly planar embedding for G_{1011} .

Proof. Let $G \in \mathcal{F}_{n,3}$. Let \widehat{G} denote the proper dual of G , and let G^* denote the truncated dual. Recall that G^* is the subgraph of \widehat{G} obtained by deleting the vertex of \widehat{G} associated with the infinite face and all its incident edges. By definition, \widehat{G} is planar, and thus G^* is non-crossing. The planar embedding of G described in Proposition 5.10 partitions the vertices of G^* into those between the spine and path A , and those between the spine and path B . Hence G^* is bipartite.

We use the Euler characteristic formula to verify the vertex count for G^* , and show that it has the requisite number of edges to be a tree. By Proposition 2.6, we know G has $n + 1$ vertices and $2n + 1$ edges. Then by the Euler characteristic formula, we have that \widehat{G} has

$$2 - |V(G)| + |E(G)| = 2 - (n + 1) + (2n + 1) = n + 2$$

vertices. Hence G^* has $n + 1$ vertices. To count the edges in G^* , note that the number of edges in G^* is equal to the number of edges in \widehat{G} that are not incident to the vertex associated to the infinite face. Using the planar embedding described in Proposition 5.10, the edges in \widehat{G} that are incident to the vertex associated to the infinite face correspond to the edges in G that comprise paths A and B . Since the spine and paths A and B partition $E(G)$, we have

$$\begin{aligned} |E(G^*)| &= |E(G)| - (\# \text{ edges in path } A) - (\# \text{ edges in path } B) \\ &= (\# \text{ edges in spine of } G) \\ &= n. \end{aligned}$$

Thus G^* is a graph with $n + 1$ vertices and n edges, so it is a tree. □

Note that the drawing conventions described in the proof of Proposition 5.10 yield a bipartite non-crossing tree whose distinguished sets are separated by a horizontal line (the spine). When we interpret one of these truncated duals as a Hasse diagram, we get a graded poset with two ranks (rank 0 elements in the distinguished set below the spine and rank 1 elements in the distinguished set above the spine). With this in mind, we use the terms “bipartite non-crossing trees” and “rank 1 posets” interchangeably throughout the remainder of the section, though in general not all rank 1 posets are bipartite non-crossing trees.

Next, we characterize the bipartite non-crossing trees on $n + 1$ vertices that are realizable as duals of DAGs from $\mathcal{F}_{n,3}$, using the drawing convention described in the proof of Proposition 5.10. We use observations due to Mészáros and Morales [MM19], in whose work bipartite non-crossing trees arise from a completely different property of DAGs.

Proposition 5.14. *There are 2^{n-2} bipartite non-crossing trees on n vertices.*

Proof. Bipartite non-crossing trees with l left vertices and r right vertices are in bijection with weak compositions of $r - 1$ into l parts [MM19]. The bijection maps a bipartite non-crossing tree with left vertices $\{x_1, \dots, x_l\}$ and r right vertices to the weak composition $(\deg x_1 - 1, \dots, \deg x_l - 1)$ of $r - 1$. It is known that the numbers of such weak compositions is $\binom{l+r-2}{l-1}$. So if $n = l + r$, the total number of bipartite non-crossing trees with n vertices is

$$\sum_{l=1}^{n-1} \binom{n-2}{l-1} = \sum_{i=0}^{n-2} \binom{n-2}{i} = 2^{n-2}. \quad \square$$

Proposition 5.15. *Of the 2^{n-1} bipartite non-crossing trees on $n+1$ vertices, 2^{n-2} are realizable as truncated duals of DAGs in $\mathcal{F}_{n,3}$ when using the planar embedding convention described in Proposition 5.10. Specifically, the bipartite non-crossing trees that are realizable as truncated duals are those whose left-most vertex of rank 0 has degree 1.*

Proof. Let P be a bipartite non-crossing tree on $n+1$ vertices. Label its rank 0 elements as $\{x_1, \dots, x_l\}$ and its rank 1 elements as $\{y_1, \dots, y_r\}$, with subscripts increasing from left to right. Then the condition we prove is necessary and sufficient for realizability is $\deg x_1 = 1$.

Suppose P is realized as the truncated dual of a DAG $G \in \mathcal{F}_{n,3}$, where G is drawn using the conventions in Proposition 5.10. Note that G has the edges $e_1, e_2, (1, 2), (2, j)$ and $(1, k)$, where $j, k \geq 3$. Using the edge partition from the proof of Proposition 5.10, this means $(1, 2)$ and $(2, j)$ are in path A and $(1, k)$ is in path B . Then our drawing conventions give us the following regions in G :

- (i) The region bounded by e_1 and $(1, 2)$, below the spine;
- (ii) The region bounded by e_2, \dots, e_{k-1} and $(2, j)$, below the spine; and
- (iii) The region bounded by e_1, \dots, e_{j-1} and $(1, j)$, above the spine.

