COURCELLE'S THEOREM FOR TRIANGULATIONS

BENJAMIN A. BURTON AND RODNEY G. DOWNEY

Abstract. In graph theory, Courcelle's theorem essentially states that, if an algorithmic problem can be formulated in monadic second-order logic, then it can be solved in linear time for graphs of bounded treewidth. We prove such a metatheorem for a general class of triangulations of arbitrary fixed dimension d, including all triangulated d-manifolds: if an algorithmic problem can be expressed in monadic second-order logic, then it can be solved in linear time for triangulations whose dual graphs have bounded treewidth.

We apply our results to 3-manifold topology, a setting with many difficult computational problems but very few parameterised complexity results, and where treewidth has practical relevance as a parameter. Using our metatheorem, we recover and generalise earlier fixed-parameter tractability results on taut angle structures and discrete Morse theory respectively, and prove a new fixed-parameter tractability result for computing the powerful but complex Turaev-Viro invariants on 3-manifolds.

1. Introduction

Parameterised complexity is a relatively new and highly successful framework for understanding the computational complexity of "hard" problems for which we do not have a polynomial-time algorithm [14]. The key idea is to measure the complexity not just in terms of the input size (the traditional approach), but also in terms of additional *parameters* of the input or of the problem itself. The result is that, even if a problem is (for instance) NP-hard, we gain a richer theoretical understanding of those classes of inputs for which the problem is still tractable, and we acquire new practical tools for solving the problem in real software.

For example, finding a Hamiltonian cycle in an arbitrary graph is NP-complete, but for graphs of fixed treewidth $\leq k$ it can be solved in linear time in the input size [14]. In general, a problem is called *fixed-parameter tractable* in the parameter k if, for any class of inputs where k is universally bounded, the running time becomes polynomial in the input size.

Treewidth in particular (which roughly measures how "tree-like" a graph is [29]) is extremely useful as a parameter. A great many graph problems are known to be fixed-parameter tractable in the treewidth, in a large part due to Courcelle's celebrated "metatheorem" [11, 12]: for *any* decision problem P on graphs, if P can be framed using monadic second-order logic, then P can be solved in *linear time* for graphs of universally bounded treewidth $\leq k$.

2000 *Mathematics Subject Classification.* Primary 57Q15, 68Q25; Secondary 68W05.

Key words and phrases. Triangulations, parameterised complexity, 3-manifolds, discrete Morse theory, Turaev-Viro invariants.

The first author is supported by the Australian Research Council under the Discovery Projects funding scheme (projects DP1094516, DP110101104), and the second author is supported by the Marsden fund of New Zealand.

The motivation behind this paper is to develop the tools of parameterised complexity for systematic use in the field of geometric topology, and in particular for 3-manifold topology. This is a field with natural and fundamental algorithmic problems, such as determining whether two knots or two triangulations are topologically equivalent, and in three dimensions such problems are often decidable but extremely complex [28].

Parameterised complexity is appealing as a theoretical framework for identifying when "hard" topological problems can be solved quickly. Unlike average-case complexity or generic complexity, it avoids the need to work with *random inputs*—something that still poses major difficulties for 3-manifold topology [15]. The viability of this framework is shown by recent parameterised complexity results in topological settings such as knot polynomials [25, 26], angle structures [8], discrete Morse theory [6], and the enumeration of 3-manifold triangulations [7].

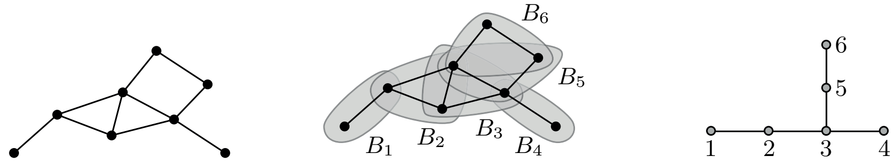
The treewidth parameter plays a key role in all of the aforementioned results. For topological problems whose input is a triangulation $\mathcal{T}$, we measure the treewidth of the *dual graph* $\mathcal{D}(\mathcal{T})$, whose nodes describe top-dimensional simplices of $\mathcal{T}$, and whose arcs show how these simplices are joined together along their facets. In 3-manifold topology this parameter has a natural interpretation, and there are common settings in which the treewidth remains small; see Section 5 for details.

Our main result in this paper is a Courcelle-like metatheorem for use with triangulations. Specifically, in Section 4 we describe a form of monadic second-order logic for use with triangulations of fixed dimension d, and we show that all problems expressible in this logical framework are fixed-parameter tractable in the treewidth of the dual graph of the input triangulation (Theorem 4.8).

Section 5 gives several applications of this metatheorem. We recover earlier results on taut angle structures [8] and discrete Morse theory [6], generalise the latter result to arbitrary dimension (Theorem 5.7), and prove a new result on computing the Turaev-Viro invariants of 3-manifolds (Theorem 5.9). These new results on discrete Morse theory and Turaev-Viro invariants have significant practical potential; see Section 5 for further discussion.

We prove our main result in two stages. In Section 3 we translate several variants of Courcelle's theorem from simple graphs to the more flexible setting of edge-coloured graphs. In Section 4 we show how to use an edge-coloured graph to encode the full structure of a triangulation (using a coloured variant of the well-known *Hasse diagram*), and using this we translate the variants of Courcelle's theorem up to the final setting of triangulations.

We emphasise that our results depend crucially on how we define a triangulation. We do not allow arbitrary simplicial complexes, where low-dimensional faces can be joined or "pinched" together independently of the larger simplices to which they belong. Instead our triangulations are formed purely by joining together d-simplices along their $(d-1)$-dimensional facets. This definition is flexible enough to encompass any reasonable concept of a triangulated d-manifold. Indeed, it covers structures more general than simplicial complexes, such as the highly efficient *one-vertex triangulations* [20] and *ideal triangulations* [30] favoured by many 3-manifold topologists.



(a) A simple graph G (b) The bags of a tree decomposition (c) The underlying tree

Figure 1. A simple graph and a corresponding tree decomposition

2. Preliminaries

2.1. **Treewidth.** Throughout this paper we work with several common classes of graphs; we briefly outline these classes here. We use the terms *node* and *arc* when working with graphs to avoid confusion with the vertices and edges of triangulations.

A *simple graph* $G = (V, E)$ is a finite set V of nodes and a finite set E of arcs, where each arc is an unordered pair $\{v, w\}$ of distinct nodes $v, w \in V$ (so we do not allow loops that join a node with itself, or multiple arcs between the same two nodes). A *multigraph* $G = (V, E)$ is a finite set V of nodes and a finite multiset of arcs, where each arc is an unordered pair $\{v, w\}$ of nodes $v, w \in V$ and we allow $v = w$ (so loops and multiple arcs are allowed). An *edge-coloured graph* $G = (V, E, C)$ is a finite set V of nodes, a finite set C of *colours* and a finite set E of arcs, where each arc is a pair $(\{v, w\}, c)$ with $v, w \in V$, $c \in C$ and $v \neq w$ (i.e., each arc is given a colour, loops are not allowed, and multiple arcs between the same two nodes are only allowed if they are assigned different colours). Unless otherwise specified, any general statement about *graphs* refers to all of these classes.

For any graph G, the *size* of the graph is denoted by $|G|$. The size counts the total number of nodes and arcs, i.e., $|G| = |V| + |E|$.

The treewidth of a graph G, introduced by Robertson and Seymour [29], essentially measures how far G is from being a tree: any tree will have treewidth 1 (the smallest possible), and a complete graph will have treewidth $|V| - 1$ (the largest possible for a given V). Graphs of small treewidth are often easier to work with, as Courcelle's theorem (described below) so strikingly shows. The full definition is as follows.

Given a simple graph or multigraph $G = (V, E)$, a *tree decomposition* of G consists of a (finite) tree T and *bags* $B_\tau \subseteq V$ for each node τ of T that satisfy the following constraints:

- each $v \in V$ belongs to some bag B_τ;
- for each arc of G, its two endpoints v, w belong to some common bag B_τ;
- for each $v \in V$, the bags containing v correspond to a connected subtree of T.

The *width* of this tree decomposition is $\max |B_\tau| - 1$, and the *treewidth* of G is the smallest width of any tree decomposition of G, which we denote by $\mathrm{tw}(G)$. Figure 1 illustrates a simple graph G and a corresponding tree decomposition of width 2.

Although computing treewidth is NP-complete, it is also fixed-parameter tractable:

Theorem 2.1 (Bodlaender [3])**.** *There exists a computable function f and an algorithm which, given a simple graph $G = (V, E)$ of treewidth $k = \mathrm{tw}(G)$, can compute a tree decomposition of width k in time $f(k) \cdot |V|$.*

More precisely, we have $f(k) \in 2^{k^{O(1)}}$; see [3, 16] for details.

2.2. Monadic second-order logic and Courcelle's theorem. Monadic second-order logic, or MSO logic, is our framework for making statements about graphs. Here we give a brief overview of the key concepts as they appear in the context of simple graphs; see a standard text such as [16] for further details. What we describe here is sometimes called *extended* MSO logic, or MS_2 logic; this highlights the fact that we can access arcs directly through variables and sets, and not just indirectly through a binary relation on nodes.

MSO logic supports:

- all of the standard boolean operations of propositional logic: $\wedge$ (and), $\vee$ (or), $\neg$ (negation), $\rightarrow$ (implication), and so on;
- variables to represent nodes, arcs, sets of nodes, or sets of arcs of a graph;
- the standard quantifiers from first-order logic: $\forall$ (the universal quantifier), and $\exists$ (the existential quantifier), which may be applied to any of these variable types;
- the binary equality relation $=$, which can be applied to nodes, arcs, sets of nodes, or sets of arcs;
- the binary inclusion relation $\in$, which can relate nodes to sets of nodes, or arcs to sets of arcs;
- the binary incidence relation $inc(e, v)$, which encodes the fact that e is an arc, v is a node, and v is one of the two endpoints of e;
- the binary adjacency relation $adj(v, v')$, which encodes the fact that v and v' are the two endpoints of some common arc.

By convention, we use lower-case letters $v, w, \ldots$ to represent nodes and $e, f, \ldots$ to represent arcs, and upper-case letters $V, W, \ldots$ to represent sets of nodes and $E, F, \ldots$ to represent sets of arcs.

Note that sets are simply a convenient representation of unary relations on nodes and arcs. A distinguishing feature of MSO logic is that we can quantify over unary relations (i.e., we can quantify over set variables as outlined above).

We use the notation $\phi(x_1, \ldots, x_t)$ to denote an MSO formula with t free variables (i.e., variables not bound by $\forall$ or $\exists$ quantifiers). An *MSO sentence* is an MSO formula with no free variables at all. If G is a simple graph and ϕ is a MSO sentence, we use the notation $G \models \phi$ to indicate that the interpretation of ϕ in the graph G is a true statement.

We give three variants of Courcelle's theorem, which relate to algorithms for (i) *decision* problems on graphs; (ii) *optimising* quantities on graphs; and (iii) *evaluating* functions on graphs. Each variant works with MSO formulae in different ways, which we now outline in turn.

