Artificial Intelligence 317 (2023) 103880

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Solving infinite-domain CSPs using the patchwork property ☆

Konrad K. Dabrowski [a,*], Peter Jonsson [c], Sebastian Ordyniak [b], George Osipov [c]

[a] *School of Computing, Newcastle University, UK*
[b] *School of Computing, University of Leeds, UK*
[c] *Department of Computer and Information Science, Linköpings Universitet, Sweden*

ARTICLE INFO

Article history:
Received 3 July 2021
Received in revised form 17 May 2022
Accepted 2 February 2023
Available online 6 February 2023

Keywords:
Constraint satisfaction problem
Parameterized complexity
Treewidth
Infinite domain
Lower bound

ABSTRACT

The constraint satisfaction problem (CSP) has important applications in computer science and AI. In particular, infinite-domain CSPs have been intensively used in subareas of AI such as spatio-temporal reasoning. Since constraint satisfaction is a computationally hard problem, much work has been devoted to identifying restricted problems that are efficiently solvable. One way of doing this is to restrict the interactions of variables and constraints, and a highly successful approach is to bound the treewidth of the underlying primal graph. Bodirsky & Dalmau (2013) [14] and Huang et al. (2013) [47] proved that CSP(Γ) can be solved in $n^{f(w)}$ time (where n is the size of the instance, w is the treewidth of the primal graph and f is a computable function) for certain classes of constraint languages Γ. We improve this bound to $f(w) \cdot n^{O(1)}$, where the function f only depends on the language Γ, for CSPs whose basic relations have the patchwork property. Hence, such problems are fixed-parameter tractable and our algorithm is asymptotically faster than the previous ones. Additionally, our approach is not restricted to binary constraints, so it is applicable to a strictly larger class of problems than that of Huang et al. However, there exist natural problems that are covered by Bodirsky & Dalmau's algorithm but not by ours, and we begin investigating ways of generalising our results to larger families of languages. We also analyse our algorithm with respect to its running time and show that it is optimal (under the Exponential Time Hypothesis) for certain languages such as Allen's Interval Algebra.

1. Introduction

The constraint satisfaction problem over a constraint language Γ (CSP(Γ)) is the problem of finding a variable assignment which satisfies a set of constraints, where each constraint is constructed from a relation in Γ. This problem can be used to model many problems encountered in computer science and AI, see e.g. Rossi et al. [73] or Dechter [30]. The CSP is computationally hard in the general case; if the variable domains are finite, then the problem is NP-complete, and otherwise it may be of arbitrarily high complexity [15]. Hence, identifying tractable problems is of great practical interest.

Tractable fragments have historically been identified using two different methods: either (1) restrict the relations that are allowed in the constraint language or (2) restrict how variables and constraints interact in problem instances. We focus

☆ Parts of this article appeared in the proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021) [27].

* Corresponding author.
E-mail addresses: konrad.dabrowski@newcastle.ac.uk (K.K. Dabrowski), peter.jonsson@liu.se (P. Jonsson), sordyniak@gmail.com (S. Ordyniak), george.osipov@liu.se (G. Osipov).

https://doi.org/10.1016/j.artint.2023.103880

on the second kind of restrictions in this article; these are often referred to as *structural restrictions*. One common way of studying structural restrictions is via the *primal graph*: this graph has the variables as its vertices with two of them joined by an edge if they occur together in the scope of a constraint. The graph parameter treewidth [11,72] has proven to be very useful in this context, since many NP-hard graph problems are tractable on instances with bounded treewidth. The treewidth of the primal graph has been extensively used in the study of finite-domain CSPs. It is known that the problem is *fixed-parameter tractable* (fpt), i.e. it can be solved in $f(w+d) \cdot n^{O(1)}$ time, where n is the size of the instance, w is the treewidth of the primal graph, d is the domain size and f is some computable function. This was proven by Gottlob et al. [40]; see also Samer & Szeider [74] for a more general treatment.

Let us now consider infinite-domain CSPs. For certain classes of constraint languages, Bodirsky & Dalmau [14, Corollary 1] proved that CSP(Γ) can be solved in $n^{O(w)}$ time (where the exact expression in the $O(w)$ term may depend on the constraint language), while Huang et al. [47, Theorem 6] obtained the bound $O(w^3 n \cdot \mathrm{e}^{w^2 \log n}) = n^{O(w^2)}$. These results prove the weaker property of membership in the complexity class XP. Algorithms with a running time bounded by $n^{f(w)}$ are obviously polynomial-time when w is fixed. However, since w appears in the exponent, such algorithms become impractical (even for small w) when large instances are considered. It is significantly better if a problem is fpt and can be solved in time $f(w) \cdot n^{O(1)}$, since the order of the polynomial in n does not depend at all on w.

Our main result is an fpt algorithm for CSPs over infinite domains where the underlying basic relations have the patchwork property [59], parameterized by treewidth. Several important CSPs (such as RCC8 and Allen's Interval Algebra [19,59]) are known to have this property. The patchwork property ensures that the union of two satisfiable instances of the CSP, whose constraints agree on their common variables, is also satisfiable. With the above discussion in mind, it is clear that our algorithm has better computational properties than the two previous ones. We will now briefly compare the applicability of the algorithms; more about this can be found in Section 7. Bodirsky & Dalmau's algorithm (BD) works for constraint languages that are *ω-categorical* (in fact, it even works for languages where only the core is ω-categorical), while Huang, Li & Renz's algorithm (HLR) works for languages with binary relations that have the *atomic network amalgamation property* (aNAP). Our algorithm has a wider applicability than HLR, since aNAP implies the patchwork property and our algorithm is not restricted to binary relations. We note that it does not seem difficult to extend the HLR algorithm so that it works for non-binary relations; such an extension would lead to an XP algorithm and our proposed algorithm remains faster.

The relation to BD is more complex, since there are problems that are covered by BD but not by our algorithm. We note that in some cases our algorithm is not applicable to a CSP directly, but one can find an equivalent problem that has the patchwork property via *homogenisation* — one example is the *Branching Time Algebra* [4] — the details are discussed later on. However, there are cases where homogenisation is not applicable, so the exact dividing line is unfortunately unclear.

Note that somewhat surprisingly the situation for finite-domain CSPs in the context of primal treewidth is rather similar to the situation for infinite CSPs having the patchwork property. That is, finite-domain CSPs are fixed-parameter tractable parameterized by primal treewidth plus domain size, but known to be W[1]-hard and in XP parameterized by primal treewidth alone [61]. Therefore, if one fixes a particular finite constraint language, finite-domain CSPs are fixed-parameter tractable parameterized by primal treewidth alone because the domain size is bounded for every finite constraint language.

In Section 2, we introduce some necessary preliminaries. The remainder of this article is divided into three distinct parts. In the first part (Sections 3 and 4), we present our main algorithm and prove that it achieves the required time bound. Our algorithm is based on dynamic programming and it is quite different from the BD algorithm since BD is based on a transformation to Datalog. The connection with HLR is more subtle: HLR is a recursive algorithm based on ideas by Darwiche [28], which like our algorithm, uses concrete tree decompositions. The important difference can be seen in the time complexity, where HLR is an XP algorithm and ours is fixed-parameter tractable. We complement our algorithmic results with tight lower bounds based on the Exponential Time Hypothesis: these show that the existence of significantly faster algorithms is not possible in certain special cases such as Allen's Interval Algebra.

In the second part (Section 5), we analyse the applicability of our algorithm. Even though the patchwork property is well known within the CSP community, there are not many formalisms that have been proven to have this property. By using certain model-theoretical concepts, we obtain an alternative way of identifying constraint languages with the patchwork property. Based on this, we demonstrate how to apply our results on constraint languages that are definable in $(\mathbb{Q}; <)$ (with applications in, for instance, temporal reasoning and scheduling) and phylogeny languages (which are useful in bioinformatics). These classes of languages give rise to CSPs with non-binary relations. Such relations have, unfortunately, not been well studied in AI, since the focus has almost exclusively been on binary relations. The phylogeny languages are particularly interesting in this respect, since the basic relations themselves are non-binary.

In the third part (Section 6), we study how our fpt result can be generalised to languages that do not have the patchwork property. One concrete example of an interesting language without the patchwork property is the previously mentioned Branching Time Algebra. We use model-theoretic tools to achieve one possible generalisation: we show that *homogenisation* can be used to extend our fpt result to certain classes of languages that do not have the patchwork property, and this extended result covers the so-called *Branching Time Algebra* [4].

A preliminary version of this article appeared as a conference paper in the Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-2021). The present version is significantly expanded and the major differences are as follows: (1) properties that make the algorithm applicable have been crystallised in a more abstract way, allowing us to solve a wider class of problems, namely those CSPs where the relations have different arities and, under certain conditions, CSPs with infinite constraint languages, (2) a fine-grained analysis of the running time has been conducted, allowing us to prove optimality of our algorithms (under the Exponential Time Hypothesis) for several interesting cases, and (3) a study of applicability of our approach to infinite-domains CSPs without the patchwork property has been initiated.

2. Preliminaries

In this section we introduce the necessary prerequisites.

2.1. Relational structures

A *(relational) signature* τ is a set of symbols, each with an associated natural number called their *arity*. A *(relational) τ-structure* $\mathbf{A}$ consists of a set D (the domain), together with relations $R^{\mathbf{A}} \subseteq D^k$ for each k-ary symbol $R \in \tau$. A structure is *countable* if its domain is a countable set. To avoid overly complex notation, we sometimes do not distinguish between the symbol R for a relation and the relation $R^{\mathbf{A}}$ itself. We also allow ourselves to view relational structures as sets and, for instance, write expressions like $R \in \mathbf{A}$.

Let $\mathbf{A}$ be a τ-structure over a domain D. We say that $\mathbf{A}$ is *k-ary* if every relation in $\mathbf{A}$ has arity k. Define $\mathbf{A}^{=k} = \bigcup\{R \in \mathbf{A} \mid R \text{ has arity } k\}$, i.e. $\mathbf{A}^{=k}$ is the union of all k-ary relations in $\mathbf{A}$.

The relations in $\mathbf{A}$ are *jointly exhaustive* (JE) if for all $k \geq 2$, $\mathbf{A}^{=k}$ is either empty or equal to D^k. They are *pairwise disjoint* (PD) if $R \cap R' = \varnothing$ for all distinct $R, R' \in \mathbf{A}$. In other words, the relations are JEPD if the nonempty subsets $\mathbf{A}^{=k}$ partition D^k.

The following concept is also used in the literature: the relations of $\mathbf{A}$ are JE$^+$ if there is a natural number $d \geq 2$ such that $\mathbf{A}^{=k} = D^k$ for all $k \in \{2, \ldots, d\}$, and $\mathbf{A}^{=k}$ is empty otherwise. Note that any set of JE relations can be augmented with the total relations D^k for all $k \in \{2, \ldots, d\}$ where $\mathbf{A}^{=k}$ is empty, thus making the (trivially) extended structure JE$^+$. Also note that every JE$^+$ structure is also JE, and that a k-ary structure cannot be JE$^+$ unless $k = 2$. The difference between JE and JE$^+$ reflects different traditions within the CSP community. Some researchers have concentrated on k-ary structures and for them the JE property is natural. Others have concentrated on structures with mixed arities and then the JE$^+$ property becomes natural.

Denote the equality relation over the domain D by $Eq_2 = \{(d, d) \mid d \in D\}$. The relations of $\mathbf{A}$ are *jointly diagonalizable* (JD) if $\bigcup\{R \in \mathbf{A} \mid R \subseteq Eq_2\} = Eq_2$. Note that JD holds vacuously if the equality relation is included in $\mathbf{A}$, and that a k-ary structure can only be JD if $k = 2$.

2.2. Logic

Let $\mathbf{A}$ be a τ-structure. First-order formulas ϕ over $\mathbf{A}$ (or, for short, $\mathbf{A}$-formulas) are defined using the logical symbols of universal and existential quantification, disjunction, conjunction, negation, equality, bracketing, variable symbols, the relation symbols from τ, and the symbol $\perp$ for the truth-value false. First-order formulas over $\mathbf{A}$ can be used to define relations: for a formula $\phi(x_1, \ldots, x_k)$ with free variables $x_1, \ldots, x_k$, the corresponding relation R is the set of all k-tuples $(t_1, \ldots, t_k) \in D^k$ such that $\phi(t_1, \ldots, t_k)$ is true in $\mathbf{A}$. In this case we say that R is *first-order definable* in $\mathbf{A}$. Our definitions of relations are always parameter-free, i.e. we do not allow the use of domain elements within them, and they are always quantifier-free. We may thus assume (without loss of generality) that all formulas defining relations are in disjunctive normal form (DNF). A formula is in DNF if it is a disjunction of one or more conjunctions of one or more *atomic formulas* of the type $R(\bar{x})$ or $\neg R(\bar{x})$, where $R \in \mathbf{A} \cup \{=\}$ and $\bar{x}$ is a sequence of variables. The conjunctions of atomic formulas are referred to as *terms*.

The most common way of using JEPD relations in AI-relevant CSPs is via the constraint language $\mathbf{A}^{\vee=}$, where $\mathbf{A}$ is a k-ary structure. For every subset S of relations in $\mathbf{A}$, $\mathbf{A}^{\vee=}$ contains the relation that is the union of the relations in S. The set $\mathbf{A}^{\vee=}$ contains the unions of all subsets of $\mathbf{A}$. Equivalently, $R \in \mathbf{A}^{\vee=}$ if R can be written as a disjunction $R_1(\bar{x}) \vee \cdots \vee R_p(\bar{x})$ where $R_1, \ldots, R_p \in \mathbf{A}$ and $\bar{x} = (x_1, \ldots, x_k)$. This definition demands that all relations in $\mathbf{A}$ have the same arity. Since we want to study sets of relations that are more expressive than $\mathbf{A}^{\vee=}$, we let $\langle \mathbf{A} \rangle_{\mathrm{b}}$ denote the set of relations that are definable by quantifier-free formulas that *only* contain the relations in $\mathbf{A}$, i.e. one is not allowed to use the equality relation $=$ unless it is a member of $\mathbf{A}$. The members of $\langle \mathbf{A} \rangle_{\mathrm{b}}$ are sometimes referred to as *Boolean combinations* of the relations in $\mathbf{A}$. When the relations in $\mathbf{A}$ are k-ary for some k, $\mathbf{A}^{\vee=} \subsetneq \langle \mathbf{A} \rangle_{\mathrm{b}}$ in general. Let us consider a concrete example. Let R and S denote 3-ary relations. The relation $T_1(x, y, z)$ defined by $R(x, y, z) \vee S(x, y, z)$ is a member of both $\{R, S\}^{\vee=}$ and $\langle \{R, S\} \rangle_{\mathrm{b}}$. The relation $T_2(x, y, z)$ defined by $R(x, y, z) \vee R(y, z, x)$ is a member of $\langle \{R, S\} \rangle_{\mathrm{b}}$, but it is not a member of $\{R, S\}^{\vee=}$.

