Tractable Classes of Binary CSPs Defined by Excluded Topological Minors[*]

David A. Cohen
Dept. of Computer Science,
Royal Holloway,
Univ. of London, UK
dave@cs.rhul.ac.uk

Martin C. Cooper
IRIT, Univ. of Toulouse,
31062 Toulouse, France
cooper@irit.fr

Peter G. Jeavons & Stanislav Živný
Dept. of Computer Science,
Univ. of Oxford, UK
peter.jeavons@cs.ox.ac.uk
standa.zivny@cs.ox.ac.uk

Abstract

The binary Constraint Satisfaction Problem (CSP) is to decide whether there exists an assignment to a set of variables which satisfies specified constraints between pairs of variables. A CSP instance can be presented as a labelled graph (called the microstructure) encoding both the forms of the constraints and where they are imposed. We consider subproblems defined by restricting the allowed form of the microstructure. One form of restriction that has previously been considered is to forbid certain specified substructures (patterns). This captures some tractable classes of the CSP, but does not capture the well-known property of acyclicity. In this paper we introduce the notion of a topological minor of a binary CSP instance. By forbidding certain patterns as topological minors we obtain a compact mechanism for expressing several novel tractable classes, including new generalisations of the class of acyclic instances.

1 Introduction

The binary Constraint Satisfaction Problem (CSP) is to decide whether it is possible to find an assignment to a set of variables which satisfies a specified set of constraints between certain pairs of variables. This paradigm has been applied in diverse application areas such as Artificial Intelligence, Bioinformatics and Operations Research [Rossi *et al.*, 2006].

As the CSP is known to be NP-complete, much theoretical work has been devoted to the identification of tractable subproblems. Important tractable cases have been identified by restricting the set of pairs where the constraints are imposed (sometimes called the constraint *structure*) [Grohe, 2007; Marx, 2013]. Other tractable cases have been identified by restricting the forms of constraints (sometimes called the constraint *language*) [Cohen and Jeavons, 2003; Barto and Kozik, 2014]. Work on both of these areas is now very far advanced: a full complexity dichotomy for any structural or language restriction now requires the classification of just one remaining family of languages [Barto and Kozik, 2014].

A CSP instance can be represented as a labelled graph (known as the *microstructure*), encoding *both* the constraint structure and the constraint language. By restricting the allowed form of the microstructure it is possible to impose simultaneous restrictions on both the structure and the language of an instance, and hence obtain a more refined collection of subproblems, allowing a more detailed complexity analysis. Currently, very little is known about the complexity of the subproblems obtained by restricting both the structure and the language of the set of instances at the same time.

Following a well-established line of research in graph theory, restrictions on the microstructure of an instance can be defined by considering local "obstructions" or "forbidden patterns". In this paper we introduce the notion of a topological minor of a binary CSP instance. By forbidding certain patterns as topological minors we provide a compact mechanism for expressing several novel tractable classes, including new generalisations of the class of instances with acyclic structure.

2 Definitions

A binary CSP instance requires the assignment of values from some specified finite domain to a finite set of variables $\{X_1, \ldots, X_n\}$. Each pair of variables, X_i and X_j is constrained by a constraint relation R_{ij}. A constraint is *non-trivial* if it is not the Cartesian product of the domains of the two variables. A *solution* to a binary CSP instance is an assignment s of values to variables, such that, for each constraint R_{ij}, $\langle s(X_i), s(X_j) \rangle \in R_{ij}$

The *constraint graph* of an instance I is $G_I = \langle V_I, E_I \rangle$, where $V_I = \{X_1, \ldots, X_n\}$ is the set of variables of I and E_I is the set of pairs $\{X_i, X_j\}$ for which R_{ij} is non-trivial.

One possible presentation of a binary CSP instance is as a labelled graph whose vertices are the set of possible variable-value assignments. This (labelled) graph is known as the (coloured) microstructure [Jégou, 1993; Angelsmark and Thapper, 2005; Cohen *et al.*, 2006; Salamon and Jeavons, 2008; Cohen *et al.*, 2012; El Mouelhi *et al.*, 2013]. An n-variable binary CSP instance I in this microstructure presentation is an n-partite graph $\langle A_1, \ldots, A_n, E^+ \rangle$, where the ith part A_i corresponds to the set of possible assignments $\langle X_i, a \rangle$ to variable X_i and there is an edge in E^+ between $\langle X_i, a \rangle \in A_i$ and $\langle X_j, b \rangle \in A_j$ if and only if $(a, b) \in R_{ij}$. We refer to individual variable-value assignments, such as $\langle X_i, a \rangle$ as *points*. If X is some variable of the instance we use the notation A_X

to represent the set of possible assignments to X. Thus A_i and A_{X_i} are synonyms.

An instance I can also be presented as a *negative microstructure* which is the n-partite labelled graph $\langle A_1,\ldots,A_n,E^-\rangle$, where there is an edge in E^- between points $p \in A_i$ and $q \in A_j$ (for $i \neq j$) if and only if there is no edge between p and q in E^+.