Taking the truncated dual of G gives a vertex for each region. The first two regions are rank 0, so they are labelled x_1 and x_2 . The third region is rank 1, so it is labelled y_1 . Since the first two regions in G share the edge e_1 , we have an edge (x_1, y_1) in P . Since the second two regions in G share the edge e_2 , we have an edge (x_2, y_1) in P . Then P cannot have any edge of the form (x_1, y_i) for $i \geq 2$, as it would cross the edge (x_2, y_1) or cross into the unbounded region, which does not correspond to a vertex in the truncated dual. Thus $\deg x_1 = 1$.

Conversely, suppose $\deg x_1 = 1$. We construct a DAG $G \in \mathcal{F}_{n,3}$ so that $G^* = P$. Begin with $n+1$ vertices and spine edges e_1, \dots, e_n . Associate each relation in P with a spine edge, going from left to right. This is a well-defined association since P is non-crossing. For each rank 0 element x_i in P , let e_j and e_k denote the spine edges associated to the left-most and right-most relations of x_i . Then draw the edge $(j, k+1)$ in G as a downward-facing semicircular arc. For x_1 , draw $(1, 2)$ and for the right-most rank 0 vertex x_l , draw $(j, n+1)$. These edges will form the path A as described in Proposition 5.10. Then for each rank 1 element y_i in P , let e_j

and e_k denote the spine edges associated to the left-most and right-most relations of y_i . Then draw the edge $(j, k + 1)$ in G as a upward-facing semicircular arc. For y_1 , draw $(1, k + 1)$ and for the right-most rank 1 vertex y_r , draw $(j, n + 1)$. These edges will form the path B as described in Proposition 5.10. By construction, we have $G^* = P$.

Given a bipartite non-crossing tree on $n + 1$ vertices where $\deg x_1 = 1$, delete the vertex x_1 and edge (x_1, y_1) , which will give us a bipartite non-crossing tree on n vertices. Similarly, given any bipartite non-crossing tree on n vertices, we can add a single vertex x_0 and edge (x_0, y_1) to get a bipartite non-crossing tree on $n + 1$ vertices where the left-most rank 0 vertex has degree 1. Thus bipartite non-crossing trees on $n + 1$ vertices that are realizable as truncated duals of DAGs in $\mathcal{F}_{n,3}$ are in bijection with bipartite non-crossing trees on n vertices. It follows from Proposition 5.14 that there are 2^{n-2} such bipartite non-crossing trees. \square

In Section 3, we described a bijection between length $n - 2$ binary strings and DAGs in $\mathcal{F}_{n,3}$. In this section, we describe (truncated) dualization, which is a bijection between DAGs in $\mathcal{F}_{n,3}$ and bipartite non-crossing trees on $n + 1$ vertices whose left-most vertex of rank 0 has degree 1. This implies the existence of a composite bijection between length $n - 2$ binary strings and bipartite non-crossing trees on $n + 1$ vertices whose left-most vertex of rank 0 has degree 1. We now describe this bijection, and leave it as a straightforward exercise to verify that the bijection is correct.

Proposition 5.16. *Given a length $n - 2$ binary string β , we can construct a corresponding bipartite non-crossing tree T_β with $n + 1$ vertices whose left-most vertex of rank 0 has degree 1 in the following way:*

- (i) *Begin with an initial edge e_0 from rank 0 to rank 1.*
- (ii) *From the rank 1 vertex incident to e_0 , construct a path of edges where each edge in the path corresponds to a 1 in β .*
- (iii) *Attach a terminal edge after the final 1-corresponding edge.*
- (iv) *For each 0 in β , find the 1s that surround it and attach an edge between the corresponding path edges in the poset.*

Conversely, given a bipartite non-crossing tree T with $n + 1$ vertices whose left-most vertex of rank 0 has degree 1, we can construct a corresponding length $n - 2$ binary string β_T in the following way:

- (i) *Trace a path from the left-most vertex of rank 0 as far right as possible.*
- (ii) *Ignoring the first and last edge in this path, write a 1 in β_T for any remaining edges in the path.*
- (iii) *For any edges not in the path, we place a 0 in β_T between the 1s that correspond to the path edges.*

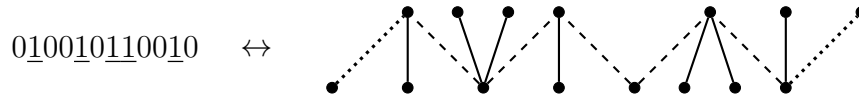


Figure 5.5: The corresponding binary string and bipartite non-crossing tree of Example 5.17.

