# Hardness of Finding Combinatorial Shortest Paths on Graph Associahedra

**Takehiro Ito**
Graduate School of Information Sciences, Tohoku University, Japan

**Naonori Kakimura**
Faculty of Science and Technology, Keio University, Yokohama, Japan

**Naoyuki Kamiyama**
Institute of Mathematics for Industry, Kyushu University, Fukouka, Japan

**Yusuke Kobayashi**
Research Institute for Mathematical Sciences, Kyoto University, Japan

**Shun-ichi Maezawa**
Department of Mathematics, Tokyo University of Science, Japan

**Yuta Nozaki**
Faculty of Environment and Information Sciences, Yokohama National University, Japan
SKCM$^2$, Hiroshima University, Japan

**Yoshio Okamoto**
Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan

## Abstract

We prove that the computation of a combinatorial shortest path between two vertices of a graph associahedron, introduced by Carr and Devadoss, is NP-hard. This resolves an open problem raised by Cardinal. A graph associahedron is a generalization of the well-known associahedron. The associahedron is obtained as the graph associahedron of a path. It is a tantalizing and important open problem in theoretical computer science whether the computation of a combinatorial shortest path between two vertices of the associahedron can be done in polynomial time, which is identical to the computation of the flip distance between two triangulations of a convex polygon, and the rotation distance between two rooted binary trees. Our result shows that a certain generalized approach to tackling this open problem is not promising. As a corollary of our theorem, we prove that the computation of a combinatorial shortest path between two vertices of a polymatroid base polytope cannot be done in polynomial time unless P = NP. Since a combinatorial shortest path on the matroid base polytope can be computed in polynomial time, our result reveals an unexpected contrast between matroids and polymatroids.

**2012 ACM Subject Classification** Mathematics of computing → Combinatorics; Theory of computation → Problems, reductions and completeness

**Keywords and phrases** Graph associahedra, combinatorial shortest path, NP-hardness, polymatroids

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2023.82

**Category** Track A: Algorithms, Complexity and Games

**Related Version** *Full Version*: `https://arxiv.org/abs/2304.14782`

**Funding** *Takehiro Ito*: JSPS KAKENHI Grant Numbers JP18H04091, JP19K11814, JP20H05793.
*Naonori Kakimura*: JSPS KAKENHI Grant Numbers JP20H05795, JP21H03397, JP22H05001.
*Naoyuki Kamiyama*: JSPS KAKENHI Grant Number JP20H05795.
*Yusuke Kobayashi*: JSPS KAKENHI Grant Numbers JP20K11692, JP20H05795, JP22H05001.
*Shun-ichi Maezawa*: JSPS KAKENHI Grant Numbers JP20H05795, JP22K13956.
*Yuta Nozaki*: JSPS KAKENHI Grant Numbers JP20H05795, JP20K14317, JP23K12974.
*Yoshio Okamoto*: JSPS KAKENHI Grant Numbers JP20H05795, JP20K11670, JP23K10982.

**Acknowledgements** This work was motivated by the talks of Vincent Pilaud and Jean Cardinal at Workshop "Polytope Diameter and Related Topics," held online on September 2, 2022. We thank them for inspiration. We are also grateful to Yuni Iwamasa and Kenta Ozeki for the discussion and to the anonymous reviewers for their helpful comments.

# 1 Introduction

Graph associahedra were introduced by Carr and Devadoss [8]. These polytopes generalize associahedra. In an associahedron, each vertex corresponds to a binary tree over a set of $n$ elements, and each edge corresponds to a rotation operation between two binary trees. For the historical account of associahedra, see the introduction of the paper by Ceballos, Santos, and Ziegler [9].

A binary tree can be obtained from a labeled path. Let $V = \{1, 2, \ldots, n\}$ be the set of vertices of the path, and $E = \{\{i, i+1\} \mid 1 \leq i \leq n-1\}$ be the set of edges of the path. To construct a labeled binary tree, we choose an arbitrary vertex from the path. Let it be $i \in V$. Then, the removal of $i$ from the path results in at most two connected components: the left subpath and the right subpath, which may be empty. Then, in the corresponding binary tree, we set $i$ as a root, and append recursively a binary tree of the left subpath as a left subtree and a binary tree of the right subpath as a right subtree. Note that in this construction, each node of the binary tree is labeled by a vertex of the path.

In the construction of graph associahedra, we follow the same idea. Since we are only interested in the graph structure of graph associahedra in this work, we only describe their vertices and edges. To define a graph associahedron, we first fix a connected undirected graph $G = (V, E)$.[1] Then, in the $G$-associahedron, the vertices correspond to the so-called elimination trees of $G$, and the edges correspond to swap operations between two elimination trees. The following description follows that of Cardinal, Merino, and Mütze [5].

An *elimination tree* of a connected undirected graph $G = (V, E)$ is a rooted tree defined as follows. It has $V$ as the vertex set and is composed of a root $v \in V$ that has as children elimination trees for each connected component of $G - v$ (Figure 1). A swap from an elimination tree $T$ to another elimination tree $T'$ of $G$ is defined as follows. Let $v$ be a non-root vertex of $T$, and let $u$ be the parent of $v$ in $T$. Denote by $H$ the subgraph of $G$ induced by the subtree rooted at $v$ in $T$. Then, the swap of $u$ with $v$ transforms $T$ to $T'$ as follows. (1) The tree $T'$ has $v$ as the parent of $u$, and $T'$ has $v$ as a child of the parent of $u$ in $T$. (2) The subtrees rooted at $u$ in $T$ remain subtrees rooted at $u$ in $T'$. (3) A subtree $S$ rooted at $v$ in $T$ remains a subtree rooted at $v$ in $T'$, unless the vertices of $S$ belong to the same connected component of $H - v$ as $u$, in which case $S$ becomes a subtree rooted at $u$ in $T'$. The $G$-associahedron for a claw $G$ is given in Figure 2. Note that a swap operation is reversible.

In this paper, among graph properties of graph associahedra, we concentrate on the computation of a combinatorial shortest path (i.e., the graph-theoretic distance) between two vertices of the polytope, which we call the combinatorial shortest path problem on graph associahedra. In this problem, we are given a graph $G$ and two elimination trees $T, T'$ of $G$, and want to compute the shortest length of a graph-theoretic path from $T$ to $T'$ on the $G$-associahedron. In the literature, we only find the studies in the case where $G$ is a complete graph or (a generalization of) a star. When $G$ is a complete graph, the $G$-associahedron is called a permutahedron, and each of its vertices corresponds to a permutation. Since an edge corresponds to an adjacent transposition, the graph-theoretic distance between two vertices is equal to the number of inversions between the corresponding permutations. This can be computed in polynomial time. When $G$ is a star, the $G$-associahedron is called stellohedron [26]. Recently, Cardinal, Pournin, and Valencia-Pabon [7] gave a polynomial-

[1] A graph associahedron can also be defined for disconnected graphs, but in this paper, we concentrate on connected graphs.

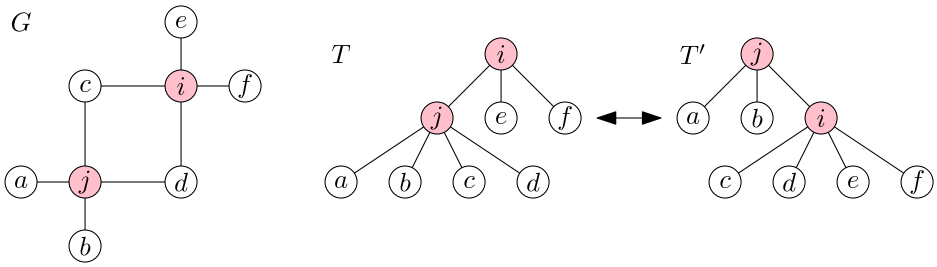


**Figure 1** An example of elimination trees. Two trees $T$ and $T'$ are elimination trees of the graph $G$. The tree $T'$ is obtained from $T$ by the swap of $i$ with $j$. The example is borrowed from Cardinal, Merino, and Mütze [5].

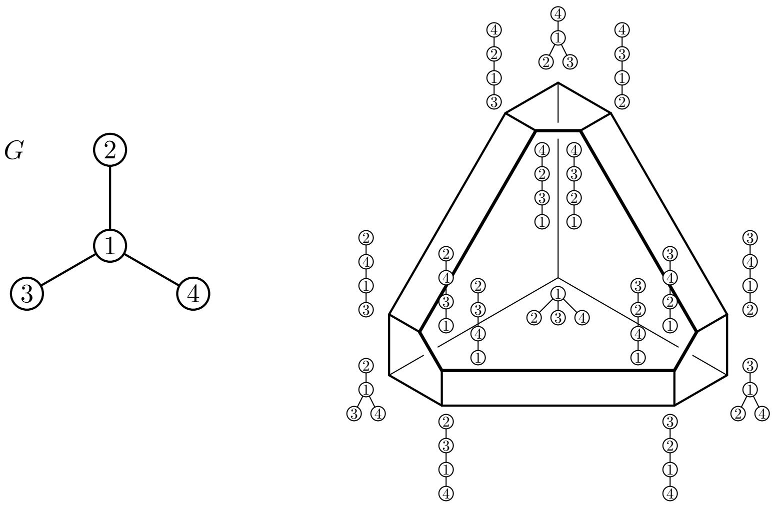


**Figure 2** An example of a graph associahedron. Each vertex of the polytope corresponds to an elimination tree of the graph $G$.

time algorithm to compute a combinatorial shortest path on stellohedra, and they generalize the algorithm when $G$ is a complete split graph (i.e., a graph obtained from a star by replacing the center vertex with a clique).