Patterns and Sub-Patterns: We now generalise the notion of microstructure and negative microstructure to obtain *patterns*: a pattern is a labelled n-partite graph which has a set of positive edges, E^+, and a set of negative edges, E^-.

A binary CSP instance can be seen as a special kind of pattern where the parts correspond to the variables of the instance and there is exactly one positive or negative edge between each pair of possible assignments to each pair of distinct variables. Positive edges connect assignments that are allowed by the constraint on the corresponding pair of variables, and negative edges connect assignments that are disallowed by this constraint.

A general pattern $P = \langle A_1,\ldots,A_n,E^+,E^-\rangle$ can in some ways be viewed as a partially specified instance: there may be pairs of points $p \in A_i, q \in A_j$ (with $i \neq j$) such that there is neither a positive edge nor a negative edge between p and q. (However, there can also be *both* a positive and a negative edge between p and q, see below.) A point is said to be *isolated* if it does not belong to any edges in E^+ or E^-.

A pattern $P' = \langle A_1',\ldots,A_n',E'^+,E'^-\rangle$ is a *homomorphic image* of a pattern $P = \langle A_1,\ldots,A_n,E^+,E^-\rangle$ if there exists a surjective mapping $f : \bigcup_{i=1}^n A_i \rightarrow \bigcup_{i=1}^n A_i'$ such that

- $\forall p,q \in \bigcup_{k=1}^n A_k$, p and q belong to the same part A_i if and only if $f(p)$ and $f(q)$ belong to the same part A_j',

- $\forall i,j \in \{1,\ldots,n\}$ with $i \neq j$, $\forall p \in A_i$, $\forall q \in A_j$: $\{p,q\} \in E^+ \Rightarrow \{f(p),f(q)\} \in E'^+$ and $\{p,q\} \in E^- \Rightarrow \{f(p),f(q)\} \in E'^-$.

Note that forming a homomorphic image of a pattern allows the parts to be renamed, and compatible points within the same part to be merged.

We will say that a pattern P occurs as a *sub-pattern* of a pattern Q if Q can be transformed into a homomorphic image of P by a sequence of the following *substructure operations*:

- removal of (positive or negative) edges,

- removal of isolated points, and

- removal of empty parts.

This notion of occurring as a sub-pattern has led to the discovery of several new tractable classes [Cooper *et al.*, 2010; Cohen *et al.*, 2012; Cooper and Escamocher, 2012]: in each case the class is defined by a simple forbidden pattern, i.e., as the set of binary CSP instances in which the given pattern does *not* occur as a sub-pattern.

Topological Minors We now present a new operation on patterns which allows us to define the notion of a topological minor of a pattern (and hence of a binary CSP instance). This new operation is analogous to the operation of eliminating subdivisions (vertices of degree 2) that is used to define a topological minor of a graph [Diestel, 2010]. However, since patterns contain two kinds of edges, the definition is slightly more complicated.

This new operation, path reduction, will sometimes lead to the introduction of edges in $E^+ \cup E^-$. Hence, we relax the restriction that E^+ and E^- be disjoint.

In a pattern $P = \langle A_1,\ldots,A_n,E^+,E^-\rangle$, we say that two parts A_i,A_j are *directly connected* if there is at least one (positive or negative) edge $\{p,q\} \in E^+ \cup E^-$ with $p \in A_i$ and $q \in A_j$.

If A_i,A_j are not directly connected and A_k is directly connected only to A_i and A_j, then the following operation can be performed, which is known as *path reduction*:

1. $\forall p \in A_i, \forall q \in A_j$: if $\exists r \in A_k$ such that $\{p,r\},\{r,q\} \in E^+$, then introduce a new positive edge $\{p,q\}$,

2. $\forall p \in A_i, \forall q \in A_j$: if $\exists r,s \in A_k$ such that $\{p,r\},\{s,q\} \in E^-$, then introduce a new negative edge $\{p,q\}$,

3. remove the part A_k and all edges containing points in A_k.

Positive and negative edges are treated differently in this definition; this is because for $p \in A_i$ and $q \in A_j$ to be part of a solution to the sub-instance on variables X_i,X_j,X_k, the points p and q must both be compatible with some common point $r \in A_k$, whereas p and q may be incompatible if they are each incompatible with some point in A_k, not necessarily the same point. This asymmetry in the definition seems to lead to more useful results.

A pattern P occurs as a *topological minor* of a pattern Q if Q can be transformed into a homomorphic image of P by a sequence of substructure operations and path reductions.

3 Some basic tractability results

We use the notation $\mathrm{CSP}_{\overline{\mathrm{SP}}}(P)$ to represent the set of binary CSP instances in which the pattern P does not occur as a sub-pattern. We use the notation $\mathrm{CSP}_{\overline{\mathrm{TM}}}(P)$ to represent the set of binary CSP instances in which the pattern P does not occur as a topological minor. For each pattern P there are therefore two distinct notions of tractability, defined as follows.

Definition 3.1 *A pattern P is* sub-pattern tractable *if there is a polynomial-time algorithm to solve $CSP_{\overline{SP}}(P)$. A pattern P is* topological-minor tractable *if there is a polynomial-time algorithm to solve $CSP_{\overline{TM}}(P)$.*

The following lemma follows directly from the definitions.