Note that if a set of k-ary relations $\mathbf{A}$ is finite and JEPD, then we may assume that first-order formulas are negation-free: any negated relation can be replaced by the disjunction of all other relations: $\neg R(\bar{x}) \equiv \bigvee_{S \in \mathbf{A} \setminus \{R\}} S(\bar{x})$.

2.3. Constraint satisfaction

Let $\mathbf{A}$ be a τ-structure with domain D. The Constraint Satisfaction Problem over $\mathbf{A}$ (CSP($\mathbf{A}$)) is defined as follows:

Instance: A set V of variables and a set C of *constraints* of the form $R(v_1, \ldots, v_k)$, where $R \in \mathbf{A}$ is a relation of arity k and $v_1, \ldots, v_k \in V$.

Question: Is there an assignment $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for every $R(v_1, \ldots, v_k) \in C$?

We say that a CSP($\mathbf{A}$) instance (V, C) is *satisfiable* if it is a yes-instance of CSP($\mathbf{A}$), and we say that the function $f : V \to D$ (that witnesses the satisfiability of (V, C)) is a *solution*. The structure $\mathbf{A}$ is often referred to as the *constraint language*. We say that a constraint language $\mathbf{A}$ is *finite* if it contains a finite number of relations. Let $\mathbf{A}$ be a finite constraint language with JEPD relations. Consider a finite $\mathbf{\Gamma} \subseteq \langle \mathbf{A} \rangle_b$ and let $\mathcal{I} = (V, C)$ be an instance of CSP($\mathbf{\Gamma}$). Recall that every relation used in a constraint in C can be defined by a DNF $\mathbf{A}$-formula that involves only positive atomic formulas of the form $R(\bar{x})$, where $R \in \mathbf{A}$. A *certificate* for $\mathcal{I}$ is a satisfiable instance $\mathcal{C} = (V, C')$ of CSP($\mathbf{A}$) that *implies* every constraint in C, i.e. for every $R(v_1, \ldots, v_k)$ in C, there is a term in the definition of this constraint (as a DNF $\mathbf{A}$-formula) such that all constraints in this term are in C'.

Proposition 1. *An instance of CSP($\mathbf{\Gamma}$) admits a certificate if and only if it is satisfiable.*

Now assume CSP($\mathbf{A}$) is decidable. Then, there is an algorithm deciding whether an instance (V, C') of CSP($\mathbf{A}$) is a certificate for an instance (V, C) of CSP($\mathbf{\Gamma}$): first, check that (V, C') is satisfiable, and then, for all $R(v_1, \ldots, v_k) \in C$, verify that (V, C') implies $R(v_1, \ldots, v_k)$ by considering every term in the definition of R and checking if it is included in C'. Note that the length of the DNF formula defining R depends only on the arity of R and $|\mathbf{A}|$, which are both bounded by constants since $\mathbf{\Gamma}$ and $\mathbf{A}$ are finite languages. Thus, if CSP($\mathbf{A}$) is solvable in polynomial time, then the certificate test can also be performed in polynomial time.

An instance of CSP($\mathbf{A}$) is *complete* if it contains a constraint over every k-tuple of (not necessarily distinct) variables for every k such that there is a relation in $\mathbf{A}$ of arity k. A certificate is complete if it is a complete instance of CSP($\mathbf{A}$). Since the relations in $\mathbf{A}$ are JE, any certificate can be extended to a complete one. Thus, we can assume that all certificates are complete.

Example 2. Consider the structure $\mathbf{Q} = (\mathbb{Q}; <, >, =)$, i.e. the rationals under the natural ordering. The relation $B = \{(x, y, z) \in \mathbb{Q}^3 \mid x < y < z \vee z < y < x\}$ is known as the *betweenness* relation and it is a member of $\langle \mathbf{Q} \rangle_b$. Let $I = (\{w, x, y, z\} \mid \{B(w, x, y), B(x, y, z)\})$ be an instance of CSP($\{B\}$). The instance I is satisfiable and this is witnessed by the solution $f(w) = 0$, $f(x) = 1$, $f(y) = 2$, $f(z) = 3$. A certificate for this instance is $\{w < x, x < y, y < z\}$. A complete certificate is

$$\begin{array}{llll} w = w, & w < x, & w < y, & w < z, \\ x > w, & x = x, & x < y, & x < z, \\ y > w, & y > x, & y = y, & y < z, \\ z > w, & z > x, & z > y, & z = z. \end{array}$$

For any instance $\mathcal{I} = (V, C)$ of CSP and any set of variables $U \subseteq V$, define $C[U] \subseteq C$ to include all constraints whose scope is in U. We say that $\mathcal{I}[U] = (U, C[U])$ is the *subinstance* of $\mathcal{I}$ induced by U. We will also say that $\mathcal{I}[U]$ is obtained by *projecting* $\mathcal{I}$ *onto* U. Properties of certificates (including completeness) are preserved under projections. We formalise this observation below.

Proposition 3. *If $\mathcal{C}$ is a certificate for $\mathcal{I} = (V, C)$, then $\mathcal{C}[U]$ is a certificate for $\mathcal{I}[U]$ for all $U \subseteq V$. If $\mathcal{C}$ is complete, then $\mathcal{C}[U]$ is also complete.*

2.4. Parameterized complexity

In parameterized algorithmics [33,37,66] the runtime of an algorithm is studied with respect to the input size n and a parameter $p \in \mathbb{N}$. The basic idea is to find a parameter that describes the structure of the instance such that the combinatorial explosion can be confined to this parameter. In this respect, the most favourable complexity class is FPT (*fixed-parameter tractable*), which contains all problems that can be decided by an algorithm running in $f(p) \cdot n^{O(1)}$ time, where f is a computable function. Problems that can be solved in this time are said to be *fixed-parameter tractable* (fpt). The more general class XP contains all problems decidable in $n^{f(p)}$ time, i.e. the problems solvable in polynomial time when the parameter p is bounded. Clearly, FPT $\subseteq$ XP. Moreover, the inclusion is strict (see e.g. [37]).

We will concentrate on one well-known parameter in this article: the *treewidth* of the *primal graph*. Thus, if we state that some problem is fpt, then we always mean with respect to this parameter. The primal graph of an instance of a CSP is the undirected graph whose vertices coincide with the variables of the instance, and where two vertices are joined by an edge if they occur in the scope of the same constraint. Treewidth is based on *tree decompositions*: a tree decomposition (T, X) of an undirected graph $G = (V, E)$ consists of a rooted tree T and a mapping X from the nodes of T to the subsets of V. The subsets $X(t)$ are called *bags*. T_t stands for the subtree rooted at t, while V_t is the set of all variables occurring in the bags of T_t, i.e. $V_t = \bigcup_{s \in V(T_t)} X(s)$. A tree decomposition fulfils the following properties:

1. $\bigcup_{t \in V(T)} X(t) = V$.
2. If $(u, v) \in E$, then $u, v \in X(t)$ for some $t \in V(T)$.
3. For any $t_1, t_2, t_3 \in T$, if t_2 lies on the path between t_1 and t_3, then $X(t_1) \cap X(t_3) \subseteq X(t_2)$.

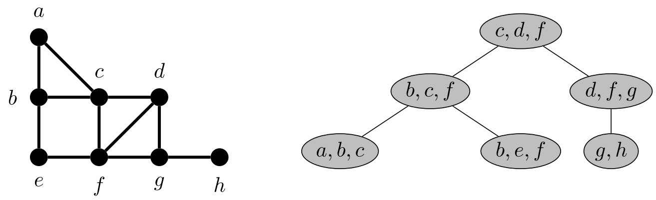

Fig. 1. A graph (left) and an optimal tree decomposition of the graph (right).

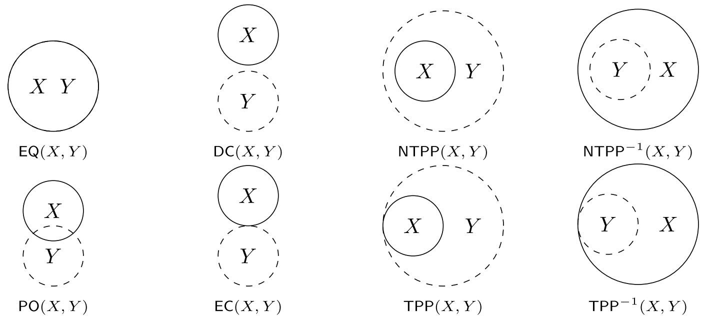

Fig. 2. Illustration of the relations of RCC8 with two-dimensional disks.

The width of a tree decomposition T is defined as $\max\{|X(t)| : t \in V(T)\} - 1$. The treewidth of a graph G is the minimum width among all tree decomposition of G. See also Fig. 1 for an example of an optimal tree decomposition of a graph. It is NP-complete to determine whether a graph has treewidth at most k [6], but when k is fixed the graphs with treewidth k can be recognised and corresponding tree decompositions can be constructed in linear time [20].

2.5. Qualitative spatial and temporal reasoning

We will consider several well-known formalisms for qualitative spatial and temporal reasoning. All of them can be defined as $\mathbf{B}^{\vee=}$ via a binary constraint language $\mathbf{B}$ with JEPD relations. We begin with a well-known spatial formalism. The region connection calculus (RCC8) [70] is a formalism for qualitative spatial reasoning, where the basic objects (referred to as *regions*) are non-empty regular closed subsets of a connected regular topological space. The regions do not have to be internally connected, that is, they may consist of different disconnected pieces. $\mathbf{B}_{\mathrm{RCC8}}$ contains eight relations: EQ (equal), PO (partial overlap), DC (disconnected), EC (externally connected), NTPP (non-tangential proper part), its converse NTPP^{-1}, TPP (tangential proper part) and its converse TPP^{-1}. See Fig. 2 for examples. RCC5 is a variant of RCC8 where one is not able to distinguish regions from their topological closure, i.e. the distinction between boundary points and interior points is ignored. The disconnectedness relations DC and EC are replaced by DR $=$ DC $\cup$ EC (distinct from), the tangential and non-tangential proper part relations TPP and NTPP are replaced by PP $=$ TPP $\cup$ NTPP (proper part), and PP^{-1} is defined analogously.

It is important to note that the exact choice of relations for representing a reasoning problem as a CSP may be crucial, and we will return to this issue in Sections 5 and 6. The importance of choosing the right representation can be illustrated even with simple formalisms such as RCC5 and RCC8. RCC8 can be represented with structures $\mathbf{A}$ and $\mathbf{B}$ such that CSP($\mathbf{A}$) is the same computational problem as CSP($\mathbf{B}$), while $\mathbf{A}$ and $\mathbf{B}$ are very different from a model-theoretical point of view. Such differences may be important when, for instance, constructing algorithms: the model-theoretical properties may simplify the algorithm construction process considerably. A concrete example is Bodirsky and Wölfl's [19] XP algorithm for RCC8 parameterized by treewidth of the primal graph. By choosing the right representation of RCC8, they could directly utilise the algorithmic framework of Bodirsky and Dalmau [14] and thus avoid unnecessary work. Another issue is based on the fact that there are structures $\mathbf{A}$ and $\mathbf{B}$ that look like suitable representations of RCC5 or RCC8 (for instance, by having the "right" composition tables), but have different CSPs. This may be highly confusing when modelling various problems, and different representations may obviously lead to problems having different computational complexity. Bodirsky and Jonsson discuss issues like these in the context of RCC5 [16, Section 2.5.2] with the aid of fairly natural structures having domains such as the non-empty subsets of the integers and the non-empty open disks in $\mathbb{R}^2$.

For RCC5 and RCC8, we will thus exclusively use the representations suggested by Bodirsky & Wölfl [19], whose CSP coincides with the standard interpretation of RCC relations. The representation is based on *Fraïssé's theorem* [38] applied to RCC8 instances and it is (unfortunately) not easy to visualize.

The choice of representation for the temporal formalisms we present below is, fortunately, much easier: the natural representations via concrete objects in $\mathbb{Q}^d$ have proven to capture the intended computational problems and at the same time have advantageous model-theoretical properties. We will consequently use these representations throughout the article.

Table 1
The thirteen basic relations in Allen's Interval Algebra. The endpoint relations $I^- < I^+$ and $J^- < J^+$ that are valid for all intervals I and J are omitted.

Basic relation		Example	Endpoints
I precedes J	p	iii	$I^+ < J^-$
J preceded-by I	pi	jjj	
I meets J	m	iiii	$I^+ = J^-$
J met-by I	mi	jjjj	
I overlaps J	o	iiii	$I^- < J^- < I^+ < J^+$
J overlapped-by I	oi	jjjj	
I during J	d	iii	$I^- > J^-$,
J includes I	di	jjjjjjj	$I^+ < J^+$
I starts J	s	iii	$I^- = J^-$,
J started-by I	si	jjjjjjj	$I^+ < J^+$
I finishes J	f	iii	$I^+ = J^+$,
J finished-by I	fi	jjjjjjj	$I^- > J^-$
I equals J	e	iiii jjjj	$I^- = J^-$, $I^+ = J^+$

1. *Allen's Interval Algebra* (IA) [3] is a temporal reasoning formalism where one considers relations between intervals of the form $I = [I^-, I^+]$, where $I^-, I^+ \in \mathbb{Q}$, $I^- < I^+$ are the start and end points, respectively. The language $\mathbf{B}_{\text{IA}}$ consists of thirteen basic relations illustrated in Table 1.
2. The d-dimensional *Block Algebra* (BA_d) [9] is a generalisation of IA to d-dimensional boxes with sides parallel to the coordinate axes. The relations in $\mathbf{B}_{\text{BA}}$ are d-tuples of IA relations, each one applied in the corresponding dimension.
3. The *Cardinal Direction Calculus* (CDC) [57] is a formalism for spatial reasoning with points on the plane as the basic objects. The relations in $\mathbf{B}_{\text{CDC}}$ correspond to eight cardinal directions (North, East, South, West and four intermediate ones) plus the equality relation. They can be viewed as pairs (R_1, R_2) for all choices of $R_1, R_2 \in \{<, =, >\}$, where each relation applies to the corresponding coordinate. See Table 2 for the correspondence between cardinal directions and pairs (R_1, R_2).

