The complexity of detecting taut angle structures on triangulations*

Benjamin A. Burton† Jonathan Spreer‡

Author's self-archived version
Available from `http://www.maths.uq.edu.au/~bab/papers/`

Abstract

There are many fundamental algorithmic problems on triangulated 3-manifolds whose complexities are unknown. Here we study the problem of finding a taut angle structure on a 3-manifold triangulation, whose existence has implications for both the geometry and combinatorics of the triangulation. We prove that detecting taut angle structures is **NP**-complete, but also fixed-parameter tractable in the treewidth of the face pairing graph of the triangulation. These results have deeper implications: the core techniques can serve as a launching point for approaching key decision problems such as unknot recognition and prime decomposition of 3-manifolds.

1 Introduction

Much work in 3-dimensional topology is driven by algorithmic problems. Examples include *unknot recognition* (testing whether a knot in $\mathbb{R}^3$ is trivial), *3-sphere recognition* (testing whether a triangulated 3-manifold is a topological sphere), *connected sum decomposition* (decomposing a 3-manifold into "prime" pieces), *JSJ decomposition* (decomposing a 3-manifold into pieces with geometric structures), and the *homeomorphism problem* (testing whether two triangulated 3-manifolds are topologically equivalent).

Many of these algorithms are new; for instance, 3-sphere recognition was only solved in 1992 by Rubinstein [29], and the homeomorphism problem was only solved in 2003 with Perelman's proof of the geometrisation conjecture [19], which ties together many complex sub-algorithms by many different authors [16]. Some algorithms, such as unknot recognition, 3-sphere recognition and connected sum decomposition, have been implemented [6] but require exponential time; others are currently so slow and so complex that they have never been implemented at all.

In this paper we consider the computational complexity of problems such as these in 3-dimensional topology, where many important questions remain wide open. For instance, it is a major open question as to whether unknot recognition and 3-sphere recognition can be solved in polynomial time. Both problems are known to lie in **NP** [14, 31], and in recent announcements both problems also lie in **co-NP** if the generalised Riemann hypothesis holds [13, 21]. Nevertheless, current state-of-the-art algorithms for both problems still require exponential time.

There is one prominent hardness result in this area, due to Agol, Hass and Thurston, involving *knot genus*: if we generalise unknot recognition to computing the genus of a knot, *and* we generalise the ambient space from $\mathbb{R}^3$ to an arbitrary 3-manifold, then the problem becomes **NP**-complete [1]. The underlying proof technique also applies to problems relating to least-area surfaces [1, 10].

Beyond the results cited above, very little is known about the computational complexity of difficult algorithmic problems such as these in 3-dimensional topology.

In this paper we address the problem of finding a *taut angle structure* on a triangulated 3-manifold (as outlined below). In particular, we show that this problem is both **NP**-complete and fixed-parameter tractable. This is the first parameterised complexity result in areas relating to difficult 3-manifold recognition/decomposition problems, and the first such NP-completeness result that is not based on the Agol-Hass-Thurston construction. More importantly, the techniques that we describe here offer a potential launching point for obtaining such results in the related setting of *normal surface theory*, a key ingredient in all of the decomposition and recognition problems outlined above. We discuss these possibilities further in Section 5.

Taut angle structures were introduced by Lackenby [22], and offer a bridge between the combinatorial structure of a triangulation and the geometric structure of the underlying manifold. Taut angle structures are com-

*The first author is supported by the Australian Research Council under the Discovery Projects funding scheme (projects DP1094516 and DP110101104).

†School of Mathematics and Physics, The University of Queensland, Brisbane, Australia. `bab@maths.uq.edu.au`

‡School of Mathematics and Physics, The University of Queensland, Brisbane, Australia. `j.spreer@uq.edu.au`

binatorial objects that act as limiting cases of the more general *angle structures*, as introduced by Rivin [27, 28] and Casson; these in turn act as linear analogues of *complete hyperbolic structures*, which play an important role in recognising and distinguishing triangulated hyperbolic 3-manifolds. Despite their simple discrete combinatorial description, taut angle structures can in the right setting lead to strict angle structures [18] and then complete hyperbolic structures [11], which in general are highly desirable but also potentially elusive.

Specifically, a *taut angle structure* on a 3-manifold triangulation $\mathcal{T}$ assigns interior angles $\{0, 0, 0, 0, \pi, \pi\}$ to the six edges of each tetrahedron of $\mathcal{T}$, so that the two π angles are opposite in each tetrahedron, and so that around each edge of the overall triangulation the sum of angles is 2π. The decision problem that we study in this paper is as follows:

PROBLEM 1.1. (TAUT ANGLE STRUCTURE) *Given an orientable 3-manifold triangulation $\mathcal{T}$ with no boundary faces, determine whether there exists a taut angle structure on $\mathcal{T}$. We measure the size of the input by the number of tetrahedra in $\mathcal{T}$, which we denote by n.*

Our first main theorem is the following:

THEOREM 1.1. *The problem* TAUT ANGLE STRUCTURE *is* ***NP****-complete.*

We prove this in Section 3 using a reduction from the **NP**-complete problem MONOTONE 1-IN-3 SAT [30]. In MONOTONE 1-IN-3 SAT we have boolean variables $x_1, \ldots, x_t$ and clauses of the form $x_i \vee x_j \vee x_k$, and we must determine whether the variables can be assigned true/false values so that one and only one of the three variables in each clause is true.

The proof involves an explicit piecewise construction of a 3-manifold triangulation that represents a given instance of MONOTONE 1-IN-3 SAT. We use three types of building blocks, which represent (i) variables x_i; (ii) the duplication of variables; and (iii) clauses $x_i \vee x_j \vee x_k$. Finding such building blocks—particularly (ii) and (iii)—was a major challenge in constructing the proof, and was performed with significant assistance from the software package *Regina* [6, 8].

In Section 4 we present additional results on *parameterised complexity*. Introduced by Downey and Fellows [9], parameterised complexity studies which aspects of an **NP**-complete problem make it difficult, and identifies classes of inputs for which fast algorithms can nonetheless be found.

Our parameters are based on the *face pairing graph* of the input triangulation $\mathcal{T}$ (that is, the dual 1-skeleton of $\mathcal{T}$). Denoted $\Gamma(\mathcal{T})$, the face pairing graph is the multigraph whose nodes represent tetrahedra of $\mathcal{T}$, and whose arcs represent pairs of tetrahedron faces that are joined together.

For TAUT ANGLE STRUCTURE, we identify two parameters of interest: the cutwidth of $\Gamma(\mathcal{T})$, and the treewidth of $\Gamma(\mathcal{T})$. We define these concepts precisely in Section 2, but in essence the cutwidth measures the worst "bottleneck" of parallel arcs in an optimal left-to-right layout of nodes, and the treewidth measures how "tree-like" the graph is. Our results are the following:

THEOREM 1.2. *Let $\mathcal{T}$ be a 3-manifold triangulation with n tetrahedra, where the graph $\Gamma(\mathcal{T})$ has cutwidth $\leq k$, and for which a corresponding layout of nodes is known. Then* TAUT ANGLE STRUCTURE *can be solved for $\mathcal{T}$ in $O(nk \cdot 3^{3k/2})$ time.*

THEOREM 1.3. *Let $\mathcal{T}$ be a 3-manifold triangulation with n tetrahedra, where the graph $\Gamma(\mathcal{T})$ has treewidth $\leq k$, and for which a corresponding tree decomposition with $O(n)$ tree nodes is known. Then* TAUT ANGLE STRUCTURE *can be solved for $\mathcal{T}$ in $O(nk \cdot 3^{7k})$ time.*

Because treewidth $\leq$ cutwidth (as shown in [3]), the latter result is more powerful. Moreover, if we fix an upper bound on the treewidth k, there is a known linear-time algorithm to test whether a graph has treewidth $\leq k$ and, if so, to compute a corresponding tree decomposition with $O(n)$ tree nodes [4]. Therefore Theorem 1.3 shows that, in the case of bounded treewidth, we can solve TAUT ANGLE STRUCTURE in *linear time* in the input size n. That is:

COROLLARY 1.1. TAUT ANGLE STRUCTURE *is linear-time fixed-parameter tractable, where the parameter is taken to be the treewidth of the face pairing graph of the input triangulation.*

For 3-manifold triangulations the treewidth of $\Gamma(\mathcal{T})$ is a natural parameter, and there are well-known families of triangulations for which the treewidth remains small. Moreover, our fixed-parameter tractability result is consistent with experimental observations from running other, more complex algorithms over small-treewidth triangulations. We discuss these issues further in Section 5.

Throughout this paper we work in the word RAM model, where simple arithmetical operations on $(\log n)$-bit integers are assumed to take constant time.

2 Preliminaries

2.1 Triangulations By a *3-manifold triangulation*, we mean a collection of n abstract tetrahedra, some or all of whose faces are affinely identified or "glued

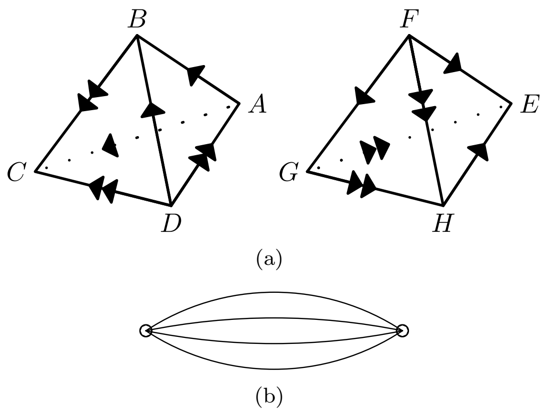

Figure 1: The figure eight knot complement and its face pairing graph

together” in pairs. As a consequence of these face gluings, many tetrahedron edges may become identified together; we refer to the result as a single *edge of the triangulation*, and likewise with vertices.

This is a purely combinatorial definition: there are no geometric constraints (such as embeddability in some $\mathbb{R}^d$), and the result need not be a simplicial complex. We may glue together two faces of the same tetrahedron if we like. A single edge of the triangulation might appear as multiple edges of the same tetrahedron, and likewise with vertices. It is common to work with *one-vertex triangulations*, where all vertices of all tetrahedra become identified as a single point.

The only constraints are the following. Each tetrahedron face must be identified with one and only one partner (we call these *internal faces*), or with nothing at all (we call these *boundary faces*). Moreover, no edge may be identified with itself in reverse as a result of the face gluings. Any edge on a boundary face is called a *boundary edge*, and all others are called *internal edges*.