2.2.1. *Decision problems.*

Example 2.2 (3-colourability)**.** The following MSO sentence expresses the fact that a simple graph is 3-colourable:

$$\begin{aligned}
&\exists V_1 \exists V_2 \exists V_3\ \forall v \forall w \\
&(v \in V_1 \vee v \in V_2 \vee v \in V_3) \wedge \\
&\neg((v \in V_1 \wedge v \in V_2) \vee (v \in V_2 \wedge v \in V_3) \vee (v \in V_1 \wedge v \in V_3)) \wedge \\
&adj(v, w) \rightarrow \neg((v \in V_1 \wedge w \in V_1) \vee (v \in V_2 \wedge w \in V_2) \vee (v \in V_3 \wedge w \in V_3)).
\end{aligned}$$

The sets V_1, V_2, V_3 indicate which nodes are assigned each of the three available colours. The second and third lines ensure that V_1, V_2 and V_3 partition the nodes, and the final line ensures that any two adjacent nodes are coloured differently.

Theorem 2.3 (Courcelle [11, 12])**.** *Given a simple graph $G = (V, E)$, its treewidth $k = \mathrm{tw}(G)$ and a fixed MSO sentence ϕ, there exists a computable function f and an algorithm for testing whether $G \models \phi$ that runs in time $f(k, |\phi|) \cdot |G|$.*

Corollary 2.4. *For any fixed MSO sentence ϕ and any class K of simple graphs with universally bounded treewidth, it is possible to test whether $G \models \phi$ for graphs $G \in K$ in time $O(|G|)$.*

In other words, testing for ϕ is linear-time fixed-parameter tractable in the treewidth. Note that, thanks to Theorem 2.1, we do not need to supply an explicit tree decomposition of the input graph G in advance.

For the remainder of this paper we will formulate our results using the language of Corollary 2.4, omitting explicit references to the underlying function f.

2.2.2. *Optimisation problems.* A *restricted MSO extremum problem* consists of an MSO formula $\phi(A_1, \ldots, A_t)$ with free set variables $A_1, \ldots, A_t$, and a rational linear function $g(x_1, \ldots, x_t)$. Its interpretation is as follows: given a simple graph G as input, we are asked to minimise $g(|A_1|, \ldots, |A_t|)$ over all sets $A_1, \ldots, A_t$ for which $G \models \phi(A_1, \ldots, A_t)$, where $|A_i|$ as usual denotes the number of objects in the set A_i.

Example 2.5 (Dominating set)**.** The well-known problem *dominating set* asks for the smallest set of nodes D in a given graph G for which every node in G is either in D or adjacent to some node in D.

To formulate this as a restricted MSO extremum problem, we use a single free set variable D, and minimise the linear function $g(|D|) = |D|$ under the following MSO constraint:

$$\forall v \; \exists w \; (v \in D) \vee (w \in D \wedge adj(v, w)).$$

Courcelle's theorem has been extended to work with such problems:

Theorem 2.6 (Arnborg, Lagergren and Seese [2])**.** *For any restricted MSO extremum problem P and any class K of simple graphs with universally bounded treewidth, it is possible to solve P for graphs $G \in K$ in time $O(|G|)$ under the uniform cost measure.*

Recall that the *uniform cost measure* assumes that elementary arithmetic operations run in constant time; see a standard text such as [1] for details. Again, Theorem 2.1 ensures that we do not need to supply an explicit tree decomposition in advance.

Arnborg et al. [2] prove more general results: for instance, they allow evaluation functions on weighted graphs (where integer or rational weights on the nodes and/or arcs are supplied with the problem instance), they allow additional constants to be given with the problem instance, and they discuss non-linear extremum problems and enumeration problems. For simplicity we restrict our attention here to the restricted class of extremum problems as described above.

2.2.3. *Evaluation problems.* We move now to evaluation problems, which are generalised counting problems: essentially, we assign a value to each solution to some MSO-defined problem on a graph, and then sum these values over all solutions. Evaluation problems come in both additive or multiplicative variants, and counting problems correspond to multiplicative variants in which all solutions have value 1.

More precisely, an *MSO evaluation problem* is defined as follows. The problem consists of an MSO formula $\phi(A_1, \ldots, A_t)$ with t free set variables $A_1, \ldots, A_t$. The input to the problem is a simple graph $G = (V, E)$, together with t weight functions $w_1, \ldots, w_t \colon V \sqcup E \to R$ on nodes and/or arcs, where R is some ring or field. The problem then asks us to compute one of the quantities

$$\sum_{G \models \phi(A_1,\ldots,A_t)} \left\{ \sum_{i=1}^{t} \sum_{x_i \in A_i} w_i(x_i) \right\} \quad \text{or} \quad \sum_{G \models \phi(A_1,\ldots,A_t)} \left\{ \prod_{i=1}^{t} \prod_{x_i \in A_i} w_i(x_i) \right\};$$

we refer to these two variants as *additive* and *multiplicative* evaluation problems respectively. For both problems, the outermost sum is over all solutions $A_1, \ldots, A_t$ that satisfy the MSO formula ϕ on the graph G.

We now come to our third variant of Courcelle's theorem, which extends earlier work of Courcelle and Mosbah [10] and Arnborg et al. [2].

Theorem 2.7 (Courcelle, Makowsky and Rotics [9]). *For any MSO evaluation problem P and any class K of simple graphs with universally bounded treewidth, it is possible to solve P for graphs $G \in K$ in time $O(|G|)$ under the uniform cost measure.*

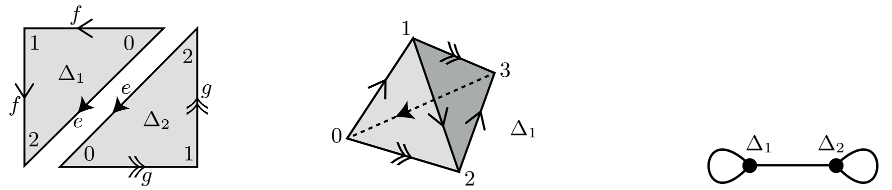
Here we interpret the uniform cost measure to allow constant-time arithmetic operations over the ring or field R. Courcelle et al. [9] only prove this explicitly for $t = 1$ free variable only; however, they note that the generalisation to a sequence of t free variables (for fixed t) is obvious.

2.3. **Triangulations.** We now describe the general class of d-dimensional triangulations upon which our metatheorem operates. In essence, these triangulations are formed by identifying (or "gluing") facets of d-simplices in pairs. This definition does not cover all simplicial complexes (in which lower-dimensional faces can also be identified independently), but it does encompass any reasonable definition of a triangulated d-manifold; moreover, it allows more general structures that simplicial complexes do not, such as the highly efficient "1-vertex triangulations" and "ideal triangulations" that are often found in algorithmic 3-manifold topology [20, 30]. The details follow.

Let $d \in \mathbb{N}$. A *d-dimensional triangulation* consists of a collection of abstract d-simplices $\Delta_1, \ldots, \Delta_n$, some or all of whose facets[1] are affinely identified (or "glued") in pairs. Each facet F of a d-simplex may only be identified with at most one other facet F' of a d-simplex; this may be another facet of the *same* d-simplex, but it cannot be F itself. Those facets that are not identified with any other facet together form the *boundary* of the triangulation.

Consider any integer i with $0 \leq i < d$. There are $\binom{d+1}{i+1}$ distinct i-faces of each simplex $\Delta_1, \ldots, \Delta_n$. As a consequence of the facet identifications, some of these i-faces become identified with each other; we refer to each class of identified i-faces as a single *i-face of the triangulation.* As usual, 0-faces and 1-faces are called *vertices*

[1] Recall that a *facet* of a d-simplex is a $(d-1)$-dimensional face.



(a) A Klein bottle $\mathcal{K}$ (b) A one-tetrahedron solid torus (c) The dual graph $\mathcal{D}(\mathcal{K})$

FIGURE 2. Examples of triangulations

and *edges* respectively. A *simplex of the triangulation* explicitly refers to one of the d-simplices $\Delta_1, \ldots, \Delta_n$ (not a smaller-dimensional face), and for convenience we also refer to these as *d-faces of the triangulation*.

A *d-manifold triangulation* is simply a d-dimensional triangulation whose underlying topological space is a d-manifold when using the quotient topology.

By convention, we label the vertices of each simplex as $0, \ldots, d$. We also arbitrarily label the vertices of each i-face of the triangulation as $0, \ldots, i$ (so, for instance, for $i = 1$ this corresponds to placing an arbitrary direction on each edge). Note that there are many possible ways in which the vertex labels of an i-face of the triangulation might correspond to vertex labels on the constituent simplices.

Example 2.8. Figure 2(a) illustrates a 2-manifold triangulation with $n = 2$ simplices whose underlying topological space is a Klein bottle. As indicated by the arrowheads, we identify the following pairs of facets (i.e., edges):

$$\Delta_1 : 02 \longleftrightarrow \Delta_2 : 20, \quad \Delta_1 : 01 \longleftrightarrow \Delta_1 : 12, \quad \Delta_2 : 01 \longleftrightarrow \Delta_2 : 12.$$

The resulting triangulation has one vertex (since all three vertices of Δ_1 and all three vertices of Δ_2 become identified together), and three edges (labelled e, f, g in the diagram).

If we label each edge so that vertices 0 and 1 are at the base and the tip of the arrow respectively, then vertex 0 of edge e corresponds to the individual triangle vertices $\Delta_1 : 0$ and $\Delta_2 : 2$, and vertex 1 of edge e corresponds to $\Delta_1 : 2$ and $\Delta_2 : 0$.

Figure 2(b) illustrates a 3-manifold triangulation with just $n = 1$ simplex whose underlying topological space is the solid torus $B^2 \times S^1$ [20]. Here we identify facets $\Delta_1 : 012 \longleftrightarrow \Delta_1 : 123$, and leave the other two facets as boundary. The resulting triangulation has just one vertex, three edges, and three 2-faces.

Let $\mathcal{T}$ be a d-dimensional triangulation. The *size* of $\mathcal{T}$, denoted $|\mathcal{T}|$, is the number of simplices (i.e., d-faces) in $\mathcal{T}$. Note that the total number of faces of *any* dimension is at most $2^{d+1}|\mathcal{T}|$, and is hence linear in $|\mathcal{T}|$ for fixed dimension d.

The *dual graph* of $\mathcal{T}$, denoted $\mathcal{D}(\mathcal{T})$, is the multigraph whose nodes correspond to simplices and whose arcs correspond to identified pairs of facets. In particular, $\mathcal{D}(\mathcal{T})$ has precisely $|\mathcal{T}|$ nodes, and each node has degree at most $d+1$. Loops may occur in $\mathcal{D}(\mathcal{T})$ if two facets of the same simplex are identified, and parallel arcs may occur if different facets of some simplex Δ_i are identified with (different) facets of the same simplex Δ_j. Figure 2(c) illustrates the dual graph of the Klein bottle from Example 2.8.

3. Edge-coloured graphs

Here we prove that the three variants of Courcelle's theorem on simple graphs (Corollary 2.4, Theorem 2.6 and Theorem 2.7) also hold for edge-coloured graphs with a fixed number of colours. Although the results here are unsurprising and draw on standard techniques, they are necessary as a stepping stone for Section 4, where we use edge-coloured graphs to encode the full structure of a triangulation.