Table 2
The basic relations of Cardinal Direction Calculus.

=	N	E	S	W	NE	SE	SW	NW
$(=, =)$	$(=, >)$	$(>, =)$	$(=, <)$	$(<, =)$	$(>, >)$	$(>, <)$	$(<, <)$	$(<, >)$

3. The main algorithm

The goal of this section is to present an fpt algorithm that is applicable to a wide range of interesting CSPs. This section is divided into four parts: we describe the patchwork property and the algorithm in the first two parts, analyse concrete time bounds in the third part, and discuss infinite constraint languages in the fourth part.

3.1. The patchwork property

The basic CSP property underlying our algorithm is the following.

Definition 4 *(Lutz and Miličić [59]).* A JEPD constraint language **A** has the *patchwork property (PP)* if, for every pair of complete satisfiable instances $\mathcal{I}_1 = (V_1, C_1)$ and $\mathcal{I}_2 = (V_2, C_2)$ of CSP(**A**) such that $\mathcal{I}_1[V_1 \cap V_2] = \mathcal{I}_2[V_1 \cap V_2]$, the instance $(V_1 \cup V_2, C_1 \cup C_2)$ is also satisfiable.

We want to underline the importance of the completeness condition in the previous definition: for example, consider the JEPD constraint language $(<, =, >)$ with domain $\mathbb{Q}$ and the two satisfiable incomplete instances $(\{a, x, b\}, \{a < x, x < b\})$ and $(\{a, y, b\}, \{a > y, y > b\})$. The intersection of these instances contains no constraints, so it is trivially satisfiable. However, their union is not satisfiable since the constraints imply that $a < b$ and $a > b$ hold simultaneously.

Several prominent formalisms for qualitative spatial and temporal reasoning in AI are known to have the patchwork property. For example, the JEPD basic relations of Allen's Interval Algebra, the Block Algebra, and the Cardinal Direction Calculus have the patchwork property [46,59], assuming that the standard representations from Section 2.5 are used. The picture is slightly more complex for RCC8 and RCC5. Lutz and Miličić [59] show that RCC8 restricted to the real plane has the patchwork property, and Huang [46] points out that this result can be lifted to the multi-dimensional case via Bodirsky

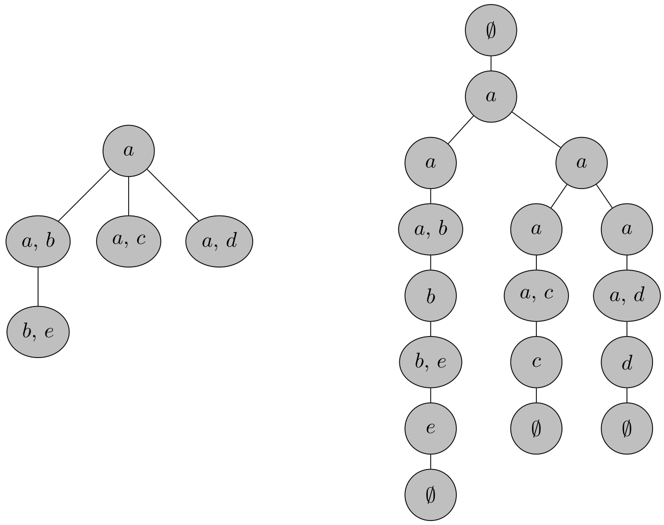

Fig. 3. A tree decomposition (left) and a corresponding nice tree decomposition (right).

& Wölfl's [19] representation. Baader & Rydval [8] also point this out in a more general setting and we will come back to their results in Section 5. We refer the reader who is interested in general model-theoretical properties of spatial reasoning formalisms to the papers by Düntsch and Li [34,35].

The following lemma (concerning complete certificates) is a direct consequence of the patchwork property.

Lemma 5. *Let* $\mathbf{A}$ *be a finite set of JEPD relations with the patchwork property and assume that* $\mathbf{\Gamma} \subseteq \langle \mathbf{A} \rangle_{\mathrm{b}}$ *is finite. For any two satisfiable instances* $\mathcal{I}_1 = (V_1, C_1)$ *and* $\mathcal{I}_2 = (V_2, C_2)$ *of CSP*$(\mathbf{\Gamma})$ *admitting complete certificates* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *such that* $\mathcal{C}_1[V_1 \cap V_2] = \mathcal{C}_2[V_1 \cap V_2]$, *the instance* $(V_1 \cup V_2, C_1 \cup C_2)$ *is also satisfiable.*

Proof. Let $\mathcal{C}_1 = (V_1, C_1')$, $\mathcal{C}_2 = (V_2, C_2')$, and define the instance $\mathcal{C}_\cup = (V_1 \cup V_2, C_1' \cup C_2')$ of CSP($\mathbf{A}$). We claim that $\mathcal{C}_\cup$ is a certificate for $(V_1 \cup V_2, C_1 \cup C_2)$. First, we note that $\mathcal{C}_\cup$ is satisfiable by Definition 4. Now consider a constraint $c \in C_1 \cup C_2$. Then c is in one of the following sets: $C_1 \setminus C_2$, $C_1 \setminus C_2$ or $C_1 \cap C_2$. In the first case, $\mathcal{C}_\cup[V_1] = \mathcal{C}_1$ implies c. Similarly, in the second case $\mathcal{C}_\cup[V_2] = \mathcal{C}_2$ implies c. Finally, if c is in the intersection of C_1 and C_2, then $\mathcal{C}_\cup[V_1 \cap V_2] = \mathcal{C}_1[V_1 \cap V_2] = \mathcal{C}_2[V_1 \cap V_2]$ implies c. Thus, $\mathcal{C}_\cup$ implies $(V_1 \cup V_2, C_1 \cup C_2)$. □

3.2. Algorithm

To simplify the presentation of our algorithm, we will use a particular kind of tree decomposition. A tree decomposition is *nice* (see also Fig. 3 for an illustration of a nice tree decomposition) if it fulfils the following properties:

- $X(r) = \varnothing$ and $X(\ell) = \varnothing$ for the root r and all leaf nodes ℓ in T.
- Every non-leaf node in T is one of the following types:
  - An **introduce node**: a node t with exactly one child t' such that $X(t) = X(t') \cup \{v\}$ for some $v \in V \setminus X(t')$.
  - A **forget node**: a node t with exactly one child t' such that $X(t) = X(t') \setminus \{w\}$ for some $w \in V \cap X(t')$.
  - A **join node**: a node t with exactly two children t_1 and t_2 such that $X(t) = X(t_1) = X(t_2)$.

Given a tree decomposition T of width ω of an n-vertex graph, one can construct a nice tree decomposition of the same width and with $O(\omega n)$ nodes in linear time [21]. Note that nice tree decompositions are merely a structured type of tree decomposition and their only purpose is to simplify the presentation of dynamic programming algorithms using treewidth.

We are now ready to present the fpt algorithm.

Theorem 6. *Let* $\mathbf{A}$ *be a finite constraint language with JEPD relations and the patchwork property. Assume CSP*$(\mathbf{A})$ *is decidable. For any finite constraint language* $\mathbf{\Gamma} \subseteq \langle \mathbf{A} \rangle_{\mathrm{b}}$, *CSP*$(\mathbf{\Gamma})$ *is fpt parameterized by the treewidth of the primal graph.*

Proof. Let $\mathcal{I} = (V, C)$ be an instance of CSP($\mathbf{\Gamma}$) and assume (T, X) is a nice tree decomposition of its primal graph. The algorithm works as follows: for every node $t \in T$, we compute the set $R(t)$ consisting of all certificates for $\mathcal{I}[V_t]$ projected onto $X(t)$. Clearly, $\mathcal{I}$ is satisfiable if and only if $R(r) \neq \varnothing$, where r is the root of T. We compute $R(t)$ using dynamic programming from the leaves upwards, i.e. a node is processed only if all its children have already been processed.

To start, we set $R(\ell) = \{(\varnothing, \varnothing)\}$ for all leaf nodes $\ell \in T$. Since the decomposition is nice, we only need to consider three cases.

1. If t is an introduce node with a child t', we enumerate certificates $\mathcal{C}$ for $\mathcal{I}[X(t)]$ and add $\mathcal{C}$ to $R(t)$ if $\mathcal{C}[X(t')]$ is in $R(t')$.
2. If t forgets a variable w and has a child t', then $R(t)$ is obtained by enumerating certificates in $R(t')$ and removing w together with all constraints involving it from the certificate.
3. Finally, if t joins nodes t_1 and t_2, then set $R(t) = R(t_1) \cap R(t_2)$ (recall that we may assume the certificates $\mathcal{C}$ that we consider are complete).

This completes description of the algorithm. To show correctness, we prove the following claim for every $t \in T$.

Claim 6.1. *$\mathcal{C}$ is a certificate for $\mathcal{I}[V_t]$ if and only if $\mathcal{C}[X(t)] \in R(t)$.*

We prove the claim by induction. In the base case, $R(\ell) = \{(\varnothing, \varnothing)\}$ is indeed the set of all certificates for $\mathcal{I}[V_\ell]$ for all leaves ℓ in T, since $V_\ell = X(\ell) = \varnothing$.

If t is an introduce node with child t', consider a certificate $\mathcal{C}$ for $\mathcal{I}[V_t]$. Note that $\mathcal{C}[V_{t'}]$ is a certificate for $\mathcal{I}[V_{t'}]$, so $\mathcal{I}[X(t')] \in R(t')$ by the inductive hypothesis. Furthermore, $\mathcal{C}[X(t)]$ is a certificate for $\mathcal{I}[X(t)]$, thus the algorithm adds it to $R(t)$. In the opposite direction, consider $\mathcal{K} \in R(t)$ and observe that, by construction, there is a certificate $\mathcal{C}'$ for $\mathcal{I}[V_{t'}]$ such that $\mathcal{K}[X(t')] = \mathcal{C}'[X(t')]$. Since $\mathcal{K}$ is a certificate for $X(t)$ and $X(t) \cap V_{t'} = X(t')$, the union of $\mathcal{K}$ and $\mathcal{C}'$ is a certificate for $\mathcal{I}[V_t]$ by the patchwork property, and $\mathcal{K}$ is precisely its projection onto $X(t)$.

If t is a forget node with a child t', consider a certificate $\mathcal{C}$ for $\mathcal{I}[V_{t'}]$ and note that, since $V_t = V_{t'}$, it is also a certificate for $\mathcal{I}[V_t]$. By the inductive hypothesis, $\mathcal{C}[X(t')] \in R(t')$, hence, the algorithm adds $\mathcal{C}[X(t)]$ to $R(t)$. In the opposite direction, consider $\mathcal{K}[X(t)] \in R(t)$ derived from $\mathcal{K} \in R(t')$ and note that the inductive hypothesis implies that $\mathcal{K}$ and, subsequently, $\mathcal{K}[X(t)]$ are projections of a certificate for $\mathcal{I}[V_t]$.

If t joins nodes t_1 and t_2, consider a certificate $\mathcal{C}$ for $\mathcal{I}[V_t]$. Note that it is also a certificate for $\mathcal{I}[V_{t_1}]$ and $\mathcal{I}[V_{t_2}]$, since $V_{t_1}, V_{t_2} \subseteq V_t$. By the inductive hypothesis, $\mathcal{C}[X(t_1)] = \mathcal{C}[X(t_2)] = \mathcal{C}[X(t)] \in R(t_1) \cap R(t_2)$ and the algorithm adds it to $R(t)$. In the opposite direction, consider $\mathcal{K} \in R(t) = R(t_1) \cap R(t_2)$. By the inductive hypothesis, there are certificates $\mathcal{C}_1$ for $\mathcal{I}[V_{t_1}]$ and $\mathcal{C}_2$ for $\mathcal{I}[V_{t_2}]$ such that $\mathcal{C}_1[X(t)] = \mathcal{C}_2[X(t)] = \mathcal{K}$. By the third property of tree decompositions, $V_{t_1} \cap V_{t_2} \subseteq X(t)$, thus the union of $\mathcal{C}_1$ and $\mathcal{C}_2$ is a certificate for $\mathcal{I}[V_t]$ by Lemma 5, and $\mathcal{K}$ is precisely its projection onto $X(t)$.

We continue with the time complexity of the algorithm. Let w denote the width of the decomposition (T, X), let k denote the maximum arity of relations in $\mathbf{A}$, and assume that $\tau(m)$ is the time required to enumerate certificates for an instance of CSP($\mathbf{\Gamma}$) with m variables. Note that since $\mathbf{A}$ and $\mathbf{\Gamma}$ are finite, the function τ depends only on the number of variables. Furthermore, $\tau(m)$ is an upper bound on the number of complete satisfiable instances of CSP($\mathbf{A}$) with m variables.

Claim 6.2. *For every $t \in T$, the computation of $R(t)$ requires at most $\tau(w+1)^2 \cdot w^{O(k)}$ time.*

First, note that any certificate with n variables contains at most $\sum_{i=1}^{k} n^i \leq kn^k$ constraints. Furthermore, $\tau(|X(t)|)$ is an upper bound on $|R(t)|$. Finally, taking a projection of a certificate onto $U \subseteq V$ requires $|U|^{O(k)}$ time. If t is an introduce node, the computation of $R(t)$ requires at most

$$\tau(w+1)|R(t')| \cdot |X(t')|^{O(k)} \leq \tau(w+1)^2 \cdot w^{O(k)}$$

time, i.e. $\tau(w+1)$ time to enumerate all potential certificates of $\mathcal{I}[X(t)]$, $|X(t')|^{O(k)}$ time to compute the projection of any such certificate onto $X(t')$, and $|R(t')|$ time to check whether this projection is in $R(t')$. If t is a forget node, the computation requires at most

$$|R(t')| \cdot |X(t)|^{O(k)} \leq \tau(w+1) \cdot w^{O(k)}$$

time, i.e. $|X(t)|^{O(k)}$ time to compute the projection of every certificate in $R(t')$ onto $X(t)$. Finally, if t joins nodes t_1 and t_2, the computation of $R(t)$ takes at most

$$|R(t_1)||R(t_2)| \cdot O(k|X(t)|^k) \leq \tau(w+1)^2 \cdot w^{O(k)}$$

time, where $w^{O(k)}$ accounts for the time required to compare two certificates.