On the other hand, it is a tantalizing open problem whether a combinatorial shortest path can be computed in polynomial time when $G$ is a path. In this case, the graph-theoretic distance corresponds to the rotation distance between two binary trees. By Catalan correspondences, this is equivalent to the flip distance between two triangulations of a convex polygon. A possible strategy to resolve this open problem is to generalize the problem and solve the generalized problem. In our case, a generalization is achieved by considering graph associahedra for general graphs.

Our main result states that the combinatorial shortest path problem on $G$-associahedra is NP-hard when $G$ is also given as part of the input. This implies that the strategy mentioned above is bound to fail.

First, we formally state the problem Combinatorial Shortest Path on Graph Associahedra as follows.

| Combinatorial Shortest Path on Graph Associahedra |
|---|
| Input: A graph $G$ and two elimination trees $T_{\mathrm{ini}}, T_{\mathrm{tar}}$ of $G$ |
| Output: The distance between $T_{\mathrm{ini}}$ and $T_{\mathrm{tar}}$ on the graph of the $G$-associahedron |

Our first theorem states the NP-hardness of COMBINATORIAL SHORTEST PATH ON GRAPH ASSOCIAHEDRA. This solves an open problem raised by Cardinal (see [3, Section 4.2]).

▶ **Theorem 1.** *COMBINATORIAL SHORTEST PATH ON GRAPH ASSOCIAHEDRA is* NP*-hard.*

Theorem 1 yields the following corollary, which is related to polymatroids introduced by Edmonds [13]. A pair $(U, \rho)$ of a finite set $U$ and a function $\rho\colon 2^U \to \mathbb{R}$ is called a *polymatroid* if $\rho$ satisfies the following conditions: (P1) $\rho(\emptyset) = 0$; (P2) if $X \subseteq Y \subseteq U$, then $\rho(X) \leq \rho(Y)$; (P3) if $X, Y \subseteq U$, then $\rho(X \cup Y) + \rho(X \cap Y) \leq \rho(X) + \rho(Y)$. The function $\rho$ is called the *rank function* of the polymatroid $(U, \rho)$.

For a polymatroid $(U, \rho)$, we define the *base polytope* of $(U, \rho)$ as

$$\mathbf{B}(\rho) := \{x \in \mathbb{R}^U \mid x(X) \leq \rho(X)\,(\forall\, X \subseteq U),\, x(U) = \rho(U)\},$$

where we define $x(X) := \sum_{u \in X} x(u)$ for each subset $X \subseteq U$. Then, $\mathbf{B}(\rho)$ is a polytope because $0 \leq \rho(U) - \rho(U \setminus \{u\}) = x(U) - \rho(U \setminus \{u\}) \leq x(u) \leq \rho(\{u\})$ for every element $e \in E$.

A polymatroid is seen as a polyhedral generalization of a matroid. For example, the greedy algorithm for matroids can be treated as an algorithm to maximize a linear function over the base polytope of a matroid,[2] and the algorithm is readily generalized to the base polytope of a polymatroid. A lot of combinatorial properties of the base polytopes of matroids also hold for the base polytopes of polymatroids. Since it is known that a combinatorial shortest path on the base polytope of a matroid can be computed in polynomial time [18], we are interested in generalizing this result to polymatroids, which leads to the following problem definition.

| COMBINATORIAL SHORTEST PATH ON POLYMATROIDS |
|---|
| Input: An oracle access to a polymatroid $(U, \rho)$ and two extreme points $x_{\text{ini}}, x_{\text{tar}}$ of the base polytope $\mathbf{B}(\rho)$ |
| Output: The distance between $x_{\text{ini}}$ and $x_{\text{tar}}$ on $\mathbf{B}(\rho)$ |

We note that a polymatroid $(U, \rho)$ is given as an oracle access that returns the value $\rho(X)$ for any set $X \subseteq U$. The running time of an algorithm is also measured in terms of the number of oracle calls. This is a standard assumption when we deal with polymatroids [16] since if we would input the function $\rho$ as a table of the values $\rho(X)$ for all $X \subseteq U$, then it would already take at least $2^{|U|}$ space. We also note that the adjacency of two extreme points of the base polytope of a polymatroid can be tested in polynomial time [31].

The next theorem states that this problem is hard, which is proved as a corollary of Theorem 1, and reveals an unexpected contrast between matroids and polymatroids.

▶ **Theorem 2.** *There exists no polynomial-time algorithm for COMBINATORIAL SHORTEST PATH ON POLYMATROID unless* P = NP.

Our proof relies on the fact that graph associahedra can be realized as the base polytopes of some polymatroids [26]. However, we need careful treatment since in the reduction we require the rank function of our polymatroid to be evaluated in polynomial time. To this end, we give an explicit inequality description of the realization of a graph associahedron due to Devadoss [12],[3] which is indeed the base polytope of a polymatroid.

[2] This can further be seen as a generalization of Kruskal's algorithm for the minimum spanning tree problem.

[3] We note here that the original definition of a graph associahedron by Carr and Devadoss [8] does not give explicit vertex coordinates of the polytope. Therefore, we rely on the realization by Devadoss [12] who gave the explicit vertex coordinates.

## Related Work

Paths on polytopes have been studied thoroughly. One of the initial motivations for this research direction is to design and understand path-following algorithms for linear optimization such as simplex methods. In his chapter of *Handbook of Discrete and Computational Geometry* [17], Kalai stated as an open problem "Find an efficient routing algorithm for convex polytopes." Here, a routing algorithm means one that finds a short path from a given initial vertex to a given target vertex.

Paths on graph associahedra have been receiving much attention. The diameter is perhaps the most frequently studied quantity, which is defined as the maximum distance between two vertices of the polytope. A famous result by Sleator, Tarjan, and Thurston [30] states that the diameter of the $(n-1)$-dimensional associahedron (i.e., a graph associahedron of an $n$-vertex path) is at most $2n-6$ when $n \geq 11$ and this bound is tight for all sufficiently large $n$. This bound is refined by Pournin [27], who proved that the diameter of the $(n-1)$-dimensional associahedron is exactly $2n-6$ when $n \geq 11$.

For a general $n$-vertex graph $G$, Manneville and Pilaud [24] proved that the diameter of $G$-associahedron is at most $\binom{n}{2}$ and at least $\max\{m, 2n-20\}$, when $m$ is the number of edges of $G$. The upper bound is attained by the case where $G$ is a complete graph (i.e., the $G$-associahedron is a permutahedron). When $G$ is an $n$-vertex star (i.e., $K_{1,n-1}$), $n \geq 6$, Manneville and Pilaud [24] showed that the diameter is $2n-2$. When $G$ is a cycle (i.e., the polytope is a cyclohedron), Pournin [28] gave the asymptotically exact diameter. When $G$ is a tree, Manneville and Pilaud [24] gave the upper bound $O(n \log n)$ while Cardinal, Langerman, and Pérez-Lantero [4] gave an example of trees for which the diameter is $\Omega(n \log n)$ (such an example is chosen as a complete binary tree). Cardinal, Pournin, and Valencia-Pabon [6] proved that the diameter is $\Theta(m)$ for $m$-edge trivially perfect graphs, and gave upper and lower bounds for the diameter in terms of treewidths, pathwidths, and treedepths of graphs. Berendsohn [2] proved that the diameter is $\Theta(n + mH)$ for a caterpillar with $n$ spine vertices, $m$ leg vertices, and the Shannon entropy $H$ of the so-called leg distribution.

To the authors' knowledge, the complexity of computing the diameter of graph associahedra has not been investigated. When polytopes are not restricted to graph associahedra, a few hardness results have been known. Frieze and Teng [15] proved that computing the diameter of a polytope, given by inequalities, is weakly NP-hard. Sanità [29] proved that computing the diameter of the fractional matching polytope of a given graph is strongly NP-hard. Kaibel and Pfetsch [19] raised an open problem about the complexity of computing the diameter of simple polytopes.

The computation of a combinatorial shortest path on a $G$-associahedron has also been studied. It is a long-standing open problem whether a combinatorial shortest path in an associahedron (i.e., a $G$-associahedron when $G$ is a path) can be computed in polynomial time. Polynomial-time algorithms are only known when $G$ is a complete graph (folklore), a star, or a complete split graph (Cardinal, Pournin, and Valencia-Pabon [7]). When $G$ is a path, a polynomial-time approximation algorithm of factor two [11] and fixed-parameter algorithms when the distance is a parameter [10, 20, 21, 23] are known.

Since a combinatorial shortest path on an associahedron is equivalent to a shortest flip sequence of triangulations of a convex polygon, the computation of a shortest flip sequence of triangulations has been studied in more general setups. For triangulations of point sets, the problem is NP-hard [22] and even APX-hard [25]. For triangulations of simple polygons, the problem is also NP-hard [1].

Elimination trees have appeared in a lot of branches of mathematics and computer science. For a good summary, see Cardinal, Merino, and Mütze [5].

## Technique Overview

To prove the hardness of the combinatorial shortest path problem on graph associahedra, we first consider a "weighted" version of the combinatorial shortest path problem on graph associahedra, which is newly introduced in this paper for our reduction. In this problem, each vertex of a given graph has a positive weight, and the swap of two vertices incurs the weight that is defined as the product of the weights of these two vertices. The weight of a swap sequence is defined as the sum of weights of swaps in the sequence. As our intermediate theorem, we prove that this weighted version is strongly NP-hard.