The *link* of a vertex V of the triangulation is the frontier of a small regular neighbourhood of V. If the link of V is a closed surface but not a sphere, we call V an *ideal vertex*. Any triangulation with one or more ideal vertices is called an *ideal triangulation*.

Although the neighbourhood of an ideal vertex is not locally $\mathbb{R}^3$ (and so ideal triangulations do not represent 3-manifolds per se), topologists often use ideal triangulations as an economical way to represent 3-manifolds with boundary (obtained by truncating the ideal vertices) or non-compact 3-manifolds (obtained by deleting the ideal vertices). Because of this, ideal triangulations are ubiquitous in the study of hyperbolic 3-manifolds.

Figure 1(a) illustrates Thurston's famous ideal triangulation of the figure eight knot complement [33]. There are $n = 2$ tetrahedra, labelled $ABCD$ and $EFGH$, with the following face gluings:

$$ABC \longleftrightarrow FGE; \qquad ABD \longleftrightarrow HEF;$$
$$ACD \longleftrightarrow HEG; \qquad BCD \longleftrightarrow GHF.$$

As a consequence of these face gluings, we obtain two edges of the triangulation, indicated by the two types of arrowhead in the diagram. All vertices of all tetrahedra become identified as a single ideal vertex of the triangulation, whose link is a torus. It can be shown that truncating this vertex does indeed yield the figure eight knot complement (i.e., the 3-manifold with torus boundary obtained by deleting a small neighbourhood of the figure eight knot from the 3-sphere).[1]

The *size* of a triangulation is measured by the number of tetrahedra n. To input a triangulation, one presents the list of face gluings (as illustrated above), which requires $O(n \log n)$ bits.

2.2 Taut angle structures Let $\mathcal{T}$ be a 3-manifold triangulation with no boundary faces. A *taut angle structure* on $\mathcal{T}$ assigns interior angles $\{0, 0, 0, 0, \pi, \pi\}$ to the six edges of each tetrahedron of $\mathcal{T}$, so that the two π angles are opposite in each tetrahedron, and so that around each edge of the triangulation the sum of angles is 2π. Geometrically, a taut structure shows how the tetrahedra can be consistently “flattened” throughout the triangulation. Here we use the nomenclature of Hodgson et al. [15]—our taut angle structures are slightly more general than the original taut structures of Lackenby [22], who also requires consistent coorientations on the 2-faces of the triangulation.

To illustrate, we can place a taut angle structure on Figure 1(a) by assigning π to the opposite edges AC and BD of the first tetrahedron, and to the opposite edges EG and FH of the second tetrahedron. It is easily seen that both edges of the triangulation (the single arrowheads versus the double arrowheads) receive the angle π exactly twice each.

We refer to the two π edges in each tetrahedron as *marked*. Combinatorially, a taut structure simply involves choosing two opposite edges of each tetrahedron to mark, in such a way that every edge of the triangulation is marked exactly twice.

A simple Euler characteristic calculation shows that, in a triangulation with no boundary faces, a taut angle structure can only exist if every vertex link is a torus or a Klein bottle. That is, taut angle structures require *ideal triangulations*.

[1] To highlight the efficiency of ideal triangulations: the smallest known *non-ideal* triangulation of the figure eight knot complement (using boundary faces instead of an ideal vertex) requires $n = 10$ tetrahedra.

Here we generalise this definition to support triangulations with boundary (which become important as we piece together triangulations for our **NP**-completeness proof). If $\mathcal{T}$ is any 3-manifold triangulation (with or without boundary faces), then a taut angle structure on $\mathcal{T}$ involves choosing two opposite edges of each tetrahedron to mark, so that every internal edge of the triangulation is marked *exactly* twice, and every boundary edge of the triangulation is marked *at most* twice.

Let $\mathcal{T}$ and $\mathcal{T}'$ be 3-manifold triangulations for which $\mathcal{T}$ is a subcomplex of $\mathcal{T}'$ (i.e., $\mathcal{T}'$ is obtained from $\mathcal{T}$ by adding new tetrahedra and/or additional face gluings). If τ and τ' are taut angle structures on $\mathcal{T}$ and $\mathcal{T}'$ respectively, we say that τ' *extends* τ if they both assign the same interior angles to the tetrahedra from $\mathcal{T}$ (i.e., the tetrahedra that belong to both triangulations).

2.3 Face pairing graphs The *face pairing graph* of a 3-manifold triangulation $\mathcal{T}$, denoted $\Gamma(\mathcal{T})$, is the multigraph whose nodes represent tetrahedra, and whose arcs represent pairs of tetrahedron faces that are glued together. A face pairing graph may contain loops (if two faces of the same tetrahedron are glued together), and/or multiple edges (if two tetrahedra are joined together along more than one face).

If every face of $\mathcal{T}$ is internal, then $\Gamma(\mathcal{T})$ is a 4-valent graph. Figure 1(b) shows the face pairing graph of the figure eight knot complement as presented in Figure 1(a).

In our parameterised complexity analysis, we measure both the cutwidth and the treewidth of $\Gamma(\mathcal{T})$. These concepts are defined as follows [9, 20]:

Definition 2.1. (Cutwidth) *A* cut *of a graph G is a partition of its nodes into two disjoint subsets N_1 and N_2. The set of arcs with one endpoint in N_1 and the other in N_2 is called the* cutset*, and the number of arcs in the cutset is referred to as the* width *of the cut (N_1, N_2).*

The cutwidth *of G is the smallest k for which there exists an ordering (or* layout*) $\nu_1, \ldots, \nu_n$ of the nodes of G such that the width of every cut $(\{\nu_1, \ldots \nu_i\}, \{\nu_{i+1}, \ldots \nu_n\})$ is at most k.*

Definition 2.2. (Treewidth) *A* tree decomposition *of a graph G is a tree T and a collection of* bags *$\{X_i \mid i \text{ is a node of } T\}$. Each bag X_i is a subset of nodes of G, and we require: (i) every node of G is contained in at least one bag X_i; (ii) for each edge of G, some bag X_i contains both its endpoints; and (iii) for all nodes i, j, k of T, if j lies on the unique path from i to k in T, then $X_i \cap X_k \subseteq X_j$.*

The width *of a tree decomposition is defined as $\max |X_i| - 1$, and the* treewidth *of G is the minimum width over all tree decompositions.*

In essence, cutwidth measures the worst "bottleneck" of parallel arcs in an optimal left-to-right layout of nodes that is chosen to make this bottleneck as small as possible, and treewidth measures how far G is from being a tree (in particular, a tree always has treewidth 1). Bodlaender shows that cutwidth $\geq$ treewidth [3]; on the other hand, there are graphs with bounded treewidth and arbitrarily large cutwidth, and so these two parameters measure genuinely different features.

Computing cutwidth and treewidth are both NP-complete [2, 12]. However, for fixed k it can be decided in linear time whether a given graph has cutwidth $\leq k$ and/or treewidth $\leq k$ [4, 32].

3 NP-completeness

In this section we prove Theorem 1.1, i.e., that TAUT ANGLE STRUCTURE is **NP**-complete. As stated earlier, we do this using a reduction from the **NP**-complete problem MONOTONE 1-IN-3-SAT [30].

The overall structure of the proof is as follows. Throughout this section, let $\mathcal{M}$ be a given instance of MONOTONE 1-IN-3 SAT, with t variables $x_1, \ldots, x_t$, and c clauses each of the form $x_i \vee x_j \vee x_k$. We say that $\mathcal{M}$ is *solvable* if and only if there is some assignment of true/false values to the variables so that exactly one of the three terms in each clause is true.

Our strategy is to build a corresponding triangulation $\mathcal{T}_{\mathcal{M}}$ that has a taut angle structure if and only if $\mathcal{M}$ is solvable. We build $\mathcal{T}_{\mathcal{M}}$ by hooking together three types of gadgets, all of which are triangulations with boundary faces: (i) *variable gadgets*, each with two choices of taut angle structure that represent true or false respectively for a single variable x_i of $\mathcal{M}$; (ii) *fork gadgets* that allow us to propagate this choice for x_i to several clauses simultaneously; and (iii) *clause gadgets* that connect three variable gadgets and support an overall taut angle structure if and only if precisely one of the three corresponding variable choices is true.

These gadgets have 2, 21 and 4 tetrahedra respectively, and we describe and analyse them in Sections 3.1, 3.2 and 3.3. We then finish off the proof of Theorem 1.1 in Section 3.4, which is a simple matter of hooking the gadgets together and observing that the entire construction can be done in polynomial time.

We hook the gadgets together along tori: each such torus consists of two faces, three edges and one vertex. To facilitate lemmas and proofs, we assign *types* a, b and c to the three edges of each such torus Θ, as illustrated in Figure 2 (we explicitly describe these edge types for each gadget).

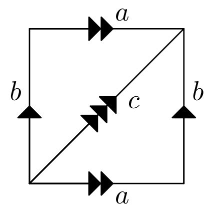

Figure 2: A two-face torus Θ with edge types a, b and c

For any taut angle structure τ, the *boundary pattern* of τ on the torus Θ is the triple (m_a, m_b, m_c), where m_a, m_b and m_c count the number of markings on the edges of type a, b and c respectively. By definition of a taut angle structure, each of $m_a, m_b, m_c \in \{0, 1, 2\}$. We use boundary patterns to represent true/false values of variables in $\mathcal{M}$: in particular, the boundary pattern $(2, 0, 0)$ represents *true*, and the boundary pattern $(0, 2, 0)$ represents *false*.

3.1 The variable gadget A variable gadget is a triangulation with torus boundary that has precisely two taut angle structures: one with boundary pattern $(2, 0, 0)$ (representing *true*), and the other with boundary pattern $(0, 2, 0)$ (representing *false*). We first define the gadget, and then prove the necessary properties.

The construction is simple: we use a $(1, 3, 4)$ *layered solid torus*, a two-tetrahedron instance of a more general and much-studied family of solid torus triangulations [17]. The details are as follows.

Construction 3.1. (Variable gadget) *To build a variable gadget, we begin with the tetrahedron Δ_1 whose vertices are labelled A, B, C, D, and we identify faces ABD and BDC (the rear faces in the diagram). This is the well-known one-tetrahedron triangulation of the solid torus [5, 17], and has three boundary edges $AB = BD = DC$; $AD = BC$; and CA, as illustrated in Figure 3(a).*

We now take a second tetrahedron Δ_2 with vertices labelled E, F, G, H, and glue the remaining boundary faces of Δ_1 to Δ_2 by identifying ABC with EFH and ACD with FGH as indicated in Figure 3(b). This is the two-tetrahedron $(1, 3, 4)$ layered solid torus, with three boundary edges and one internal edge. On the boundary we assign edge types $a \to AB = BD = DC = EF = HG$; $b \to CA = HE = GF$; and $c \to EG$. For completeness, the one internal edge is $AD = BC = FH$.