3.1. **MSO logic on edge-coloured graphs.** For any fixed number of colours $k \in \mathbb{N}$, we can extend MSO logic to the setting of edge-coloured graphs as follows. We allow all of the constructs of MSO logic on simple graphs, as outlined in Section 2.2. In additional, we support:

- k unary colour relations $col_1(e), \ldots, col_k(e)$, where $col_i(e)$ encodes the fact that e is an arc of the ith colour;
- k binary adjacency relations $adj_1(v, v'), \ldots, adj_k(v, v')$, where $adj_i(v, v')$ encodes the fact that v and v' are the two endpoints of some common arc of the ith colour.

Note that the relations adj_i are for convenience only, and can easily be encoded using the other constructs available.

As usual, if $G = (V, E, C)$ is an edge-coloured graph with $|C| = k$ colours, and ϕ is an MSO sentence using the additional constructs outlined above, the notation $G \models \phi$ indicates that the interpretation of ϕ in the graph G is a true statement.

We define extremum and evaluation problems exactly as before: a restricted MSO extremum problem minimises a rational function over sizes of set variables as in Section 2.2.2, and an MSO evaluation problem computes an additive or multiplicative quantity using weights on the nodes and/or arcs as in Section 2.2.3.

3.2. **Metatheorems on edge-coloured graphs.** Here we translate Courcelle's theorem and its variants to edge-coloured graphs. The basic idea is, for any edge-coloured graph G, to construct an associated simple graph $\overline{G}$ for which:

- the size $|\overline{G}|$ is linear in $|G|$;
- the treewidth $\mathrm{tw}(\overline{G})$ is at worst linear in $\mathrm{tw}(G)$; and
- MSO formulae on G translate to MSO formulae on $\overline{G}$.

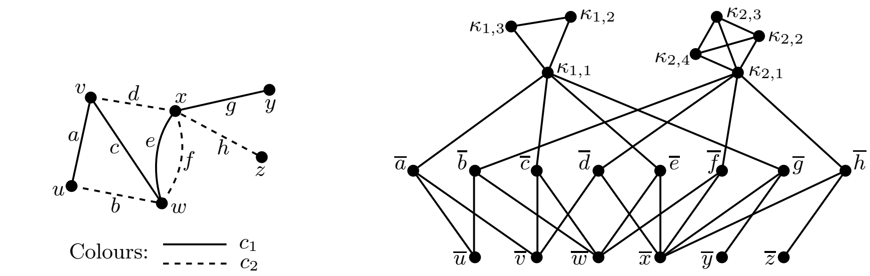
The coloured variants of Courcelle's theorem then fall out naturally, as seen in Theorem 3.5. The details follow.

Construction 3.1. *Let $G = (V, E, C)$ be any edge-coloured graph, with colours $C = \{c_1, \ldots, c_k\}$. We construct the associated simple graph $\overline{G}$ as follows:*

- *For each colour c_i we add $i + 2$ nodes $\kappa_{i,1}, \ldots, \kappa_{i,i+2}$ to $\overline{G}$ plus all $\binom{i+2}{2}$ possible arcs between them. In other words, we insert k cliques of sizes $3, \ldots, k + 2$.*
- *For each node $v \in V$ we add a corresponding node $\overline{v}$ to $\overline{G}$.*
- *For each arc $e = \{(v, w), c\} \in E$ we add a corresponding node $\overline{e}$ to $\overline{G}$, plus three arcs $\{\overline{e}, \overline{v}\}$, $\{\overline{e}, \overline{w}\}$, and $\{\overline{e}, \kappa_{c,1}\}$.*

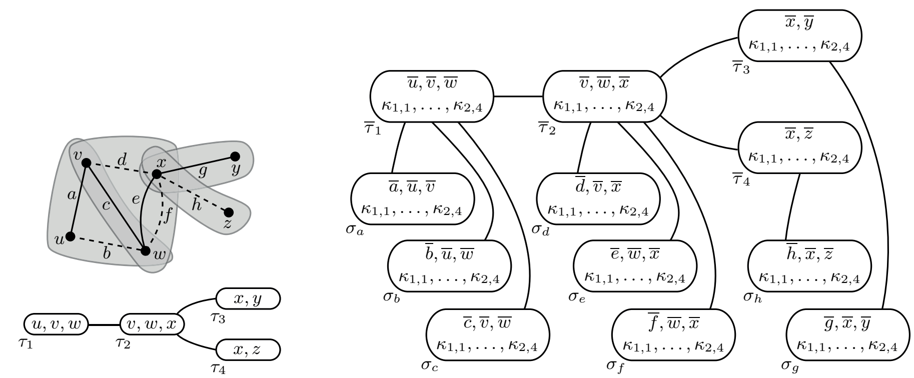
Figure 3 illustrates this construction. It is immediate from the definition of an edge-coloured graph (see Section 2.1) that $\overline{G}$ is indeed a simple graph as claimed.

Lemma 3.2. *For any edge-coloured graph $G = (V, E, C)$ with $|C| = k$ colours, we have $|\overline{G}| = |V| + 4|E| + \binom{k+4}{3} - 4$.*



(a) The edge-coloured graph G (b) The associated simple graph $\overline{G}$

FIGURE 3. Converting an edge-coloured graph into a simple graph



(a) A tree decomposition of G (b) The corresponding tree decomposition of $\overline{G}$

FIGURE 4. Bounding the treewidth of the simple graph $\overline{G}$

Proof. It is clear by construction that the number of nodes of $\overline{G}$ is $|V| + |E| + \sum_{i=3}^{k+2} i = |V| + |E| + \binom{k+3}{2} - 3$, and the number of arcs of $\overline{G}$ is $3|E| + \sum_{i=3}^{k+2} \binom{i}{2} = 3|E| + \binom{k+3}{3} - 1$. The result then follows from the identity $\binom{k+3}{2} + \binom{k+3}{3} = \binom{k+4}{3}$. $\square$

Lemma 3.3. *For any edge-coloured graph $G = (V, E, C)$ with $|C| = k$ colours, we have* $\operatorname{tw}(\overline{G}) \leq \operatorname{tw}(G) + \binom{k+3}{2} - 1$.

Proof. Suppose we have a tree decomposition of G with underlying tree T. From this we build a tree decomposition of $\overline{G}$ with underlying tree $\overline{T}$ as follows (see Figure 4 for an illustration):

(1) In the original tree decomposition, let τ be a node of T and let $B_\tau = \{v_1, \ldots, v_m\}$ be the corresponding bag. In the new tree decomposition, we create a corresponding node $\overline{\tau}$ of $\overline{T}$, whose bag $B_{\overline{\tau}}$ contains the corresponding nodes $\overline{v}_1, \ldots, \overline{v}_m$ plus all of the clique nodes $\kappa_{i,j}$.
(2) For each arc $e = \{(v, w), c\}$ of G we create a new node σ_e of $\overline{T}$, whose corresponding bag B_{σ_e} contains $\overline{e}$, $\overline{v}$ and $\overline{w}$ plus all of the clique nodes $\kappa_{i,j}$.
(3) We connect the nodes of $\overline{T}$ as follows. For each pair of adjacent nodes τ_1, τ_2 of T, we connect the corresponding nodes $\overline{\tau}_1, \overline{\tau}_2$ in $\overline{T}$. For each arc

$e = \{(v, w), c\}$ of G, we connect the corresponding node σ_e of $\overline{T}$ to one (and only one) arbitrarily chosen node $\overline{\tau}$ for which $\overline{v}, \overline{w} \in B_{\overline{\tau}}$.

Note that the choice in step (3) is always possible: since we began with a tree decomposition of G, some node τ of T must have $v, w \in B_\tau$ and thus $\overline{v}, \overline{w} \in B_{\overline{\tau}}$.

It is simple to show that this construction does indeed yield a tree decomposition of $\overline{G}$: all necessary properties of a tree decomposition follow directly from the construction above and the fact that we began with a tree decomposition of G.

It remains to compute the bag sizes in our new tree decomposition. For each bag $B_{\overline{\tau}}$ created in step (1), we have $|B_{\overline{\tau}}| = |B_\tau| + \sum_{i=3}^{k+2} i = |B_\tau| + \binom{k+3}{2} - 3$. For each bag B_{σ_e} created in step (2), we have $|B_{\sigma_e}| = 3 + \sum_{i=3}^{k+2} i = \binom{k+3}{2}$. Therefore $\text{tw}(\overline{G}) \leq \max\left\{\text{tw}(G) + \binom{k+3}{2} - 3, \binom{k+3}{2} - 1\right\} \leq \text{tw}(G) + \binom{k+3}{2} - 1$. □

Lemma 3.4. *For any fixed $k \in \mathbb{N}$, every MSO formula $\phi(x_1, \ldots, x_t)$ on edge-coloured graphs with k colours has a corresponding MSO formula $\overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ on simple graphs for which, for any edge-coloured graph $G(V, E, C)$ with $|C| = k$ colours:*

- *Any assignment of values to $x_1, \ldots, x_t$ for which $G \models \phi(x_1, \ldots, x_t)$ yields a corresponding assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ for which $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$, and this assignment is obtained as follows:*
  - *if $x_i = v$ for some node $v \in V$, then $\overline{x}_i = \overline{v}$;*
  - *if $x_i = e$ for some arc $e \in E$, then $\overline{x}_i = \overline{e}$ (note that $\overline{x}_i$ is a node of $\overline{G}$, not an arc);*
  - *if x_i is a set $\{v_i\} \subseteq V$ or $\{e_i\} \subseteq E$, then $\overline{x}_i$ is the corresponding set $\{\overline{v}_i\}$ or $\{\overline{e}_i\}$.*
- *Conversely, any assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ for which $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ is derived from an assignment of values to $x_1, \ldots, x_t$ for which $G \models \phi(x_1, \ldots, x_t)$, as described above.*

In particular, this means that if ϕ is a sentence then $G \models \phi$ if and only if $\overline{G} \models \overline{\phi}$, and if ϕ has free variables then the solutions to $G \models \phi(x_1, \ldots, x_t)$ are in bijection with the solutions to $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

Proof. The proof works in three stages: (i) we develop additional "helper" relations to constrain the roles that variables can play in $\overline{\phi}$; (ii) we translate each piece of the formula ϕ so that statements about G in ϕ become statements about $\overline{G}$ in $\overline{\phi}$, and thus solutions to $G \models \phi(x_1, \ldots, x_t)$ become solutions to $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$; and then (iii) we add additional constraints to $\overline{\phi}$ to avoid any additional (and unwanted) solutions to $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

In the first stage, we develop the following "helper" unary relations using MSO logic on simple graphs. These relations only make sense when interpreting $\overline{\phi}$ on a simple graph $\overline{G}$ that was built using Construction 3.1; in other contexts they are well-defined but meaningless.