To this end, we reduce the NP-hard problem of finding a balanced minimum $s$-$t$ cut in a graph [14] to the weighted version of the combinatorial shortest path problem on graph associahedra. In the balanced minimum $s$-$t$ cut problem, we want to determine whether there exists a minimum $s$-$t$ cut of a given graph $G$ that is a bisection of the vertex set. In the reduction, we construct a vertex-weighted graph $H$ from $G$ and two elimination trees $T_{\mathrm{ini}}$, $T_{\mathrm{tar}}$ of $H$. The weighted graph $H$ is constructed by replacing $s$ and $t$ by large cliques, subdividing each edge, and duplicating each vertex; the weights are assigned so that the subdivision vertices receive small weights, and duplicated vertices and vertices in large cliques receive large weights. Elimination trees $T_{\mathrm{ini}}$ and $T_{\mathrm{tar}}$ are constructed so that swapping two vertices of large weights is forbidden and identifying a few vertices of small weights (that corresponds to a balanced minimum $s$-$t$ cut of $G$) gives a shortest path.

In the second step, we reduce the weighted version to the original unweighted version of the problem. To this end, a vertex of weight $w$ is replicated by a clique of size $w$. We want to make sure that a swap of the vertices $u, v$ of weights $w(u), w(v)$, respectively in the weighted instance is mapped to consecutive $w(u) \cdot w(v)$ swaps of the vertices of cliques that correspond to $u$ and $v$ in the constructed unweighted instance. This is proved via the useful operation of projections combined with the averaging argument.

## 2 Preliminaries

For a positive integer $k$, let $[k]$ denote $\{1, 2, \ldots, k\}$.

For a graph $G = (V, E)$ and two elimination trees $T_{\mathrm{ini}}$ and $T_{\mathrm{tar}}$ of $G$, we say that a sequence $\mathbf{T} = \langle T_0, T_1, \ldots, T_\ell \rangle$ of elimination trees of $G$ is a *reconfiguration sequence from $T_{\mathrm{ini}}$ to $T_{\mathrm{tar}}$* if $T_0 = T_{\mathrm{ini}}$, $T_\ell = T_{\mathrm{tar}}$, and $T_i$ is obtained from $T_{i-1}$ by applying a single swap operation for $i \in [\ell]$. We sometimes regard $\mathbf{T}$ as a sequence of swap operations if no confusion may arise. The length of $\mathbf{T}$ is defined as the number $\ell$ of swaps in $\mathbf{T}$, which we denote $\mathrm{length}(\mathbf{T})$. Then, Combinatorial Shortest Path on Graph Associahedra is the problem of finding a reconfiguration sequence $\mathbf{T}$ from $T_{\mathrm{ini}}$ to $T_{\mathrm{tar}}$ that minimizes $\mathrm{length}(\mathbf{T})$.

When $u \in V$ is a child of $v \in V$ in an elimination tree $T$, an operation swapping $u$ and $v$ is represented by $\mathbf{swap}(u, v)$. Note that we distinguish $\mathbf{swap}(u, v)$ and $\mathbf{swap}(v, u)$. For an elimination tree $T$ and for a vertex $v \in V(T)$, let $\mathbf{anc}_T(v)$ (resp. $\mathbf{des}_T(v)$) denote the set of all ancestors (descendants) of $v$ in $T$. Note that $u \in \mathbf{anc}_T(v)$ if and only if $v \in \mathbf{des}_T(u)$. Note also that neither $\mathbf{anc}_T(v)$ nor $\mathbf{des}_T(v)$ contains $v$. We say that distinct vertices $u$ and $v$ are *comparable* in $T$ if $u \in \mathbf{anc}_T(v)$ or $v \in \mathbf{anc}_T(u)$. Otherwise, they are called *incomparable* in $T$. A linear ordering $\prec$ on $V$ defines an elimination tree $T$ uniquely so that $u \in \mathbf{anc}_T(v)$ implies $u \prec v$.

Let $G = (V, E)$ be an undirected graph. For $X \subseteq V$, let $\delta_G(X)$ denote the set of edges between $X$ and $V \setminus X$. For $s, t \in V$, we say that $X \subseteq V$ is an *$s$-$t$ cut* if $s \in X$ and $t \notin X$. An edge set $F \subseteq E$ is called an *$s$-$t$ cut set* if $F = \delta_G(X)$ for some $s$-$t$ cut $X \subseteq V$. A *minimum $s$-$t$ cut* is an $s$-$t$ cut $X$ minimizing $|\delta_G(X)|$. For $X \subseteq V$, let $G[X]$ denote the subgraph induced by $X$ and let $E[X]$ denote its edge set.

# 3 Hardness of the Weighted Problem

We consider a weighted variant of Combinatorial Shortest Path on Graph Associahedra, which we call Weighted Combinatorial Shortest Path on Graph Associahedra. In the problem, we are given a graph $G = (V, E)$, two elimination trees $T_{\mathrm{ini}}$ and $T_{\mathrm{tar}}$, and a weight function $w\colon V \to \mathbb{Z}_{>0}$. For $u, v \in V$, the *weight* of $\mathbf{swap}(u, v)$ is defined as $w(u) \cdot w(v)$. This value is sometimes denoted by $w(\mathbf{swap}(u, v))$. The *weighted length* (or simply the *weight*) of a reconfiguration sequence $\mathbf{T}$ is defined as the total weight of swaps in $\mathbf{T}$, which we denote by $\mathrm{length}_w(\mathbf{T})$. The objective of Weighted Combinatorial Shortest Path on Graph Associahedra is to find a reconfiguration sequence $\mathbf{T}$ from $T_{\mathrm{ini}}$ to $T_{\mathrm{tar}}$ that minimizes $\mathrm{length}_w(\mathbf{T})$.

In this section, we show that the weighted variant is strongly NP-hard.

▶ **Theorem 3.** *Weighted Combinatorial Shortest Path on Graph Associahedra is strongly* NP*-hard, that is, it is* NP*-hard even when the input size is* $\sum_{v \in V} w(v)$.

## 3.1 Reduction

To show Theorem 3, we reduce Balanced Minimum $s$-$t$ Cut to Weighted Combinatorial Shortest Path on Graph Associahedra. In Balanced Minimum $s$-$t$ Cut, the input consists of a connected graph $G = (V, E)$ with $s, t \in V$, and the objective is to determine whether $G$ contains a minimum $s$-$t$ cut $X$ with $|X| = |V \setminus X|$. Without loss of generality, we may assume that $|V|$ is even. Let $V = \{s, t, v_1, v_2, \dots, v_{2n}\}$ and $E = \{e_1, \dots, e_m\}$, where $|V| = 2n + 2$ and $|E| = m$. It is known that Balanced Minimum $s$-$t$ Cut is NP-hard [14]. For an instance of Balanced Minimum $s$-$t$ Cut, we construct an instance of Weighted Combinatorial Shortest Path on Graph Associahedra as follows.

Let $N$ be a sufficiently large integer (e.g., $N = 10n^3m$). We first subdivide each edge $e \in E$ by introducing a new vertex $u_e$. Then, for each $v \in V$, we introduce a copy $v'$ of $v$. We replace $s$ with a clique $\{s_1, \dots, s_{N^3}\}$ of size $N^3$ and replace $t$ with another clique $\{t_1, \dots, t_{N^3}\}$ of size $N^3$. Let $H$ be the obtained graph. Formally, the graph $H = (V(H), E(H))$ is defined as follows:

$$
\begin{aligned}
V(H) &= (V \setminus \{s,t\}) \cup \{v' \mid v \in V\} \cup \{u_e \mid e \in E\} \cup \{s_1, \dots, s_{N^3}\} \cup \{t_1, \dots, t_{N^3}\},\\
E(H) &= \{\{v, u_e\} \mid v \in V \setminus \{s,t\},\ e \in \delta_G(v)\} \cup \{\{v', u_e\} \mid v \in V,\ e \in \delta_G(v)\}\\
&\qquad \cup \{\{s_i, s_j\} \mid i, j \in [N^3],\ i \neq j\} \cup \{\{t_i, t_j\} \mid i, j \in [N^3],\ i \neq j\}\\
&\qquad \cup \{\{s_i, u_e\} \mid i \in [N^3],\ e \in \delta_G(s)\} \cup \{\{t_i, u_e\} \mid i \in [N^3],\ e \in \delta_G(t)\}.
\end{aligned}
$$

Define $w\colon V(H) \to \mathbb{Z}_{>0}$ as follows:

$$
\begin{aligned}
w(v) &= N && (v \in V \setminus \{s,t\}),\\
w(v') &= N^8 && (v \in V),\\
w(u_e) &= 1 && (e \in E),\\
w(s_i) = w(t_i) &= N^4 && (i \in [N^3]).
\end{aligned}
$$

The initial elimination tree $T_{\mathrm{ini}}$ is defined by the following linear ordering:

$$
\begin{aligned}
&v_1 \prec \cdots \prec v_{2n} \prec s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_{N^3} \prec t_{N^3}\\
&\qquad \prec u_{e_1} \prec \cdots \prec u_{e_m} \prec v'_1 \prec \cdots \prec v'_{2n} \prec s' \prec t'.
\end{aligned}
$$