Observation 3.1. *The variable gadget is a layered solid torus, as described in [17]. In particular, it has just one vertex, whose link is a disc, and its two boundary faces join together to form a torus.*

These properties are true of any layered solid torus;

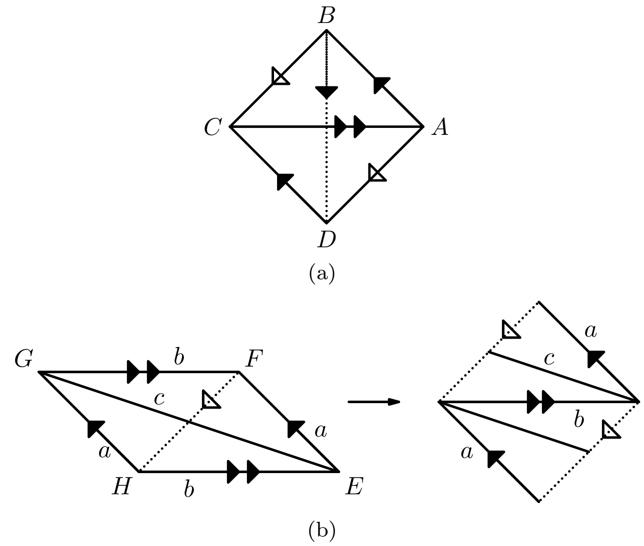

Figure 3: Building a variable gadget

see [5, 17] for more information on the general layered solid torus construction (the details of which are not important here). All of the claims above can also be verified computationally using the software package *Regina* [6, 8].

Lemma 3.1. *The variable gadget supports precisely two taut angle structures: one with boundary pattern $(2, 0, 0)$, and one with boundary pattern $(0, 2, 0)$.*

A theoretical proof is given in the appendix; however, again this is easy to verify computationally using *Regina*, which can enumerate all taut angle structures on a given triangulation.

3.2 The fork gadget A fork gadget is a triangulation that allows us to duplicate a variable x_i from our monotone 1-in-3 sat instance $\mathcal{M}$. Specifically, we can attach a fork gadget to some boundary torus Θ of some triangulation $\mathcal{T}$ with a taut angle structure τ; as a result it produces two new boundary tori that both inherit the *same* boundary pattern with which τ meets Θ. The details are as follows.

Construction 3.2. (Fork gadget) *To build a fork gadget, we begin with an annular prism; that is, the prism over a disc with a hole cut out of the centre, as illustrated in Figure 4(a). We triangulate this prism with 21 tetrahedra: the precise triangulation is important, and is spelled out explicitly in the appendix. As a consequence, this triangulates the outer cylinder with four triangles and four vertices A, B, C, D, and triangulates the inner cylinder with two triangles and two vertices E, F.*

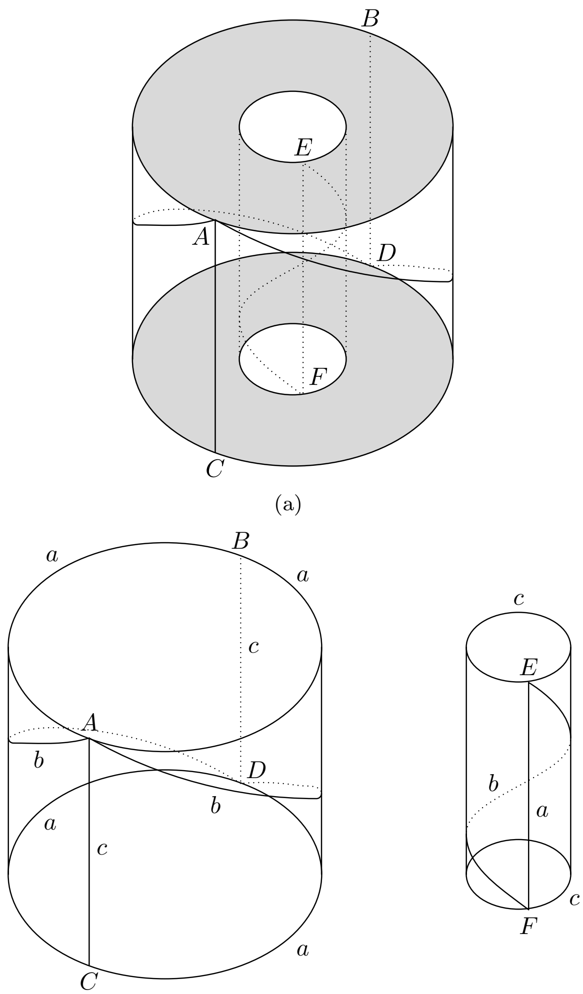

Figure 4: Building a fork gadget

We then glue the top of the prism to the bottom, effectively creating a "hollow" solid torus; that is, a manifold with a torus boundary component on the outside and another torus boundary component on the inside. We assign edge types a, b, c to the three edges of the inner torus, and also to the six edges of the outer torus; the precise labellings are shown in Figure 4(b).

Lemma 3.2. *Let $\mathcal{T}$ be a 3-manifold triangulation, two of whose boundary faces form a two-triangle torus Θ with the usual a, b, c edge types. Let $\mathcal{T}'$ be the new triangulation obtained by attaching a fork gadget to Θ along the right-hand side of the outer torus of the fork gadget, as illustrated in Figure 5(a), so that the edge types a, b, c match. Then $\mathcal{T}'$ is a 3-manifold triangulation with four new boundary faces that form two disjoint two-triangle tori Θ', Θ'', as illustrated in Figure 5(b).*

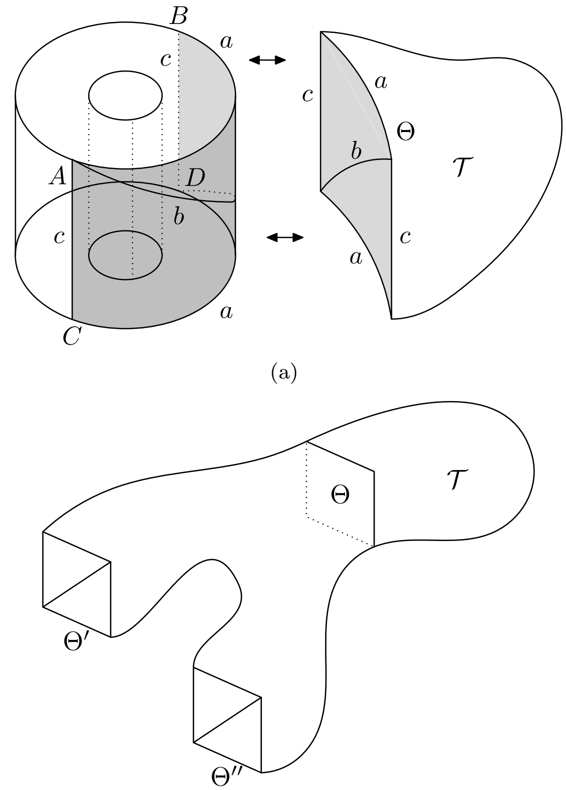

Figure 5: Attaching a fork gadget to the 3-manifold triangulation $\mathcal{T}$

Let τ be a taut angle structure on $\mathcal{T}$ that meets the original boundary torus Θ in one of the patterns $(2, 0, 0)$ or $(0, 2, 0)$. Then we can extend τ through the fork gadget to obtain a taut angle structure τ' on $\mathcal{T}'$. Moreover, every such extension τ' meets the new boundary tori Θ', Θ'' in the same boundary pattern with which τ meets Θ.

In other words: a fork gadget allows us to duplicate a boundary torus in a way that also *duplicates the boundary patterns* of taut angle structures. We prove this lemma in the appendix; again we note that these claims can be verified computationally using the software package *Regina*.

Remark. Constructing the fork gadget was the most difficult aspect of this paper. It involved an interplay between theory and computation, and the triangulation includes substructures explicitly tailored to eliminate unwanted extensions of τ. See the full version of this paper for details.

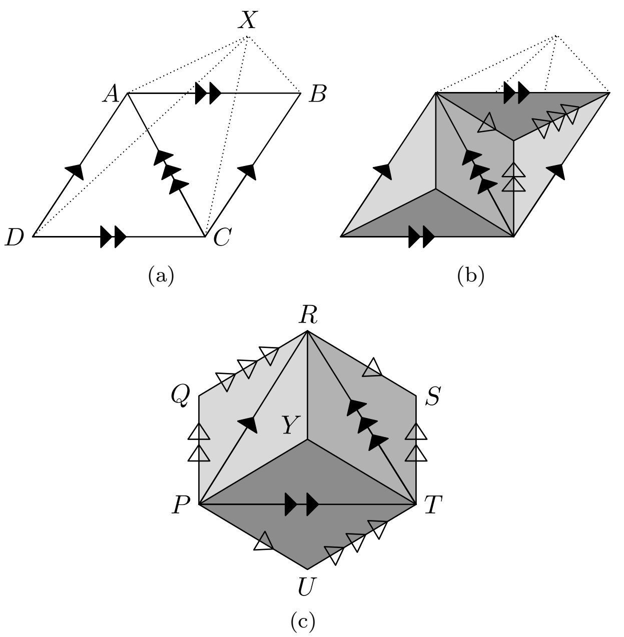

Figure 6: Building a clause gadget

3.3 The clause gadget A clause gadget represents a clause $x_i \vee x_j \vee x_k$ from our MONOTONE 1-IN-3 SAT instance $\mathcal{M}$. We can attach it to three boundary tori of some triangulation $\mathcal{T}$ with a taut angle structure τ, and τ will only extend through the clause gadget if exactly one of its boundary patterns on the tori is $(2,0,0)$ (*true*), and the other two are $(0,2,0)$ (*false*). The details are as follows.