There are $O(\omega n)$ nodes in the tree T, so the algorithm solves CSP($\mathbf{\Gamma}$) in $\tau(w+1)^2 \cdot w^{O(k)} \cdot O(\omega n) \in \tau(w+1)^2 \cdot w^{O(k)} \cdot n$ time. The $\tau(w+1)^2 \cdot w^{O(k)}$ term depends only on the parameter w, hence CSP($\mathbf{\Gamma}$) is fpt. □

3.3. Concrete time bounds

We continue by taking a closer look at some classical CSPs for qualitative spatial and temporal reasoning.

Corollary 7. *CSP($\mathbf{B}^{\vee=}$) is solvable in*

1. $2^{O(w^2)} \cdot O(n)$ *time if* $\mathbf{B}$ *is* $\mathbf{B}_{\mathrm{RCC5}}$ *or* $\mathbf{B}_{\mathrm{RCC8}}$,
2. $2^{O(w \log w)} \cdot O(n)$ *time if* $\mathbf{B}$ *is* $\mathbf{B}_{\mathrm{IA}}$, $\mathbf{B}_{\mathrm{BA}_d}$ *or* $\mathbf{B}_{\mathrm{CDC}}$.

Proof. Consider Claim 6.2 in Theorem 6. Since the languages under consideration are JEPD and binary, the total number of instances of CSP($\mathbf{B}$) with w variables is $|\mathbf{B}|^{w^2} = 2^{O(w^2)}$, since $|\mathbf{B}|$ is constant. Solving instances of CSP($\mathbf{B}_{\mathrm{RCC8}}$) and CSP($\mathbf{B}_{\mathrm{RCC5}}$) takes polynomial time [71], so $\tau(w) = 2^{O(w^2)}$. This yields the result for RCC5 and RCC8.

For the remaining cases, we need a tighter bound on $\tau(w)$. We show that the number of complete certificates for these problems is at most $2^{O(w \log w)}$. Observe that an ordered partition of a set S of size n is a surjective function $\pi : S \to \{1, \ldots, r\}$ for some $r \in \{1, \ldots, n\}$. Any two elements of S can be compared with the usual relations $\{<, =, >\}$ according to the values assigned to them by π. Observe that there are at most $n^n = 2^{O(n \log n)}$ ordered partitions of S.

Every complete satisfiable instance of CSP($\mathbf{B}_{\mathrm{IA}}$) corresponds to a unique ordered partition of the endpoints of the intervals (see e.g. [77]); this is because a complete instance provides a relation between every pair of distinct intervals, which in turn provides an ordering of the corresponding endpoints of the intervals. For an instance with w variables (i.e. $2w$ endpoints), there are at most $2^{O(w \log w)}$ such partitions. Thus, an instance of CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$) with w variables admits at most $2^{O(w \log w)}$ complete certificates. Given an ordered partition on the endpoints of the intervals, there is a polynomial-time procedure that recovers the corresponding complete satisfiable instance of CSP($\mathbf{B}_{\mathrm{IA}}$), if one exists: for every variable, check that its left endpoint precedes its right endpoint – if not, then there is no corresponding instance; otherwise, deduce the relation between every pair of variables according to the ordered partition of their endpoints. The last step works since $\mathbf{B}_{\mathrm{IA}}$ is JEPD. Finally, observe that generating all (unordered) partitions of a set takes $O(1)$ amortised time per partition [48] and generating all permutations takes $O(1)$ time per permutation [75]. Thus, $\tau(w) = 2^{O(w \log w)}$ for CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$).

The Block Algebra BA$_d$ can be viewed as an extension of Allen's Interval Algebra to d dimensions, and the complete certificates correspond to d ordered partitions of the endpoints. We have $((2w)^{2w})^d = 2^{O(w \log w)}$ since d is fixed, so $\tau(w) = 2^{O(w \log w)}$ for CSP($\mathbf{B}_{\mathrm{BA}_d}^{\vee=}$).

Every satisfiable instance of the CSP($\mathbf{B}_{\mathrm{CDC}}$) corresponds to two ordered partitions – one for the x coordinates and one for the y coordinates. There are $(w^w)^2 = 2^{O(w \log w)}$ such pairs of partitions, so $\tau(w) = 2^{O(w \log w)}$ for CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$). □

We want to make a few remarks at this point. First of all, the proof of Corollary 7 also shows that CSP($\mathbf{\Gamma}$) for any finite $\mathbf{\Gamma} \in \langle \mathbf{B} \rangle_{\mathrm{b}}$ is solvable in $2^{O(w^2)} \cdot O(n)$ time if $\mathbf{B}$ is $\mathbf{B}_{\mathrm{RCC5}}$ or $\mathbf{B}_{\mathrm{RCC8}}$, and in $2^{O(w \log w)} \cdot O(n)$ time if $\mathbf{B}$ is $\mathbf{B}_{\mathrm{IA}}$, $\mathbf{B}_{\mathrm{BA}_d}$ or $\mathbf{B}_{\mathrm{CDC}}$. Another important observation is that CSP($\mathbf{B}_{\mathrm{RCC5}}$) and CSP($\mathbf{B}_{\mathrm{RCC8}}$) are known to be solvable in $2^{O(n \log n)}$ time [51], but this result does not carry over to certificate enumeration and it is thus not sufficient for improving the $2^{O(w^2)} \cdot O(n)$ bound for CSP($\mathbf{B}_{\mathrm{RCC5}}$) and CSP($\mathbf{B}_{\mathrm{RCC8}}$).

Observation 8. *There are* $2^{\Theta(w^2)}$ *complete satisfiable instances of CSP($\mathbf{B}_{\mathrm{RCC5}}$) with* w *variables.*

Proof. First, note that there are $|\mathbf{B}_{\mathrm{RCC5}}|^{\binom{w}{2}} = 2^{O(w^2)}$ not necessarily satisfiable instances of CSP($\mathbf{B}_{\mathrm{RCC5}}$) with w variables. Now, consider complete instances (V, C) of this problem with $V = \{v_1, \ldots, v_w\}$, where the constraints over each pair of variables are either DR or PO, chosen arbitrarily. We claim that every such instance is satisfiable, and since there are $2^{\binom{w}{2}} = 2^{O(w^2)}$ of them, this yields the result.

Recall that in RCC5, the domain consists of all subsets of a topological space. Note that the subsets need not be internally connected. We refer to internally connected subsets as *regions*. To prove the claim, we construct an assignment f that assigns a subset of disjoint regions to every variable. For convenience, we consider two sets of regions: X_i for all $i \in \{1, \ldots, w\}$ and $Y_{i,j}$ for all $i, j \in \{1, \ldots, w\}$ with $i < j$. First, we set $f(v_i) = \{X_i\}$ for all $i \in \{1, \ldots, w\}$. Then, for every pair $i, j \in \{1, \ldots, w\}$ with $i < j$ such that PO(v_i, v_j) is in C, we add $Y_{i,j}$ to both $f(v_i)$ and $f(v_j)$.

If DR(v_i, v_j) is in C, then $f(v_i) \cap f(v_j) = \varnothing$, so $f(v_i)$ and $f(v_j)$ are disjoint. Otherwise, if PO(v_i, v_j) is in C, then $f(v_i) \cap f(v_j) \neq \varnothing$, $f(v_i) \setminus f(v_j) \neq \varnothing$ and $f(v_j) \setminus f(v_i) \neq \varnothing$, so $f(v_i)$ and $f(v_j)$ partially overlap. Thus, f is a satisfying assignment for (V, C) and this completes the proof. □

RCC8 is a generalisation of RCC5, so the same result holds for RCC8.

3.4. Infinite constraint languages

So far we have considered only finite constraint languages $\mathbf{\Gamma} \in \langle \mathbf{B} \rangle_{\mathrm{b}}$. If the language $\mathbf{\Gamma}$ is infinite, the representation of relations becomes problematic since the maximal arity is no longer bounded by a constant. In this case, the DNF formulas defining relations may be arbitrarily large. The time complexity of checking whether an instance of CSP($\mathbf{B}$) is a certificate for an instance of CSP($\mathbf{\Gamma}$) depends on the size of this representation. To circumvent this difficulty, we consider two possibilities – an oracle model and a restricted version of CSP, where the scope of every constraint contains only distinct variables. We remark that in both cases the fpt algorithm from Theorem 6 solves CSP($\mathbf{\Gamma}$) even if $\mathbf{\Gamma}$ is infinite.

In the oracle model we assume that, given a constraint $R(\bar{v})$ with $R \in \mathbf{\Gamma}$, the time complexity of checking whether an instance $\mathcal{C}$ of CSP($\mathbf{B}$) implies $R(\bar{v})$ is in polynomial time in the size of $\mathcal{C}$, i.e. it is independent of the representation of R. Clearly, Claim 6.2 still holds in this case since the time to compute the record $R(t)$ only depends on $|X(t)|$. Hence, CSP($\mathbf{\Gamma}$) is fpt in the oracle model even if $\mathbf{\Gamma}$ is infinite.

Alternatively, we can restrict CSP($\mathbf{\Gamma}$) by disallowing repeated variables in the scopes of its constraints. Note that if the language $\mathbf{B}$ is JD, e.g. if $\mathbf{B}$ contains equality, then this restriction does not affect the expressive power of the language: one can introduce many copies of a variable x by adding constraints $x = x', x' = x'', \ldots$ and use those in place of repeated variables. Note however that these additional constraints impact the structure of the instance. When the primal treewidth of an instance of CSP($\mathbf{\Gamma}$) is bounded by w and the scopes of constraints contain distinct variables, no constraint can have arity larger than $w + 1$. Thus, the size of the DNF formula defining any relation in this instance is bounded by a function of w and the computation of the record $R(t)$ only depends on $|X(t)|$. Hence, repetition-free CSP($\mathbf{\Gamma}$) is in fpt even if $\mathbf{\Gamma}$ is infinite.

4. Tight lower bounds

By Corollary 7, CSPs for Cardinal Direction Calculus, Allen's Interval Algebra and Block Algebra admit algorithms running in $2^{O(w \log w)} \cdot n^{O(1)}$ time on instances with primal treewidth w. It is natural to ask whether the dependence on w can be improved. In this section we provide evidence that such an improvement is improbable by establishing tight lower bounds on the running time of the algorithms for these problems assuming the Exponential Time Hypothesis (ETH) [49]. 3-Satisfiability is the problem of deciding whether a propositional formula in conjunctive normal form with at most three literals in each term admits a satisfying assignment. The ETH is a standard complexity assumption that rules out the existence of an algorithm for 3-Satisfiability running in subexponential time, i.e. there is no algorithm that runs in $2^{o(n)}$ time where n is the number of variables.

We start by showing a reduction from the following problem to CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$):

Vertex Colouring

Instance: (G, k), where G is an undirected graph with vertex set $V(G)$ and edge set $E(G)$, and $k \in \mathbb{N}$
Question: Does there exist $\chi : V(G) \to \{1, \ldots, k\}$ such that $\chi(u) \neq \chi(v)$ for all $(u, v) \in E(G)$?

Such a mapping χ is said to be a *proper k-colouring* of G. If such χ exists, we say that G is *k-colourable*. Lokshtanov et al. [58] introduced a framework for showing ETH-based lower bounds on parameterized algorithms and used it to prove the following theorem:

Theorem 9. Vertex Colouring *cannot be solved in* $2^{o(w \log w)} \cdot n^{O(1)}$ *time on graphs with treewidth w unless the ETH fails.*

Before we present our reduction, we prove two well-known useful observations about treewidth.

Lemma 10. *Graphs with treewidth w are* $(w + 1)$*-colourable.*

Proof. The treewidth of a graph G is alternatively characterised as $\min\{\omega(H) - 1 \mid G \text{ is a subgraph of } H, H \text{ is chordal}\}$, where $\omega(H)$ is the size of the largest clique in H (e.g. see [32, Chapter 12.4]). Chordal graphs are perfect graphs and this implies that H is colourable with $\omega(H) = w + 1$ colours (e.g. see [32, Chapter 5.5]). Since G is a subgraph of H, it is also $(w + 1)$-colourable. □

An induced subgraph H of G is a graph such that $V(H) \subseteq V(G)$ and $(u, v) \in E(H)$ if and only if both $(u, v) \in E(G)$ and $u, v \in V(H)$.

Lemma 11. *Let H be an induced subgraph of G and let w be the treewidth of H. Then the treewidth of G is at most* $w + |V(G)| - |V(H)|$.

Proof. Denote $V(G) \setminus V(H)$ by U. Given a tree decomposition (T, X) of H of width w, observe that adding all variables in U to every bag $X(t)$ yields a tree decomposition of G. Clearly, the maximum number of variables in a bag of the resulting tree decomposition is $w + 1 + |U|$, so the treewidth of G is at most $w + |U| = w + |V(G)| - |V(H)|$. □

Now we are ready to establish our first lower bound.

Theorem 12. *CSP*($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) *cannot be solved in* $2^{o(w \log w)} \cdot n^{O(1)}$ *time on instances with primal treewidth w unless the ETH fails.*

Proof. Let (G, k) be an instance of the Vertex Colouring problem. We construct an instance $\mathcal{I}$ of CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) that is satisfiable if and only if G is k-colourable.

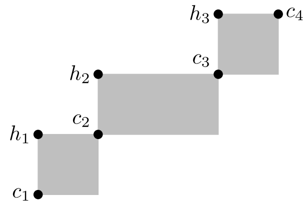

Fig. 4. Construction from the proof of Theorem 12 for $k = 4$.