- *is_node*(x) indicates that a variable in $\overline{\phi}$ represents a node of the original edge-coloured graph G; that is, $x = \overline{v}$ for some $v \in V$.
- *is_arc*(x) indicates that a variable in $\overline{\phi}$ represents an arc of the original edge-coloured graph G; that is, $x = \overline{e}$ for some arc $e \in E$.

- For each $i = 1, \ldots, k$, $is\_col_i(x)$ indicates that a variable in $\overline{\phi}$ represents one of the nodes of the clique representing the ith colour; that is, $x = \kappa_{i,j}$ for some j.

These relations are simple (but messy) to piece together using MSO logic. For each fixed i, the relation $is\_col_i(x)$ is true if and only if x represents a node that belongs to an i-clique but not an $(i+1)$-clique. The relation $is\_arc(x)$ is true if and only if x is a node that does not belong to a triangle, but is adjacent to a node that does. The relation $is\_node(x)$ is true if and only if x is a node for which none of $is\_arc(x)$ or $is\_col_i(x)$ are true.

In the second stage, we translate each piece of ϕ to a piece of $\overline{\phi}$ as follows:

- Each variable x_i in ϕ is replaced by $\overline{x}_i$ in $\overline{\phi}$.
- Standard logical operations (such as $\wedge$, $\vee$, $\neg$, $\rightarrow$ and so on), standard quantifiers ($\forall$ and $\exists$), and the equality and inclusion relations ($=$ and $\in$) are copied directly from ϕ to $\overline{\phi}$.
- The incidence relation $inc(e, v)$ in ϕ is replaced by the phrase $is\_arc(e) \wedge is\_node(v) \wedge adj(e, v)$ in $\overline{\phi}$.
- Each colour relation $col_i(e)$ in ϕ is replaced by the phrase $\exists v \;\; is\_arc(e) \wedge is\_col_i(v) \wedge adj(e, v)$ in $\overline{\phi}$.
- The adjacency relation $adj(v, v')$ in ϕ is replaced by the phrase $\exists e \;\; is\_node(v) \wedge is\_node(v') \wedge adj(v, e) \wedge adj(v', e) \wedge v \neq v'$ in $\overline{\phi}$.
- Each coloured adjacency relation $adj_i(v, v')$ in ϕ can be replaced with an equivalent statement in ϕ using $inc(\cdot, \cdot)$ and $col_i(\cdot)$, and then translated to $\overline{\phi}$ as described above.

It follows directly from this translation process that, if an assignment of values to $x_1, \ldots, x_t$ gives $G \models \phi(x_1, \ldots, x_t)$, then the corresponding assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ as described in the lemma statement gives $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

In the third stage, we eliminate any additional and unwanted solutions to $\overline{G} \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$, by insisting that for every variable x in ϕ, the translated variable $\overline{x}$ in $\overline{\phi}$ must still represent a node, arc, set of nodes or set of arcs in the source graph G. This ensures that logical statements about such variables $\overline{x}_i$ in $\overline{\phi}$ translate correctly back to logical statements about variables x_i in ϕ. Specifically, for each variable x in ϕ (either bound or free), we add the corresponding clause to $\overline{\phi}$:

$$is\_node(\overline{x}) \vee is\_arc(\overline{x}) \vee [\forall z \;\; z \in \overline{x} \rightarrow is\_node(z)] \vee [\forall z \;\; z \in \overline{x} \rightarrow is\_arc(z)]. \quad \square$$

Theorem 3.5. *Let K be any class of edge-coloured graphs with fixed colour set $C = \{c_1, \ldots, c_k\}$ and with universally bounded treewidth. Then:*

- *For any fixed MSO sentence ϕ, it is possible to test whether $G \models \phi$ for edge-coloured graphs $G \in K$ in time $O(|G|)$;*
- *For any restricted MSO extremum problem P, it is possible to solve P for edge-coloured graphs $G \in K$ in time $O(|G|)$ under the uniform cost measure;*
- *For any MSO evaluation problem P, it is possible to solve P for edge-coloured graphs $G \in K$ in time $O(|G|)$ under the uniform cost measure.*

Proof. Let P be any such problem, and let ϕ be its underlying MSO sentence or formula. We first use Lemma 3.4 to translate ϕ to $\overline{\phi}$, yielding a new problem $\overline{P}$ in the setting of simple graphs. Then, for any edge-coloured graph G given as input to P, we construct the simple graph $\overline{G}$ and solve $\overline{P}$ for $\overline{G}$ instead. Lemma 3.4 ensures that the solutions to both problems are the same.

For extremum problems, we note that the sizes of the sets in each solution to ϕ are equal to the sizes of the sets in the corresponding solution to $\overline{\phi}$ (i.e., the value of the extremum does not change). For evaluation problems, the input weight on a node v or arc e of G becomes the same input weight on the corresponding node $\overline{v}$ or $\overline{e}$ of $\overline{G}$; all remaining nodes and arcs of $\overline{G}$ are assigned trivial weights (0 or 1 according to whether the evaluation problem is additive or multiplicative).

It is clear from Construction 3.1 that we can build the simple graph $\overline{G}$ in time $O(|G|)$, and Lemmata 3.2 and 3.3 ensure that $\overline{G}$ has universally bounded treewidth and $O(|G|)$ size. The result now follows directly from the three original variants of Courcelle's theorem (Corollary 2.4, Theorem 2.6 and Theorem 2.7). □

4. Triangulations

In this section we prove our main result, that all three variants of Courcelle's theorem hold for triangulations of fixed dimension (Theorem 4.8).

4.1. **MSO logic on triangulations.**

Our first task is to extend MSO logic to the setting of d-dimensional triangulations, for fixed dimension $d \in \mathbb{N}$. Here we define MSO logic to support:

- all of the standard boolean operations of propositional logic ($\wedge$, $\vee$, $\neg$, $\rightarrow$, and so on);
- for each $i = 0, \ldots, d$, variables to represent i-faces of a triangulation, or sets of i-faces of a triangulation;
- the standard quantifiers ($\forall$ and $\exists$) and the binary equality and inclusion relations ($=$ and $\in$), which may be applied to any of these variable types;
- for each $i = 0, \ldots, d-1$ and for each ordered sequence $\pi_0, \ldots, \pi_i$ of distinct integers from $\{0, \ldots, d\}$, a subface relation $\leq_{\pi_0 \ldots \pi_i}$.

The relation $(f \leq_{\pi_0 \ldots \pi_i} s)$ indicates that f is an i-face of the triangulation, s is a simplex of the triangulation, and that f is identified with the subface of s formed by the simplex vertices $\pi_0, \ldots, \pi_i$, in a way that vertices $0, \ldots, i$ of the face f correspond to vertices $\pi_0, \ldots, \pi_i$ of the simplex s.

Example 4.1. Recall the Klein bottle example illustrated in Figure 2(a). Here the three edges e, f, g satisfy the subface relations

$$\begin{array}{lll} e \leq_{02} \Delta_1 & f \leq_{01} \Delta_1 & g \leq_{01} \Delta_2 \\ e \leq_{20} \Delta_2 & f \leq_{12} \Delta_1 & g \leq_{12} \Delta_2. \end{array}$$

The triangulation has only one vertex (since all vertices of Δ_1 and Δ_2 are identified together); call this vertex v. Then v satisfies all possible subface relations

$$\begin{array}{lll} v \leq_{0} \Delta_1 & v \leq_{1} \Delta_1 & v \leq_{2} \Delta_1 \\ v \leq_{0} \Delta_2 & v \leq_{1} \Delta_2 & v \leq_{2} \Delta_2. \end{array}$$

Notation. In general, we will adopt the convention that lower-case letters $s, t, \ldots$ represent simplices and $f^{(i)}, g^{(i)}, \ldots$ represent i-faces, and that upper-case letters $S, T, \ldots$ and $F^{(i)}, G^{(i)}, \ldots$ represent sets of simplices and sets of i-faces respectively.

Example 4.2 (Orientability)**.** Consider $d = 2$, i.e., the case of triangulated surfaces. We will construct an MSO sentence ϕ stating that a triangulation represents an *orientable* surface.

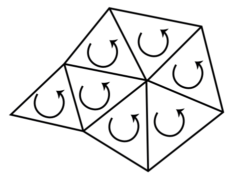


FIGURE 5. Adjacent triangles have compatible orientations

Recall that a 2-dimensional triangulation is orientable if and only if each triangle can be assigned an orientation (clockwise or anticlockwise) so that adjacent triangles have compatible orientations, as illustrated in Figure 5.

The sentence ϕ is given below. Here S_+ and S_- are variables denoting sets of simplices with clockwise and anticlockwise orientations respectively. To encode the compatibility constraint on adjacent triangles, we introduce variables u and v to represent two adjacent simplices, and $f^{(1)}$ to represent the common edge along which they are joined.

$$\begin{aligned}
&\exists S_+ \exists S_- \forall s\ \forall f^{(1)}\ \forall u\ \forall v \\
&(s \in S_+ \vee s \in S_-) \wedge \neg(s \in S_+ \wedge s \in S_-) \wedge \\
&\left[(f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{01} v \wedge u \neq v) \to ((u \in S_+ \wedge v \in S_-) \vee (u \in S_- \wedge v \in S_+))\right] \wedge \\
&\left[(f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{10} v) \qquad\quad \to ((u \in S_+ \wedge v \in S_+) \vee (u \in S_- \wedge v \in S_-))\right] \wedge \\
&\left[(f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{02} v) \qquad\quad \to ((u \in S_+ \wedge v \in S_+) \vee (u \in S_- \wedge v \in S_-))\right] \wedge \\
&\left[(f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{20} v) \qquad\quad \to ((u \in S_+ \wedge v \in S_-) \vee (u \in S_- \wedge v \in S_+))\right] \wedge \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\left[(f^{(1)} \leq_{21} u \wedge f^{(1)} \leq_{21} v \wedge u \neq v) \to ((u \in S_+ \wedge v \in S_-) \vee (u \in S_- \wedge v \in S_+))\right].
\end{aligned}$$

The first line of this sentence is just quantifiers, and the second line ensures that S_+ and S_- partition the simplices. The remaining lines encode the fact that adjacent simplices must have compatible orientations. They do this by iterating through all 6×6 possible ways in which $f^{(1)}$ could appear as both an edge of u and an edge of v (making u and v adjacent), and in each case a simple parity check determines whether u and v must have the same orientation (as in the case $f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{02} v$) or opposite orientations (as in the case $f^{(1)} \leq_{01} u \wedge f^{(1)} \leq_{20} v$).

For the six cases where u and v use the *same* subface relation (e.g., the first and last cases above), we add the additional requirement $u \neq v$ to ensure that triangles u and v lie on opposite sides of the edge $f^{(1)}$.

We return now to notation and definitions. If $\mathcal{T}$ is a d-dimensional triangulation and ϕ is an MSO sentence as outlined above, the notation $\mathcal{T} \models \phi$ indicates (as usual) that the interpretation of ϕ in the triangulation $\mathcal{T}$ is a true statement.