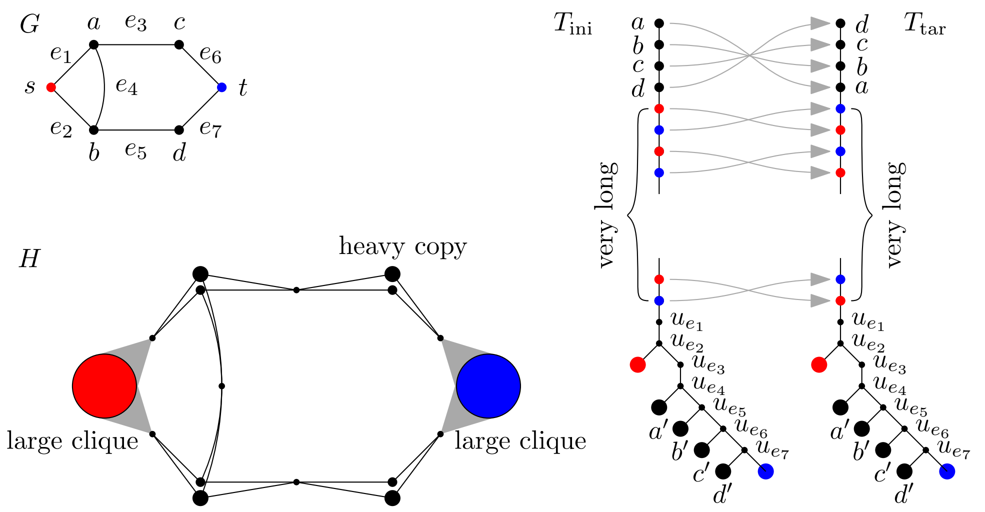


**Figure 3** Reduction for Theorem 3.

Note that, in $T_{\mathrm{ini}}$, the vertices $v_1, \dots, v_{2n}, s_1, t_1, s_2, t_2, \dots, s_{N^3}, t_{N^3}$ are aligned on a path, while the other elements are not necessarily aligned sequentially. The target elimination tree $T_{\mathrm{tar}}$ is the elimination tree defined by the following linear ordering:

$$\begin{aligned} &v_{2n} \prec \cdots \prec v_1 \prec t_1 \prec s_1 \prec t_2 \prec s_2 \prec \cdots \prec t_{N^3} \prec s_{N^3} \\ &\quad \prec u_{e_1} \prec \cdots \prec u_{e_m} \prec v'_1 \prec \cdots \prec v'_{2n} \prec s' \prec t'. \end{aligned}$$

We consider an instance $(H, w, T_{\mathrm{ini}}, T_{\mathrm{tar}})$ of Weighted Combinatorial Shortest Path on Graph Associahedra. In this instance, we reverse the ordering of the first $2n$ elements and reverse the ordering of $s_i$ and $t_i$ for each $i$. See Figure 3 for an illustration.

To prove Theorem 1, it suffices to show the following proposition.

▶ **Proposition 4.** *Let $\lambda$ be the minimum size of an $s$-$t$ cut set in $G$. There is a reconfiguration sequence from $T_{\mathrm{ini}}$ to $T_{\mathrm{tar}}$ of weight less than $4\lambda N^7 + (n^2 - n + 1)N^2$ if and only if $G$ has a minimum $s$-$t$ cut $X$ with $|X| = |V \setminus X|$.*

## 3.2 Proof of Proposition 4

### Sufficiency ("if" part)

Suppose that $G$ has a minimum $s$-$t$ cut $X$ with $|X| = |V \setminus X| = n + 1$. Let $U = \{u_e \mid e \in \delta_G(X)\}$. Note that $|U| = |\delta_G(X)| = \lambda$. Starting from $T_{\mathrm{ini}}$, we swap an element in $U$ and its parent repeatedly so that we obtain an elimination tree $T_1$ in which each element in $U$ is an ancestor of $V(H) \setminus U$. See Figure 4. The total weight of swaps from $T_{\mathrm{ini}}$ to $T_1$ is at most $|U|(2nN + 2N^7 + m)$. Since $G - \delta_G(X)$ consists of two connected components, so does $H - U$. Thus, $T_1 - U$ consists of two elimination trees $T_s$ and $T_t$ such that $T_s$ contains $(X \setminus \{s\}) \cup \{s_1, \dots, s_{N^3}\} \cup \{u_e \mid e \in E[X]\} \cup \{v' \mid v \in X\}$ and $T_t$ contains $((V \setminus X) \setminus \{t\}) \cup \{t_1, \dots, t_{N^3}\} \cup \{u_e \mid e \in E[V \setminus X]\} \cup \{v' \mid v \in V \setminus X\}$.

In $T_s$, by swapping $u$ and $v$ for every pair of $u, v \in X \setminus \{s\}$, we obtain an elimination tree in which $v_i$ is an ancestor of $v_j$ for $v_i, v_j \in X \setminus \{s\}$ with $i > j$. The total weight of these swaps is $\binom{|X|-1}{2} \cdot N^2$. Similarly, by applying swaps with weight $\binom{|V \setminus X|-1}{2} \cdot N^2$ to $T_t$, we obtain an elimination tree in which $v_i$ is an ancestor of $v_j$ for $v_i, v_j \in (V \setminus X) \setminus \{t\}$ with $i > j$. Let $T_2$ be the elimination tree obtained from $T_1$ by applying these operations.

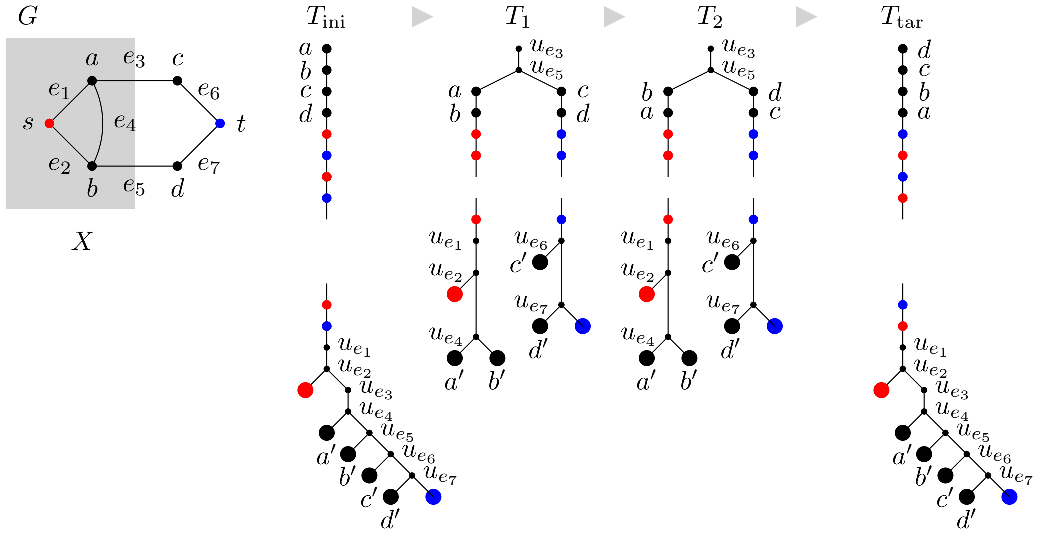


**Figure 4** A reconfiguration sequence from $T_{\text{ini}}$ to $T_{\text{tar}}$.

Starting from $T_2$, we swap an element in $U$ and its child repeatedly so that we obtain an elimination tree $T_{\text{tar}}$. This can be done by applying swaps whose total weight is at most $|U|(2nN + 2N^7 + m)$.

Therefore, the total weight of the above swaps from $T_{\text{ini}}$ to $T_{\text{tar}}$ is at most

$$\begin{aligned}
&2|U|(2nN + 2N^7 + m) + \binom{|X|-1}{2} \cdot N^2 + \binom{|V \setminus X|-1}{2} \cdot N^2 \\
&\quad = 4\lambda N^7 + n(n-1)N^2 + 4\lambda nN + 2\lambda m \\
&\quad < 4\lambda N^7 + (n^2 - n + 1)N^2,
\end{aligned}$$

where we note that $|U| = \lambda$ and $|X| = |V \setminus X| = n + 1$. This shows the sufficiency.

## Necessity (“only if” part)

Let $\mathbf{T}$ be a reconfiguration sequence from $T_{\text{ini}}$ to $T_{\text{tar}}$ whose weight is less than $4\lambda N^7 + (n^2 - n + 1)N^2$. Since this weight is less than $N^8$, we observe the following.

▶ **Observation 5.** *For $v \in V$, vertex $v'$ is not swapped with other vertices in $\mathbf{T}$. For $i, j \in [N^3]$, none of* $\mathbf{swap}(s_i, s_j)$, $\mathbf{swap}(t_i, t_j)$, $\mathbf{swap}(s_i, t_j)$, *and* $\mathbf{swap}(t_i, s_j)$ *is applied in* $\mathbf{T}$.

By Observation 5, we cannot swap $s_1$ and $t_1$ directly, and hence $\mathbf{T}$ contains an elimination tree $T^*$ in which $s_1$ and $t_1$ are incomparable. Then, there exists a vertex set $V^* \subseteq V(H)$ such that $s_1$ and $t_1$ are contained in different connected components of $H - V^*$, and each vertex in $V^*$ is an ancestor of $s_1$ and $t_1$ in $T^*$. By Observation 5 again, $V^*$ does not contain $v'$ for $v \in V$, that is, $V^* \subseteq V \cup \{u_e \mid e \in E\}$. Note that removing $V^* \cap V$ does not affect the connectedness of $H$ since each vertex $v \in V^* \cap V$ has its copy $v'$ in $H$. Let

$$F := \{e \in E \mid u_e \in V^*\}.$$

Then, $s$ and $t$ are contained in different connected components of $G - F$, i.e., $F$ contains an $s$-$t$ cut set in $G$.