Table 3
Transformation of CDC relations into Allen relations.

CDC	Definition	Allen
=	$x_1 = y_1$ and $x_2 = y_2$	e
N	$x_1 = y_1$ and $x_2 > y_2$	si
E	$x_1 > y_1$ and $x_2 = y_2$	f
S	$x_1 = y_1$ and $x_2 < y_2$	s
W	$x_1 < y_1$ and $x_2 = y_2$	fi
NE	$x_1 > y_1$ and $x_2 > y_2$	{oi,mi,pi}
SE	$x_1 > y_1$ and $x_2 < y_2$	d
SW	$x_1 < y_1$ and $x_2 < y_2$	{p,m,o}
NW	$x_1 < y_1$ and $x_2 > y_2$	di

First, we introduce variables z_v for all $v \in V(G)$, variables c_i for all colours $i \in \{1, \ldots, k\}$, and auxiliary variables h_j for all $j \in \{1, \ldots, k-1\}$. Then, we add the following constraints:

(C1) $c_i\{\mathrm{SW}\}c_{i+1}$, $h_i\{\mathrm{N}\}c_i$ and $h_i\{\mathrm{W}\}c_{i+1}$ for all $i \in \{1, \ldots, k-1\}$;
(C2) $z_v\{=, \mathrm{NE}\}c_1$, $z_v\{=, \mathrm{SW}\}c_k$ for all $v \in V(G)$;
(C3) $z_v\{\mathrm{SW}, =, \mathrm{NE}\}c_i$ for all $v \in V(G)$ and $i \in \{2, \ldots, k-1\}$;
(C4) $z_v\{\overline{\mathrm{SE}}\}h_j$ for all $v \in V(G)$ and $j \in \{1, \ldots, k-1\}$, where $\overline{\mathrm{SE}}$ is the negation of SE, i.e. $\overline{\mathrm{SE}} = \{\mathrm{S}, \mathrm{SW}, \mathrm{W}, \mathrm{NW}, \mathrm{N}, \mathrm{NE}, \mathrm{E}, =\}$;
(C5) $z_u\{\mathrm{SW}, \mathrm{NE}\}z_v$ for all $(u, v) \in E(G)$.

Towards proving correctness of the reduction, let $\chi : V(G) \to \{1, \ldots, k\}$ be a proper k-colouring of G. Define an assignment f for $\mathcal{I}$ by setting $f(c_i) = (i, i)$ for all $i \in \{1, \ldots, k\}$, $f(h_j) = (j, j+1)$ for all $j \in \{1, \ldots, k-1\}$ and $f(z_v) = (\chi(v), \chi(v))$ for all $v \in V(G)$. It is straightforward to verify that the constraints (C1)–(C4) are satisfied by this assignment. The constraints in (C5) are also satisfied because χ is a proper colouring of G.

In the opposite direction, let f be a satisfying assignment for $\mathcal{I}$. Assume that $f(c_i) = (a_i, b_i)$ for all $i \in \{1, \ldots, k\}$. Constraints (C1) imply that $f(h_j) = (a_j, b_{j+1})$ for all $j \in \{1, \ldots, k-1\}$. Fig. 4 shows an example of the construction. Consider $z_v = (x_v, y_v)$ for an arbitrary $v \in V(G)$. By (C2) and (C3), z_v can only take values inside the rectangles with corners (a_i, b_i), (a_i, b_{i+1}), (a_{i+1}, b_i) and (a_{i+1}, b_{i+1}) (shaded in the figure) excluding the boundary except for the bottom left and top right corners. Constraints (C4) forbid z_v from taking values inside the rectangles, which leaves only the corners as possible values. Thus, $f(z_v) \in \{f(c_1), \ldots, f(c_k)\}$ for all $v \in V(G)$, and we can define a colouring χ by setting $\chi(v) = i$ whenever $f(z_v) = f(c_i)$. Note that Constraints (C5) imply that $f(x_u) \neq f(x_v)$ whenever $(u, v) \in E(G)$. Therefore χ is a proper colouring.

Now consider the structure of the instance $\mathcal{I}$. Denote the primal treewidth of $\mathcal{I}$ by w and the treewidth of G by w_G. Observe that the primal graph of $\mathcal{I}$ consists of G with $2k-1$ additional vertices for $c_1, \ldots, c_k$ and $h_1, \ldots, h_{k-1}$. By Lemma 11, $w \leq w_G + 2k - 1$. Furthermore, by Lemma 10, G can be coloured with at most $w_G + 1$ colours. Thus, we can safely assume that $k \leq w_G$ and, consequently, $w < 3w_G$. Therefore, if CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) admits a $2^{o(w \log w)} \cdot n^{O(1)}$ algorithm, then so does the Vertex Colouring and this contradicts the ETH by Theorem 9. $\square$

We continue by establishing lower bounds on CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$) and CSP($\mathbf{B}_{\mathrm{BA}_d}^{\vee=}$). To prove the result, we use the following lemma:

Lemma 13. *There is a polynomial-time reduction from CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) to CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$) that preserves the primal graph of the instance.*

Proof. Let (V, C) be an instance of CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) and suppose f is a satisfying assignment. Denote $f(v)$ by (v_1, v_2) for all $v \in V$. We may assume without loss of generality that $v_1 < v_2$ for all $v \in V$, since all points in the image of f can be translated into the second quadrant of the coordinate plane, where $v_1 < 0$ and $v_2 > 0$. Thus, every pair of points (v_1, v_2) can be viewed as an interval $[v_1, v_2]$. With this in mind, we produce an instance (V', C') of CSP($\mathbf{B}_{\mathrm{CDC}}^{\vee=}$) with $V' = V$ by converting every relation in C into a $\mathbf{B}_{\mathrm{IA}}^{\vee=}$ relation according to the rules in Table 3. The disjunction of any subset of the CDC relations is obtained by taking the disjunction of their converted counterparts.

Equivalence of (V, C) and (V', C') follows from the definitions of the basic relations of Cardinal Direction Calculus and Allen's Interval Algebra. Clearly, the reduction requires polynomial time. Furthermore, there is a constraint in C' over a pair of variables if and only if there is a constraint in C over the same pair of variables. Thus, (V, C) and (V', C') have the same primal graph. $\square$

Corollary 14. *CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$) and CSP($\mathbf{B}_{\mathrm{BA}_d}^{\vee=}$) cannot be solved in $2^{o(w \log w)} \cdot n^{O(1)}$ time on instances with primal treewidth w unless the ETH fails.*

Proof. The result for CSP($\mathbf{B}_{\mathrm{IA}}^{\vee=}$) follows directly by combining Theorem 12 with Lemma 13. Note that $\mathbf{B}_{\mathrm{BA}_d}$ generalises $\mathbf{B}_{\mathrm{IA}}$ (namely, $\mathbf{B}_{\mathrm{IA}}$ is $\mathbf{B}_{\mathrm{BA}_d}$ for $d = 1$). Thus, the lower bound also holds for CSP($\mathbf{B}_{\mathrm{BA}_d}^{\vee=}$). $\square$

5. Applications based on patchwork

We analyse the applicability of our fpt result (Theorem 6) in this section. The patchwork property has not been directly verified for many formalisms—the list in Corollary 7 is quite meagre. However, it has been verified implicitly for wide classes of relations, and this is something that can be exploited. We first connect the patchwork property with the *amalgamation property* and *homogeneous* structures. This allows us to use model-theoretical concepts and results to identify interesting classes of relations that have the patchwork property. In the final step, we demonstrate how these ideas can be used on concrete examples — we study constraint languages that are first-order definable in $(\mathbb{Q}; <)$ and phylogeny languages.

5.1. Patchwork, amalgamation and homogeneity

When analysing PP from a model-theoretic angle, it is convenient to view CSPs in terms of homomorphisms. A *homomorphism* for τ-structures $\mathbf{A}, \mathbf{B}$ is a mapping $h : \mathbf{A} \to \mathbf{B}$ that preserves each relation of $\mathbf{A}$, i.e. if $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$ for some k-ary relation symbol $R \in \tau$, then $(h(a_1), \ldots, h(a_k)) \in R^{\mathbf{B}}$. Let $\mathbf{B}$ be a structure with a (not necessarily finite) signature τ. CSP($\mathbf{B}$) is then the following decision problem:

Instance. A finite τ-structure $\mathbf{A}$.
Question. Is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$?

It is well known that this definition coincides with the definition given earlier; this is, for instance, discussed in [16, Section 2]. We will use an analogue of subinstances for τ-structures: a τ-structure $\mathbf{A}$ is a *substructure* of a τ-structure $\mathbf{B}$ if and only if (1) the domain of $\mathbf{A}$ is a subset of the domain of $\mathbf{B}$ and (2) for each $R \in \tau$, the tuple $\bar{a}$ is in $R^{\mathbf{A}}$ if and only if $\bar{a}$ is in $R^{\mathbf{B}}$. Equivalently, $R^{\mathbf{A}} = R^{\mathbf{B}} \cap A^n$, where n is the arity of R. We need several kinds of homomorphisms in what follows. A *strong* homomorphism additionally satisfies the only if direction in the definition of a homomorphism, i.e. it also preserves the complements of relations. An *embedding* is an injective strong homomorphism. An *isomorphism* is a surjective (and thus bijective) embedding, and an *automorphism* is an isomorphism from $\mathbf{A}$ to itself. Let Aut($\mathbf{A}$) denote the set of automorphisms of $\mathbf{A}$. It is easy to verify that Aut($(\mathbb{Q}; <, =, >)$) contains the function $f(x) = a + x$ for arbitrary $a \in \mathbb{Q}$ and the function $g(x) = b \cdot x$ for every rational number $b > 0$. However, the set of automorphisms contains many other types of functions.

We connect the definition of patchwork with the *amalgamation property* (AP). A class $\mathcal{K}$ of τ-structures has AP if for every $\mathbf{B}_1, \mathbf{B}_2 \in \mathcal{K}$ such that their maximal common substructure $\mathbf{A}$ contains all elements that are both in $\mathbf{B}_1$ and $\mathbf{B}_2$, there exists $\mathbf{C} \in \mathcal{K}$ (called an *amalgam*) and embeddings $f_1 : \mathbf{B}_1 \to \mathbf{C}$ and $f_2 : \mathbf{B}_2 \to \mathbf{C}$ such that $f_1(a) = f_2(a)$ for every $a \in \mathbf{A}$. Let $\mathbf{D}$ be a countable τ-structure. Age($\mathbf{D}$) denotes the class of all finite τ-structures that embed into $\mathbf{D}$. Various connections between patchwork and amalgamation concepts have been hinted upon in the literature many times (see e.g. Bodirsky and Jonsson [16], Huang [46], and Li et al. [55,56]) but the details have not been clearly spelled out. Baader & Rydval [8] proved the following result.

Theorem 15. *Let $\mathbf{D}$ be a JE$^+$PDJD structure (as defined in Section 2.1). If Age($\mathbf{D}$) has the amalgamation property, then $\mathbf{D}$ has the patchwork property.*

Their results do not apply directly to structures that are k-ary. We complement Theorem 15 by showing that the same implication holds for k-ary JEPD structures that contain the k-ary equality relation.

Theorem 16. *Let $\mathbf{D}$ be a k-ary JEPD τ-structure with domain D and assume that the k-ary equality relation $Eq_k = \{(d, \ldots, d) \in D^k \mid d \in D\}$ is in $\mathbf{D}$. If Age($\mathbf{D}$) has the amalgamation property, then $\mathbf{D}$ has the patchwork property.*

Proof. Consider the instances $I_1 = (V_1, C_1)$, $I_2 = (V_2, C_2)$ of CSP($\mathbf{D}$) in Definition 4 as τ-structures $\mathbf{I}_1$, $\mathbf{I}_2$. Note that the intersection $I_1[V_1 \cap V_2] = I_2[V_1 \cap V_2]$ viewed as a τ-structure $\mathbf{A}$ is the maximal common substructure of $\mathbf{I}_1$, $\mathbf{I}_2$ and contains all elements that appear in both of them. To apply AP, we need to show that $\mathbf{I}_1$ and $\mathbf{I}_2$ embed into $\mathbf{D}$. Recall that an embedding is an injective strong homomorphism.

The remainder of the proof applies for all $i \in \{1, 2\}$. Since I_i is satisfiable, there is a homomorphism $h_i : \mathbf{I}_i \to \mathbf{D}$. Additionally, I_i is complete and $\mathbf{D}$ has JEPD relations, so for all $R \in \tau$, $(h_i(x_1), \ldots, h_i(x_k)) \in R^{\mathbf{D}}$ implies that the constraint $R(x_1, \ldots, x_k)$ is in C_i and it is satisfied. Hence, h_i is a strong homomorphism. To show that it is injective, we observe that for all $x, y \in \mathbf{I}_i$, if $Eq_k(x, y, \ldots, y) \in C_i$, then $x = y$. Otherwise, by completeness, there is another $R \in \tau$ such that $R(x, y, \ldots, y) \in C_i$. By PD, $R \cap Eq_k = \varnothing$, so $x \neq y$. Thus, h_i is injective, and ergo, an embedding.

We know that $\mathbf{I}_1, \mathbf{I}_2 \in \mathrm{Age}(\mathbf{D})$ so, by AP, the amalgam of $\mathbf{I}_1$ and $\mathbf{I}_2$ is also in $\mathrm{Age}(\mathbf{D})$. Note that the structure $\mathbf{C}$ defined by $(V_1 \cup V_2, C_1 \cup C_2)$ embeds into the amalgam. Hence, it is homomorphic to $\mathbf{D}$ and $(V_1 \cup V_2, C_1 \cup C_2)$ is satisfiable. □

Theorems 15 and 16 allow us to relate PP to some properties and results that have been successfully used in the study of CSPs. To this end, we will use *homogeneity*. A homogeneous structure $\mathbf{A}$ is a countable structure such that for every isomorphism $f : \mathbf{B} \to \mathbf{C}$ between finite substructures $\mathbf{B}, \mathbf{C}$ of $\mathbf{A}$, there is an automorphism f' of $\mathbf{A}$ extending f. Intuitively speaking, a homogeneous structure enjoys the following property: the surroundings of two isomorphic substructures always look *very* similar. Homogeneity thus implies that the structure has an extremely high degree of symmetry. The following result is part of the classical *Fraïssé's Theorem* [38].