We define extremum and evaluation problems as before, though this time we must alter the latter definition so that weights are given to the faces of a triangulation (instead of nodes and arcs of a graph). Specifically, in the setting of MSO logic on d-dimensional triangulations:

- A *restricted MSO extremum problem* consists of an MSO formula $\phi(A_1, \ldots, A_t)$ with free set variables $A_1, \ldots, A_t$, and a rational linear function $g(x_1, \ldots, x_t)$. Its interpretation is as follows: given a d-dimensional triangulation $\mathcal{T}$ as input, we are asked to minimise $g(|A_1|, \ldots, |A_t|)$ over all sets $A_1, \ldots, A_t$ for which $\mathcal{T} \models \phi(A_1, \ldots, A_t)$.

- An *MSO evaluation problem* consists of an MSO formula $\phi(A_1, \ldots, A_t)$ with t free set variables $A_1, \ldots, A_t$. The input to the problem is a d-dimensional triangulation $\mathcal{T}$, together with t weight functions $w_1, \ldots, w_t \colon F_0 \sqcup \ldots \sqcup F_d \to R$, where F_i denotes the set of all i-faces of $\mathcal{T}$, and R is some ring or field. The problem then asks us to compute one of the quantities

$$\sum_{\mathcal{T} \models \phi(A_1,\ldots,A_t)} \left\{ \sum_{i=1}^{t} \sum_{x_i \in A_i} w_i(x_i) \right\} \quad \text{or} \quad \sum_{\mathcal{T} \models \phi(A_1,\ldots,A_t)} \left\{ \prod_{i=1}^{t} \prod_{x_i \in A_i} w_i(x_i) \right\}.$$

We note again that MSO evaluation problems include counting problems as a special case (simply use the multiplicative variant with all weights set to 1). For examples of extremum and evaluation problems, see discrete Morse matchings (Section 5.2) and the Turaev-Viro invariants (Section 5.3) respectively.

4.2. **Metatheorems on triangulations.** We begin this section by introducing coloured Hasse diagrams, which allow us to translate between triangulations and edge-coloured graphs. Following this we present and prove the three variants of Courcelle's theorem on triangulations (Theorem 4.8).

In settings such as simplicial complexes or polytopes, a Hasse diagram is a graph with a node for each face, and an arc whenever an i-face appears as a subface of an $(i+1)$-face. Here we extend Hasse diagrams to include more precise information about *how* subfaces are embedded, and to support situations in which a face f is embedded as a subface of another face f' more than once.

Definition 4.3. Let $\mathcal{T}$ be a d-dimensional triangulation. Then the *coloured Hasse diagram* of $\mathcal{T}$, denoted $\mathcal{H}(\mathcal{T})$, is the following edge-coloured graph:

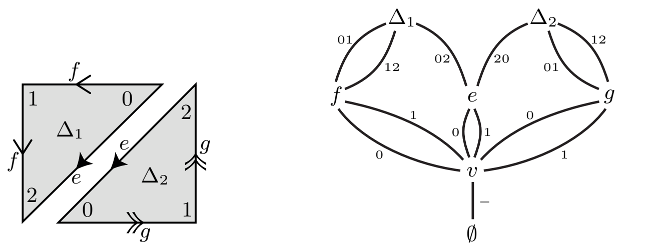
- The colours of $\mathcal{H}(\mathcal{T})$ are, for all $i = 0, \ldots, d-1$, all ordered sequences $\pi_0 \ldots \pi_i$ of distinct integers from the set $\{0, \ldots, i+1\}$ (so exactly one integer from the set is not used). We also allow an additional "empty colour", denoted by a dash $(-)$. This gives $\sum_{i=1}^{d+1} i!$ colours in total.

  For example, for $d = 2$ we use the following colours: $-, 0, 1, 01, 02, 10, 12, 20, 21$. Note that there is no colour labelled 2 in this list.
- The nodes of $\mathcal{H}(\mathcal{T})$ are the i-faces of $\mathcal{T}$, for all $i = 0, \ldots, d$. We also add an extra node for the "empty face", denoted $\emptyset$.
- The arcs of $\mathcal{H}(\mathcal{T})$ are as follows. If f is an i-face and g is an $(i+1)$-face with $f \leq_{\pi_0 \ldots \pi_i} g$, then we place an arc between nodes f and g of colour $\pi_0 \ldots \pi_i$. We also add an arc between each vertex of the triangulation and the empty node $\emptyset$, coloured by the "empty colour" $-$.

Example 4.4. Figure 6 shows a 2-dimensional Klein bottle (the same as seen earlier in Example 2.8), alongside its coloured Hasse diagram. For consistency with the earlier example, we label the common vertex as v, the three edges as e, f, g, and the two triangles as Δ_1, Δ_2.

We now establish a series of results that allow us to convert problems about triangulations into problems about edge-coloured graphs. Recall from Section 2.3 that if $\mathcal{T}$ is a triangulation then $|\mathcal{T}|$ denotes the number of top-dimensional simplices of $\mathcal{T}$, and $\mathcal{D}(\mathcal{T})$ denotes the dual graph of $\mathcal{T}$. Following the pattern established in Section 3.2, the following lemmata essentially show that for fixed dimension d:

- the size $|\mathcal{H}(\mathcal{T})|$ is linear in $|\mathcal{T}|$;



(a) A Klein bottle $\mathcal{K}$ (b) The coloured Hasse diagram $\mathcal{H}(\mathcal{K})$

FIGURE 6. An example of a coloured Hasse diagram

- the treewidth $\text{tw}(\mathcal{H}(\mathcal{T}))$ is at worst linear in $\text{tw}(\mathcal{D}(\mathcal{T}))$; and
- MSO formulae on $\mathcal{T}$ translate to MSO formulae on $\mathcal{H}(\mathcal{T})$.

The triangulation-based variants of Courcelle's theorem then follow naturally from these results, as seen in Theorem 4.8.

Lemma 4.5. *For any d-dimensional triangulation $\mathcal{T}$ of positive size, we have* $|\mathcal{H}(\mathcal{T})| \le 2^d(d+3) \cdot |\mathcal{T}|$.

Proof. Each individual d-simplex has $\binom{d+1}{i+1}$ distinct i-faces for each $i = 0, \ldots, d$, and so the triangulation has $\le \binom{d+1}{i+1}|\mathcal{T}|$ distinct i-faces in total (typically there are fewer, since individual faces of simplices are identified together in the triangulation). Therefore $\mathcal{H}(\mathcal{T})$ has at most $1 + \sum_{i=0}^{d}\binom{d+1}{i+1}|\mathcal{T}| \le 2^{d+1}|\mathcal{T}|$ nodes, including the special node $\emptyset$.

For each $i = 1, \ldots, d$, each i-face g of the triangulation has precisely $(i+1)$ subface relationships of the form $f \le_{\pi_0 \ldots \pi_{i-1}} g$ where f is an $(i-1)$-face. In addition, each 0-face (vertex) of the triangulation has exactly one arc running to the empty node $\emptyset$. Therefore $\mathcal{H}(\mathcal{T})$ has at most $\binom{d+1}{1}|\mathcal{T}| + \sum_{i=1}^{d}(i+1)\binom{d+1}{i+1}|\mathcal{T}| = 2^d(d+1)|\mathcal{T}|$ arcs.

Combining these counts gives $|\mathcal{H}(\mathcal{T})| \le 2^{d+1}|\mathcal{T}| + 2^d(d+1)|\mathcal{T}| = 2^d(d+3)|\mathcal{T}|$. □

Lemma 4.6. *For any d-dimensional triangulation $\mathcal{T}$ of positive size, we have* $\text{tw}(\mathcal{H}(\mathcal{T})) \le (2^{d+1}-1)(\text{tw}(\mathcal{D}(\mathcal{T}))+1)$.

Proof. Suppose we have a tree decomposition of $\mathcal{D}(\mathcal{T})$, with underlying tree T and bags $\{B_\tau\}$. From this we build a tree decomposition of $\mathcal{H}(\mathcal{T})$ using the same tree T but with different bags $\{B'_\tau\}$.

Specifically, let τ be a node of T. The bag B_τ contains a set of nodes of $\mathcal{D}(\mathcal{T})$, or equivalently a set of simplices $\Delta_1, \ldots, \Delta_t$ of $\mathcal{T}$. We define the corresponding bag B'_τ to contain those nodes of $\mathcal{H}(\mathcal{T})$ that denote all of $\Delta_1, \ldots, \Delta_t$, all of their subfaces (of any dimension), and also the empty face $\emptyset$.

Assume for now that this is indeed a tree decomposition of $\mathcal{H}(\mathcal{T})$ (we prove this shortly). Each Δ_i has $\le 2^{d+1}-2$ non-empty proper subfaces (possibly fewer if some of these subfaces are identified), and so $|B'_\tau| \le (2^{d+1}-1)|B_\tau|+1$. Therefore $\text{tw}(\mathcal{H}(\mathcal{T}))+1 \le (2^{d+1}-1)[\text{tw}(\mathcal{D}(\mathcal{T}))+1]+1$, which produces the final result.

It remains to show that our construction does yield a tree decomposition of $\mathcal{H}(\mathcal{T})$. It is clear that every node of $\mathcal{H}(\mathcal{T})$ belongs to a bag B'_τ. Moreover, every arc of $\mathcal{H}(\mathcal{T})$ has both endpoints in some common bag B'_τ, since each bag containing a face f will also contain every subface of f.

To prove the subtree connectivity property for the bags $\{B'_\tau\}$:

- Every bag contains $\emptyset$.
- For each simplex Δ_i, we already know from the original tree decomposition that the bags containing Δ_i form a connected subtree of T.
- Consider some i-face f of $\mathcal{T}$ for $i < d$, and let $\Delta_1, \ldots, \Delta_t$ be the simplices of $\mathcal{T}$ that contain f as a subface.

  Recall that f is in fact an equivalence class of individual i-faces of the simplices $\Delta_1, \ldots, \Delta_t$ that are identified *as a consequence of the facet gluings*. In other words, the individual appearances of f in each simplex $\Delta_1, \ldots, \Delta_t$ are linked by a series of gluings of facets of these simplices—that is, arcs of the dual graph $\mathcal{D}(\mathcal{T})$ that connect $\Delta_1, \ldots, \Delta_t$.

  The bags containing each Δ_i form a connected subtree T_i, and each arc of $\mathcal{D}(\mathcal{T})$ that joins Δ_i with Δ_j has a corresponding bag B'_τ for which $\Delta_i, \Delta_j \in B'_\tau$. Therefore these arcs effectively join the subtrees $T_1, \ldots, T_t$ together, and so all bags containing f form a single (larger) connected subtree of T. □

We emphasise that the proof of Lemma 4.6 makes critical use of our definition of a d-dimensional triangulation, where we only explicitly identify *facets* of d-simplices, and all lower-dimensional face identifications are just a consequence of this. The proof above does not work for more general simplicial complexes (where lower-dimensional faces can be independently identified), but again we note that our setting here covers all reasonable definitions of a triangulated *manifold*.