Since removing $V$ does not affect the connectedness of $H$, we also observe the following.

▶ **Observation 6.** *Let $T$ and $T'$ be elimination trees in $\mathbf{T}$ and let $e_1, e_2 \in E$ be distinct edges. If $u_{e_1} \in \mathbf{anc}_T(u_{e_2})$ and $u_{e_2} \in \mathbf{anc}_{T'}(u_{e_1})$, then $\mathbf{swap}(u_{e_3}, u_{e_4})$ is applied for some $e_3, e_4 \in E$ (possibly $\{e_1, e_2\} \cap \{e_3, e_4\} \neq \emptyset$) between $T$ and $T'$.*

We divide $\mathbf{T}$ into two reconfiguration sequences $\mathbf{T}_1$ and $\mathbf{T}_2$, where $\mathbf{T}_1$ is from $T_{\text{ini}}$ to $T^*$ and $\mathbf{T}_2$ is from $T^*$ to $T_{\text{tar}}$. By symmetry, we may assume that

$$\text{length}_w(\mathbf{T}_1) \leq \frac{\text{length}_w(\mathbf{T})}{2} < 2\lambda N^7 + N^3.$$

For $i \in [N^3]$, define

$$L_i = \{e \in E \mid \mathbf{swap}(u_e, s_i) \text{ is applied in } \mathbf{T}_1\},$$
$$R_i = \{e \in E \mid \mathbf{swap}(u_e, t_i) \text{ is applied in } \mathbf{T}_1\}.$$

For $i \in [N^3]$, let $\mathbf{swap}(L_i)$ denote the set of all swaps $\mathbf{swap}(u_e, s_i)$ in $\mathbf{T}_1$ with $e \in L_i$. Similarly, let $\mathbf{swap}(R_i)$ denote the set of all swaps $\mathbf{swap}(u_e, t_i)$ in $\mathbf{T}_1$ with $e \in R_i$.

▷ Claim 7. For $i \in [N^3]$, we have the following:

- if an edge $e \in E$ is contained in the connected component of $G - L_i$ containing $s$, then $u_e \in \mathbf{des}_T(s_i)$ for any elimination tree $T$ in $\mathbf{T}_1$, and
- if an edge $e \in E$ is contained in the connected component of $G - R_i$ containing $t$, then $u_e \in \mathbf{des}_T(t_i)$ for any elimination tree $T$ in $\mathbf{T}_1$.

Proof. For each edge $e \in E$ in the connected component of $G - L_i$ containing $s$, vertices $s_i$ and $u_e$ are contained in the same connected component in $H - \{u_f \mid f \in L_i\}$. Since $u_e \in \mathbf{des}_{T_{\text{ini}}}(s_i)$ holds and $\mathbf{swap}(u_e, s_i)$ is not applied in $\mathbf{T}_1$ as $e \notin L_i$, we have that $u_e \in \mathbf{des}_T(s_i)$ for any elimination tree $T$ in $\mathbf{T}_1$. The same argument works for the second statement. ◁

To simplify the notation, let $L_0 = R_0 = F$. For $i \in [N^3] \cup \{0\}$, let $X_i \subseteq V$ be the vertex set of the connected component of $G - L_i$ containing $s$. Similarly, let $Y_i \subseteq V$ be the vertex set of the connected component of $G - R_i$ containing $t$.

▷ Claim 8. For $i, j \in [N^3] \cup \{0\}$ with $j > i$, we have the following:

**(i)** $(E[X_j] \setminus L_j) \cap L_i = \emptyset$, and
**(ii)** $(E[Y_j] \setminus R_j) \cap R_i = \emptyset$.

Proof. To show (i), assume to the contrary that there exists $e \in (E[X_j] \setminus L_j) \cap L_i$ for some $j > i$. Note that $j \in [N^3]$. Since $e \in E[X_j] \setminus L_j$, Claim 7 shows that $u_e \in \mathbf{des}_T(s_j)$ for any elimination tree $T$ in $\mathbf{T}_1$. If $i \geq 1$, then since $u_e \in \mathbf{des}_T(s_j)$ and $s_i \in \mathbf{anc}_T(s_j)$, we see that $u_e$ and $s_i$ are not adjacent in $T$. This implies that $\mathbf{swap}(u_e, s_i)$ is not applied in $\mathbf{T}_1$, which contradicts $e \in L_i$. If $i = 0$, then $e \in L_0 = F$ implies that $u_e \in \mathbf{anc}_{T^*}(s_1) \subseteq \mathbf{anc}_{T^*}(s_j)$, which contradicts $u_e \in \mathbf{des}_T(s_j)$ for any $T$. The same argument works for (ii). ◁

▷ Claim 9. $X_0 \supseteq X_1 \supseteq X_2 \supseteq \cdots \supseteq X_{N^3}$ and $Y_0 \supseteq Y_1 \supseteq Y_2 \supseteq \cdots \supseteq Y_{N^3}$.

Proof. Let $i, j \in [N^3] \cup \{0\}$ be indices with $j > i$. Since $(E[X_j] \setminus L_j) \cap L_i = \emptyset$ by Claim 8 (i), all vertices in $X_j$ are contained in the same connected component of $G - L_i$. Since both $X_i$ and $X_j$ contain $s$, we obtain $X_j \subseteq X_i$. This shows that $X_0 \supseteq X_1 \supseteq X_2 \supseteq \cdots \supseteq X_{N^3}$. Similarly, we obtain $Y_0 \supseteq Y_1 \supseteq Y_2 \supseteq \cdots \supseteq Y_{N^3}$. ◁

▷ Claim 10. For $i \in [N^3]$, we have $|L_i| = |R_i| = \lambda$, $L_i = \delta_G(X_i)$, and $R_i = \delta_G(Y_i)$.

Proof. Since $F$ contains an $s$-$t$ cut set, it holds that $X_0 \subseteq V \setminus \{t\}$. For $i \geq 1$, since $s \in X_i \subseteq X_0 \subseteq V \setminus \{t\}$ by Claim 9, we see that $\delta_G(X_i)$ is an $s$-$t$ cut set contained in $L_i$. Similarly, $R_i$ contains an $s$-$t$ cut set in $G$. Therefore, we obtain $|L_i|, |R_i| \geq \lambda$ for any $i \in [N^3]$. By considering the weight of $\mathbf{T}_1$, we obtain

$$\begin{aligned}
2\lambda N^7 + N^3 &> \mathrm{length}_w(\mathbf{T}_1)\\
&\geq \sum_{i=1}^{N^3} (w(s_i)|L_i| + w(t_i)|R_i|)\\
&= 2\lambda N^7 + N^4 \sum_{i=1}^{N^3} ((|L_i| - \lambda) + (|R_i| - \lambda)),
\end{aligned}$$

which shows that $|L_i| = |R_i| = \lambda$ for any $i \in [N^3]$. Therefore, each of $L_i$ and $R_i$ is a minimum $s$-$t$ cut set in $G$, and hence $L_i = \delta_G(X_i)$ and $R_i = \delta_G(Y_i)$ hold. ◁

Since the total weight of $\mathbf{swap}(L_i)$ and $\mathbf{swap}(R_i)$ is at least $2\lambda N^7$ by this claim, we see that $u_e$ and $s_i$ (resp. $t_i$) are swapped exactly once in $\mathbf{T}_1$ for $e \in L_i$ (resp. $e \in R_i$) and for $i \in [N^3]$. For $i \in [N^3]$, let $T_i$ (resp. $T'_i$) be the elimination tree that appears in $\mathbf{T}_1$ after all the swaps in $\mathbf{swap}(L_i)$ (resp. $\mathbf{swap}(R_i)$) are just applied.

▷ Claim 11. Elimination trees $T'_{N^3}, T_{N^3}, T'_{N^3-1}, T_{N^3-1}, \dots, T'_1, T_1$ appear in this order in $\mathbf{T}_1$.

Proof. We first show that $T_i$ appears after $T'_i$ for $i \in [N^3]$. Let $e \in R_i$ be the edge such that $\mathbf{swap}(u_e, t_i)$ is applied just before obtaining $T'_i$. Ler $f \in L_i \setminus (R_i \setminus \{e\})$, where the existence of such $f$ is guaranteed by $|L_i| = |R_i|$. Since $R_i$ is a minimum $s$-$t$ cut set by Claim 10, we see that $G - (R_i \setminus \{e\})$ is connected. Then, for any elimination tree $T$ before $T'_i$, we have $u_{e'} \in \mathbf{des}_T(t_i)$ for any $e' \in E \setminus (R_i \setminus \{e\})$. In particular, $u_f \in \mathbf{des}_T(t_i)$. Since $s_i \in \mathbf{anc}_T(t_i)$, we see that $u_f$ and $s_i$ are not adjacent in $T$. This shows that we cannot apply $\mathbf{swap}(u_f, s_i)$ before $T'_i$. Therefore, $T_i$ appears after $T'_i$ in $\mathbf{T}_1$.

By the same argument, we can show that $T'_i$ appears after $T_{i+1}$ for $i \in [N^3-1]$. Therefore, $T'_{N^3}$, $T_{N^3}$, $T'_{N^3-1}$, $T_{N^3-1}, \dots, T'_1$, $T_1$ appear in this order. ◁

▷ Claim 12. For $i \in [N^3]$, we have the following:

- $u_e \in \mathbf{anc}_{T_i}(u_{e'})$ for any $e \in L_i$ and $e' \in E \setminus L_i$, and
- $u_e \in \mathbf{anc}_{T'_i}(u_{e'})$ for any $e \in R_i$ and $e' \in E \setminus R_i$.