Theorem 17. $\mathrm{Age}(\mathbf{A})$ *has AP when* $\mathbf{A}$ *is a countable homogeneous structure with a countable signature.*

Fraïssé's Theorem is explained in most textbooks on model theory such as Hodges [45]. Combining Theorems 15, 16 and 17 gives us the following result.

Corollary 18. *Let* $\mathbf{D}$ *denote a countable homogeneous structure with a countable signature.*

1. *If* $\mathbf{D}$ *is JE*$^+$*PDJD, then* $\mathbf{D}$ *has PP, and*
2. *if* $\mathbf{D}$ *is a* k*-ary JEPD structure that contains the* k*-ary equality relation, then* $\mathbf{D}$ *has PP.*

A large number of homogeneous structures are known from the literature (see, for example, the surveys by Macpherson [60] and Hirsch [44]) and they play an important role in CSP research. In fact, after the Feder-Vardi conjecture on finite-domain CSPs was settled (independently) by Bulatov [24] and Zhuk [80], much of the complexity-oriented work has concentrated on homogeneous infinite-domain CSPs. We note that all examples in Corollary 7 can be formulated by homogeneous structures; for instance, Hirsch [43] proved this for Allen's Interval Algebra and Bodirsky and Wölfl [19] for RCC8. A fact to keep in mind is that one may have two structures $\mathbf{A}$ and $\mathbf{B}$ such that $\mathrm{CSP}(\mathbf{A})$ is the same computational problem as $\mathrm{CSP}(\mathbf{B})$, $\mathbf{A}$ is homogeneous, but $\mathbf{B}$ is not homogeneous. This phenomenon is, for instance, discussed (in the context of RCC8) by Bodirsky and Wölfl [19] and Huang et al. [47] (in the context of temporal constraints). A straightforward example is provided by the structures $(\mathbb{Q}; <)$ and $(\mathbb{N}; <_{\mathbb{N}})$ where $<_{\mathbb{N}}$ denotes the ordering on the natural numbers. The structure $(\mathbb{Q}; <)$ is homogeneous while $(\mathbb{N}; <_{\mathbb{N}})$ is not,[1] and $\mathrm{CSP}((\mathbb{Q}; <))$ and $\mathrm{CSP}((\mathbb{N}; <_{\mathbb{N}}))$ are the same computational problems.

5.2. Examples

The machinery presented above allows us to show fpt results for large families of CSPs. Our first example is the set of CSPs $\mathcal{T}$ whose constraint languages consist of finite subsets of $\langle(\mathbb{Q}; <)\rangle_{\mathrm{b}}$. Well-known CSPs in $\mathcal{T}$ (with prominent applications in e.g. AI) are the Point Algebra [78], the ORD-Horn class [64] and certain scheduling problems [62], together with basic problems in complexity theory such as Betweenness and Cyclic Ordering [39]. Clearly, $\mathcal{T}$ contains many different CSPs based on non-binary relations and, in fact, the CSPs with binary relations are a subset of the Point Algebra and thus polynomial-time solvable [78]. One ought to observe that the CSP for Allen's Interval Algebra is *not* in $\mathcal{T}$ since its domain consists of the closed convex subsets of $\mathbb{Q}$ and not of $\mathbb{Q}$ itself, but there is a straightforward reduction from Allen's Interval Algebra to a certain problem in $\mathcal{T}$. The CSPs in $\mathcal{T}$ have been intensively studied in the literature: for instance, Bodirsky and Kára [18] proved that any CSP in $\mathcal{T}$ is either polynomial-time solvable or NP-complete.

Arbitrarily choose $\mathrm{CSP}(\Gamma)$ in $\mathcal{T}$. It is folklore that the structure $\mathbf{Q} = (\mathbb{Q}; <, >, =)$ is homogeneous (see, for instance, Example 2.1.2 in Macpherson [60] for a proof sketch). The structure $\mathbf{Q}$ is obviously JEPD and it contains the binary equality relation, so it has PP by Corollary 18. Since $\mathrm{CSP}(\mathbf{Q})$ is decidable, it follows from Theorem 6 that $\mathrm{CSP}(\Gamma)$ is fpt. This proves the following.

Proposition 19. *Every problem in* $\mathcal{T}$ *is fpt parameterized by the treewidth of the primal graph.*

We continue by illustrating Theorem 6 with a more elaborate example. *Phylogeny problems* are used for phylogenetic reconstruction in bioinformatics, but also in areas such as database theory, computational genealogy and computational

[1] Consider $f(1) = 0$, which is a trivial isomorphism between the substructures $(\{1\}; <_{\mathbb{N}})$ and $(\{0\}; <_{\mathbb{N}})$, but cannot be extended to an automorphism – there is no way to choose $f(0)$ such that $f(0) < f(1)$.

linguistics. A recent overview can be found in Warnow [79]. The problem is intuitively the following: given a partial description of a tree, is there a tree that is compatible with the given information? Many problems of this kind are NP-hard: concrete examples include the *subtree avoidance problem* [65], the *forbidden triple problem* [23] and the *quartet consistency problem* [76]. Fpt algorithms are thus an interesting option for solving phylogeny problems. Our basic idea is to rephrase phylogeny problems as CSPs and then apply Theorem 6. We formalise this below, mostly following Bodirsky et al. [17].

Let T be a *tree*, i.e. an undirected, acyclic, connected graph, and let r be the *root* of T. We only consider binary trees, i.e. all vertices except for the root have either degree 3 or 1, and the root has either degree 2 or 0. The vertex set of T is denoted by $V(T)$ and the set of leaves $L(T) \subseteq V(T)$ consists of the vertices of degree 1. For arbitrary $u, v \in V(T)$, we say that u *lies below* v if the path from u to the root r passes through v. We say that u *lies strictly below* v if u lies below v and $u \neq v$. The *youngest common ancestor* (yca) of $S \subseteq V(T)$ is the vertex u that lies above all vertices in S and has maximal distance from r; this vertex is uniquely determined by S. The *leaf structure* of T is the $\{C\}$-structure $(L(T); C)$ where $(x, y, z) \in C$ for any $x, y, z \in L(T)$ if and only if $yca(\{y, z\})$ lies strictly below $yca(\{x, y, z\})$. Following the literature on phylogeny problems, we write $x|yz$ instead of $C(x, y, z)$.

An *atomic phylogeny formula* ϕ is a conjunction of formulas of the form $x|yz$ and $x = y$. We say that ϕ with variables V is *satisfiable* if there exists a rooted binary tree T and a mapping $s : V \to L(T)$ such that ϕ is satisfied by T under s. The atomic phylogeny problem Aphyl is the computational problem with atomic phylogeny formulas as instances and the question is whether the formula is satisfiable or not. Aphyl is connected to CSPs as follows.

Theorem 20. *There exists a homogeneous $\{|, =\}$-structure $\mathbf{A}$ with a countable domain and the following property: an instance I of* Aphyl *is satisfiable if and only if I (viewed as an instance of CSP($\{|, =\}$)) homomorphically maps to $\mathbf{A}$.*

Proof. Use Proposition 2 in Bodirsky et al. [17]. □

The relation $x|yz$ will be a basic relation in the CSP we are aiming for. Since we need a JEPD set of relations as the basis for Theorem 6, the following observation (see, for instance Bodirsky et al. [17, Section 2.1]) is useful.

Observation 21. *Let x, y, z be arbitrary leaves in an arbitrarily chosen rooted binary tree. If $x|yz$, then it may be the case that $y = z$. However, $x|yz$ implies that $x \neq y$ and $x \neq z$. Hence, we either have $x|yz$, $y|xz$, $z|xy$ or $x = y = z$.*

Assume that the structure $\mathbf{A}$ in Theorem 20 has domain A and contains the relations $|'$ and $='$. Let $\mathbf{P}$ denote the structure $(A; R_1, R_2, R_3, R_4)$ where $R_1(x, y, z) \Leftrightarrow x|'yz$, $R_2(x, y, z) \Leftrightarrow y|'xz$, $R_3(x, y, z) \Leftrightarrow z|'xy$ and $R_4(x, y, z) \Leftrightarrow (x =' y =' z)$.

Proposition 22. *Let Γ be a finite subset of $\langle \mathbf{P} \rangle_b$. Then CSP($\Gamma$) is fpt parameterized by the treewidth of the primal graph.*

Proof. We know that $\mathbf{P} \subseteq \langle (A; |', =') \rangle_b$ and it is straightforward to verify that $\mathbf{P}$ is homogeneous since $(A; |', =')$ is homogeneous—all relations in $\mathbf{P}$ can be obtained by permuting the arguments of relations in $(A; |', =')$. The structure $\mathbf{P}$ is JEPD by Observation 21 and it contains the ternary equality relation. Thus, $\mathbf{P}$ has PP by Corollary 18.2. Finally, CSP($\mathbf{P}$) is solvable in polynomial time [2] and the proposition follows from Theorem 6. □

This proves that the three examples of NP-hard phylogeny problems that were discussed earlier are fpt. We exemplify this with the aid of the forbidden triple problem (the exact formulations of the other two problems as CSPs based on relations in $\langle \mathbf{P} \rangle_b$ can be found in Bodirsky et al. [17, Section 2.2]). This problem is the phylogeny problem concerning formulas

$$F(x, y, z) = \neg(x|yz)$$

and the parameterization is the primal treewidth of a conjunction of such formulas. The essence of Theorem 20 and Proposition 22 is that there exists a relation $R(x, y, z) \in \langle \mathbf{P} \rangle_b$ that exactly captures the formula F:

$$R(x, y, z) \equiv \neg R_1(x, y, z)$$

or, equivalently,

$$R(x, y, z) \equiv R_2(x, y, z) \vee R_3(x, y, z) \vee R_4(x, y, z).$$

Thus, the forbidden triple problem can be viewed as CSP($\{R\}$). The transformation from the forbidden triple problem to CSP($\{R\}$) is simply to replace each formula $F(x, y, z)$ with the constraint $R(x, y, z)$. This operation obviously preserves the treewidth of instances.

We remark that the disequality relation neq $= \{(a, b) \in A^2 \mid a \neq b\}$ is used to define relations in some phylogeny examples—note that $\text{neq}(x, y) \Leftrightarrow \neg R_4(x, x, y) \Leftrightarrow R_1(x, x, y) \vee R_2(x, x, y) \vee R_3(x, x, y)$ so the relation neq is a member of $\langle \mathbf{P} \rangle_b$.

6. Beyond patchwork

There are interesting examples of structures $\mathbf{A}$ that do not have PP. An eminent example is the *Branching Time Algebra* (BTA) [4] (also known as the *Branching Point Algebra (BPA)*) which has been used, for example, in planning [29], as the basis for temporal logics [36], and as the basis for a generalisation of Allen's Interval Algebra [69]. We note that, in particular, the complexity of the branching variant of Allen's Interval Algebra has recently gained attraction [10]. In BTA, the past of a time point is linearly ordered, but the future is only partially ordered (see Bodirsky [12, Section 5.2] for formal details). This implies that time becomes a directed tree-like structure with four basic relations $=$, $<$, $>$ and $||$, meaning "equal", "before", "after" and "unrelated", respectively.

One may formulate BTA as a CSP($\mathbf{B}_{\mathrm{BTA}}$) where $\mathbf{B}_{\mathrm{BTA}}$ is JEPD, but one cannot formulate the problem so that $\mathbf{B}_{\mathrm{BTA}}$ has PP; this follows from adapting an argument by Hirsch [44, Section 4.1]. Both CSP($\mathbf{B}_{\mathrm{BTA}}$) and CSP($\mathbf{B}_{\mathrm{BTA}}^{\vee=}$) are solvable in polynomial time [44, Section 4.2], but there are finite $\Gamma \subseteq \langle \mathbf{B}_{\mathrm{BTA}} \rangle_{\mathrm{b}}$ such that CSP(Γ) is NP-hard [22]. It is thus natural to ask whether the CSP(Γ) problem is fpt for arbitrary finite $\Gamma \subseteq \langle \mathbf{B}_{\mathrm{BTA}} \rangle_{\mathrm{b}}$. We show this in the affirmative by exploiting *homogenisability*: a homogenisable structure is a structure that can be extended with a finite number of new relations in order to make the expanded structure homogeneous [25,42]. Homogenisation has recently become an interesting tool for analysing CSPs: examples include connections between homogenisation and local consistency algorithms for CSPs [7] and applications concerning logically defined CSPs [12, Section 4.3.3]. More background information about homogenisable structures can be found in the survey by Macpherson [60] and the thesis by Ahlman [1]. We will prove a general fpt result for homogenisable structures in Theorem 28, and thus prove that Theorem 6 can indeed be generalised to certain structures that do not have PP. In particular, this result proves that CSP(Γ) is fpt whenever Γ is a finite subset of $\langle \mathbf{B}_{\mathrm{BTA}} \rangle_{\mathrm{b}}$. We divide the rest of this section into two parts, where the first part is concerned with ω-categorical structures and the second part describes how homogenisable structures can be used to obtain fpt results.