CONSTRUCTION 3.3. (CLAUSE GADGET) *To build a clause gadget, we begin with a two-triangle torus $ABCD$ and cone it to a point using two tetrahedra, as illustrated in Figure 6(a) (so the upper and lower faces* ABX *and* DCX *are joined, as are the left and right faces* ADX *and* BCX*). This makes the cone point X an ideal vertex (its link is a torus). There are two boundary faces remaining (at the front of the diagram); to each we attach a new tetrahedron, as shown in Figure 6(b). We now have six boundary faces that together form a torus, which concludes our construction.*

For convenience, Figure 6(c) presents this same boundary torus as a hexagon $PQRSTU$; this will make the attaching process easier to describe in Lemma 3.3 below.

LEMMA 3.3. *Let $\mathcal{T}$ be a 3-manifold triangulation (possibly disconnected), six of whose boundary faces form three disjoint two-triangle tori Θ_1, Θ_2, Θ_3 each with the usual a, b, c edge types. Let $\mathcal{T}'$ be the new triangulation obtained by attaching these three tori to a clause gadget along the rectangles $PQRY$, $RSTY$ and $TUPY$*

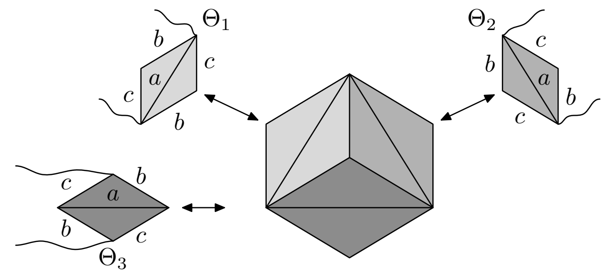

Figure 7: Attaching a clause gadget to three boundary tori of $\mathcal{T}$

respectively, as illustrated in Figure 7; in particular, so the type a edges join to PR, RT and TP, and the type b edges join to QR and PY for Θ_1, ST and RY for Θ_2, and UP and TY for Θ_3. Then $\mathcal{T}'$ is a 3-manifold triangulation.

Let τ be a taut angle structure on $\mathcal{T}$ that meets each torus $\Theta_1, \Theta_2, \Theta_3$ in one of the patterns $(2,0,0)$ or $(0,2,0)$. Then we can extend τ through the clause gadget to obtain a taut angle structure τ' on $\mathcal{T}'$ if and only if exactly one of these three boundary patterns is $(2,0,0)$.

We give a theoretical proof in the appendix, but again this can also be verified computationally.

3.4 Proving Theorem 1.1 Now that we are equipped with our various gadgets, the proof of Theorem 1.1 is straightforward. In summary: we build a variable gadget for each variable x_i of $\mathcal{M}$, duplicate its boundary torus using fork gadgets until we have one "copy" for each time x_i occurs in a clause of $\mathcal{M}$, and then hook these boundary tori together using clause gadgets. Lemmas 3.1, 3.2 and 3.3 together ensure that the resulting triangulation (which is indeed orientable with no boundary faces) has a taut angle structure if and only if $\mathcal{M}$ is solvable. Full details of the proof are given in the appendix.

4 Fixed-parameter tractability

So far we have shown that detecting taut angle structures is hard in general. However, in practice running times are often surprisingly fast. This leads us to the natural question of whether the running time can be improved if we restrict ourselves to more specific classes of triangulations.

The way we approach this question here is to prove that TAUT ANGLE STRUCTURE is fixed-parameter tractable in both the cutwidth and the treewidth of the face pairing graph of the triangulation. The precise results are given by Theorem 1.2 (for cutwidth) and Theorem 1.3 (for treewidth), which are both stated in the introduction. In this section we outline the proofs for both of these theorems. Full details of the proofs can be found in the appendix.

Note that the precise running times given here (parameterised by both the number of tetrahedra n and the cutwidth/treewidth k) assume that we are given extra information alongside our triangulation: for Theorem 1.2 we assume a left-to-right ordering (or layout) of nodes that corresponds to a cutwidth of $\leq k$, and for Theorem 1.3 we assume a tree decomposition of width $\leq k$ with $O(n)$ tree nodes. In contrast, Corollary 1.1 (that TAUT ANGLE STRUCTURE is linear-time fixed-parameter tractable in the treewidth) does not require any such information, since in the setting of bounded treewidth we can use the algorithm of Bodlaender [4] to compute such a tree decomposition in time linear in n.

4.1 Bounded cutwidth: Proving Theorem 1.2

Assume we have an ordering $v_1, \ldots, v_n$ of the nodes of $\Gamma(\mathcal{T})$ that yields cutwidth $\leq k$, and recall that every node v_i corresponds to a tetrahedron Δ_{v_i} of the triangulation $\mathcal{T}$. Following a dynamic programming approach, we examine sub-triangulations $\mathcal{T}_i$ ($1 \leq i \leq n$) defined by considering only tetrahedra $\Delta_{v_1}, \ldots, \Delta_{v_i}$. By construction, every boundary face of $\mathcal{T}_i$ corresponds to an arc in the cutset $(\{v_1, \ldots, v_i\}, \{v_{i+1}, \ldots, v_n\})$. It follows that $\mathcal{T}_i$ has at most k boundary faces and thus at most $3k/2$ boundary edges. Since each boundary edge is marked 0, 1 or 2 times, this leaves us with at most $3^{3k/2}$ possible marking patterns on the boundary of $\mathcal{T}_i$ that correspond to taut angle structures on $\mathcal{T}_i$.

Working our way through $\mathcal{T}_1, \ldots, \mathcal{T}_n = \mathcal{T}$, for each i we determine which marking patterns on the boundary of $\mathcal{T}_i$ correspond to taut angle structures on $\mathcal{T}_i$. At each step we match the previous boundary patterns on $\mathcal{T}_{i-1}$ with the three choices of markings for Δ_i, thereby computing the possible marking patterns on the boundary of $\mathcal{T}_i$. There are n such steps, each involving $O(k)$ work on at most $3^{3k/2}$ boundary patterns, giving an overall running time of $O(nk \cdot 3^{3k/2})$.

4.2 Bounded treewidth: Proving Theorem 1.3

Here we adopt a similar approach, but this time we do our dynamic programming over a tree. Suppose we are given a tree decomposition of $\Gamma(\mathcal{T})$ of width $\leq k$ with $O(n)$ tree nodes. We arbitrarily choose a root, so that our tree becomes a hierarchy of subtrees as illustrated in Figure 8.

Recall that each node ν of the tree corresponds to a *bag* of tetrahedra in the tree decomposition. We define the sub-triangulation $\mathcal{T}_\nu$ by considering only those tetrahedra that appear only in bags within the subtree rooted at ν, excluding any tetrahedra that appear in bags outside this subtree.

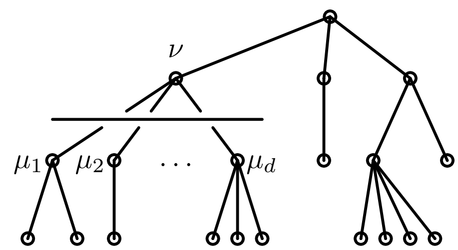

Figure 8: Dynamic programming over a tree decomposition

In a key observation, we show that each sub-triangulation $\mathcal{T}_\nu$ has $\leq 6(k+1)$ boundary edges. Working our way up the tree from the leaves to the root, we determine for each tree node ν which of the $3^{6(k+1)}$ possible marking patterns on the boundary of $\mathcal{T}_\nu$ correspond to taut angle structures on $\mathcal{T}_\nu$. Once we reach the root of the tree (where $\mathcal{T}_\nu = \mathcal{T}$), we know that $\mathcal{T}$ has a taut angle structure if and only if our list of possible marking patterns is non-empty.

The recursive step in our dynamic programming works as follows. Consider any node ν in the tree, with immediate child nodes $\mu_1, \ldots, \mu_d$. We determine the possible marking patterns on the boundary of $\mathcal{T}_\nu$ by aggregating the possible marking patterns on the boundaries of $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$, and combining these with the possible markings on any new tetrahedra in $\mathcal{T}_\nu$. This aggregation is made fast by observing that the triangulations $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$ use distinct tetrahedra, and have $\leq 6(k+1)$ boundary edges *in total*. This allows us to perform the recursive step in time $O(k \cdot 3^{7k})$, and thereby solve the entire problem in time $O(nk \cdot 3^{7k})$.

5 Discussion

Theorem 1.1 shows that even if we restrict our attention to orientable triangulations with no boundary faces, detecting taut angle structures is still **NP**-complete. However, our construction creates triangulations with many ideal vertices. It would be interesting to know if this **NP**-completeness result could be tightened to detecting taut angle structures on *one-vertex* triangulations.

We have an explicit script that uses the software package *Regina* to build the triangulation $\mathcal{T}_\mathcal{M}$ for a given MONOTONE 1-IN-3 SAT instance $\mathcal{M}$. In the full version of this paper we discuss this script further and describe some of the 3-manifolds that it produces.

Theorems 1.2 and 1.3 help explain why taut angle structures are relatively easy to detect in practice [15]: there are many triangulations $\mathcal{T}$ for which $\Gamma(\mathcal{T})$ has small cutwidth and/or treewidth.[2] In the closed setting, for instance, the conjectured minimal triangulations of many Seifert fibred spaces have extremely small treewidth [24, 26], and common building blocks such as layered solid tori and triangular prisms have treewidth 1 and 3 respectively. Small cutwidth and treewidth triangulations have also been found fast to work with in other settings, such as normal surface theory [7].

It is worth considering whether we can find a more powerful parameter than the treewidth of the face pairing graph. In the more general setting of constraint satisfaction problems, it is known (under certain hypotheses) that treewidth essentially yields the best algorithms [25]. In our setting, however, we are also subject to strong topological constraints that are difficult to analyse in a purely combinatorial framework, and that may provide new opportunities for optimisation.

Looking forward: the frameworks in this paper for **NP**-completeness and fixed-parameter tractability have significant potential for use with *normal surface theory*, which is the central algorithmic machine for solving problems such as unknot recognition, 3-sphere recognition, prime decomposition, and many more. A key feature of both taut angle structures and normal surfaces is that they correspond to vertices of a high-dimensional polytope subject to simple combinatorial constraints derived from the tetrahedra [18, 23]. Moreover, both taut angle structures and normal surfaces can be incrementally "extended" through different sections of the triangulation according to constraints derived from the local face gluings—a technique used throughout this paper.

Although normal surfaces are more numerous and more difficult to work with, this common foundation gives us hope that the techniques developed here could, with further research, be used to tackle some of the fundamental open complexity problems in knot theory and 3-manifold topology.

References

[1] Ian Agol, Joel Hass, and William Thurston, *The computational complexity of knot genus and spanning area*, Trans. Amer. Math. Soc. **358** (2006), no. 9, 3821–3850 (electronic).