Lemma 3.2 *If a pattern P is sub-pattern tractable then P is topological-minor tractable.*

The two patterns shown in Figure 1(a) and 1(b) are known to be sub-pattern tractable [Cooper and Živný, 2012; Escamocher, 2013]. Hence, they are also topological-minor tractable, by Lemma 3.2.

Lemma 3.3 *If pattern P occurs as a topological minor in pattern Q, and Q is topological-minor tractable, then P is topological-minor tractable.*

Proof: Suppose that P occurs in pattern Q as a topological minor. From the transitivity of the occurrence (as a topological minor) relation, we can deduce that $\mathrm{CSP}_{\overline{\mathrm{TM}}}(P) \subseteq \mathrm{CSP}_{\mathrm{TM}}(Q)$. It follows, directly, that if Q is topological-minor tractable, then P is also topological-minor tractable. ∎
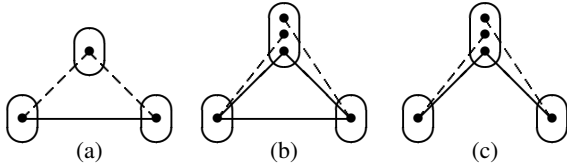
Figure 1: Patterns which define tractable classes when excluded as sub-patterns. Parts are shown as ovals, points as filled circles, positive edges as solid lines and negative edges as dashed lines.
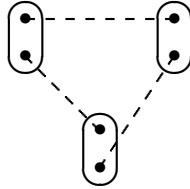

Figure 2: A pattern which defines the class of acyclic binary CSP instances when forbidden as a topological minor.

As a simple example, we can deduce from Lemma 3.3 that Figure 1(c) is topological-minor tractable, since Figure 1(c) occurs in Figure 1(b) as a topological minor and we have already seen that Figure 1(b) is topological-minor tractable.

4 Acyclicity and bounded treewidth

One important tractable class of binary CSP instances is the class of instances whose constraint graph is acyclic [Freuder, 1982]. However, this class cannot be defined by a finite set of forbidden sub-patterns, as the following result shows.

Lemma 4.1 *The class of acyclic binary CSP instances cannot be characterised by a finite set of forbidden sub-patterns.*

Proof: Let T be a finite set of patterns which do not occur in any acyclic CSP instance, and let k be the maximum number of parts in any pattern from T. Consider an instance I with variables $\{X_1, \ldots, X_{k+1}\}$ and non-trivial constraints between X_i and X_{i+1} for $i = 1, \ldots, k$ and between X_{k+1} and X_1. No pattern from T occurs as a sub-pattern in I, since any sub-pattern of I with at most k variables is acyclic. Since I is not acyclic we are done. ∎

On the other hand, it is straightforward to characterise the class of acyclic instances by forbidding a single pattern as a topological minor. For any graph G on vertices $\{v_1, \ldots, v_n\}$, we can create a corresponding pattern P_G with a part A_i for each vertex v_i of G and, for each edge $e = \{v_i, v_j\}$ of G, a negative edge in P_G between points $p_e \in A_i$ and $q_e \in A_j$, these points being distinct for each edge e. For example, when G is a 3-cycle, the corresponding pattern P_G is shown in Figure 2. Every edge in the constraint graph corresponds to a non-trivial constraint relation, that forbids at least one pair of points. Hence, the pattern P_G is a topological minor of a binary CSP instance I if and only if G is a topological minor of

the constraint graph of I. Hence forbidding the pattern shown in Figure 2 as a topological minor exactly defines the class of binary CSP instances whose constraint graph is acyclic.

This idea can easily be extended to any of the tractable classes of binary CSP instances defined by imposing any fixed bound on the treewidth of the constraint graph [Freuder, 1985]. The graph minor theorem [Robertson and Seymour, 2004] implies that for any fixed $k \geq 1$ there is a finite set O_k of graphs such that the class of graphs of treewidth at most k is precisely the class of graphs excluding all graphs from the set O_k as topological minors [Diestel, 2010]. Consequently, for any $k \geq 1$ the class of binary CSP instances of treewidth at most k can be defined by a finite set of patterns that are forbidden as topological minors.

5 Tractability via Tutte decomposition

In this section we will describe a generic scheme for proving tractability of classes of binary CSP instances defined by a forbidden topological minor. We will use this scheme in the proofs of Proposition 6.4 and Theorem 7.1.

If U is a set of vertices of a graph G, we write $G[U]$ for the induced graph on U. We say that $\langle U_1, U_2 \rangle$ is a *separation* of G if $G = G[U_1] \cup G[U_2]$. The *separator* of the separation $\langle U_1, U_2 \rangle$ is $U_1 \cap U_2$ and its *order* is $|U_1 \cap U_2|$. The *torso* of U_1 in the separation $\langle U_1, U_2 \rangle$ is obtained from the induced graph $G[U_1]$ by adding every edge between the vertices of the separator. A *Tutte decomposition* of a graph G is a tree T where each node is labelled with a subset of the vertices of G, each arc induces a separation of G of order at most two, and the torso of each node is three-connected, or a cycle, or has at most 2 vertices. Every finite graph has a Tutte decomposition of this kind [Tutte, 1966].