6.1. ω-categoricity

We first remind the reader of the definition of ω-categoricity: the *(first-order) theory* of a τ-structure $\mathbf{A}$, which is denoted by Th($\mathbf{A}$), is the set of all first-order τ-sentences, i.e. formulas without free variables, that are satisfied by $\mathbf{A}$, and $\mathbf{A}$ is said to be ω-categorical if Th($\mathbf{A}$) has exactly one model up to isomorphism. The concept of ω-categoricity plays a key role in the study of complexity aspects of CSPs [12], but it is also important from an AI perspective [43,46,50]. Examples of such structures include all structures with a finite domain and all structures that were presented in Section 2.5. The relation between ω-categorical structures and homogeneous structures can be summarised as follows. We say that a theory T admits *quantifier elimination* if for every formula $\phi(x_1, \ldots, x_n)$, there is a quantifier-free formula $\psi(x_1, \ldots, x_n)$ such that T entails

$$\forall x_1 \cdots \forall x_n (\psi(x_1, \ldots, x_n) \Leftrightarrow \phi(x_1, \ldots, x_n)).$$

Theorem 23. *Let* $\mathbf{A}$ *be a structure.*

1. *If* $\mathbf{A}$ *is* ω*-categorical, then* $\mathbf{A}$ *is homogeneous if and only if* Th($\mathbf{A}$) *admits quantifier elimination [60, Proposition 3.1.6].*
2. *If* $\mathbf{A}$ *is a homogeneous structure with finite signature, then* $\mathbf{A}$ *is* ω*-categorical [60, Corollary 3.1.3].*

A useful property of ω-categorical structures is that they can be refined into finite JE^{+}PDJD structures. Let $\mathbf{A} = (A; R_1, \ldots, R_m)$ denote a relational structure (that is not necessarily ω-categorical). The *orbit* of a tuple $\bar{a} = (a_1, \ldots, a_k) \in A^k$, denoted by Orb($\bar{a}$), is the set

$$\{(g(a_1), \ldots, g(a_n)) \mid g \in \mathrm{Aut}(\mathbf{A})\}.$$

The orbits of k-tuples in A partition the set A^k: for arbitrary $\bar{a}, \bar{b} \in A^k$, either $\mathrm{Orb}(\bar{a}) = \mathrm{Orb}(\bar{b})$ or $\mathrm{Orb}(\bar{a}) \cap \mathrm{Orb}(\bar{b}) = \emptyset$, and for every $\bar{a} \in A^k$ there exists a $\bar{c} \in A^k$ such that $\bar{a} \in \mathrm{Orb}(\bar{c})$. If $\mathbf{A}$ is ω-categorical, then the set $\{\mathrm{Orb}(\bar{a}) \mid \bar{a} \in A^k\}$ is finite for every $k \in \mathbb{N}$; this is an important consequence of a result by Engeler, Ryll-Nardzewski and Svenonius (this theorem is covered by most textbooks on model theory such as Hodges [45]). The following definition will simplify our presentation.

Definition 24. Let $\mathbf{A} = (D; R_1, \ldots, R_m)$ denote a relational structure and let $d \geq 1$. Define $\mathbf{A}^{\leq d}$ to be the relational structure over D whose relations are all orbits of at most d-ary tuples over A.

We collect a few straightforward facts about $\mathbf{A}^{\leq d}$ and we note that these are discussed in more detail by Baader & Rydval [8, Section 4]. Facts 1 and 2 are based on the observation that if a relation R contains the tuple $\bar{a} = (a_1, \ldots, a_k)$, then $\mathrm{Orb}(\bar{a}) \subseteq R$, too (by the definition of automorphisms), while Fact 3 is a direct consequence of the observation on orbit size of ω-categorical structures that was made above.

Fact 1. The structure $\mathbf{A}^{\leq d}$ is JE^{+}PDJD, but it is not a k-ary structure in general and it is not necessarily finite.

Fact 2. Every m-ary relation $R \in \mathbf{A}$ with $m \leq d$ can be viewed as the union of relations in $\mathbf{A}^{\leq d}$ or, equivalently,

$$R(x_1, \ldots, x_m) \equiv \bigvee_{S \in \mathbf{S}} S(x_1, \ldots, x_m)$$

for some $\mathbf{S} \subseteq \mathbf{A}^{\leq d}$.

Fact 3. The structure $\mathbf{A}^{\leq d}$ is finite if $\mathbf{A}$ is ω-categorical.

The following result connects structures $\mathbf{A}^{\leq d}$ with ω-categoricity, homogeneity and the patchwork property.

Theorem 25 *(Immediate consequence of Theorem 5 in Baader & Rydval [8]). Let* $\mathbf{A}$ *denote an* ω*-categorical homogeneous relational structure containing at most* d*-ary relations for some* $d \geq 2$*. Then* $\mathbf{A}^{\leq d}$ *has the patchwork property.*

6.2. Homogenisation

We will now present a result (Theorem 28) concerning fixed-parameter tractability of CSPs based on structures that do not have PP. To illustrate the result, we will come back to the branching time problem at the end of this section. The proof will use various ways of defining relations with the aid of logical formulas. Thus, in addition to full first-order logic, we also need fragments where only certain logical operators are allowed: the *existential fragment* consists of formulas built using negation, conjunction, disjunction and existential quantification only, while the *existential positive fragment* additionally disallows negation. We emphasise that it is required that the equality relation is allowed in existential (positive) definitions, which is a difference compared to the definitions underlying the operation $\langle\cdot\rangle_{\mathrm{b}}$. We begin with a decidability result.

Lemma 26. *Let* $\mathbf{A}$ *be a relational structure and assume that the relations* $R_1, \ldots, R_k$ *are existential positive definable in* $\mathbf{A}$*. If CSP(*$\mathbf{A}$*) is decidable, then CSP(*$\{R_1, \ldots, R_k\}$*) is also decidable.*

Proof. Suppose that, for $i \in \{1, \ldots, k\}$, R_i has definition

$$\phi_i(x_1, \ldots, x_m) \equiv \exists u_1 \cdots u_n \psi_i(x_1, \ldots, x_m, u_1, \ldots, u_n)$$

where ψ_i is quantifier-free. We assume (without loss of generality) that each ψ_i is in DNF. The conversion to DNF can be done without introducing any negations. Define the $(m+n)$-ary relation R'_i such that $R'_i(x_1, \ldots, x_m, u_1, \ldots, u_n) \equiv \psi_i(x_1, \ldots, x_m, u_1, \ldots, u_n)$.

Let $I = (V, C)$ be an instance of CSP($\{R_1, \ldots, R_k\}$). We first construct an equivalent instance $I' = (V', C')$ of CSP($\{R'_1, \ldots, R'_k\}$). Start by setting $V' = V$. Arbitrarily choose a constraint $R_i(x_1, \ldots, x_m)$ in C. Expand V' with n new variables $u_1, \ldots, u_n$ and add the relation $R'_i(x_1, \ldots, x_n, u_1, \ldots, u_n)$ to C'. Repeat this process for all constraints in C. It is obvious that I is satisfiable if and only if I' is satisfiable.

Recall that the formulas $\psi_1, \ldots, \psi_k$ are in DNF and that they contain no negations. If I' is satisfiable by an assignment $f : V' \to D$, then we can construct a certificate that witnesses this. For each constraint $R'_i(x_1, \ldots, x_m, u_1, \ldots, u_m)$, pick one term in ψ_i that is satisfied by f and put it into the set S. It follows that S is satisfiable if it is viewed as a CSP instance in the obvious way. Now, S only contains relations in $\mathbf{A} \cup \{=\}$ — recall that the equality relation can be used in an existential positive definition but its negation $\neg(x = y)$ cannot. For every constraint $x = y$ in S, identify the variable x with the variable y and note that the resulting set S' only contains relations in $\mathbf{A}$ and that it is satisfiable if and only if S is satisfiable. This suggests the following algorithm: enumerate all possible certificates for S' and check whether at least one of them is satisfiable. Only a finite number of certificates exist, since $\mathbf{A}$ is finite and the decidability of CSP($\mathbf{A}$) implies decidability of the satisfiability test. We conclude that CSP($\{R'_1, \ldots, R'_k\}$) is decidable and so is CSP($\{R_1, \ldots, R_k\}$). $\square$

We will now focus on structures that are *model-complete cores*. The exact definition is not important for our purposes, but a certain characterisation of ω-categorical model-complete cores is very important.

Lemma 27 *(Theorem 4.5.1 in Bodirsky [12]). A countable* ω*-categorical structure* $\mathbf{A}$ *is* model-complete *if and only if every first-order formula over* $\mathbf{A}$ *is equivalent to an existential positive formula over* $\mathbf{A}$*.*

Lemma 27 can be viewed as a restricted type of quantifier elimination. Model-complete cores are very useful when studying CSPs. It is known that every countable and ω-categorical structure $\mathbf{A}$ is homomorphically equivalent to an ω-categorical model-complete core $\mathbf{A}'$—this implies that CSP($\mathbf{A}$) and CSP($\mathbf{A}'$) can be viewed as the same problem [12, Section 1.1]. The model-complete core $\mathbf{A}'$ can, in various ways, be considered to be a more "structured" object than $\mathbf{A}$ and thus be easier to work with [12, Section 4.5]. Much of the work on the complexity of CSPs has consequently focused on model-complete cores.

We are finally ready to prove the main result of this section. We stress that the definition of homogenisation of a structure $\mathbf{A}$ requires that only a finite number of relations are added to $\mathbf{A}$. Otherwise, the resulting structure contains an infinite number of relations and this would prevent us from applying Theorem 6.

Theorem 28. *Let $\mathbf{B}$ be a countably infinite ω-categorical structure with finite signature, and assume that $\mathbf{B}$ is a model-complete core and that CSP($\mathbf{B}$) is decidable. If $\mathbf{B}$ is homogenisable by relations that are first-order definable in $\mathbf{B}$, then CSP($\mathbf{\Gamma}$) is fpt parameterized by the treewidth of the primal graph for arbitrary finite $\mathbf{\Gamma} \subseteq \langle \mathbf{B} \rangle_{\mathrm{b}}$.*

Proof. Let $\mathbf{C}$ denote the finite homogeneous expansion of $\mathbf{B}$ and let d denote the maximal arity of relations in $\mathbf{C}$. The structure $\mathbf{C}$ is ω-categorical by Theorem 23.2.

Claim 28.1. *CSP($\mathbf{C}^{\leq d}$) is decidable.*

We say that a relation $R \subseteq A^k$ is *preserved* by a function $f : A \to A$ if for every $(a_1, \dots, a_k) \in R$, $(f(a_1), \dots, f(a_k))$ is also in R. Let $\mathcal{O}$ be an orbit of k-tuples of $\mathbf{C}$. It follows immediately from the definition of orbits that $\mathcal{O}$ is preserved by every function in Aut($\mathbf{C}$). This implies that $\mathcal{O}$ is first-order definable in $\mathbf{C}$ [12, Proposition 4.2.9], since $\mathbf{C}$ is a countable ω-categorical structure. A direct consequence is that $\mathcal{O}$ is also first-order definable in $\mathbf{B}$, since $\mathbf{C}$ is a first-order definable expansion of $\mathbf{B}$.

Recall that $\mathbf{B}$ is a countable ω-categorical model-complete core, so every first-order formula over $\mathbf{B}$ is logically equivalent to an existential positive formula over $\mathbf{B}$ by Lemma 27. We conclude that $\mathcal{O}$ has an existential positive definition in $\mathbf{B}$. With this in mind, it follows that the relations in $\mathbf{C}^{\leq d}$ are existential positive definable in $\mathbf{B}$, since $\mathbf{C}^{\leq d}$ only contains relations that are orbits of tuples of $\mathbf{C}$. Lemma 26 thus implies that CSP($\mathbf{C}^{\leq d}$) is decidable, since $\mathbf{C}^{\leq d}$ only contains a finite number of relations by Fact 3.

Claim 28.2. $\mathbf{C}^{\leq d}$ *has PP.*

The structure $\mathbf{C}$ is an ω-categorical homogeneous structure that contains at most d-ary relations, so it has PP by Theorem 25.

Claim 28.3. $\mathbf{C}^{\leq d}$ *is JEPD.*

The structure $\mathbf{C}^{\leq d}$ is JE$^+$PDJD by Fact 1.

Theorem 6 combined with Claims 28.1-28.3 and the fact that $\mathbf{C}^{\leq d}$ is a finite structure implies that CSP($\mathbf{\Theta}$) is fpt for arbitrary $\mathbf{\Theta} \subseteq \langle \mathbf{C}^{\leq d} \rangle)_{\mathrm{b}}$. The structure $\mathbf{B}$ is ω-categorical, so Facts 2 and 3 imply that every relation in $\mathbf{B}$ can be viewed as a finite union of relations in $\mathbf{B}^{\leq d}$. We know that $\mathbf{B} \subseteq \mathbf{C}$, so $\mathbf{C}^{\leq d}$ consequently contains one relation for each orbit of d-tuples in $\mathbf{B}$, i.e. $\mathbf{B}^{\leq d} \subseteq \mathbf{C}^{\leq d}$. This implies that every relation in $\langle \mathbf{B} \rangle_{\mathrm{b}}$ has a logically equivalent relation in $\langle \mathbf{C}^{\leq d} \rangle_{\mathrm{b}}$. We may (without loss of generality since $\mathbf{\Gamma}$ is finite) assume that we have a pre-computed table that, for every $R \in \mathbf{\Gamma}$, contains the corresponding relation R' in $\langle \mathbf{C}^{\leq d} \rangle_{\mathrm{b}}$. Given an instance (V, C) of CSP($\mathbf{\Gamma}$), we can thus convert it in polynomial time into an equivalent instance of CSP($\mathbf{\Theta}'$) where $\mathbf{\Theta}'$ is a finite subset of $\langle \mathbf{C}^{\leq d} \rangle_{\mathrm{b}}$. We conclude that CSP($\mathbf{\Gamma}$) is fpt. $\square$

Let us now return to the branching time example that we discussed in the beginning of this section. Consider a structure $\mathbf{B}_{\mathrm{BTA}} = (B; =, <, >, ||)$ such that the Branching Time Algebra problem is the same computational problem as CSP($\mathbf{B}_{\mathrm{BTA}}$). Two suitable structures have been pointed out by Bodirsky [12, Section 5.2]: they are referred to as $\mathbb{S}$ and $\mathbb{T}$. We do not consider $\mathbb{S}$ in what follows, since it is not a model-complete core. The structure $\mathbb{T}$, though, is a countably infinite model-complete core that is ω-categorical. Hence, we let $\mathbf{B}_{\mathrm{BTA}}$ coincide with $\mathbb{T}$. Note that Bodirsky views these structures as having three relations $\leq$, $\neq$ and $||$, where $||$ allows both that two elements are unrelated or that they are equal. This difference is irrelevant in our setting; for instance, $x < y$ holds if and only if both $x \leq y$ and $x \neq y$ hold. Now, $\mathbf{B}_{\mathrm{BTA}}$ is a countably infinite model-complete core that is ω-categorical and, additionally, Bodirsky et al. [13] have shown that $\mathbf{B}_{\mathrm{BTA}}$ expanded by the relation

$$\{(x, y, z) \in B^3 \mid \exists u \in B\, ((u < x \lor u = x) \land (u > z \lor u||y) \land (z > u \lor z||u))\}$$

is homogeneous. Clearly, this relation is first-order definable in $\mathbf{B}_{\mathrm{BTA}}$. We know that $\mathbf{B}_{\mathrm{BTA}}$ is polynomial-time solvable, so CSP($\mathbf{\Gamma}$) is fpt for arbitrary finite $\mathbf{\Gamma} \subseteq \langle \mathbf{B}_{\mathrm{BTA}} \rangle_{\mathrm{b}}$ by Theorem 28.