Proof. Let $T$ be an elimination tree in $\mathbf{T}_1$ just before $T_i$. Then, there exists an edge $f \in L_i$ such that $T_i$ is obtained from $T$ by applying $\mathbf{swap}(u_f, s_i)$. Since $G - (L_i \setminus \{f\})$ is connected by Claim 10, we have $u_e \in \mathbf{anc}_T(s_i)$ for $e \in L_i \setminus \{f\}$ and $u_{e'} \in \mathbf{des}_T(s_i)$ for $e' \in E \setminus (L_i \setminus \{f\})$. Therefore, after applying $\mathbf{swap}(u_f, s_i)$, we obtain $u_e \in \mathbf{anc}_{T_i}(f)$ for $e \in L_i \setminus \{f\}$ and $u_{e'} \in \mathbf{des}_{T_i}(f)$ for $e' \in E \setminus L_i$. This shows that $u_e \in \mathbf{anc}_{T_i}(u_{e'})$ for any $e \in L_i$ and $e' \in E \setminus L_i$. By the same argument, we obtain $u_e \in \mathbf{anc}_{T'_i}(u_{e'})$ for any $e \in R_i$ and $e' \in E \setminus R_i$. ◁

▷ Claim 13. $X_1 = V \setminus Y_1$.

Proof. Observe that $X_0$ and $Y_0$ are disjoint since $F = L_0 = R_0$ contains an $s$-$t$ cut set in $G$. Since $X_1 \subseteq X_0$ and $Y_1 \subseteq Y_0$ by Claim 9, we see that $X_1$ and $Y_1$ are disjoint. To derive a contradiction, assume that $X_1 \neq V \setminus Y_1$, that is, $X_1$ and $Y_1$ are disjoint sets with $X_1 \cup Y_1 \subsetneq V$. Then, by Claim 9, we obtain $X_i \neq V \setminus Y_i$ for any $i \in [N^3]$. This shows that $L_i \neq R_i$ for any $i \in [N^3]$. Since $|L_i| = |R_i| = \lambda$, there exist $f_i \in L_i \setminus R_i$ and $f'_i \in R_i \setminus L_i$. By Claim 12, we obtain $u_{f_i} \in \mathbf{anc}_{T_i}(u_{f'_i})$ and $u_{f'_i} \in \mathbf{anc}_{T'_i}(u_{f_i})$. By Observation 6, $\mathbf{swap}(u_e, u_{e'})$ is applied for some $e, e' \in E$ between $T'_i$ and $T_i$. Since such a swap is required for each $i \in [N^3]$, by Claim 11, we have to swap pairs in $\{u_e \mid e \in E\}$ at least $N^3$ times in $\mathbf{T}_1$. Therefore, we obtain

$$\text{length}_w(\mathbf{T}_1) \geq \sum_{i=1}^{N^3}(w(s_i)|L_i| + w(t_i)|R_i|) + N^3 = 2\lambda N^7 + N^3,$$

which contradicts $\text{length}_w(\mathbf{T}_1) < 2\lambda N^7 + N^3$. ◁

▷ Claim 14. $F = \delta_G(X_1) = \delta_G(Y_1)$.

Proof. Claims 10 and 13 show that $L_1 = R_1 = \delta_G(X_1) = \delta_G(Y_1)$. This together with Claim 8 shows that $F \cap E[X_1] = F \cap E[Y_1] = \emptyset$. Since $F$ contains an $s$-$t$ cut set in $G$, we obtain $F = \delta_G(X_1) = \delta_G(Y_1)$. ◁

▷ Claim 15. Let $T$ be an elimination tree in $\mathbf{T}_1$. If two vertices $u, v \in V \setminus \{s, t\}$ are contained in the same connected component in $G - F$, then $u$ and $v$ are comparable in $T$.

Proof. By Claim 14, $G - F$ consists of two connected components $G[X_1]$ and $G[Y_1]$. We first consider the case when $u, v \in X_1 \setminus \{s\}$. By Claims 7 and 14, we obtain $u_e \in \mathbf{des}_T(s_1)$ for any $e \in E[X_1]$. Furthermore, since $\text{length}_w(\mathbf{T}_1) < 2\lambda N^7 + N^3$ holds and the total weight of $\mathbf{swap}(L_i)$ and $\mathbf{swap}(R_i)$ is $2\lambda N^7$, neither $\mathbf{swap}(s_1, u)$ nor $\mathbf{swap}(s_1, v)$ is applied in $\mathbf{T}_1$, because $w(s_1)w(u) = w(s_1)w(v) = N^5$. Therefore, we obtain $u \in \mathbf{anc}_T(s_1)$ and $v \in \mathbf{anc}_T(s_1)$, which shows that $u$ and $v$ are comparable in $T$. The same argument works when $u, v \in Y_1 \setminus \{t\}$. ◁

Since the weight of $\mathbf{T}_1$ is at least $\sum_{i=1}^{N^3}(w(s_i)|L_i| + w(t_i)|R_i|) = 2\lambda N^7$, we obtain

$$\text{length}_w(\mathbf{T}_2) = \text{length}_w(\mathbf{T}) - \text{length}_w(\mathbf{T}_1) < 2\lambda N^7 + N^3.$$

Hence, the above argument (Claims 7–15) can be applied also to the reverse sequence of $\mathbf{T}_2$. In particular, Claim 15 holds even if $\mathbf{T}_1$ is replaced with $\mathbf{T}_2$. Therefore, if two vertices $u, v \in V \setminus \{s, t\}$ are contained in the same connected component in $G - F$, then $u$ and $v$ are comparable in any elimination tree in $\mathbf{T}$. For such a pair of vertices $u$ and $v$, the only way to reverse the ordering of $u$ and $v$ is to apply $\mathbf{swap}(u, v)$ or $\mathbf{swap}(v, u)$.

Recall that $G - F$ consists of two connected components $G[X_1]$ and $G[Y_1]$ by Claim 14. Since the ordering of $v_1, \dots, v_{2n}$ are reversed from $T_{\text{ini}}$ to $T_{\text{tar}}$, we see that $\mathbf{swap}(u, v)$ or $\mathbf{swap}(v, u)$ has to be applied in $\mathbf{T}$ if $u, v \in X_1 \setminus \{s\}$ or $u, v \in Y_1 \setminus \{t\} = (V \setminus X_1) \setminus \{t\}$. Furthermore, we have to swap some elements in $\{u_e \mid e \in E\}$ and $\{s_1, t_1, \dots, s_{N^3}, t_{N^3}\}$ in $\mathbf{T}_2$, whose total weight is at least $2\lambda N^7$ in the same way as $\mathbf{T}_1$. With these observations, we evaluate the weight of $\mathbf{T}$ as follows, where we denote $k = |X_1|$ to simplify the notation:

$$\begin{aligned}
\text{length}_w(\mathbf{T}) &\geq 2\lambda N^7 + 2\lambda N^7 + \binom{|X_1| - 1}{2} \cdot N^2 + \binom{|V \setminus X_1| - 1}{2} \cdot N^2 \\
&= 4\lambda N^7 + \frac{(k-1)(k-2)}{2} N^2 + \frac{(2n-k+1)(2n-k)}{2} N^2 \\
&= 4\lambda N^7 + (k^2 - 2(n+1)k + 2n^2 + n + 1) N^2 \\
&= 4\lambda N^7 + (n^2 - n) N^2 + (k - n - 1)^2 N^2.
\end{aligned}$$

This together with $\text{length}_w(\mathbf{T}) < 4\lambda N^7 + (n^2 - n + 1)N^2$ shows that $(k - n - 1)^2 < 1$, and hence $k = n + 1$ by the integrality of $k$ and $n$.

Therefore, we obtain $|X_1| = k = n + 1$ and $|Y_1| = |V \setminus X_1| = n + 1$. Since $|\delta_G(X_1)| = |L_1| = \lambda$, this shows that $X_1$ is a desired $s$-$t$ cut in $G$.

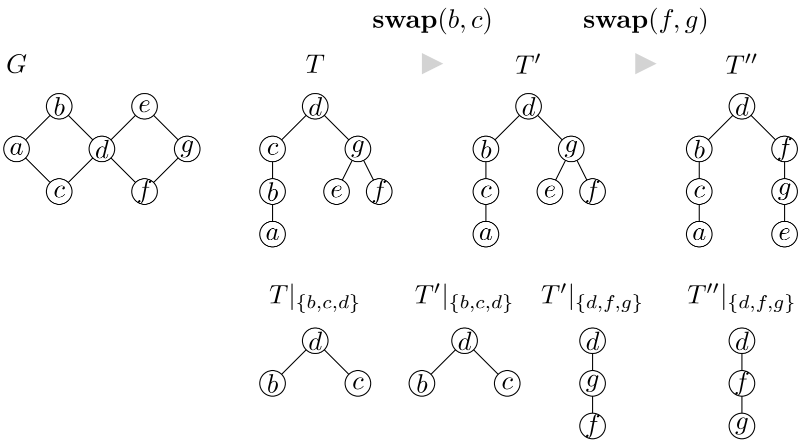


**Figure 5** An example of projections. Note that $T|_{\{b,c,d\}} = T'|_{\{b,c,d\}}$ since $b$ and $c$ are incomparable in $T|_{\{b,c,d\}}$, and $T''|_{\{d,f,g\}}$ is obtained from $T'|_{\{d,f,g\}}$ by swapping $f$ and $g$ since $f$ and $g$ are adjacent in $T'|_{\{d,f,g\}}$.