Lemma 4.7. *For any fixed dimension $d \in \mathbb{N}$, every MSO formula $\phi(x_1, \ldots, x_t)$ on d-dimensional triangulations has a corresponding MSO formula $\overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ on edge-coloured graphs with $k = \sum_{i=1}^{d+1} i!$ colours for which, for any d-dimensional triangulation $\mathcal{T}$:*

- *Any assignment of values to $x_1, \ldots, x_t$ for which $\mathcal{T} \models \phi(x_1, \ldots, x_t)$ yields a corresponding assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ for which $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$. This assignment is obtained by replacing faces of $\mathcal{T}$ with the corresponding nodes of $\mathcal{H}(\mathcal{T})$.*
- *Conversely, any assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ for which $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ is derived from an assignment of values to $x_1, \ldots, x_t$ for which $\mathcal{T} \models \phi(x_1, \ldots, x_t)$, as described above.*

As with the earlier Lemma 3.4, this in particular implies that if ϕ is a sentence then $\mathcal{T} \models \phi$ if and only if $\mathcal{H}(\mathcal{T}) \models \overline{\phi}$, and if ϕ has free variables then the solutions to $\mathcal{T} \models \phi(x_1, \ldots, x_t)$ are in bijection with the solutions to $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

Proof. Following the proof of Lemma 3.4, we work in three stages: (i) we develop "helper" relations to constrain the roles that variables can play in $\overline{\phi}$; (ii) we translate each component of ϕ to a corresponding component of $\overline{\phi}$; and (iii) we add additional constraints to $\overline{\phi}$ to avoid spurious unwanted solutions to $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

For convenience, when developing the formula $\overline{\phi}$ we give our $\sum_{i=1}^{d+1} i!$ colours the same labels that would appear in a coloured Hasse diagram. That is, we use the empty colour $-$, plus the series of colours $\pi_0 \ldots \pi_i$ as described in Definition 4.3.

In the first stage we develop "helper" unary relations $is\_face_i(x)$ for $i = 0, \ldots, d$, using MSO logic on edge-coloured graphs. These relations only makes sense when interpreting $\overline{\phi}$ on the coloured Hasse diagram of a d-dimensional triangulation; in other contexts they are well-defined but meaningless.

Each relation $is\_face_i(x)$ encodes the fact that a variable in $\overline{\phi}$ represents an i-face of the original triangulation $\mathcal{T}$. To build these relations using MSO logic:

- We encode $is\_face_0(x)$ by requiring that x is incident with an arc of the empty colour $-$ and also an arc of some other colour.
- We encode $is\_face_i(x)$ for $1 \leq i \leq d-1$ by requiring that x is incident with an arc of some colour $\pi_0 \ldots \pi_{i-1}$ and also an arc of some colour $\pi_0 \ldots \pi_i$.
- We encode $is\_face_d(x)$ by requiring that x is incident with an arc of some colour $\pi_0 \ldots \pi_{d-1}$ but no arc of any colour $\pi_0 \ldots \pi_{d-2}$.

In the second stage, we translate each piece of ϕ to a piece of $\overline{\phi}$ as follows:

- Each variable x_i in ϕ is replaced by $\overline{x}_i$ in $\overline{\phi}$.
- Standard logical operations (such as $\wedge$, $\vee$, $\neg$, $\rightarrow$ and so on), standard quantifiers ($\forall$ and $\exists$), and the equality and inclusion relations ($=$ and $\in$) are copied directly from ϕ to $\overline{\phi}$.
- Each subface relation ($f \leq_{\pi_0 \ldots \pi_i} s$) in ϕ is replaced by a long (but bounded length) phrase in $\overline{\phi}$ that ensures $is\_face_i(f) \wedge is\_face_d(s)$, and that enforces the exact subface relation by enumerating all possible chains through the Hasse diagram that pass through intermediate $(i+1), \ldots, (d-1)$-faces.

  For example, in $d = 3$ dimensions the edge-to-tetrahedron relationship $f^{(1)} \leq_{01} s$ could be encoded as:

$$
\begin{aligned}
& is\_face_1(f^{(1)}) \wedge is\_face_3(s) \wedge \\
& \big(\, \exists f^{(2)} \quad [adj_{01}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{012}(f^{(2)}, s)] \quad \vee \\
& \qquad\qquad [adj_{10}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{102}(f^{(2)}, s)] \quad \vee \\
& \qquad\qquad [adj_{02}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{021}(f^{(2)}, s)] \quad \vee \\
& \qquad\qquad [adj_{20}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{120}(f^{(2)}, s)] \quad \vee \\
& \qquad\qquad [adj_{12}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{201}(f^{(2)}, s)] \quad \vee \\
& \qquad\qquad [adj_{21}(f^{(1)}, f^{(2)}) \;\wedge\; adj_{210}(f^{(2)}, s)] \qquad \big)\,.
\end{aligned}
$$

This ensures that, if an assignment of values to $x_1, \ldots, x_t$ gives $\mathcal{T} \models \phi(x_1, \ldots, x_t)$, the corresponding assignment of values to $\overline{x}_1, \ldots, \overline{x}_t$ gives $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$.

In the third stage, we eliminate additional and unwanted solutions to $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ by insisting that, for every variable x that appears in ϕ (either bound or free), the translated variable $\overline{x}$ in $\overline{\phi}$ must represent an i-face or set of i-faces in the triangulation:

$$
\begin{aligned}
& is\_face_0(\overline{x}) \vee \ldots \vee is\_face_d(\overline{x}) \vee \\
& [\forall z \; z \in \overline{x} \rightarrow is\_face_0(z)] \vee \ldots \vee [\forall z \; z \in \overline{x} \rightarrow is\_face_d(z)]\,.
\end{aligned}
$$

This ensures that every solution to $\mathcal{H}(\mathcal{T}) \models \overline{\phi}(\overline{x}_1, \ldots, \overline{x}_t)$ translates back to a corresponding solution to $\mathcal{T} \models \phi(x_1, \ldots, x_t)$. □

Theorem 4.8. *For fixed dimension $d \in \mathbb{N}$, let K be any class of d-dimensional triangulations whose dual graphs have universally bounded treewidth. Then:*

- *For any fixed MSO sentence ϕ, it is possible to test whether $\mathcal{T} \models \phi$ for triangulations $\mathcal{T} \in K$ in time $O(|\mathcal{T}|)$.*
- *For any restricted MSO extremum problem P, it is possible to solve P for triangulations $\mathcal{T} \in K$ in time $O(|\mathcal{T}|)$ under the uniform cost measure.*
- *For any MSO evaluation problem P, it is possible to solve P for triangulations $\mathcal{T} \in K$ in time $O(|\mathcal{T}|)$ under the uniform cost measure.*

Proof. The proof is essentially the same as for Theorem 3.5 (Courcelle's theorem for edge-coloured graphs). Let P be any such problem, and let ϕ be its underlying MSO sentence or formula. Using Lemma 4.7 we can translate ϕ to $\overline{\phi}$, yielding a new problem $\overline{P}$ on edge-coloured graphs. Then, for any triangulation $\mathcal{T}$ given as input to P, we construct the Hasse diagram $\mathcal{H}(\mathcal{T})$ and solve $\overline{P}$ for $\mathcal{H}(\mathcal{T})$ instead. For extremum problems the size of each set (and hence the value of the extremum) does not change, and for evaluation problems we copy the input weights from i-faces of $\mathcal{T}$ directly to the corresponding nodes of $\mathcal{H}(\mathcal{T})$.

Noting that d is fixed, Lemma 4.5 shows that $\mathcal{H}(\mathcal{T})$ has size $O(|\mathcal{T}|)$ and can therefore be constructed in $O(|\mathcal{T}|)$ time, and Lemma 4.6 ensures that the treewidth $\mathrm{tw}(\mathcal{H}(\mathcal{T}))$ is universally bounded. The result now follows directly from the edge-coloured variants of Courcelle's theorem (Theorem 3.5). □

5. Applications

We present several applications of our main result (Theorem 4.8); most are in the realm of *3-manifold topology*, where algorithmic questions are often solvable but highly complex, and parameterised complexity is just beginning to be explored.

- We recover two of the earliest fixed-parameter tractability results on 3-manifolds as a direct corollary of Theorem 4.8. The underlying problems are a decision problem on detecting taut angle structures [8], and an extremum problem on optimal discrete Morse matchings [6].
- We prove two new fixed-parameter tractability results. These include a d-dimensional generalisation of the discrete Morse matching result, plus a result on computing the powerful Turaev-Viro invariants for 3-manifolds.

Although "treewidth of the dual graph" might seem artificial as a parameter, this is both natural and useful for 3-manifold triangulations—there are many common constructions in 3-manifold topology that are conducive to small treewidth even when the total number of tetrahedra is large. For instance:

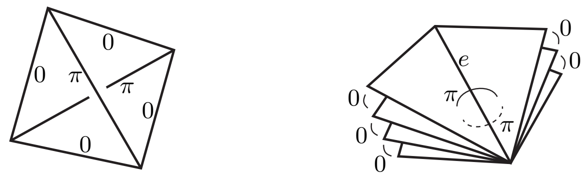
- Dehn fillings do not increase treewidth when performed in the natural way by attaching layered solid tori [20].
- The "canonical" triangulations of arbitrary Seifert fibred spaces over the sphere have a universally bounded treewidth of just two [5].
- Building a complex 3-manifold triangulation from smaller blocks with "narrow" $O(1)$-sized connections can also keep treewidth small. See for instance JSJ decompositions [21] or the bricks of Martelli and Petronio [27], where each connection involves just two triangles.

5.1. **Taut angle structures.**

Given a 3-dimensional triangulation $\mathcal{T}$, a *taut angle structure* assigns internal dihedral angles $0, 0, 0, 0, \pi, \pi$ to the six edges of each tetrahedron of $\mathcal{T}$ so that:

- the two π angles are assigned to opposite edges in each tetrahedron (as in Figure 7(a));
- for each edge e of the triangulation, the sum of all angles assigned to e is precisely 2π (as in Figure 7(b)).

Essentially, a taut angle structure shows how the tetrahedra can be "squashed flat" in a manner that is globally consistent. They are typically used in the context of *ideal triangulations* (where $\mathcal{T}$ becomes a non-compact 3-manifold after its vertices



(a) The π angles in each tetrahedron are assigned to opposite edges

(b) The angle sum around each edge is 2π

FIGURE 7. Defining a taut angle structure

are removed), and they play an interesting role in linking the combinatorics of a triangulation with the geometry and topology of the underlying 3-manifold [19, 23].