To demonstrate topological-minor tractability for a pattern P we proceed as follows. Let I be an instance in which P does not occur as a topological minor and let G_I be its constraint graph. We denote by n the number of variables in I and by d the maximum domain size of any variable in I.

Build a Tutte tree decomposition, T, of G_I, and consider any leaf C of T inducing the separation $\langle C, R \rangle$ of G_I, with separator S. Let I_C be the subinstance of I on the variables corresponding to the vertices of C, and let C_t be the torso of C. Suppose that the following two assumptions hold:

(**A1**) I_C can be solved and its solutions projected onto the separator S in polynomial time; the resulting reduced instance on R will be denoted by I_{red}.

(**A2**) P does not occur in I_{red} as a topological minor.

Then it follows that a recursive algorithm, which at each step chooses some leaf C of T, and then solves the associated subproblem I_C to obtain the reduced instance I_{red}, will solve the original instance using a polynomial (in n and d) number of calls to the polynomial-time algorithm from (**A1**).

In the proofs below we will omit the simple cases where the separator S has order less than two, or C contains at most 3 vertices, or the torso of C is a cycle (and hence has treewidth 2 and is solvable in polynomial time). Hence we will assume that C is three-connected and of size greater than three.

Finally, note that if $\{u, v\}$ is an order 2 separator for some separation $\langle C, R \rangle$ of G_I then we have the following:
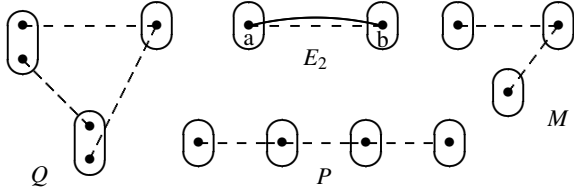
Figure 3: Patterns which define tractable classes when excluded as topological minors.

- Any path from a vertex in C to a vertex in R must pass through u or v;
- There must exist some path from u to v in $G_I[R]$, which we will denote $path_R(u,v)$.

6   Some topological-minor tractable patterns

Consider the patterns shown in Figure 3. By the classification theorem for patterns with only negative edges given in [Cohen *et al.*, 2012], the class of instances obtained by forbidding the pattern Q as a sub-pattern, $CSP_{\overline{SP}}(Q)$, is NP-complete. Similarly for pattern P. The pattern E_2 cannot occur as a sub-pattern in a CSP instance, and hence $CSP_{\overline{SP}}(E_2)$ is the class of all instances and so is trivially NP-complete. However, we will show in this section that these three patterns define tractable classes when forbidden as topological minors.

Proposition 6.1 *The patterns Q and M, shown in Figure 3, are topological-minor tractable.*

Proof: It is known that $CSP_{\overline{SP}}(M)$ can be solved in polynomial time, since M is a sub-pattern of the pattern shown in Figure 1(a) [Cooper and Živný, 2012]. Hence we only need to prove the result for Q.

Consider an instance I in which the pattern Q does not occur as a topological minor, and let G_I be the constraint graph of I. Suppose the pattern M occurs as a sub-pattern on the triple of variables (x,y,z) in I, with y being the variable at which the two negative edges meet.

Since Q does not occur as a topological minor in I, it follows that there is no path from x to z in G_I that does not pass through y. We can therefore find a tree-decomposition of G_I into components in which M does not occur as a sub-pattern and joined at such variables y which we can call articulation variables. Since $CSP_{\overline{SP}}(M)$ is tractable, any subinstance corresponding to a leaf component can be solved in polynomial time for each possible assignment to the unique articulation variable which joins it to its parent component in the tree-decomposition. This leads to the elimination of the leaf component and possible elimination of some values in the domain of this articulation variable. The original instance I can be solved in polynomial time by repeatedly solving subinstances corresponding to leaf components in this way.   ∎

Proposition 6.2 *The pattern E_2, shown in Figure 3, is topological-minor tractable.*

Proof: Since the pattern E_2 cannot occur as a sub-pattern in any instance, it can only occur as a topological minor if there is at least one path reduction operation. Hence excluding the pattern E_2 as a topological minor is equivalent to excluding the pattern in Figure 1(c) as a topological minor. In Section 2 we observed that Figure 1(c) is topological-minor tractable, so it follows that E_2 is topological-minor tractable.   ∎

Corollary 6.3 *All 2-part patterns are topological-minor tractable.*

Proof: Let P_2 be an arbitrary 2-part pattern. The function which maps all of A_1 to a and all of A_2 to b is necessarily a homomorphism from P_2 to the pattern E_2 in Figure 3, since $\{a,b\}$ is both a negative and a positive edge in E_2. By Proposition 6.2 and Lemma 3.3, it follows that P_2 is tractable.   ∎

Proposition 6.4 *The pattern P, shown in Figure 3, is topological-minor tractable.*

Proof: The proof uses the generic scheme described in Section 5, so we only need to establish the two assumptions.