[2] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods **8** (1987), no. 2, 277–284.

[3] Hans L. Bodlaender, *Classes of graphs with bounded tree-width*, Technical Report RUU-CS-86-22, Utrecht University, 1986.

[4] ———, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput. **25** (1996), no. 6, 1305–1317.

[5] Benjamin A. Burton, *Face pairing graphs and 3-manifold enumeration*, J. Knot Theory Ramifications **13** (2004), no. 8, 1057–1101.

[6] ———, *Introducing Regina, the 3-manifold topology software*, Experiment. Math. **13** (2004), no. 3, 267–272.

[7] ———, *Optimizing the double description method for normal surface enumeration*, Math. Comp. **79** (2010), no. 269, 453–484.

[8] Benjamin A. Burton, Ryan Budney, William Pettersson, et al., *Regina: Software for 3-manifold topology and normal surface theory*, `http://regina.sourceforge.net/`, 1999–2012.

[9] R. G. Downey and M. R. Fellows, *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag, New York, 1999.

[10] Nathan M. Dunfield and Anil N. Hirani, *The least spanning area of a knot and the optimal bounding chain problem*, SCG '11: Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry, ACM, 2011, pp. 135–144.

[11] David Futer and François Guéritaud, *From angled triangulations to hyperbolic structures*, Interactions Between Hyperbolic Geometry, Quantum Topology and Number Theory, Contemp. Math., vol. 541, Amer. Math. Soc., Providence, RI, 2011, pp. 159–182.

[12] Michael R. Garey and David S. Johnson, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979, A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[13] Joel Hass, *New results on the complexity of recognizing the 3-sphere*, Oberwolfach Rep. **9** (2012), no. 2, 1425–1426.

[14] Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger, *The computational complexity of knot and link problems*, J. Assoc. Comput. Mach. **46** (1999), no. 2, 185–211.

[15] 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.

[16] William Jaco, *The homeomorphism problem: Classification of 3-manifolds*, Lecture notes, Available from `http://www.math.okstate.edu/~jaco/pekinglectures.htm`, 2005.

[17] William Jaco and J. Hyam Rubinstein, *0-efficient triangulations of 3-manifolds*, J. Differential Geom. **65** (2003), no. 1, 61–168.

[18] Ensil Kang and J. Hyam Rubinstein, *Ideal triangulations of 3-manifolds II; Taut and angle structures*, Algebr. Geom. Topol. **5** (2005), 1505–1533.

[19] Bruce Kleiner and John Lott, *Notes on Perelman's*

[2] In contrast, there are at present no conjectured examples of manifolds that do *not* have small treewidth triangulations. Such a conjecture would likely be extremely difficult to prove.

papers, Geom. Topol. **12** (2008), no. 5, 2587–2855.

[20] Ephraim Korach and Nir Solel, *Tree-width, path-width, and cutwidth*, Discrete Appl. Math. **43** (1993), no. 1, 97–101.

[21] Greg Kuperberg, *Knottedness is in NP, modulo GRH*, Preprint, `arXiv:1112.0845`, November 2011.

[22] Marc Lackenby, *Taut ideal triangulations of 3-manifolds*, Geom. Topol. **4** (2000), 369–395.

[23] Feng Luo and Stephan Tillmann, *Angle structures and normal surfaces*, Trans. Amer. Math. Soc. **360** (2008), no. 6, 2849–2866.

[24] Bruno Martelli and Carlo Petronio, *Complexity of geometric three-manifolds*, Geom. Dedicata **108** (2004), no. 1, 15–69.

[25] Daniel Marx, *Can you beat treewidth?*, Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2007, pp. 169–179.

[26] Sergei V. Matveev, *Tables of 3-manifolds up to complexity 6*, Max-Planck-Institut für Mathematik Preprint Series (1998), no. 67, available from `http://www.mpim-bonn.mpg.de/html/preprints/preprints.html`.

[27] Igor Rivin, *Euclidean structures on simplicial surfaces and hyperbolic volume*, Ann. of Math. (2) **139** (1994), no. 3, 553–580.

[28] ______, *Combinatorial optimization in geometry*, Adv. in Appl. Math. **31** (2003), no. 1, 242–271.

[29] J. Hyam Rubinstein, *An algorithm to recognize the 3-sphere*, Proceedings of the International Congress of Mathematicians (Zürich, 1994), vol. 1, Birkhäuser, 1995, pp. 601–611.

[30] Thomas J. Schaefer, *The complexity of satisfiability problems*, Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978), ACM, 1978, pp. 216–226.

[31] Saul Schleimer, *Sphere recognition lies in NP*, Low-dimensional and Symplectic Topology (Michael Usher, ed.), Proceedings of Symposia in Pure Mathematics, vol. 82, Amer. Math. Soc., 2011, pp. 183–214.

[32] Dimitrios M. Thilikos, Maria Serna, and Hans L. Bodlaender, *Cutwidth I: A linear time fixed parameter algorithm*, J. Algorithms **56** (2005), no. 1, 1–24.

[33] William P. Thurston, *The geometry and topology of 3-manifolds*, Lecture notes, Princeton University, 1978.

Appendix: Additional proofs

In this appendix we include detailed proofs and constructions that were omitted from the main text for reasons of space.

The variable gadget In Section 3.1 we describe the variable gadget in detail. Here we give a full proof of Lemma 3.1, which we restate below. We refer to the labels and diagrams from Section 3.1 throughout.

LEMMA 3.1. *The variable gadget supports precisely two taut angle structures: one with boundary pattern* $(2,0,0)$*, and one with boundary pattern* $(0,2,0)$*.*

Proof. Consider the internal edge $AD = BC = FH$. Since edges AD and BD are opposite in tetrahedron Δ_1, the only way to mark this internal edge *twice* is to mark edges AD and BC of tetrahedron Δ_1, and to not mark edge FH in tetrahedron Δ_2.

Therefore our choice for Δ_1 is forced, and we are left with two options for what to mark in Δ_2. We could either mark EF and HG, which yields a taut structure with boundary pattern $(2,0,0)$, or we could mark EH and FG, which yields a taut angle structure with boundary pattern $(0,2,0)$.

	Face 123	Face 124	Face 134	Face 234
Δ_1	$\Delta_{18}:312$	$\Delta_{13}:312$	$\Delta_{18}:324$	$\Delta_4\ :234$
Δ_2	$\Delta_7\ :342$	$\Delta_{20}:342$	$\Delta_{20}:312$	$\Delta_{11}:234$
Δ_3	$\Delta_{19}:312$	$\Delta_8\ :342$	$\Delta_{16}:321$	*Outer L*
Δ_4	*Outer L*	$\Delta_9\ :124$	*Outer R*	$\Delta_1\ :234$
Δ_5	$\Delta_{16}:342$	$\Delta_{15}:321$	$\Delta_{19}:324$	*Outer R*
Δ_6	*Inner*	$\Delta_{12}:342$	$\Delta_{21}:312$	*Upper*
Δ_7	*Inner*	$\Delta_{11}:132$	$\Delta_8\ :134$	$\Delta_2\ :312$
Δ_8	$\Delta_{14}:324$	$\Delta_{14}:321$	$\Delta_7\ :134$	$\Delta_3\ :412$
Δ_9	*Upper*	$\Delta_4\ :124$	*Upper*	$\Delta_{10}:234$
Δ_{10}	$\Delta_{13}:324$	$\Delta_{11}:124$	$\Delta_{11}:134$	$\Delta_9\ :234$
Δ_{11}	$\Delta_7\ :142$	$\Delta_{10}:124$	$\Delta_{10}:134$	$\Delta_2\ :234$
Δ_{12}	$\Delta_{21}:423$	$\Delta_{13}:124$	$\Delta_{13}:134$	$\Delta_6\ :412$
Δ_{13}	$\Delta_1\ :241$	$\Delta_{12}:124$	$\Delta_{12}:134$	$\Delta_{10}:213$
Δ_{14}	$\Delta_8\ :421$	$\Delta_{15}:124$	$\Delta_{15}:134$	$\Delta_8\ :213$
Δ_{15}	$\Delta_5\ :421$	$\Delta_{14}:124$	$\Delta_{14}:134$	*Lower*
Δ_{16}	$\Delta_3\ :431$	$\Delta_{17}:124$	$\Delta_{17}:134$	$\Delta_5\ :312$
Δ_{17}	*Lower*	$\Delta_{16}:124$	$\Delta_{16}:134$	*Lower*
Δ_{18}	$\Delta_1\ :231$	$\Delta_{19}:124$	$\Delta_{19}:134$	$\Delta_1\ :314$
Δ_{19}	$\Delta_3\ :231$	$\Delta_{18}:124$	$\Delta_{18}:134$	$\Delta_5\ :314$
Δ_{20}	$\Delta_2\ :341$	$\Delta_{21}:124$	$\Delta_{21}:134$	$\Delta_2\ :412$
Δ_{21}	$\Delta_6\ :341$	$\Delta_{20}:124$	$\Delta_{20}:134$	$\Delta_{12}:231$

Table 1: The 21-tetrahedron triangulation of the prism over the annulus

The fork gadget In Section 3.2 we outline the construction of the fork gadget, but we do not give the precise 21-tetrahedron triangulation of the annular prism. Here we present this 21-tetrahedron triangulation in full. Once again we refer to the labels and diagrams from Section 3.2.