Problem 5.1. TAUT ANGLE STRUCTURE *is the decision problem whose input is a 3-dimensional triangulation $\mathcal{T}$, and whose output is* `true` *or* `false` *according to whether there exists a taut angle structure on $\mathcal{T}$.*

Naïvely, this problem can be solved using an exponential-sized combinatorial search through all $3^{|\mathcal{T}|}$ possible assignments of angles—in each tetrahedron there are three choices for which pair of opposite edges receive the angles π. For small treewidth triangulations, however, it has been shown that we can do much better:

Theorem 5.2 (B.-Spreer [8])**.** *The problem* TAUT ANGLE STRUCTURE *is linear-time fixed-parameter tractable, where the parameter is the treewidth of the dual graph of the input triangulation.*

More specifically, for any class K of 3-dimensional triangulations whose dual graphs have universally bounded treewidth $\leq k$, we can solve TAUT ANGLE STRUCTURE *for any input triangulation $\mathcal{T} \in K$ in time $O(3^{7k}k \cdot |\mathcal{T}|)$.*

We observe now that we can recover the linear-time fixed-parameter tractability result directly from Theorem 4.8 (though we do not recover the precise "constant" $3^{7k}k$). All we need to do is express the existence of a taut angle structure using MSO logic. The full MSO sentence is messy, and so we merely outline the key ideas:

- To encode the assignment of angles, we introduce set variables T_1, T_2, T_3 that partition the tetrahedra of the input triangulation $\mathcal{T}$. The set T_1 represents the tetrahedra with π angles on opposite edges 01 and 23, set T_2 represents the tetrahedra with π angles on opposite edges 02 and 13, and set T_3 represents the tetrahedra with π angles on opposite edges 03 and 12.
- To enforce the 2π angle sum criterion, we introduce an edge variable $f^{(1)}$ and ensure that $\forall f^{(1)}$, the edge $f^{(1)}$ is assigned the value π precisely twice. Here a *single* assignment of π to $f^{(1)}$ corresponds to some simplex s for which:

  $s \in T_1$ and $\left(f^{(1)} \leq_{01} s \vee f^{(1)} \leq_{10} s \vee f^{(1)} \leq_{23} s \vee f^{(1)} \leq_{32} s\right)$; or
  $s \in T_2$ and $\left(f^{(1)} \leq_{02} s \vee f^{(1)} \leq_{20} s \vee f^{(1)} \leq_{13} s \vee f^{(1)} \leq_{31} s\right)$; or
  $s \in T_3$ and $\left(f^{(1)} \leq_{03} s \vee f^{(1)} \leq_{30} s \vee f^{(1)} \leq_{12} s \vee f^{(1)} \leq_{21} s\right)$.

  We must piece together the full MSO clause with care, since $f^{(1)}$ may receive two assignments of π from the same simplex (e.g., a simplex $s \in T_1$ for which both $f^{(1)} \leq_{01} s$ and $f^{(1)} \leq_{23} s$).

By formalising these ideas into a full MSO sentence, the linear-time fixed-parameter tractability of TAUT ANGLE STRUCTURE follows directly from the main result of this paper (Theorem 4.8).

5.2. **Discrete Morse matchings.** In essence, discrete Morse theory offers a way to study the "topological complexity" of a triangulation. The idea is to effectively quarantine the topological content of a triangulation into a small number of "critical faces"; the remainder of the triangulation then becomes "padding" that is topologically unimportant.

A key problem in discrete Morse theory is to find an *optimal Morse matching*, where the number of critical faces is as small as possible. Solving this problem yields important topological information [13], and has a number of practical applications [18, 24].

We give a very brief overview of the key concepts here; see the very accessible paper [17] for details. Morse matchings are typically defined in terms of the Hasse diagram of a simplicial complex. Here we work with the coloured Hasse diagram instead, which allows us to port the necessary concepts to the more general d-dimensional triangulations that we use in this paper.

Definition 5.3. Let $\mathcal{T}$ be a d-dimensional triangulation with coloured Hasse diagram $\mathcal{H}(\mathcal{T})$. Let $V^{(i)}$ denote the set of nodes of $\mathcal{H}(\mathcal{T})$ that correspond to i-faces of $\mathcal{T}$ for each $i = 0, \ldots, d$, and let E denote the set of arcs of $\mathcal{H}(\mathcal{T})$.

A *Morse matching* on $\mathcal{T}$ is a set of arcs $M \subseteq E$ with the following properties:

- The arcs in M are *disjoint* (i.e., no two are incident with a common node), and no arc in M is incident with the empty node $\emptyset$.
- For each $i = 0, \ldots, d-1$, there are no *alternating cycles* between $V^{(i)}$ and $V^{(i+1)}$. That is, $\mathcal{H}(\mathcal{T})$ has no cycle whose nodes alternate between $V^{(i)}$ and $V^{(i+1)}$, and whose arcs alternate between M and $E \backslash M$.

For a given Morse matching M, a *critical i-face* of $\mathcal{T}$ is an i-face whose corresponding node $v \in V^{(i)}$ is not incident with any arc $e \in M$. We let $c(M)$ denote the total number of critical faces, so $c(M) = \sum_{i=0}^{d} |V^{(i)}| - 2|M|$.

Example 5.4. Consider the coloured Hasse diagram seen earlier in Figure 6. This diagram has several pairs of parallel arcs, none of which can be used in a Morse matching (since this would yield an alternating cycle of length two). Therefore the largest Morse matching possible has just $|M| = 1$ arc (either the arc from e to Δ_1, or the arc from e to Δ_2). Such a matching has $c(M) = 4$ critical faces (one 2-face, two 1-faces, and one 0-face).

Problem 5.5. d-DIMENSIONAL OPTIMAL MORSE MATCHING *is the extremum problem whose input is a d-dimensional triangulation $\mathcal{T}$, and whose output is the minimum of $c(M)$ over all Morse matchings M on $\mathcal{T}$.*

Even in dimension $d = 3$ the related decision problem is NP-complete [22], but again for small treewidth triangulations we can do better:

Theorem 5.6 (B.-Lewiner-Paixão-Spreer [6])**.** *The problem* 3-DIMENSIONAL OPTIMAL MORSE MATCHING *is linear-time fixed-parameter tractable, where the parameter is the treewidth of the dual graph of the input triangulation.*

More specifically, for any class K of 3-dimensional triangulations whose dual graphs have universally bounded treewidth $\leq k$, we can solve the problem for any input triangulation $\mathcal{T} \in K$ in time $O(4^{k^2+k} \cdot k^3 \cdot \log k \cdot |\mathcal{T}|)$.

Again we show now that we can recover the linear-time fixed-parameter tractability result directly from Theorem 4.8 (but not the precise "constant" $4^{k^2+k} \cdot k^3 \cdot \log k$). Moreover, we generalise this to arbitrary dimensions:

Theorem 5.7. *For fixed dimension $d \in \mathbb{N}$ and any class K of d-dimensional triangulations whose dual graphs have universally bounded treewidth, we can solve* d-DIMENSIONAL OPTIMAL MORSE MATCHING *for triangulations $\mathcal{T} \in K$ in time $O(\mathcal{T})$ under the uniform cost measure.*

Proof. To formulate this as an MSO extremum problem on triangulations:

- For each $i = 0, \ldots, d$ we introduce a set variable $V^{(i)}$ and a clause to ensure that $V^{(i)}$ contains all i-faces of $\mathcal{T}$. These trivial set variables are used in the rational function that we seek to minimise (see below).
- To encode a Morse matching, we use a family of set variables $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$ for each $i = 1, \ldots, d$ and for each sequence $\pi_0 \ldots \pi_{i-1}$ of distinct integers in $\{0, \ldots, i\}$. Each variable $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$ holds all i-faces of $\mathcal{T}$ that are matched with an $(i-1)$-face using an arc of $\mathcal{H}(\mathcal{T})$ of colour $\pi_0 \ldots \pi_{i-1}$.
- For each set $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$, we add a clause to ensure that the corresponding arcs exist in $\mathcal{H}(\mathcal{T})$: $\forall v^{(i)}\ \exists v^{(i-1)}\ v^{(i)} \in W^{(i)}_{\pi_0\ldots\pi_{i-1}} \to v^{(i-1)} \leq_{\pi_0\ldots\pi_{i-1}} v^{(i)}$.
- We add a (long) clause to ensure that each face of $\mathcal{T}$ is involved in at most one matching.
- For each $i = 1, \ldots, d$, we add a (very long) clause to ensure that there are no alternating cycles between i-faces and $(i-1)$-faces. Here we encode an alternating cycle using a second family of set variables $X^{(i)}_{\pi_0\ldots\pi_{i-1}}$ that represent arcs present in the cycle but not the matching, where:
  - for each i and each $\pi_0 \ldots \pi_{i-1}$ the sets $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$ and $X^{(i)}_{\pi_0\ldots\pi_{i-1}}$ are disjoint;
  - each i-face $v^{(i)}$ appears in either none of the sets $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$ or $X^{(i)}_{\pi_0\ldots\pi_{i-1}}$, or else exactly one of the sets $W^{(i)}_{\pi_0\ldots\pi_{i-1}}$ and one of the sets $X^{(i)}_{\pi_0\ldots\pi_{i-1}}$;
  - each $(i-1)$-face $v^{(i-1)}$ is involved in precisely zero or two relationships of the form $v^{(i-1)} \leq_{\pi_0\ldots\pi_{i-1}} v^{(i)}$ where $v^{(i)} \in W^{(i)}_{\pi_0\ldots\pi_{i-1}} \cup X^{(i)}_{\pi_0\ldots\pi_{i-1}}$.

We can now encode d-DIMENSIONAL OPTIMAL MORSE MATCHING as the extremum problem that asks to minimise $\sum_i |V^{(i)}| - 2\sum_i |W^{(i)}_{\pi_0\ldots\pi_{i-1}}|$, and the result follows from Theorem 4.8. □

5.3. **The Turaev-Viro invariants.** The Turaev-Viro invariants are an infinite family of topological invariants of 3-manifolds [31]. For every triangulation $\mathcal{T}$ of a closed 3-manifold, there is an invariant $|\mathcal{T}|_{r,q_0}$ for each integer $r \geq 3$ and each $q_0 \in \mathbb{C}$ for which q_0 is a $(2r)$th root of unity and q_0^2 is a primitive rth root of unity. A key property (which justifies the name "invariants") is that the value of $|\mathcal{T}|_{r,q_0}$ depends only upon the topology of the underlying 3-manifold, and not the particular choice of triangulation $\mathcal{T}$.

The Turaev-Viro invariants can be expressed as sums over combinatorial objects on $\mathcal{T}$, and so (unlike many other 3-manifold invariants) lend themselves well to computation. Moreover, they have proven extremely useful in practical software settings for distinguishing between different 3-manifolds [4, 28]. However, they have a major drawback: computing $|\mathcal{T}|_{r,q_0}$ requires time $O(r^{2|\mathcal{T}|} \times \mathrm{poly}(|\mathcal{T}|))$ under existing algorithms, and so is feasible only for small triangulations and/or small r.

In Theorem 5.9 we show again that we can do much better for small treewidth triangulations: for fixed r, computing $|\mathcal{T}|_{r,q_0}$ is linear-time fixed-parameter tractable, with (as usual) the treewidth of the dual graph as the parameter.

Before proving this result, we give a short outline of how the Turaev-Viro invariants $|\mathcal{T}|_{r,q_0}$ are defined. The formula is relatively detailed and so our summary here is very brief, with just enough information to prove the subsequent theorem. For more information, including motivations and topological context for these invariants, we refer the reader to references such as [28, 31].

Definition 5.8. Let $\mathcal{T}$ be a closed 3-manifold triangulation, and let V, E and S denote the set of all vertices, edges and tetrahedra of $\mathcal{T}$. Let $r \geq 3$ be an integer, and let $q_0 \in \mathbb{C}$ be a $(2r)$th root of unity for which q_0^2 is a primitive rth root of unity. We define the Turaev-Viro invariant $|\mathcal{T}|_{r,q_0}$ as follows.