(**A1**)   Let M be the pattern consisting of two intersecting negative edges, shown in Figure 3. Suppose that M occurs in I_C as a sub-pattern on two *disjoint* triples of variables (x,y,z) and (x',y',z'). As explained in Section 5, we assume that C_t is 3-connected. It follows by Menger's theorem [Dirac, 1966] that there are three disjoint paths from x to x' in C_t. There must be one of these paths, π, which does not pass through y or y'. There must be a subpath σ of π which begins at x or z and ends at x' or z' (or vice versa, i.e., which begins at x' or z' and ends at x or z) and which does not pass through any other variables in $\{x,y,z,x',y',z'\}$. Without loss of generality, suppose that σ joins x to x'. But then P occurs as a topological minor on the extended path σ^+ given by $z \to y \to x, \sigma, x' \to y' \to z'$.

But this implies that P occurs as a topological minor in I, since if σ^+ passes by the edge $\{u,v\}$ in the torso C_t, this edge can be replaced by $path_R(u,v)$ which is a path from u to v in R, whose existence was noted in Section 5. Since this contradicts our initial assumption, we can deduce that M does not occur in I_C as a sub-pattern on two *disjoint* triples.

We can therefore deduce that all pairs of triples of variables (x,y,z), (x',y',z') on which M occurs as a sub-pattern in I_C intersect, i.e. $\{x,y,z\} \cap \{x',y',z'\} \neq \emptyset$. Now, consider an arbitrary triple of variables (x,y,z) on which M occurs as a sub-pattern. It follows that the instance which results after any instantiation of the three variables x,y,z contains no occurrence of M, since for each triple of variables (x',y',z') on which M occurs in I_C, at least one of its variables has been eliminated by instantiation.

Thus, after instantiation of at most three variables, I_C does not contain M as a sub-pattern. This also holds for any version of I_C obtained by instantiating the variables u,v. Since, as noted earlier, $CSP_{\overline{SP}}(M)$ is tractable, we can therefore determine in polynomial time which instantiations of u,v can be extended to a solution of I_C. We add a negative edge in I between points representing assignments p,q to variables u,v if

p, q cannot be extended to a solution to I_C, and then we delete all variables in C from I. Overall, we have proved that I_{red} can be constructed in polynomial time.

(**A2**): Suppose, for a contradiction, that we introduce the pattern P as a topological minor when reducing I to I_{red}. This occurrence of P must use the edge $\{u, v\}$. During the reduction from I to I_{red}, we can introduce negative (but not positive) edges between points $p \in A_u$ and $q \in A_v$. Suppose that a negative edge $\{p, q\}$ is introduced by the reduction from I to I_{red}. But this can only be the case if there was a path $\pi = (u, w_1, \ldots, w_t, v)$ in the graph C with a sequence of negative edges in I_C linking p to q via variables $w_1, \ldots, w_k$ in I_C: $\{p, s_1\}$, $\{t_1, s_2\}$, $\{t_2, s_3\}$ ... $\{t_k, q\}$ with $\forall i \in \{1, \ldots, k\}$, $s_i, t_i \in A_{w_i}$. This means that we can replace the edge $\{u, v\}$ in the occurrence of P in I_{red} by a path (obtained from I_π for some path π in C by deleting unnecessary points and edges) so that P occurs as a topological minor in the original instance I. This contradiction shows that we cannot introduce P as a topological minor by reducing I to I_{red}.

Note that the number of instances of $\text{CSP}_{\overline{\text{SP}}}(M)$ that need to be solved is $O(nd^5)$. ∎

7 Topological-minor tractable patterns extending acyclicity

In this Section we show how the well-known tractable class of acyclic instances can be generalised to obtain larger tractable classes defined by forbidding certain topological minors.

Theorem 7.1 *Let P_2 be any sub-pattern tractable pattern on three variables $\{x, y, z\}$ where there is at most one negative edge between A_x and A_y, and between A_y and A_z, and no edges between A_x and A_z.*

Let P be a pattern on variables $\{x, y, z, w\}$ obtained by extending P_2. The pattern P has six new points $p_1, p_2 \in A_x$, $q_1, q_2 \in A_w$, and $r_1, r_2 \in A_z$, together with three new negative edges $\{p_1, r_1\}$, $\{p_2, q_1\}$, $\{q_2, r_2\}$ (see Figure 4).

Any such P is a topological-minor tractable pattern.

Proof: The proof uses the generic scheme described in Section 5, so we only need to establish the two assumptions.