Table 1 lists the individual face gluings for the annular prism as shown in Figure 4(a). There are 21 tetrahedra labelled $\Delta_1,\ldots,\Delta_{21}$, and the four vertices of each tetrahedron are labelled $1,2,3,4$. Each row of the table represents a tetrahedron, and each column

	Face 123	Face 124	Face 134	Face 234
Δ_1	$\Delta_{18} : 312$	$\Delta_{13} : 312$	$\Delta_{18} : 324$	$\Delta_4 : 234$
Δ_2	$\Delta_7 : 342$	$\Delta_{20} : 342$	$\Delta_{20} : 312$	$\Delta_{11} : 234$
Δ_3	$\Delta_{19} : 312$	$\Delta_8 : 342$	$\Delta_{16} : 321$	*Outer L*
Δ_4	*Outer L*	$\Delta_9 : 124$	*Outer R*	$\Delta_1 : 234$
Δ_5	$\Delta_{16} : 342$	$\Delta_{15} : 321$	$\Delta_{19} : 324$	*Outer R*
Δ_6	*Inner*	$\Delta_{12} : 342$	$\Delta_{21} : 312$	$\Delta_{15} : 432$
Δ_7	*Inner*	$\Delta_{11} : 132$	$\Delta_8 : 134$	$\Delta_2 : 312$
Δ_8	$\Delta_{14} : 324$	$\Delta_{14} : 321$	$\Delta_7 : 134$	$\Delta_3 : 412$
Δ_9	$\Delta_{17} : 213$	$\Delta_4 : 124$	$\Delta_{17} : 234$	$\Delta_{10} : 234$
Δ_{10}	$\Delta_{13} : 324$	$\Delta_{11} : 124$	$\Delta_{11} : 134$	$\Delta_9 : 234$
Δ_{11}	$\Delta_7 : 142$	$\Delta_{10} : 124$	$\Delta_{10} : 134$	$\Delta_2 : 234$
Δ_{12}	$\Delta_{21} : 423$	$\Delta_{13} : 124$	$\Delta_{13} : 134$	$\Delta_6 : 412$
Δ_{13}	$\Delta_1 : 241$	$\Delta_{12} : 124$	$\Delta_{12} : 134$	$\Delta_{10} : 213$
Δ_{14}	$\Delta_8 : 421$	$\Delta_{15} : 124$	$\Delta_{15} : 134$	$\Delta_8 : 213$
Δ_{15}	$\Delta_5 : 421$	$\Delta_{14} : 124$	$\Delta_{14} : 134$	$\Delta_6 : 432$
Δ_{16}	$\Delta_3 : 431$	$\Delta_{17} : 124$	$\Delta_{17} : 134$	$\Delta_5 : 312$
Δ_{17}	$\Delta_9 : 213$	$\Delta_{16} : 124$	$\Delta_{16} : 134$	$\Delta_9 : 134$
Δ_{18}	$\Delta_1 : 231$	$\Delta_{19} : 124$	$\Delta_{19} : 134$	$\Delta_1 : 314$
Δ_{19}	$\Delta_3 : 231$	$\Delta_{18} : 124$	$\Delta_{18} : 134$	$\Delta_5 : 314$
Δ_{20}	$\Delta_2 : 341$	$\Delta_{21} : 124$	$\Delta_{21} : 134$	$\Delta_2 : 412$
Δ_{21}	$\Delta_6 : 341$	$\Delta_{20} : 124$	$\Delta_{20} : 134$	$\Delta_{12} : 231$

Table 2: The complete triangulation of the fork gadget

represents one of its four faces. For instance, the top-left cell of table indicates that the face with vertices $1, 2, 3$ of tetrahedron Δ_1 is glued to the face with vertices $3, 1, 2$ (in that order) of tetrahedron Δ_{18} (the same gluing can be seen from the other side in the fourth-last row of the table).

There are 12 faces on the boundary of the annular prism: three on the upper annulus with vertices A, B, E (marked *Upper* in the table); three on the lower annulus with vertices C, D, F (marked *Lower* in the table); four on the outer cylinder with vertices A, B, C, D (two on the left-hand side of the diagram marked *Outer L*, and two on the right-hand side of the diagram marked *Outer R*), and finally two on the inner cylinder with vertices E, F (marked *Inner* in the table).

The final stage of the construction is to glue the upper annulus to the lower annulus. The result is shown in Table 2, where only six boundary faces remain (on the outer and inner cylinders). This triangulation is the fork gadget in its entirety.

For reference, the six boundary faces as shown in Figure 4(b) are as follows:

- the upper triangle ABD on the left-hand side of the outer cylinder is $\Delta_4 : 213$;
- the lower triangle ACD on the left-hand side of the outer cylinder is $\Delta_3 : 243$;
- the upper triangle ABD on the right-hand side of the outer cylinder is $\Delta_4 : 413$;
- the lower triangle ACD on the right-hand side of the outer cylinder is $\Delta_5 : 423$;
- the upper triangle EFE on the inner cylinder is $\Delta_6 : 312$;
- the lower triangle FEF on the inner cylinder is $\Delta_7 : 321$.

We now give a full proof of Lemma 3.2, which we restate below. Because the fork gadget contains 21 tetrahedra, a theoretical analysis of the possible taut angle structures would be onerous; therefore we use the software package *Regina* to assist with our analysis. Once again, we refer to the labels and diagrams from Section 3.2.

LEMMA 3.2. *Let $\mathcal{T}$ be a 3-manifold triangulation, two of whose boundary faces form a two-triangle torus Θ with the usual a, b, c edge types. Let $\mathcal{T}'$ be the new triangulation obtained by attaching a fork gadget to Θ along the right-hand side of the outer torus of the fork gadget, as illustrated in Figure 5(a), so that the edge types a, b, c match. Then $\mathcal{T}'$ is a 3-manifold triangulation with four new boundary faces that form two disjoint two-triangle tori Θ', Θ'', as illustrated in Figure 5(b).*

Let τ be a taut angle structure on $\mathcal{T}$ that meets the original boundary torus Θ in one of the patterns $(2, 0, 0)$ or $(0, 2, 0)$. Then we can extend τ through the fork gadget to obtain a taut angle structure τ' on $\mathcal{T}'$. Moreover, every such extension τ' meets the new boundary tori Θ', Θ'' in the same boundary pattern with which τ meets Θ.

Proof. It is simple to see that the new triangulation $\mathcal{T}'$ satisfies our conditions for a 3-manifold triangulation: the only new edge identification that results from the gluing the fork gadget to the torus Θ is that the vertical edges AC and BD on the outer cylinder become identified together (note that the upper right-hand edge AB and the lower right-hand edge CD are already identified, and the gluing to Θ is consistent with this). In particular, no edge becomes glued to itself in reverse.

The inner cylinder (with vertices $E = F$) remains unchanged, and becomes the first new torus boundary component Θ' (recall again that the upper edge of the inner cylinder is already glued to the lower edge within the fork gadget). As the vertical edges AC and BD become identified, the two remaining faces on the left-hand side of the outer cylinder form a second boundary torus Θ'', with one horizontal edge $a \to AB = CD$,

Edge	τ_1	τ_2	τ_3	τ_4
Inner cylinder				
Vertical edge (a)	0	0	2	2
Diagonal edge (b)	2	2	0	0
Horizontal edge (c)	0	0	0	0
Outer cylinder				
Horizontal left edge (a, on Θ'')	1	0	2	1
Horizontal right edge (a, meets Θ)	1	2	0	1
Diagonal left edge (b, on Θ'')	1	2	0	1
Diagonal right edge (b, meets Θ)	1	0	2	1
Vertical front edge ($c \to AC$)	0	0	0	0
Vertical rear edge ($c \to BD$)	0	0	0	0

Table 3: Edge markings from the four taut angle structures on the fork gadget

one diagonal edge $b \to AD$, and one vertical edge $c \to AC = BD$.

We now examine how the taut structure τ can be extended through the fork gadget. By entering Table 2 as a triangulation into the software package *Regina* [6, 8][3] and enumerating all taut angle structures, we find that the fork gadget has precisely four taut angle structures $\tau_1, \tau_2, \tau_3, \tau_4$. Table 3 lists the number of times that each τ_i marks each edge on the inner and outer cylinder of the fork gadget; the edges are identified by their labels as shown in Figures 4(b) and 5(a).

From here the proof is a simple matter of chasing edge markings around the diagram.

- Suppose that τ meets the torus Θ in the boundary pattern $(2, 0, 0)$.

  Consider edge a on the torus Θ, which becomes an internal edge of the final triangulation $\mathcal{T}'$. Since edge a on Θ already has two markings, and since this edge is joined to the horizontal right edge a on the outer cylinder of the fork gadget, this latter edge must have *zero* markings within the fork gadget. This means that the only compatible taut angle structure within the fork gadget is τ_3.

  Likewise, edge b on the torus Θ becomes an internal edge of $\mathcal{T}'$. Since edge b on Θ has no markings, it requires two markings from within the fork gadget; we see from Table 3 that τ_3 provides this as required.

  The vertical edge c on the torus Θ becomes the boundary edge c of the new boundary torus Θ''. Since this edge receives no markings from either Θ or τ_3, it has no markings in the final triangulation $\mathcal{T}'$.

  This shows that combining τ_3 with τ gives us a taut angle structure on $\mathcal{T}'$; that is, τ can indeed be extended through the fork gadget (and this is the only one way of doing so). We now examine the boundary patterns that arise on the new boundary tori Θ' and Θ''.

  We have already seen above that edge c on the torus Θ'' receives no markings at all. The remaining edges of Θ' and Θ'' are all new to the fork gadget, and so any markings on them must come from τ_3. Reading these figures from Table 3, we see that the inner torus Θ' receives a final boundary pattern of $(2, 0, 0)$, and the outer torus Θ'' likewise receives a final boundary pattern of $(2, 0, 0)$.

- Suppose instead that τ meets the torus Θ in the boundary pattern $(0, 2, 0)$.

  We can follow a similar argument as before. This time edge b on Θ already has two markings, and so the diagonal right edge b on the outer cylinder of the fork gadget must have no markings within the fork gadget, forcing us to choose τ_2. As before we see that the markings from τ_2 are consistent on the gluing torus Θ, and leave the new boundary tori Θ' and Θ'' with boundary patterns $(0, 2, 0)$ and $(0, 2, 0)$ respectively.

Therefore any such taut angle structure τ on $\mathcal{T}$ can be extended through the fork gadget, and the resulting boundary patterns on both new tori Θ' and Θ'' will be identical to the original boundary pattern on Θ.

The clause gadget We come now to the clause gadget, which is described in detail in Section 3.3. Here we prove Lemma 3.3, and once more we refer the reader to the labels and diagrams given in Section 3.3.

LEMMA 3.3. *Let $\mathcal{T}$ be a 3-manifold triangulation (possibly disconnected), six of whose boundary faces form three disjoint two-triangle tori Θ_1, Θ_2, Θ_3 each with the usual a, b, c edge types. Let $\mathcal{T}'$ be the new triangulation obtained by attaching these three tori to a clause gadget along the rectangles $PQRY$, $RSTY$ and $TUPY$ respectively, as illustrated in Figure 7; in particular, so the type a edges join to PR, RT and TP, and the type b edges join to QR and PY for Θ_1, ST and RY for Θ_2, and UP and TY for Θ_3. Then $\mathcal{T}'$ is a 3-manifold triangulation.*

Let τ be a taut angle structure on $\mathcal{T}$ that meets each torus $\Theta_1, \Theta_2, \Theta_3$ in one of the patterns $(2, 0, 0)$ or

[3] Readers are welcome to download *Regina* and try this for themselves. However, a word of caution: *Regina* numbers its tetrahedra and vertices starting from 0, not 1. Therefore all tetrahedron labels and all vertex numbers in Table 2 must be reduced by 1.