Example 5.17. The binary string 010010110010 corresponds to the rank 1 poset in Figure 5.5. Note how the 1s correspond to the maximal path (dashed edges), without the initial and terminal edges (dotted edges), and the 0s correspond to the non-path edges (solid edges).

Next we describe how changing a 0 to a 1 affects the posets structure.

Definition 5.18. Let T be a bipartite non-crossing tree on $n + 1$ vertices whose left-most vertex of rank 0 has degree 1. Let P be the path beginning at the left-most vertex of T and ending at the right-most vertex. For an edge e_i not in the path P , we define a *flip at edge e_i* to be the operation that obtains a new bipartite non-crossing tree on $n + 1$ vertices whose left-most vertex of rank 0 has degree 1, T' , in the following way:

- (i) Split T into two connected components, the left ending with edge e_i and the right beginning after edge e_i .
- (ii) Take the (poset) dual of the right component.
- (iii) Reattach the right component to the left component to create a new tree T' where e_i is in the path beginning at the left-most vertex of T' and ending at the right-most vertex.

Example 5.19. In Figure 5.6, we perform a flip at the fifth edge from the left. First, we see our initial poset. Next, we separate it into two pieces after edge five. Then we invert the right component. Finally, we reattach the inverted piece so that edge five is in the path from left-most vertex to the right-most vertex.

The following proposition follows immediately from the bijection.

Proposition 5.20. Let β be a length $n - 2$ binary string and let T be the corresponding poset as described in Proposition 5.16. Then changing a 0 in β to a 1 is equivalent to performing a flip at the edge of T that corresponds to the 0 of β .

Example 5.21. In Figure 5.6, we see that the flip from Example 5.19 corresponds to changing b_4 from 0 to 1 in the binary string 010010110010.

Corollary 5.22. Let T be a bipartite non-crossing tree poset on $n + 1$ vertices whose left-most vertex of rank 0 has degree 1. Let P be the path beginning at the left-most vertex of T and ending at the right-most vertex. If we perform a flip at an edge of T not in P to obtain a new tree T' , then

$$e(T') \leq e(T).$$

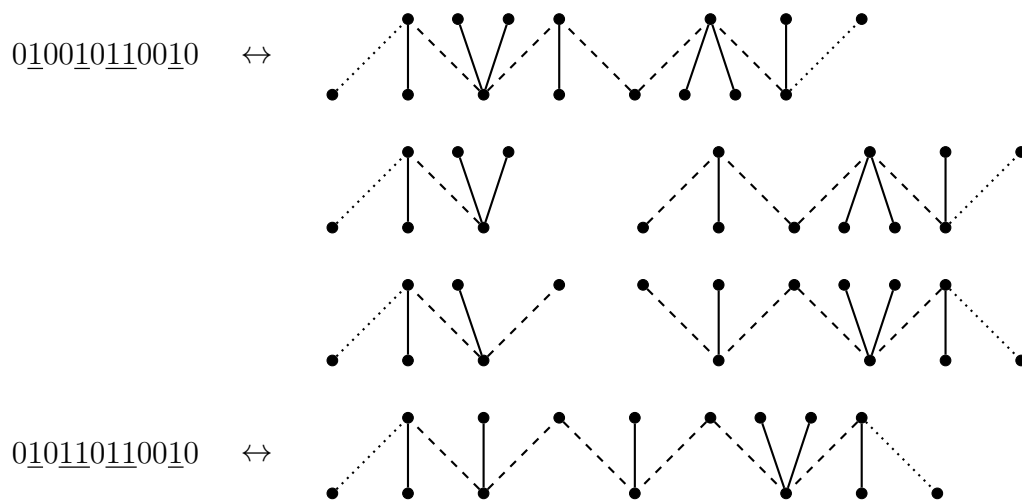


Figure 5.6: The poset operation corresponding to changing b_4 to a 1 in the binary string 010010110010.

Acknowledgements

The authors thank the anonymous referees for helpful comments.