## 4 Hardness of the Unweighted Problem (Proof of Theorem 1)

To show Theorem 1, we reduce Weighted Combinatorial Shortest Path on Graph Associahedra to Combinatorial Shortest Path on Graph Associahedra. An operation called *projection* (e.g. [6]) plays an important role in our validity proof.

### 4.1 Useful Operation: Projection

Let $G = (V, E)$ be a graph and let $T$ be an elimination tree associated with $G$. For $U \subseteq V$ such that $G[U]$ is connected, let $T|_U$ be the elimination tree associated with $G[U]$ that preserves the ordering in $T$. That is, $u \in \mathbf{anc}_{T|_U}(v)$ if and only if $u \in \mathbf{anc}_T(v)$ and $u$ are $v$ are connected in $G[U] - \mathbf{anc}_T(u)$ for $u, v \in U$. Note that such $T|_U$ is uniquely determined. We call $T|_U$ the *projection* of $T$ to $U$. See Figure 5 for illustration.

▶ **Lemma 16.** *Let $U \subseteq V$ be a vertex set such that $G[U]$ is connected. Let $T$ and $T'$ be elimination trees associated with $G$ such that $T'$ is obtained from $T$ by applying* $\mathbf{swap}(u, v)$*, where $u, v \in V$.*

1. *If $\{u, v\} \subseteq U$, then either $T'|_U = T|_U$ or $T'|_U$ is obtained from $T|_U$ by applying* $\mathbf{swap}(u, v)$*.*
2. *Otherwise, $T'|_U = T|_U$.*

**Proof.** Since all the vertices in $V \setminus U$ are removed when we construct $T|_U$, $\mathbf{swap}(u, v)$ affects $T|_U$ only if $\{u, v\} \subseteq U$, which proves the second item. For the first item, suppose that $\{u, v\} \subseteq U$. Then, $u$ and $v$ are adjacent or incomparable in $T|_U$. If they are adjacent, then $T'|_U$ is obtained from $T|_U$ by applying $\mathbf{swap}(u, v)$. If they are incomparable, then $T'|_U = T|_U$. ◀

### 4.2 Reduction

Suppose we are given a graph $G = (V, E)$, two elimination trees $T_{\mathrm{ini}}$ and $T_{\mathrm{tar}}$, and a weight function $w\colon V \to \mathbb{Z}_{>0}$, which form an instance of Weighted combinatorial Shortest Path on Graph Associahedra. Then, we replace each vertex $v \in V$ with a clique of size $w(v)$. Formally, consider a graph $G' = (V', E')$ such that $V' = \{v_i \mid v \in V,\ i \in \{1, \ldots, w(v)\}\}$, and $\{u_i, v_j\} \in E'$ if $\{u, v\} \in E$ or $u = v$. Let $T'_{\rm ini}$ (resp. $T'_{\rm tar}$) be the elimination tree obtained from $T_{\rm ini}$ (resp. $T_{\rm tar}$) by replacing a vertex $v \in V$ with a path $v_1, v_2, \dots, v_{w(v)}$. That is, for distinct $u, v \in V$, there is an arc $(u, v)$ in $T_{\rm ini}$ (resp. $T_{\rm tar}$) if and only if $(u_{w(u)}, v_1)$ is an arc of $T'_{\rm ini}$ (resp. $T'_{\rm tar}$). Note that the obtained elimination tree is associated with $G'$. This defines an instance of Combinatorial Shortest on Graph Associahedra.

## 4.3 Validity

In what follows, we show that the obtained instance of Combinatorial Shortest Path on Graph Associahedra has a reconfiguration sequence of length at most $\ell$ if and only if the original instance of Weighted Combinatorial Shortest Path on Graph Associahedra has a reconfiguration sequence $\mathbf{T}$ with $\mathrm{length}_w(\mathbf{T}) \le \ell$.

### Sufficiency ("if" part)

Suppose that the original instance of Weighted Combinatorial Shortest Path on Graph Associahedra has a reconfiguration sequence $\mathbf{T}$ from $T_{\rm ini}$ to $T_{\rm tar}$. Then, we construct a reconfiguration sequence $\mathbf{T}'$ from $T'_{\rm ini}$ to $T'_{\rm tar}$ by replacing each swap $\mathbf{swap}(u, v)$ in $\mathbf{T}$ with $w(u) \cdot w(v)$ swaps $\{\mathbf{swap}(u_i, v_j) \mid i \in [w(u)],\ j \in [w(v)]\}$. This gives a reconfiguration sequence from $T'_{\rm ini}$ to $T'_{\rm tar}$ whose length is $\mathrm{length}_w(\mathbf{T})$, which shows the sufficiency.

### Necessity ("only if" part)

Suppose that the obtained instance of Combinatorial Shortest Path on Graph Associahedra has a reconfiguration sequence $\mathbf{T}'$ from $T'_{\rm ini}$ to $T'_{\rm tar}$ of length at most $\ell$. For any $v \in V$, since $v_1, \dots, v_{w(v)}$ form a clique, they are comparable in any elimination tree in $\mathbf{T}'$. Furthermore, since $v_1, \dots, v_{w(v)}$ are aligned in this order in both $T'_{\rm ini}$ and $T'_{\rm tar}$, we may assume that $\mathbf{swap}(v_i, v_j)$ is not applied in $\mathbf{T}'$ for any $i, j \in [w(v)]$.

Let $\Phi$ be the set of all maps $\phi\colon V \to \mathbb{Z}$ such that $\phi(v) \in \{1, \dots, w(v)\}$ for any $v \in V$. Note that $|\Phi| = \prod_{v \in V} w(v)$. For $\phi \in \Phi$, define $U_\phi = \{v_{\phi(v)} \mid v \in V\}$. Note that $G'[U_\phi]$ is isomorphic to $G$, and hence it is connected. By projecting each elimination tree in $\mathbf{T}'$ to $U_\phi$, we obtain a sequence of elimination trees. Lemma 16 shows that this forms a reconfiguration sequence, say $\mathbf{T}_\phi$, if we remove duplications when the same elimination tree appears consecutively. Since $G'[U_\phi]$ is isormorphic to $G$, by idenfitying $v_{\phi(v)}$ with $v$ for each $v \in V$, we can regard $\mathbf{T}_\phi$ as a recofiguration sequence from $T_{\rm ini}$ to $T_{\rm tar}$. That is, $\mathbf{T}_\phi$ is regarded as a feasible solution of the original instance of Weighted Combinatorial Shortest Path on Graph Associahedra.

In what follows, we consider reconfiguration sequences $\{\mathbf{T}_\phi \mid \phi \in \Phi\}$ and show that a desired sequence exists among them. Suppose that $\mathbf{swap}(u_i, v_j)$ is applied in $\mathbf{T}'$, where $u, v \in V$, $i \in [w(u)]$, and $j \in [w(v)]$. Then, Lemma 16 shows that the corresponding swap operation $\mathbf{swap}(u_i, v_j)$, which is identified with $\mathbf{swap}(u, v)$, is applied in $\mathbf{T}_\phi$ only if $\phi(u) = i$ and $\phi(v) = j$. Thus, such a swap is applied in at most $|\{\phi \in \Phi \mid \phi(u) = i,\ \phi(v) = j\}| = |\Phi|/(w(u) \cdot w(v))$ sequences in $\{\mathbf{T}_\phi \mid \phi \in \Phi\}$. Therefore, we obtain

$$\begin{aligned}\sum_{\phi \in \Phi} \mathrm{length}_w(\mathbf{T}_\phi) &= \sum_{\phi \in \Phi} \sum_{\mathbf{swap}(u,v) \in \mathbf{T}_\phi} w(\mathbf{swap}(u, v)) \\ &\le \sum_{\mathbf{swap}(u_i,v_j) \in \mathbf{T}'} w(\mathbf{swap}(u, v)) \cdot \frac{|\Phi|}{w(u) \cdot w(v)} \\ &= \mathrm{length}(\mathbf{T}') \cdot |\Phi| \le \ell \cdot |\Phi|,\end{aligned}$$

where each reconfiguration sequence is regarded as a multiset of swaps. Therefore,

$$\min_{\phi\in\Phi}(\mathrm{length}_w(\mathbf{T}_\phi)) \leq \frac{1}{|\Phi|}\sum_{\phi\in\Phi}\mathrm{length}_w(\mathbf{T}_\phi) \leq \ell.$$

Hence, there exists $\phi \in \Phi$ such that $\mathbf{T}_\phi$ is a desired sequence. This shows the necessity.

Therefore, the weighted problem can be reduced to the unweighted problem, and hence Theorem 3 implies Theorem 1.

## 5 Hardness for Polymatroids (Proof of Theorem 2)

In this section, we give a proof sketch of Theorem 2.

We reduce Combinatorial Shortest Path on Graph Associahedra to Combinatorial Shortest Path on Polymatroids. Assume that we are given an instance $G = (V, E)$, $T_{\mathrm{ini}}$, and $T_{\mathrm{tar}}$ of Combinatorial Shortest Path on Graph Associahedra. To this end, we construct a polymatroid $(V, f)$ satisfying the following conditions.

1. $\mathbf{B}(f)$ is a realization of the $G$-associahedron.
2. For each subset $X \subseteq V$, we can evaluate the value $f(X)$ in time bounded by a polynomial in the size of $G$.
3. We can find the extreme points $x_{\mathrm{ini}}, x_{\mathrm{tar}}$ of $\mathbf{B}(f)$ corresponding to $T_{\mathrm{ini}}, T_{\mathrm{tar}}$, respectively, in time bounded by a polynomial in the size of $G$.