$(0,2,0)$. *Then we can extend* τ *through the clause gadget to obtain a taut angle structure* τ' *on* $\mathcal{T}'$ *if and only if exactly one of these three boundary patterns is* $(2,0,0)$.

Proof. Once again it is simple to see that $\mathcal{T}'$ satisfies our conditions for a 3-manifold triangulation: as a result of the torus gluings, we obtain new edge identifications on the clause gadget as follows: $QP = RY = ST$; $QR = PY = UT$; and $RS = YT = PU$. None of these identifications cause an edge to be glued to itself in reverse.

Now let τ be a taut angle structure on $\mathcal{T}$ as described in the statement of the lemma. We examine how we might extend τ through the clause gadget.

- Consider the three edges AB, BC and CA. These become three distinct internal edges after the gluings, and so they require six markings between them. However, every pair of opposite edges in every tetrahedron from the clause gadget only includes one of these three edges, and so between them these three edges can only receive at most four markings from within the clause gadget itself.

  Therefore edges AB, BC and CA must receive at least two markings from the external structure τ. Since AB, BC and CA are all joined to edges of type a on the tori $\Theta_1, \Theta_2, \Theta_3$, it follows that at least one of the boundary patterns on these three tori must be $(2,0,0)$.

- Now consider the three edges $QP = RY = ST$, $QR = PY = UT$ and $RS = YT = PU$. Again these become three distinct internal edges after the gluings, and so again they require six markings between them. However, this time they can only receive *two* markings from within the clause gadget: these edges do not meet the "cone tetrahedra" from Figure 6(a) at all, and the two tetrahedra that we attach in Figure 6(b) feature these edges only once for any pair of opposite edges.

  Therefore edges $QP = RY = ST$, $QR = PY = UT$ and $RS = YT = PU$ need at least four markings between them from the external structure τ. Since these edges are all joined to edges of types b and c on the tori $\Theta_1, \Theta_2, \Theta_3$, it follows that at least two of the boundary patterns on these tori must be $(0,2,0)$.

We have now established that, if there is any hope to extend τ through the clause gadget, exactly one of the three boundary patterns on the three tori Θ_1, Θ_2 and Θ_3 must be $(2,0,0)$. We must still show that such an extension is possible.

Our clause gadget is symmetric, and so without loss of generality we can suppose that τ meets Θ_1 in the boundary pattern $(2,0,0)$, and meets Θ_2 and Θ_3 in the boundary pattern $(0,2,0)$. We mark edges within the clause gadget as follows: for the "cone tetrahedra" in Figure 6(a), we mark edge pairs AB/CX and DC/AX. For each of the two tetrahedra that we attach in Figure 6(b), we mark AC and the corresponding opposite edge.

From here it is a simple matter to follow the edge markings through the clause gadget and verify that every edge of the clause gadget receives exactly two markings in total; that is, τ can indeed be extended through the clause gadget as required.

Proving Theorem 1.1 In Section 3.4 we outline a proof of Theorem 1.1, which we restate below. Here we give the full details of this proof.

THEOREM 1.1. *The problem* TAUT ANGLE STRUCTURE *is* ***NP****-complete.*

Proof. First we note that TAUT ANGLE STRUCTURE is clearly in **NP**: if a taut angle structure exists, then the corresponding edge markings form a linear-sized certificate that is simple to verify in small polynomial time.

To show that TAUT ANGLE STRUCTURE is **NP**-complete, we give a polynomial reduction from MONOTONE 1-IN-3 SAT. Let $\mathcal{M}$ be an instance of MONOTONE 1-IN-3 SAT, as described at the beginning of Section 3, with t variables $x_1, \ldots, x_t$, and with c clauses each of the form $x_i \vee x_j \vee x_k$. For simplicity we assume that each variable appears in at least one clause (otherwise it can be harmlessly removed). We build a corresponding triangulation $\mathcal{T}_{\mathcal{M}}$ as follows:

(i) For each variable x_i, we construct a variable gadget V_i.

(ii) For each variable x_i, suppose that x_i appears n_i times in total amongst the clauses of $\mathcal{M}$ (so $\sum n_i = 3c$). Beginning with V_i, we attach a fork gadget to the boundary torus of V_i, then attach another fork gadget to one of the new boundary tori and so on, until we have attached $n_i - 1$ fork gadgets in total. Each time we attach a fork gadget we ensure that the boundary edge labels a, b, c match, as described in Lemma 3.2. The result is a connected triangulation with n_i distinct two-triangle boundary tori; we denote this triangulation by W_i.

(iii) For each clause $x_i \vee x_j \vee x_k$, we construct a clause gadget and attach one of the boundary tori from

W_i, one of the boundary tori from W_j, and one of the boundary tori from W_k. Again we ensure that the boundary edge labels a, b, c match, as described in Lemma 3.3.

By Observation 3.1 and Lemmas 3.2 and 3.3, the resulting object $\mathcal{T}_{\mathcal{M}}$ is a 3-manifold triangulation; moreover, it is simple to see from the construction that $\mathcal{T}_{\mathcal{M}}$ is orientable and has no remaining boundary faces. The total number of tetrahedra in $\mathcal{T}_{\mathcal{M}}$ is $2t + 21\sum_{i=1}^{t}(n_i - 1) + 4c = 67c - 19t$, and the construction is easy to perform in small polynomial time in t and c.

All that remains is to show that $\mathcal{T}_{\mathcal{M}}$ has a taut angle structure if and only if $\mathcal{M}$ is solvable:

- Suppose that $\mathcal{M}$ is solvable. By Lemma 3.1, we can assign a taut angle structure to each V_i with boundary pattern $(2,0,0)$ or $(0,2,0)$ according to whether x_i is true or false respectively. By Lemma 3.2, this extends to a taut angle structure on each W_i where *every* boundary pattern on W_i is $(2,0,0)$ or $(0,2,0)$ according to whether x_i is true or false respectively. Finally, because each clause contains exactly one true variable, Lemma 3.3 shows that these taut angle structures extend through the clause gadgets, giving a taut angle structure on the full triangulation $\mathcal{T}_{\mathcal{M}}$.

- Suppose that $\mathcal{T}_{\mathcal{M}}$ has a taut angle structure τ. By Lemma 3.1, restricting τ to each variable gadget V_i must give one of the boundary patterns $(2,0,0)$ or $(0,2,0)$; we set the corresponding variable x_i to true or false accordingly. For each i, Lemma 3.2 shows that τ must meet every boundary torus of W_i in the same pattern as for V_i; that is, $(2,0,0)$ if we set x_i to true, or $(0,2,0)$ if we set x_i to false. Finally, because we know that τ extends through the clause gadgets, Lemma 3.3 shows that each clause must have exactly one variable x_i set to true; that is, $\mathcal{M}$ is solvable.

Therefore our construction is indeed a polynomial reduction from MONOTONE 1-IN-3 SAT to TAUT ANGLE STRUCTURE, and so TAUT ANGLE STRUCTURE is **NP**-complete.

Fixed-parameter tractability Here we give full proofs of Theorems 1.2 and 1.3, both restated below. In both proofs we assume that we have access to the full skeleton of the triangulation $\mathcal{T}$ (i.e., we know which tetrahedron edges are identified and which tetrahedron vertices are identified); such information is easily computed using linear time depth-first search techniques.

THEOREM 1.2. *Let $\mathcal{T}$ be a 3-manifold triangulation with n tetrahedra, where the graph $\Gamma(\mathcal{T})$ has cutwidth $\leq k$, and for which a corresponding layout of nodes is known. Then* TAUT ANGLE STRUCTURE *can be solved for $\mathcal{T}$ in $O(nk \cdot 3^{3k/2})$ time.*

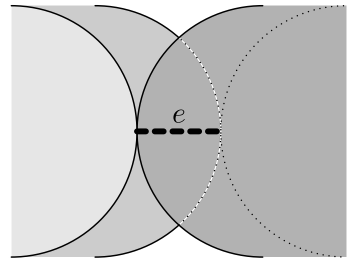

Figure 9: A pinched edge in the boundary of $\mathcal{T}_i$

Proof. Let the given layout of nodes of the face pairing graph $\Gamma(\mathcal{T})$ be $v_1, \ldots, v_n$, so that no cut $C_i = (\{v_1, \ldots, v_i\},\ \{v_{i+1}, \ldots, v_n\})$ has width more than k.

Recall that every node v_i of $\Gamma(\mathcal{T})$ corresponds to a tetrahedron Δ_{v_i} of the triangulation $\mathcal{T}$, and that every arc in the cutset for C_i is an arc of $\Gamma(\mathcal{T})$, and represents a triangle of $\mathcal{T}$.

Following a dynamic programming approach, we define sub-triangulations $\mathcal{T}_1, \ldots, \mathcal{T}_n$, where the sub-triangulation $\mathcal{T}_i$ contains only the tetrahedra $\Delta_{v_1}, \ldots, \Delta_{v_i}$. We maintain all face gluings between these tetrahedra, but if a tetrahedron Δ_{v_x} is glued to a tetrahedron Δ_{v_y} in the full triangulation $\mathcal{T}$ with $x \leq i < y$, then this will simply appear as a boundary face of Δ_{v_x} in $\mathcal{T}_i$.

A triangulation $\mathcal{T}_i$ might contain "pinched edges", which occur when multiple edges on the boundary of the sub-triangulation $\mathcal{T}_i$ correspond to the same edge of $\mathcal{T}$ (see edge e in Figure 9). We happily accept such anomalies and consider these edges identical in $\mathcal{T}_i$; as a result the number of edges on the boundary of $\mathcal{T}_i$ is *at most* (but might not be equal to) $3/2$ the number of faces.

By construction, every boundary face of $\mathcal{T}_i$ corresponds to an arc in the cutset C_i, and so it follows that $\mathcal{T}_i$ has $\leq k$ boundary faces and thus $\leq 3k/2$ boundary edges. Since any taut angle structure on $\mathcal{T}_i$ must mark each boundary edge 0, 1 or 2 times, there can be at most $3^{3k/2}$ different patterns of markings on the boundary of $\mathcal{T}_i$ that correspond to taut angle structures on $\mathcal{T}_i$.