Let $I = \{0, 1/2, 1, 3/2, \ldots, (r-2)/2\}$, so $|I| = r - 1$. A triple $(i, j, k) \in I^3$ is called *admissible* if $i+j+k \in \mathbb{Z}$, $i \leq j+k$, $j \leq i+k$, $k \leq i+j$, and $i+j+k \leq r-2$.

A *colouring* θ of $\mathcal{T}$ is a map $\theta\colon E \to I$; in other words, we "colour" each edge of $\mathcal{T}$ with an element of I. A colouring θ is called *admissible* if, for each 2-face f of $\mathcal{T}$, the three edges $e_1, e_2, e_3 \in E$ bounding f give rise to an admissible triple $(\theta(e_1), \theta(e_2), \theta(e_3))$.

We make use of several complex constants that depend on r and q_0. We do not need to give their exact values here; details can be found in [31]. These constants are: $\alpha \in \mathbb{C}$; $\beta_i \in \mathbb{C}$ for each $i \in I$; and $\gamma_{i,j,k,\ell,m,n} \in \mathbb{C}$ for each $i, j, k, \ell, m, n \in I$. Note that these constants do not depend upon the specific triangulation $\mathcal{T}$.

We use these constants as follows. Let θ be an admissible colouring of $\mathcal{T}$. For each vertex $v \in V$ we define $|v|_\theta = \alpha$. For each edge $e \in E$ we define $|e|_\theta = \beta_{\theta(e)}$. For each tetrahedron $\Delta \in S$ we define $|\Delta|_\theta = \gamma_{\theta(e_{01}),\theta(e_{02}),\theta(e_{12}),\theta(e_{23}),\theta(e_{13}),\theta(e_{03})}$, where $e_{ij} \in E$ is the edge of the triangulation that joins vertices i and j of Δ. For the entire triangulation we define $|\mathcal{T}|_\theta = \left(\prod_{v \in V} |v|_\theta\right) \cdot \left(\prod_{e \in E} |e|_\theta\right) \cdot \left(\prod_{\Delta \in S} |\Delta|_\theta\right)$.

Finally, we define the Turaev-Viro invariant $|\mathcal{T}|_{r,q_0} = \sum_\theta |\mathcal{T}|_\theta$, where this sum is taken over all admissible colourings θ of $\mathcal{T}$.

Theorem 5.9. *For any fixed integer $r \geq 3$ and any class K of closed 3-manifold triangulations whose dual graphs have universally bounded treewidth, we can compute any Turaev-Viro invariant $|\mathcal{T}|_{r,q_0}$ for any closed 3-manifold triangulation $\mathcal{T} \in K$ in time $O(\mathcal{T})$ under the uniform cost measure.*

Proof. We prove this by framing the computation of $|\mathcal{T}|_{r,q_0}$ as a multiplicative MSO evaluation problem.

For our MSO formula ϕ, we define several free variables that together encode an admissible colouring θ of $\mathcal{T}$. These include:

- A single set variable V that stores all vertices of $\mathcal{T}$.
- A set variable E_i for each $i \in I$. Each set E_i represents those edges of $\mathcal{T}$ that are assigned the colour i.
- A set variable $S_{i,j,k,\ell,m,n}$ for each $i, j, k, \ell, m, n \in I$ where the triples (i, j, k), (k, ℓ, m), (m, n, i) and (j, ℓ, n) are all admissible. Each set $S_{i,j,k,\ell,m,n}$ represents those tetrahedra whose edges 01, 02, 12, 23, 13 and 03 are assigned colours i, j, k, ℓ, m, n respectively, which means that every 2-face of such a tetrahedron will be coloured by an admissible triple.

We insert clauses into our MSO formula to ensure that:

- The set variable V contains precisely all vertices of $\mathcal{T}$.

- The set variables E_i together partition the edges of $\mathcal{T}$, and the set variables $S_{i,j,k,\ell,m,n}$ together partition the tetrahedra of $\mathcal{T}$.
- The colourings $\{E_i\}$ and $\{S_{i,j,k,\ell,m,n}\}$ are consistent. This involves clauses $\forall s \forall e^{(1)}\ [s \in S_{i,j,k,\ell,m,n} \wedge (e^{(1)} \leq_{01} s \vee e^{(1)} \leq_{10} s)] \to e^{(1)} \in E_i$ and many others of a similar form, where $e^{(1)}$ is an edge variable and s is a tetrahedron variable.

It follows that the solutions to $\mathcal{T} \models \phi$ correspond precisely to the admissible colourings of $\mathcal{T}$.

We now assign weights for our evaluation problem as follows:

- For the set variable V, we assign a weight of α to each vertex of $\mathcal{T}$, and a weight of 1 to all other faces of all other dimensions.
- For each set variable E_i, we assign a weight of β_i to each edge of $\mathcal{T}$, and a weight of 1 to all other faces of all other dimensions.
- For each set variable $S_{i,j,k,\ell,m,n}$, we assign a weight of $\gamma_{i,j,k,\ell,m,n}$ to each tetrahedron of $\mathcal{T}$, and a weight of 1 to all other faces of all other dimensions.

By Definition 5.8, the Turaev-Viro invariant $|\mathcal{T}|_{r,q_0}$ is precisely the solution to the multiplicative evaluation problem defined by this MSO formula and these weights. Note that, whilst the MSO formula depends only on r (which is fixed), the weights depend on both r and q_0 (where q_0 is supplied with the input).

The result now follows directly from Theorem 4.8. □

References

1. Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1975, Second printing, Addison-Wesley Series in Computer Science and Information Processing.
2. Stefan Arnborg, Jens Lagergren, and Detlef Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms **12** (1991), no. 2, 308–340.
3. Hans L. Bodlaender, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput. **25** (1996), no. 6, 1305–1317.
4. Benjamin A. Burton, *Structures of small closed non-orientable 3-manifold triangulations*, J. Knot Theory Ramifications **16** (2007), no. 5, 545–574.
5. ———, *Computational topology with Regina: Algorithms, heuristics and implementations*, Geometry and Topology Down Under (Craig D. Hodgson, William H. Jaco, Martin G. Scharlemann, and Stephan Tillmann, eds.), Contemporary Mathematics, no. 597, Amer. Math. Soc., Providence, RI, 2013.
6. Benjamin A. Burton, Thomas Lewiner, João Paixão, and Jonathan Spreer, *Parameterized complexity of discrete Morse theory*, SCG '13: Proceedings of the 29th Annual Symposium on Computational Geometry, ACM, 2013, pp. 127–136.
7. Benjamin A. Burton and William Pettersson, *Fixed parameter tractable algorithms in combinatorial topology*, Preprint, `arXiv:1402.3876`, February 2014.
8. Benjamin A. Burton and Jonathan Spreer, *The complexity of detecting taut angle structures on triangulations*, SODA '13: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2013, pp. 168–183.
9. B. Courcelle, J. A. Makowsky, and U. Rotics, *On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic*, Discrete Appl. Math. **108** (2001), no. 1-2, 23–52, International Workshop on Graph-Theoretic Concepts in Computer Science (Smolenice Castle, 1998).
10. B. Courcelle and M. Mosbah, *Monadic second-order evaluations on tree-decomposable graphs*, Theoret. Comput. Sci. **109** (1993), no. 1-2, 49–82, International Workshop on Computing by Graph Transformation (Bordeaux, 1991).
11. Bruno Courcelle, *On context-free sets of graphs and their monadic second-order theory*, Graph-Grammars and their Application to Computer Science (Warrenton, VA, 1986), Lecture Notes in Comput. Sci., vol. 291, Springer, Berlin, 1987, pp. 133–146.

12. ———, *Graph rewriting: An algebraic and logic approach*, Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 193–242.
13. Tamal K. Dey, Herbert Edelsbrunner, and Sumanta Guha, *Computational topology*, Advances in Discrete and Computational Geometry (South Hadley, MA, 1996), Contemp. Math., vol. 223, Amer. Math. Soc., Providence, RI, 1999, pp. 109–143.
14. R. G. Downey and M. R. Fellows, *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag, New York, 1999.
15. Nathan M. Dunfield and William P. Thurston, *Finite covers of random 3-manifolds*, Invent. Math. **166** (2006), no. 3, 457–521.
16. J. Flum and M. Grohe, *Parameterized complexity theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
17. Robin Forman, *A user's guide to discrete Morse theory*, Sém. Lothar. Combin. **48** (2002), Art. B48c, 35.
18. David Günther, Jan Reininghaus, Hubert Wagner, and Ingrid Hotz, *Efficient computation of 3D Morse-Smale complexes and persistent homology using discrete Morse theory*, Vis. Comput. **28** (2012), no. 10, 959–969.
19. Craig D. Hodgson, J. Hyam Rubinstein, Henry Segerman, and Stephan Tillmann, *Veering triangulations admit strict angle structures*, Geom. Topol. **15** (2011), no. 4, 2073–2089.
20. William Jaco and J. Hyam Rubinstein, *0-efficient triangulations of 3-manifolds*, J. Differential Geom. **65** (2003), no. 1, 61–168.
21. William H. Jaco and Peter B. Shalen, *Seifert fibered spaces in* 3*-manifolds*, Mem. Amer. Math. Soc. **21** (1979), no. 220, viii+192.
22. Michael Joswig and Marc E. Pfetsch, *Computing optimal Morse matchings*, SIAM J. Discrete Math. **20** (2006), no. 1, 11–25 (electronic).
23. Marc Lackenby, *Taut ideal triangulations of 3-manifolds*, Geom. Topol. **4** (2000), 369–395.
24. Thomas Lewiner, Helio Lopes, and Geovan Tavares, *Applications of Forman's discrete morse theory to topology visualization and mesh compression*, IEEE Trans. Vis. Comput. Graphics **10** (2004), no. 5, 499–508.
25. J. A. Makowsky, *Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width*, Discrete Appl. Math. **145** (2005), no. 2, 276–290.
26. J. A. Makowsky and J. P. Mariño, *The parameterized complexity of knot polynomials*, J. Comput. Syst. Sci. **67** (2003), no. 4, 742–756.
27. Bruno Martelli and Carlo Petronio, *A new decomposition theorem for 3-manifolds*, Illinois J. Math. **46** (2002), 755–780.
28. Sergei Matveev, *Algorithmic topology and classification of 3-manifolds*, Algorithms and Computation in Mathematics, no. 9, Springer, Berlin, 2003.
29. Neil Robertson and P. D. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, J. Algorithms **7** (1986), no. 3, 309–322.
30. William P. Thurston, *The geometry and topology of 3-manifolds*, Lecture notes, Princeton University, 1978.
31. Vladimir G. Turaev and Oleg Y. Viro, *State sum invariants of* 3*-manifolds and quantum* $6j$*-symbols*, Topology **31** (1992), no. 4, 865–902.

School of Mathematics and Physics, The University of Queensland, Brisbane QLD 4072, Australia

E-mail address: bab@maths.uq.edu.au

School of Mathematics, Statistics and Operations Research, Victoria University, New Zealand.

E-mail address: Rod.Downey@msor.vuw.ac.nz