(**A1**) Suppose first that P_2 occurs in I_C as a sub-pattern on the path of variables x, y, z. As explained in Section 5, we assume that C_t is three connected. Then, by Menger's theorem there are three disjoint paths π_1, π_2, π_3 from x to z in C_t. There must be two of these paths, say π_1 and π_2, which do not pass through y. But this implies that P occurs as a topological minor in I, since if either π_1 or π_2 passes by the edge $\{u, v\}$ in the torso C_t, this edge can be replaced by $\text{path}_R(u, v)$ which is a path from u to v in R, whose existence was shown in Section 5. Since this contradicts our initial assumption, we can assume that I_C does *not* contain P_2 as a sub-pattern. This also holds for any version of I_C obtained by instantiating the variables u, v. Therefore we can determine in polynomial time which instantiations of u, v can be extended to a solution of I_C. We add a negative edge in I between points representing assignments p, q to variables u, v if p, q cannot be extended to a solution to I_C, and then we delete all variables in C from
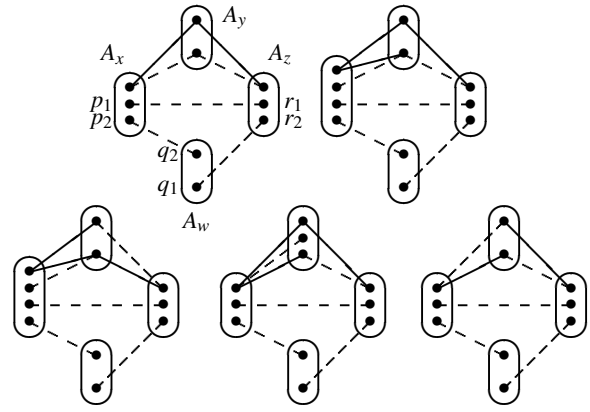

Figure 4: Topological-minor tractable patterns derived from sub-pattern tractable patterns.

I. Overall, we have proved that I_{red} can be constructed in polynomial time.

(**A2**) Suppose, for a contradiction, that we introduce the pattern P as a topological minor when reducing I to I_{red}. This occurrence of P must use the edge $\{u, v\}$. During the reduction from I to I_{red}, we can introduce negative (but not positive) edges between points $p \in A_u$ and $q \in A_v$. Observe that, by definition, P contains at most one negative edge between any pair of variables. Suppose that a negative edge $\{p, q\}$ is introduced by the reduction from I to I_{red}. But this can only be the case if there was a path $\pi = (u, w_1, \ldots, w_t, v)$ in the graph C with a sequence of negative edges in I_C linking p to q via variables $w_1, \ldots, w_k$ in I_C: $\{p, s_1\}$, $\{t_1, s_2\}$, $\{t_2, s_3\}$ ... $\{t_k, q\}$ with $\forall i \in \{1, \ldots, k\}$, $s_i, t_i \in A_{w_i}$. Furthermore, in I_{red}, if there is a positive edge from $p' \in A_u$ to $q' \in A_v$ then there is necessarily a solution to I_C including the assignments p' and q' (and hence a solution on the subinstance I_π of I_C on the path $\pi = (u, w_1, \ldots, w_t, v)$ in C. This means that we can replace the edge $\{u, v\}$ in the occurrence of P in I_{red} by a path (obtained from I_π for some path π in C by deleting unnecessary points and edges) so that P occurs in the original instance I. This contradiction shows that we cannot introduce P as a topological minor by reducing I to I_{red}.

Note that the number of instances of $\text{CSP}_{\overline{\text{SP}}}(P_2)$ that need to be solved is $O(nd^2)$. ∎

Theorem 7.1, tells us that 2-constraint patterns on three variables that are known to be sub-pattern-tractable [Cooper and Escamocher, 2012] can all be used as building blocks of topological-minor tractable patterns by adding two paths of negative edges. The resulting topological-minor tractable patterns are shown in Figure 4. For each of these patterns P, the pattern shown in Figure 2 is a sub-pattern and hence a topological minor of P. Thus, by the transitivity of occurrence as a topological minor, each tractable class $\text{CSP}_{\overline{\text{TM}}}(P)$ necessarily contains all acyclic binary CSP instances.

Since detecting whether a fixed pattern occurs as a sub-pattern can be done in polynomial time [Cohen *et al.*, 2012], and constructing a Tutte decomposition is also polynomial-time, these tractable classes can be identified in polynomial-
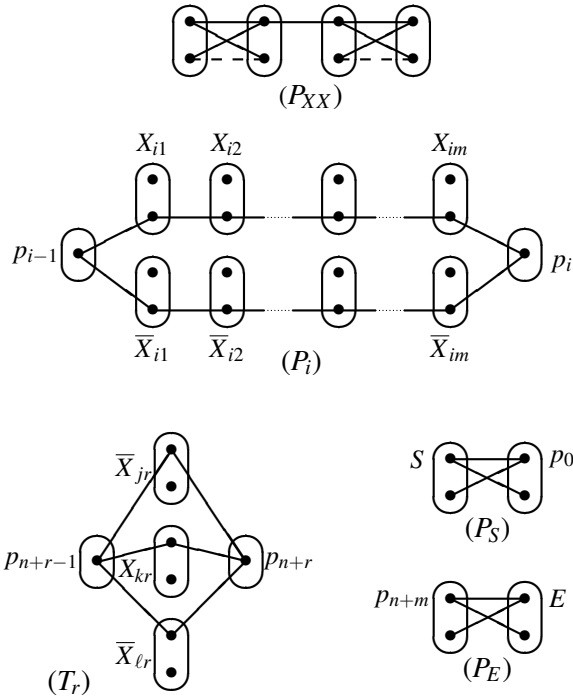
Figure 5: The minor pattern P_{XX} and the building blocks for the CSP instances in which detecting it is NP-hard.
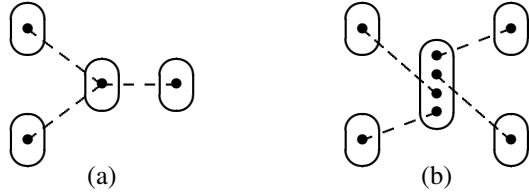