Following our dynamic programming strategy, we work through the triangulations in the order $\mathcal{T}_1, \ldots, \mathcal{T}_n$, and for each i we compute precisely which boundary marking patterns on $\mathcal{T}_i$ correspond to taut angle structures on $\mathcal{T}_i$:

- For $\mathcal{T}_1$ we simply try all three choices of markings on the tetrahedron Δ_{v_1}, identify which of these

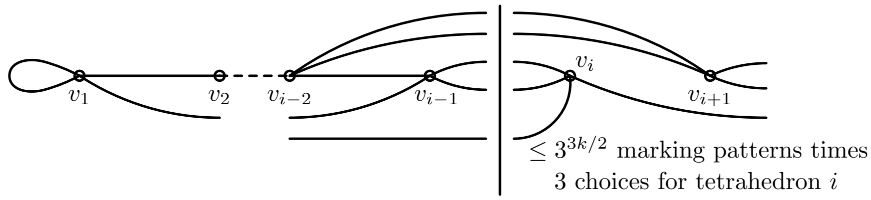

Figure 10: Finding taut angle structures under bounded cutwidth

form a taut angle structure on $\mathcal{T}_1$, and store the resulting marking patterns on the boundary.

- For $\mathcal{T}_i$, we consider each of the $\leq 3^{3k/2}$ boundary marking patterns on $\mathcal{T}_{i-1}$ that yields a taut angle structure on $\mathcal{T}_{i-1}$, and attempt to combine these with each of the three choices of markings on the new tetrahedron Δ_{v_i}. We discard any combination that marks an edge more than twice, or that marks an internal edge less than twice; the remaining combinations yield taut angle structures on $\mathcal{T}_i$, and we store them in our solution set for $\mathcal{T}_i$. See Figure 10 for an illustration of this procedure.

Since the final triangulation $\mathcal{T}_n = \mathcal{T}$ has no boundary faces (and hence no boundary edges), the full triangulation $\mathcal{T}$ has a taut angle structure if and only if the solution set for $\mathcal{T}_n$ contains the empty marking pattern (as opposed to no marking patterns at all).

Taking into account that in each step the number of boundary patterns to consider is at most $3^{3k/2}$, that there are only three choices of markings for the tetrahedron Δ_{v_i}, and that we can update and test each boundary marking pattern in $O(k)$ time, it follows that each step can be performed in $O(k \cdot 3^{3k/2})$ time overall. The total running time for all n steps of the algorithm is therefore $O(nk \cdot 3^{3k/2})$.

THEOREM 1.3. *Let $\mathcal{T}$ be a 3-manifold triangulation with n tetrahedra, where the graph $\Gamma(\mathcal{T})$ has treewidth $\leq k$, and for which a corresponding tree decomposition with $O(n)$ tree nodes is known. Then* TAUT ANGLE STRUCTURE *can be solved for $\mathcal{T}$ in $O(nk \cdot 3^{7k})$ time.*

Proof. Recall that each node ν of the tree corresponds to a bag of nodes in $\Gamma(\mathcal{T})$; that is, a bag of tetrahedra. We arbitrarily choose a root for the tree, so that the tree becomes a hierarchy of subtrees as illustrated in Figure 8.

As in the TAUT ANGLE STRUCTURE problem statement, we assume that $\mathcal{T}$ has no boundary faces. For each node ν of the tree, we define the sub-triangulation $\mathcal{T}_\nu$ by considering only those tetrahedra that appear only in bags within the subtree rooted at ν; that is, we exclude any tetrahedron that appears in any bag outside this subtree. As in the previous proof, we maintain all face gluings between these tetrahedra; however, if a tetrahedron $\Delta \in \mathcal{T}_\nu$ is glued to a tetrahedron $\Delta' \notin \mathcal{T}_\nu$ in the full triangulation $\mathcal{T}$ then this will simply appear as a boundary face of Δ in $\mathcal{T}_\nu$. Once again a triangulation $\mathcal{T}_\nu$ might contain "pinched edges"; again we happily accept such anomalies and consider these edges identical in $\mathcal{T}_\nu$, which means that the number of edges on the boundary of $\mathcal{T}_\nu$ is at most (but not necessarily equal to) 3/2 the number of faces.

We now make a series of observations:

(i) *Each sub-triangulation $\mathcal{T}_\nu$ has at most $4(k+1)$ boundary faces and at most $6(k+1)$ boundary edges.*

If ν is the root node then this is trivial (since $\mathcal{T}_\nu = \mathcal{T}$ contains no boundary faces at all). Otherwise, let η be the parent node of ν in the tree. Any boundary face of $\mathcal{T}_\nu$ must correspond to some gluing between tetrahedra $\Delta \in \mathcal{T}_\nu$ and $\Delta' \notin \mathcal{T}_\nu$, which in turn corresponds to some arc α in the face pairing graph. Because Δ only appears in bags within the subtree rooted at ν, and because one of these bags must contain both endpoints of the arc α, it follows that Δ' appears in some bag within the subtree at ν also.

By the definition of tree decomposition, since Δ' appears in some bag within the subtree at ν and also in some bag outside this subtree, it must also appear in the bag at the parent node η. There are $\leq k+1$ tetrahedra in the bag at η and so $\leq 4(k+1)$ possibilities for the face of Δ' that is joined to Δ. Therefore there are $\leq 4(k+1)$ such boundary faces of $\mathcal{T}_\nu$.

Finally, since the number of edges on the boundary surface is $\leq$ 3/2 the number of faces, $\mathcal{T}_\nu$ has $\leq 6(k+1)$ boundary edges.

(ii) *Suppose the tree node ν has child nodes $\mu_1, \ldots, \mu_d$, as illustrated in Figure 8. Then no two triangulations $\mathcal{T}_{\mu_i}, \mathcal{T}_{\mu_j}$ have any tetrahedra in common.*

This is true by definition of $\mathcal{T}_{\mu_i}$, since no tetrahedron in $\mathcal{T}_{\mu_i}$ can appear in the subtree rooted at μ_j (which lies outside the subtree rooted at μ_i).

(iii) *Suppose the tree node ν has child nodes $\mu_1, \ldots, \mu_d$, as illustrated in Figure 8. Then the triangulations $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$ have $\leq 4(k+1)$ boundary faces in total. Moreover, they have $\leq 6(k+1)$ boundary edges in total, even if we count each edge repeatedly for each $\mathcal{T}_{\mu_i}$ that contains it.*

As in the argument for (i) above, each boundary face of each triangulation $\mathcal{T}_{\mu_i}$ corresponds to a face of some tetrahedron Δ' in the bag at the parent node ν. Moreover, from (ii) the triangulations $\mathcal{T}_{\mu_i}$ contain distinct tetrahedra, and so each such boundary face can only appear in *one* of the $\mathcal{T}_{\mu_i}$. Therefore the triangulations $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$ have $\leq 4(k+1)$ boundary faces between them, and these boundary faces are all distinct.

From above, each $\mathcal{T}_{\mu_j}$ has at most $3/2$ as many boundary edges as it has boundary faces. It therefore follows that $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$ have $\leq 6(k+1)$ boundary edges between them, even if we count repeated edges multiple times.

Our algorithm for solving TAUT ANGLE STRUCTURE is based on dynamic programming over the tree, and operates as follows. For each triangulation $\mathcal{T}_\nu$, we compute the number of marking patterns on the boundary of $\mathcal{T}_\nu$ that correspond to taut angle structures on $\mathcal{T}_\nu$. Since there are $\leq 6(k+1)$ boundary edges on $\mathcal{T}_\nu$, there are $\leq 3^{6(k+1)}$ such possible marking patterns.

We work our way from the leaves of the tree up to the root, computing these boundary patterns on each triangulation $\mathcal{T}_\nu$ as we go:

- If ν is a leaf node, we simply try all 3^{k+1} possible markings on the $\leq k+1$ tetrahedra in the bag at ν, which takes $O(3^{k+1})$ time. For each combination that yields a taut angle structure, we record the corresponding marking pattern on the boundary.

- If ν is not a leaf node then let $\mu_1, \ldots, \mu_d$ be its immediate children in the tree, as illustrated in Figure 8. Let $\mathcal{T}'$ be the (possibly disconnected) triangulation obtained by combining the tetrahedra from $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$.

  For each combination of boundary marking patterns on $\mathcal{T}_{\mu_1}, \ldots, \mathcal{T}_{\mu_d}$, we combine these into a single boundary marking pattern on $\mathcal{T}'$ (if any boundary edges are repeated then we sum the corresponding markings). By observation (iii) above, we can form each such combination in $O(k)$ time. If each $\mathcal{T}_{\mu_i}$ has b_i boundary edges, the total number of combinations that we form is $\leq 3^{b_1} \ldots 3^{b_d} = 3^{b_1+\ldots+b_d} \leq 3^{6(k+1)}$.

  We discard any combination that marks a boundary edge more than twice in total, or that marks an internal edge less than twice. Any combination b' that survives must correspond to a taut angle structure on $\mathcal{T}'$ (we simply combine the taut angle structures on each $\mathcal{T}_{\mu_i}$, which we can do because the $\mathcal{T}_{\mu_i}$ have no tetrahedra in common). We now combine b' with all possible markings on the *new* tetrahedra in $\mathcal{T}_\nu$ that are not already present in $\mathcal{T}'$; there are $\leq k+1$ new tetrahedra because they must all belong to the bag at ν. Again we discard combinations that mark an edge more than twice, or that mark an internal edge less than twice; any combination that remains must arise from a taut angle structure on $\mathcal{T}_\nu$, whereupon we add its boundary marking pattern to our solution set.

  In summary, we obtain $\leq 3^{6(k+1)}$ marking patterns on $\mathcal{T}'$ which we combine with 3^{k+1} choices of markings for the new tetrahedra in $\mathcal{T}_\nu$, giving a grand total of $\leq 3^{7(k+1)}$ combinations overall. Adding in a factor of k to merge marking patterns and test for bad edge markings, the overall running time of this step is $O(k \cdot 3^{7(k+1)}) = O(k \cdot 3^{7k})$.

Let ρ be the root node. Since $\mathcal{T}_\rho = \mathcal{T}$ has no boundary faces (and hence no boundary edges), the full triangulation $\mathcal{T}$ has a taut angle structure if and only if the solution set for $\mathcal{T}_\rho$ contains the empty marking pattern (as opposed to no marking patterns at all). Since there are $O(n)$ nodes in our tree decomposition, the total running time for the algorithm is $O(nk \cdot 3^{7k})$.