References

- [BBB⁺24] Matias von Bell, Benjamin Braun, Kaitlin Bruegge, Derek Hanely, Zachery Peterson, Khrystyna Serhiyenko, and Martha Yip. Triangulations of flow polytopes, ample framings, and gentle algebras. *Selecta Math. (N.S.)*, 30(3):Paper No. 55, 34, 2024. doi:10.1007/s00029-024-00942-6.
- [BGDH⁺18] Carolina Benedetti, Rafael S. González D’León, Christopher R. H. Hanusa, Pamela E. Harris, Apoorva Khare, Alejandro H. Morales, and Martha Yip. The volume of the caracol polytope. In *30th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2018)*, volume 80B, pages Art. 87, 12, Hanover, United States, 2018. Séminaire Lotharingien de Combinatoire.
- [BGDH⁺19] Carolina Benedetti, Rafael S. González D’León, Christopher R. H. Hanusa, Pamela E. Harris, Apoorva Khare, Alejandro H. Morales, and Martha Yip. A combinatorial model for computing volumes of flow polytopes. *Trans. Amer. Math. Soc.*, 372(5):3369–3404, 2019. doi:10.1090/tran/7743.
- [BV08] Welleda Baldoni and Michèle Vergne. Kostant partitions functions and flow polytopes. *Transform. Groups*, 13(3-4):447–469, 2008. doi:10.1007/s00031-008-9019-8.

- [DHMR23] William T. Dugan, Maura Hegarty, Alejandro H. Morales, and Annie Raymond. Generalized Pitman-Stanley flow polytopes. In *35th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2023)*, volume 89B, pages Art. 80, 12, UC Davis, United States, 2023. Séminaire Lotharingien de Combinatoire.
- [DKK12] Vladimir I. Danilov, Alexander V. Karzanov, and Gleb A. Koshevoy. Coherent fans in the space of flows in framed graphs. In *24th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2012)*, volume AR of *Discrete Math. Theor. Comput. Sci. Proc.*, pages 481–490. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2012.
- [FLNU18] Bailey K. Fosdick, Daniel B. Larremore, Joel Nishimura, and Johan Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60(2):315–355, 2018. doi:10.1137/16M1087175.
- [GDHMY23] Rafael S. González D’León, Christopher R. H. Hanusa, Alejandro H. Morales, and Martha Yip. Column-convex matrices, G -cyclic orders, and flow polytopes. *Discrete Comput. Geom.*, 70(4):1593–1631, 2023. doi:10.1007/s00454-023-00518-9.
- [KMS21] Kabir Kapoor, Karola Mészáros, and Linus Setiabrata. Counting integer points of flow polytopes. *Discrete Comput. Geom.*, 66(2):723–736, 2021. doi:10.1007/s00454-021-00289-1.
- [KN09] Brian Karrer and M. E. J. Newman. Random graph models for directed acyclic networks. *Physical Review E*, 80(4), 2009. doi:10.1103/PhysRevE.80.046110.
- [Lid84] B. V. Lidskiĭ. The Kostant function of the system of roots A_n . *Funktsional. Anal. i Prilozhen.*, 18(1):76–77, 1984. doi:10.1007/bf01076370.
- [LMM19] Ricky Ini Liu, Alejandro H. Morales, and Karola Mészáros. Flow polytopes and the space of diagonal harmonics. *Canad. J. Math.*, 71(6):1495–1521, 2019. doi:10.4153/cjm-2018-007-3.
- [LMSD19] Ricky I. Liu, Karola Mészáros, and Avery St. Dizier. Gelfand-Tsetlin polytopes: a story of flow and order polytopes. *SIAM J. Discrete Math.*, 33(4):2394–2415, 2019. doi:10.1137/19M1251242.
- [MM19] Karola Mészáros and Alejandro H. Morales. Volumes and Ehrhart polynomials of flow polytopes. *Math. Z.*, 293(3-4):1369–1401, 2019. doi:10.1007/s00209-019-02283-z.
- [MMS19] Karola Mészáros, Alejandro H. Morales, and Jessica Striker. On flow polytopes, order polytopes, and certain faces of the alternating sign matrix polytope. *Discrete Comput. Geom.*, 62(1):128–163, 2019. doi:10.1007/s00454-019-00073-2.
- [MS21] Alejandro H. Morales and William Shi. Refinements and symmetries of the Morris identity for volumes of flow polytopes. *C. R. Math. Acad. Sci. Paris*, 359(7):823–851, 2021. doi:10.5802/crmath.218.

- [Sta86] Richard P. Stanley. Two poset polytopes. *Discrete Comput. Geom.*, 1(1):9–23, 1986. doi:10.1007/BF02187680.
- [vBGDMCY21] Matias von Bell, Rafael S. González D’León, Francisco A. Mayorga Cetina, and Martha Yip. On framed triangulations of flow polytopes, the ν -Tamari lattice and Young’s lattice. In *33rd International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2021)*, volume 85B, pages Art. 42, 12, Bar Ilan University, Ramat Gan, Israel, 2021. Séminaire Lotharingien de Combinatoire.