Figure 6: Patterns which define NP-complete classes when excluded as topological minors.

time. Similar remarks apply to the tractable classes identified in Proposition 6.4 and Proposition 6.1.

8 Detection of topological minors

For every fixed undirected graph H, there is an $O(|V(G)|^3)$ time algorithm that tests, given a graph G, if H is a topological minor of G [Grohe *et al.*, 2011], so the detection of topological minors in graphs is fixed parameter tractable. However, for topological minor patterns in CSP instances the situation is different. We show in this section that it is NP-complete to determine whether the fixed 4-variable pattern P_{XX} shown in Figure 5 occurs as a topological minor.

Theorem 8.1 *The problem of deciding $I \in CSP_{\overline{TM}}(P_{XX})$ is coNP-complete.*

Proof: The problem is clearly in coNP, so it suffices to give a reduction from 3-SAT to the complement of the problem from the statement of the theorem. Let I_{SAT} be an instance

of 3-SAT with variables $x_1,\ldots,x_n$ and clauses $C_1,\ldots,C_m$. We create a binary CSP instance I with variables $\{S,E\} \cup \{p_i \mid i = 0\ldots n+m\} \cup \{X_{ir}, \overline{X}_{ir} \mid i = 1\ldots n, r = 1\ldots m\}$, such that determining $I \not\in CSP_{\overline{TM}}(P_{XX})$ is equivalent to solving I_{SAT}.

Consider the diagrams P_i, T_r, P_S and P_E shown in Figure 5. The variables in diagram T_r correspond to the literals in the clause C_r, with X_{ir} corresponding to $\neg x_i$ occurring in clause r and $\overline{X}_{ir}$ corresponding to x_i occurring in clause r. That is, the example of diagram T_r in Figure 5 would correspond to the clause $C_r = x_j \vee \neg x_k \vee x_\ell$. It is implicitly meant that the index i goes from 1 to n, and the index r goes from 1 to m. The positive edges in I are the solid lines in these $n+m+2$ diagrams. Any pairs of points in I not shown to be connected with a positive edge by one of these diagrams is connected by a negative edge.

The only pairs of variables in I connected by more than one positive edge are $\{p_{n+m}, E\}$ and $\{p_0, S\}$. So, if P_{XX} occurs as a topological minor pattern in I, then the subproblems P_E and P_S must map injectively into P_{XX}. Therefore, deciding whether P_{XX} occurs as a minor pattern in I is equivalent to deciding whether there is a path π of positive edges from p_0 to p_{n+m} in I which passes through each variable at most once. We now show that this (graph) condition is equivalent to I_{SAT} having a satisfying assignment.

If such a path π exists, then for each variable x_i of I_{SAT}, it must select in P_i either the upper path through variables X_{ir} $(r = 1,\ldots,m)$ or the lower path through variables $\overline{X}_{ir}$ $(r = 1,\ldots,m)$. Thus π selects a truth value for each variable x_i: TRUE if π follows the upper of these two paths, FALSE otherwise. The positive edges in P_i $(1 \leq i \leq n)$ use different points in $A_{X_{ir}}$ (shown as the bottom of the two points) from the positive edges in T_j $(1 \leq j \leq m)$ (which use the top points); this ensures that π does indeed pass through the points $p_0,\ldots,p_{n+m}$ in this order.

The path π must pass from p_{n+r-1} to p_{n+r} by one of the three paths: through variables $\overline{X}_{jr}, X_{kr}$ or $\overline{X}_{\ell r}$, without passing through variables that have been already used by π. Thus, for π to exist it must have already assigned TRUE to one of the literals $x_j, \neg x_k$ or x_ℓ of the clause C_r.

It follows that P_{XX} occurs as a topological minor of I if and only if I has an appropriate path of positive edges, which occurs if and only if I_{SAT} is satisfiable. ∎

Theorem 8.1 shows that not all classes defined by forbidding topological minors can be identified in polynomial time. Certain uses of tractable classes require polynomial-time recognition, in particular, the automatic recognition and resolution of easy instances within general-purpose solvers. On the other hand, polynomial-time recognition of a tractable class $\mathscr{C}$ is not required for the construction of a polynomial-time solvable relaxation in $\mathscr{C}$, nor in the proof (by a human being) that a subproblem of CSP encountered in practice falls in $\mathscr{C}$.