We first argue that the conditions above suffice for our proof. Suppose the existence of a polymatroid $(V, f)$ with the properties above. Then, we may construct a polynomial-time algorithm for Combinatorial Shortest Path on Graph Associahedra with a fictitious polynomial-time algorithm for Combinatorial Shortest Path on Polymatroids as follows. Let $(G, T_{\mathrm{ini}}, T_{\mathrm{tar}})$ be an instance of Combinatorial Shortest Path on Graph Associahedra. From Properties 1 and 3, we can construct an instance $((V, f), x_{\mathrm{ini}}, x_{\mathrm{tar}})$ of Combinatorial Shortest Path on Polymatroids in polynomial time. By the fictitious polynomial-time algorithm, we can solve the instance in time bounded by a polynomial in $|V|$ and the number of oracle calls to $f$. By Property 2, this running time is bounded by a polynomial in $|V|$. Thus, we find a solution to $(G, T_{\mathrm{ini}}, T_{\mathrm{tar}})$ in polynomial time, and the proof is completed.

In our construction of such a polymatroid $(V, f)$, we use the realization of the $G$-associahedron by Devadoss [12], which can be described as follows. Let $T$ be an elimination tree of $G$. For each vertex $v \in V$, we define $T(v)$ as the vertex set of the subtree of $T$ rooted at $v$. Then, we define the vector $x^T \in \mathbb{R}^V$ by choosing the coordinate $x^T(v)$ at every vertex of $v$ from the leaves to the root according to the following rule.

- If $v$ is a leaf of $T$, then we define $x^T(v) := 0$.
- If $v$ is not a leaf of $T$, then we define $x^T(v)$ so that

$$\sum_{u\in T(v)} x^T(u) = 3^{|T(v)|-2}.$$

Define $\mathcal{E} := \{x^T \mid T \text{ is an elimination tree of } G\}$. Then, Devadoss [12] proved that the convex hull of $\mathcal{E}$ is a realization of the $G$-associahedron, and for each elimination tree $T$ of $G$, the point $x^T$ is an extreme point of the $G$-associahedron.

In our proof, we define the function $f\colon 2^V \to \mathbb{R}$ by

$$f(X) := 3^{|V|-2} - \sum_{C \in \mathcal{C}^*(X)} 3^{|C|-2}$$

for each subset $X \subseteq V$, where $\mathcal{C}^*(X)$ is the family of connected components of $G - X$ with at least two vertices.

Properties 2 and 3 above are immediate: it is not difficult to see that we can evaluate the values of the function $f$ in time bounded by a polynomial in the size of $G$; we can construct $x_{\text{ini}}$ and $x_{\text{tar}}$ from $T_{\text{ini}}$ and $T_{\text{tar}}$, respectively, as $x_{\text{ini}} = x^{T_{\text{ini}}}$ and $x_{\text{tar}} = x^{T_{\text{tar}}}$. In the full version, we prove that $(V, f)$ is a polymatroid and $\mathbf{B}(f)$ coincides with the convex hull of $\mathcal{E}$. This completes the reduction. Therefore, Theorem 2 follows from Theorem 1.

## 6 Conclusion

We prove that the combinatorial shortest path computation is hard on graph associahedra and base polytopes of polymatroids. This evaporates our hope for resolving an open problem to obtain a polynomial-time algorithm for finding a shortest flip sequence between two triangulations of convex polygons and the rotation distance between two binary trees by generalizing the setting to graph associahedra. However, that open problem is still open, and we should pursue another way of attacking it.

## References


1. Oswin Aichholzer, Wolfgang Mulzer, and Alexander Pilz. Flip distance between triangulations of a simple polygon is NP-complete. *Discret. Comput. Geom.*, 54(2):368–389, 2015. `doi:10.1007/s00454-015-9709-7`.
2. Benjamin Aram Berendsohn. The diameter of caterpillar associahedra. In Artur Czumaj and Qin Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2022, June 27-29, 2022, Tórshavn, Faroe Islands*, volume 227 of *LIPIcs*, pages 14:1–14:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.SWAT.2022.14`.
3. Maike Buchin, Anna Lubiw, Arnaud de Mesmay, and Saul Schleimer. Computation and reconfiguration in low-dimensional topological spaces (Dagstuhl Seminar 22062). *Dagstuhl Reports*, 12(2):17–66, 2022. `doi:10.4230/DagRep.12.2.17`.
4. Jean Cardinal, Stefan Langerman, and Pablo Pérez-Lantero. On the diameter of tree associahedra. *Electron. J. Comb.*, 25(4):4, 2018. `doi:10.37236/7762`.
5. Jean Cardinal, Arturo I. Merino, and Torsten Mütze. Efficient generation of elimination trees and graph associahedra. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2128–2140. SIAM, 2022. `doi:10.1137/1.9781611977073.84`.
6. Jean Cardinal, Lionel Pournin, and Mario Valencia-Pabon. Diameter estimates for graph associahedra. *Ann. Comb.*, 26:873–902, 2022. `doi:10.1007/s00026-022-00598-z`.
7. Jean Cardinal, Lionel Pournin, and Mario Valencia-Pabon. The rotation distance of brooms, 2022. `doi:10.48550/arXiv.2211.07984`.
8. Michael Carr and Satyan L. Devadoss. Coxeter complexes and graph-associahedra. *Topology and its Applications*, 153(12):2155–2168, 2006. `doi:10.1016/j.topol.2005.08.010`.
9. Cesar Ceballos, Francisco Santos, and Günter M. Ziegler. Many non-equivalent realizations of the associahedron. *Combinatorica*, 35(5):513–551, 2015. `doi:10.1007/s00493-014-2959-9`.
10. Sean Cleary and Katherine St. John. Rotation distance is fixed-parameter tractable. *Inf. Process. Lett.*, 109(16):918–922, 2009. `doi:10.1016/j.ipl.2009.04.023`.

11 Sean Cleary and Katherine St. John. A linear-time approximation algorithm for rotation distance. *J. Graph Algorithms Appl.*, 14(2):385–390, 2010. doi:10.7155/jgaa.00212.

12 Satyan L. Devadoss. A realization of graph associahedra. *Discrete Mathematics*, 309(1):271–276, 2009. doi:10.1016/j.disc.2007.12.092.

13 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In Richard Guy, Haim Hanani, Norbert Sauer, and Johanan Schönheim, editors, *Combinatorial Structures and their Applications*, pages 69–87. Gordon and Breach, New York, NY, 1970.

14 Uriel Feige and Mohammad Mahdian. Finding small balanced separators. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 375–384. ACM, 2006. doi:10.1145/1132516.1132573.

15 Alan M. Frieze and Shang-Hua Teng. On the complexity of computing the diameter of a polytope. *Comput. Complex.*, 4:207–219, 1994. doi:10.1007/BF01206636.

16 Satoru Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, Amsterdam, The Netherlands, 2nd edition, 2005.

17 Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors. *Handbook of Discrete and Computational Geometry, Third Edition*. CRC Press LLC, 2017.

18 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.

19 Volker Kaibel and Marc E. Pfetsch. Some algorithmic problems in polytope theory. In Michael Joswig and Nobuki Takayama, editors, *Algebra, Geometry, and Software Systems [outcome of a Dagstuhl seminar]*, pages 23–47. Springer, 2003. doi:10.1007/978-3-662-05148-1_2.

20 Iyad A. Kanj, Eric Sedgwick, and Ge Xia. Computing the flip distance between triangulations. *Discret. Comput. Geom.*, 58(2):313–344, 2017. doi:10.1007/s00454-017-9867-x.

21 Haohong Li and Ge Xia. An $\mathcal{O}(3.82^k)$ time FPT algorithm for convex flip distance. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 44:1–44:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.STACS.2023.44.

22 Anna Lubiw and Vinayak Pathak. Flip distance between two triangulations of a point set is NP-complete. *Comput. Geom.*, 49:17–23, 2015. doi:10.1016/j.comgeo.2014.11.001.

23 Joan M. Lucas. An improved kernel size for rotation distance in binary trees. *Inf. Process. Lett.*, 110(12-13):481–484, 2010. doi:10.1016/j.ipl.2010.04.022.

24 Thibault Manneville and Vincent Pilaud. Graph properties of graph associahedra. *Seminaire Lotharingien de Combinatoire*, 73:B73d, 2015.

25 Alexander Pilz. Flip distance between triangulations of a planar point set is APX-hard. *Comput. Geom.*, 47(5):589–604, 2014. doi:10.1016/j.comgeo.2014.01.001.

26 Alex Postnikov, Victor Reiner, and Lauren Williams. Faces of generalized permutohedra. *Doc. Math.*, 13:207–273, 2008.

27 Lionel Pournin. The diameter of associahedra. *Advances in Mathematics*, 259:13–42, 2014. doi:10.1016/j.aim.2014.02.035.

28 Lionel Pournin. The asymptotic diameter of cyclohedra. *Israel Journal of Mathematics*, 219(2):609–635, 2017. doi:10.1007/s11856-017-1492-0.

29 Laura Sanità. The diameter of the fractional matching polytope and its hardness implications. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 910–921. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00090.

30 Daniel D. Sleator, Robert E. Tarjan, and William P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc.*, 1:647–681, 1988. doi:10.1090/S0894-0347-1988-0928904-4.

31 Donald M. Topkis. Adjacency on polymatroids. *Math. Program.*, 30(2):229–237, 1984. doi:10.1007/BF02591887.