9 Discussion

Not all patterns are topological-minor tractable. For example, for the two patterns, P, shown in Figure 6, $CSP_{\overline{TM}}(P)$ is NP-hard. This is implied by the following two facts which hold for both of these particular patterns, P:

1. $\mathrm{CSP}_{\overline{\mathrm{TM}}}(P) = \mathrm{CSP}_{\mathrm{SP}}(P)$.

2. $\mathrm{CSP}_{\overline{\mathrm{SP}}}(P)$ is NP-complete [Cohen *et al.*, 2012].

One long-term goal is to characterise precisely which patterns are topological-minor tractable.

We are also investigating other applications for topological minors, such as in variable elimination [Cohen *et al.*, 2013].

The notion of a topological minor of a CSP instance, introduced here, allows a new approach to the definition of tractable classes of CSP instances. Although the examples of such classes obtained above may not prove to be especially significant in themselves, we believe that this new approach has great potential for systematically describing all tractable classes. However, to achieve further progress it may well be necessary to further refine or modify the definition of a topological minor given here. We regard this work as simply a first step towards a general topological theory of complexity for constraint satisfaction problems.

References

[Angelsmark and Thapper, 2005] Ola Angelsmark and Johan Thapper. A microstructure based approach to constraint satisfaction optimisation problems. In *Proceedings of the 18th International FLAIRS Conference*, 2005.

[Barto and Kozik, 2014] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3, 2014.

[Cohen and Jeavons, 2003] D. Cohen and P. Jeavons. Tractable constraint languages. In *Constraint Programming*, chapter 11, pages 299–331. Morgan Kaufman, 2003.

[Cohen *et al.*, 2006] David A. Cohen, Peter Jeavons, Christopher Jefferson, Karen E. Petrie, and Barbara M. Smith. Symmetry definitions for constraint satisfaction problems. *Constraints*, 11(2-3):115–137, 2006.

[Cohen *et al.*, 2012] David A. Cohen, Martin C. Cooper, Páidí Creed, Dániel Marx, and András Z. Salamon. The tractability of CSP classes defined by forbidden patterns. *J. Artif. Intell. Res. (JAIR)*, 45:47–78, 2012.

[Cohen *et al.*, 2013] David A. Cohen, Martin C. Cooper, Guillaume Escamocher, and Stanislav Živný. Variable elimination in binary CSP via forbidden patterns. In Francesca Rossi, editor, *IJCAI*. IJCAI/AAAI, 2013.

[Cooper and Escamocher, 2012] Martin C. Cooper and Guillaume Escamocher. A dichotomy for 2-constraint forbidden CSP patterns. In Jörg Hoffmann and Bart Selman, editors, *AAAI*. AAAI Press, 2012.

[Cooper and Živný, 2012] Martin C. Cooper and Stanislav Živný. Tractable triangles and cross-free convexity in discrete optimisation. *J. Artif. Intell. Res. (JAIR)*, 44:455–490, 2012.

[Cooper *et al.*, 2010] Martin C. Cooper, Peter G. Jeavons, and András Z. Salamon. Generalizing constraint satisfaction on trees: Hybrid tractability and variable elimination. *Artif. Intell.*, 174(9-10):570–584, 2010.

[Diestel, 2010] Reinhard Diestel. *Graph Theory*. Springer, fourth edition edition, 2010.

[Dirac, 1966] G. A. Dirac. Short proof of menger's graph theorem. *Mathematika*, 13(1):42–44, 1966.

[El Mouelhi *et al.*, 2013] Achref El Mouelhi, Philippe Jégou, Cyril Terrioux, and Bruno Zanuttini. Some new tractable classes of csps and their relations with backtracking algorithms. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 61–76. Springer, 2013.

[Escamocher, 2013] G. Escamocher. *Forbidden Patterns in Constraint Satisfaction Problems*. Ph.D thesis, Toulouse University, IRIT, University of Toulouse, France, 2013.

[Freuder, 1982] E.C. Freuder. A sufficient condition for backtrack-free search. *J. ACM*, 29(1):24–32, 1982.

[Freuder, 1985] E.C. Freuder. A sufficient condition for backtrack-bounded search. *Journal of the ACM*, 32:755–761, 1985.

[Grohe *et al.*, 2011] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA*, pages 479–488. ACM, 2011.

[Grohe, 2007] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1–24, March 2007.

[Jégou, 1993] Philippe Jégou. Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In *AAAI*, pages 731–736, Menlo Park, CA, USA, jul 1993. AAAI Press.

[Marx, 2013] Dániel Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM*, 60(6):42, 2013.

[Robertson and Seymour, 2004] Neil Robertson and Paul D. Seymour. Graph minors. XX. wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[Rossi *et al.*, 2006] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *The Handbook of Constraint Programming*. Elsevier, 2006.

[Salamon and Jeavons, 2008] András Z. Salamon and Peter G. Jeavons. Perfect constraints are tractable. In *CP*, volume 5202 of *Lecture Notes in Computer Science*, pages 524–528. Springer, 2008.

[Tutte, 1966] W.T. Tutte. Connectivity in graphs. Mathematical Expositions. 15. Toronto: University of Toronto Press; London: Oxford University Press. IX, 145 p. (1966)., 1966.