It may be illuminating to compare the Branching Time Algebra with its sibling – the *partial-order time algebra* (PTA). PTA has various applications in, for instance, the analysis of concurrent and distributed systems [5,54]. In PTA, both the past and the future of a time point are partially ordered. This implies that time becomes a partial order with four basic relations $=_{\mathrm{PTA}}$, $<_{\mathrm{PTA}}$, $>_{\mathrm{PTA}}$ and $||_{\mathrm{PTA}}$, meaning "equal", "before", "after" and "unrelated", respectively. The satisfiability problem for PTA can be formulated with a countable finite homogeneous structure $\mathbf{D}$ (known as the *random partial order*) such that $\mathbf{D}$ is

JEPD and CSP(**D**) is decidable. Thus, Theorem 6 is directly applicable in this case (since **D** has PP by Corollary 18) and homogenisation is not necessary. For more details concerning PTA, together with a complexity classification, see Kompatscher & Van Pham [52].

7. Discussion and future research

Huang et al. [47] proved that CSP(**A**) is in XP whenever **A** is a binary constraint language with aNAP. This property is PP restricted to binary relations, with the completeness condition replaced by the algebraic closure condition. aNAP is less restrictive than PP, so it might be preferred in practical implementations for some constraint languages. However, in the worst case, using aNAP yields no advantage over using PP, and it is only defined for binary languages. We can thus conclude that our algorithm has a larger scope of applicability than the algorithm by Huang et al.

Bodirsky & Dalmau [14] show that CSP(**A**) is in XP whenever **A** is a countable structure that is ω-categorical. There are examples of ω-categorical model-complete cores that cannot be made homogeneous by adding *any* finite set of relations. A concrete example based on the countable atomless Boolean algebra can be found in [12, Section 5.7]. We conclude that there are still ω-categorical structures **A** for which we do not know whether CSP(**A**) is fpt or not. Closing this gap is an obvious direction for future research.

There are many relevant constraint satisfaction problems where our algorithmic framework is not directly applicable. Obvious examples are constraint languages that do not enjoy the patchwork property and where homogenisation is impossible or a suitable homogenisation process is not known. Examples include the Unit Interval Algebra (i.e. Allen's Interval Algebra restricted to intervals of equal length [68]), the Simple Temporal Problem [31], the $\mathcal{OPRA}$ calculus [63], various disjunctive temporal problems [67], metric extensions of Allen's Interval Algebra [53] and certain spatial formalisms including the Cardinal Direction Calculus for extended objects by Goyal and Egenhofer [41]. We note, however, that some of these problems have been analysed according to their parameterized complexity [26] by using other methods.

Our algorithm solves CSPs over Cardinal Direction Calculus, Allen's Interval Algebra and Block Algebra in $2^{O(w \log w)}$ time. Under the Exponential Time Hypothesis, significantly improving the dependence of w is not possible in these cases. However, for RCC5 and RCC8 the running time is slower, since the number of certificates is $2^{O(w^2)}$. Either proving a tight lower bound under plausible complexity assumptions or finding a faster $2^{o(w^2)} \cdot \text{poly}(||I||)$ algorithm for RCC5 or RCC8 is an interesting future direction. We believe that such an improved algorithm would need new ideas, and the RCC5 and RCC8 algorithms by Jonsson et al. [51] do not seem to be useful in such a project (as was discussed in Section 3.3).

Another plausible way forward is to consider parameterizations that are less restrictive or orthogonal to primal treewidth. Examples that come to mind are the treewidth of the dual graph or the incidence graph and variants of hypertree width, since these have been successfully used for efficiently solving CSPs as well as other combinatorial problems. However, these parameters can be ruled out for finite constraint languages, since there the treewidth of the primal, dual and incidence graphs, as well as all variants of hypertree width, are within a constant factor of each other [74]. This is not true for infinite constraint languages, so here additional parameters are interesting to study.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The first and third authors acknowledge support from the Engineering and Physical Sciences Research Council (EPSRC project EP/V00252X/1). The second and the fourth authors were supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. In addition, the second author was partially supported by the Swedish Research Council (VR) under grants 2017-04112 and 2021-04371.

References

[1] Ove Ahlman, Limit Laws, Homogenizable Structures and Their Connections, PhD thesis, Department of Mathematics, Uppsala University, 2018.
[2] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, Jeffrey D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM J. Comput. 10 (3) (1981) 405–421.
[3] James F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1983) 832–843.
[4] Frank Anger, Peter Ladkin, Rita Rodriguez, Atomic temporal interval relations in branching time: calculation and application, in: Applications of Artificial Intelligence IX, in: SPIE, vol. 1468, 1991, pp. 122–136.
[5] Frank D. Anger, On Lamport's interprocessor communication model, ACM Trans. Program. Lang. Syst. 11 (3) (1989) 404–417.
[6] Stefan Arnborg, Derek Corneil, Andrzej Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Matrix Anal. Appl. 8 (2) (1987) 277–284.
[7] Albert Atserias, Szymon Torunczyk, Non-homogenizable classes of finite structures, in: Proc. 25th EACSL Annual Conference on Computer Science Logic (CSL-2016), 2016, pp. 16:1–16:16.
[8] Franz Baader, Jakub Rydval, Description logics with concrete domains and general concept inclusions revisited, in: Proc. 10th International Joint Conference on Automated Reasoning (IJCAR-2020), 2020, pp. 413–431.
[9] Philippe Balbiani, Jean-François Condotta, Luis Fariñas del Cerro, Tractability results in the block algebra, J. Log. Comput. 12 (5) (2002) 885–909.

[10] Alessandro Bertagnon, Marco Gavanelli, Alessandro Passantino, Guido Sciavicco, Stefano Trevisani, Branching interval algebra: an almost complete picture, Inf. Comput. 281 (2021) 104809.
[11] Umberto Bertelé, Franscesco Brioschi, Nonserial Dynamic Programming, Academic Press, 1972.
[12] Manuel Bodirsky, Complexity of Infinite-Domain Constraint Satisfaction, Lecture Notes in Logic., vol. 52, Cambridge University Press, 2021.
[13] Manuel Bodirsky, David Bradley-Williams, Michael Pinsker, András Pongrácz, The universal homogeneous binary tree, J. Log. Comput. 28 (1) (2018) 133–163.
[14] Manuel Bodirsky, Víctor Dalmau, Datalog and constraint satisfaction with infinite templates, J. Comput. Syst. Sci. 79 (1) (2013) 79–100.
[15] Manuel Bodirsky, Martin Grohe, Non-dichotomies in constraint satisfaction complexity, in: Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP-2008), 2008, pp. 184–196.
[16] Manuel Bodirsky, Peter Jonsson, A model-theoretic view on qualitative constraint reasoning, J. Artif. Intell. Res. 58 (2017) 339–385.
[17] Manuel Bodirsky, Peter Jonsson, Trung Van Pham, The complexity of phylogeny constraint satisfaction problems, ACM Trans. Comput. Log. 18 (3) (2017) 23:1–23:42.
[18] Manuel Bodirsky, Jan Kára, The complexity of temporal constraint satisfaction problems, J. ACM 57 (2) (2010) 9:1–9:41.
[19] Manuel Bodirsky, Stefan Wölfl, RCC8 is polynomial on networks of bounded treewidth, in: Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011), 2011, pp. 756–761.
[20] Hans L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317.
[21] Hans L. Bodlaender, Ton Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs, J. Algorithms 21 (2) (1996) 358–402.
[22] Mathias Broxvall, Peter Jonsson, Point algebras for temporal reasoning: algorithms and complexity, Artif. Intell. 149 (2) (2003) 179–220.
[23] David Bryant, Building Trees, Hunting for Trees, and Comparing Trees, PhD thesis, University of Canterbury, 1997.
[24] Andrei Bulatov, A dichotomy theorem for nonuniform CSPs, in: Proc. 58th Annual Symposium on Foundations of Computer Science (FOCS-2017), 2017, pp. 319–330.
[25] Jacinta Covington, Homogenizable relational structures, Ill. J. Math. 34 (4) (1990) 731–743.
[26] Konrad K. Dabrowski, Peter Jonsson, Sebastian Ordyniak, George Osipov, Disjunctive temporal problems under structural restrictions, in: Proc. 35th AAAI Conference on Artificial Intelligence (AAAI-2021), vol. 35, 2021, pp. 3724–3732.
[27] Konrad K. Dabrowski, Peter Jonsson, Sebastian Ordyniak, George Osipov, Solving infinite-domain CSPs using the patchwork property, in: Proc. 35th AAAI Conference on Artificial Intelligence (AAAI-2021), vol. 35, 2021, pp. 3715–3723.
[28] Adnan Darwiche, Recursive conditioning, Artif. Intell. 126 (1–2) (2001) 5–41.
[29] Thomas L. Dean, Mark S. Boddy, Reasoning about partially ordered events, Artif. Intell. 36 (3) (1988) 375–399.
[30] Rina Dechter, Constraint Processing, Elsevier Morgan Kaufmann, 2003.
[31] Rina Dechter, Itay Meiri, Judea Pearl, Temporal constraint networks, Artif. Intell. 49 (1–3) (1991) 61–95.
[32] Reinhard Diestel, Graph Theory, 5th edition, Graduate Texts in Mathematics, vol. 173, Springer, 2017.
[33] Rodney G. Downey, Michael R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science., Springer, 2013.
[34] Ivo Düntsch, Sanjiang Li, Extension properties of Boolean contact algebras, in: Proc. 13th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS-2012), 2012, pp. 342–356.
[35] Ivo Düntsch, Sanjiang Li, On the homogeneous countable Boolean contact algebra, Log. Log. Philos. 22 (2013) 213–251.
[36] E. Allen Emerson, Joseph Y. Halpern, "Sometimes" and "not never" revisited: on branching versus linear time temporal logic, J. ACM 33 (1) (1986) 151–178.
[37] Jörg Flum, Martin Grohe, Parameterized Complexity Theory, Springer, 2006.
[38] Roland Fraïssé, Sur certaines relations qui généralisent l'ordre des nombres rationnels, C. R. Acad. Sci. 237 (1953) 540–542.
[39] Michael R. Garey, David S. Johnson, Computers and Intractability: a Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[40] Georg Gottlob, Francesco Scarcello, Martha Sideri, Fixed-parameter complexity in AI and nonmonotonic reasoning, Artif. Intell. 138 (1–2) (2002) 55–86.
[41] Roop Goyal, Max Egenhofer, The direction-relation matrix: a representation for directions relations between extended spatial objects, in: Proc. The Annual Assembly and the Summer Retreat of University Consortium for Geographic Information Systems Science, 1997.
[42] David Hartman, Jan Hubička, Jaroslav Nešetřil, Complexities of relational structures, Math. Slovaca 65 (2) (2015) 229–246.
[43] Robin Hirsch, Relation algebras of intervals, Artif. Intell. 83 (2) (1996) 267–295.
[44] Robin Hirsch, Expressive power and complexity in algebraic logic, J. Log. Comput. 7 (3) (1997) 309–351.
[45] Wilfrid Hodges, A Shorter Model Theory, Cambridge University Press, New York, NY, USA, 1997.
[46] Jinbo Huang, Compactness and its implications for qualitative spatial and temporal reasoning, in: Proc. 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-2012), 2012.
[47] Jinbo Huang, Jason Jingshi Li, Jochen Renz, Decomposition and tractability in qualitative spatial and temporal reasoning, Artif. Intell. 195 (2013) 140–164.
[48] Semba Ichiro, An efficient algorithm for generating all partitions of the set $\{1, 2, \ldots, n\}$, J. Inf. Process. 7 (1) (1984) 41–42.
[49] Russell Impagliazzo, Ramamohan Paturi, Francis Zane, Which problems have strongly exponential complexity?, J. Comput. Syst. Sci. 63 (2001) 512–530.
[50] Peter Jonsson, Constants and finite unary relations in qualitative constraint reasoning, Artif. Intell. 257 (2018) 1–23.
[51] Peter Jonsson, Victor Lagerkvist, George Osipov, Acyclic orders, partition schemes and CSPs: unified hardness proofs and improved algorithms, Artif. Intell. 296 (2021) 103505.
[52] Michael Kompatscher, Trung Van Pham, A complexity dichotomy for poset constraint satisfaction, IfCoLog J. Log. Appl. 5 (8) (2018) 1663–1696.
[53] Andrei A. Krokhin, Peter Jeavons, Peter Jonsson, Constraint satisfaction problems on intervals and length, SIAM J. Discrete Math. 17 (3) (2004) 453–477.
[54] Leslie Lamport, The mutual exclusion problem: part I - a theory of interprocess communication, J. ACM 33 (2) (1986) 313–326.
[55] Jason Jingshi Li, Jinbo Huang, Jochen Renz, A divide-and-conquer approach for solving interval algebra networks, in: Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI-2009), 2009, pp. 572–577.
[56] Jason Jingshi Li, Tomasz Kowalski, Jochen Renz, Sanjiang Li, Combining binary constraint networks in qualitative reasoning, in: Proc. 18th European Conference on Artificial Intelligence (ECAI-2008), 2008, pp. 515–519.
[57] Gérard Ligozat, Reasoning about cardinal directions, J. Vis. Lang. Comput. 9 (1) (1998) 23–44.
[58] Daniel Lokshtanov, Dániel Marx, Saket Saurabh, Slightly superexponential parameterized problems, SIAM J. Comput. 47 (3) (2018) 675–702.
[59] Carsten Lutz, Maja Miličić, A tableau algorithm for description logics with concrete domains and general tboxes, J. Autom. Reason. 38 (1–3) (2007) 227–259.
[60] Dugald Macpherson, A survey of homogeneous structures, Discrete Math. 311 (15) (2011) 1599–1634.
[61] Dániel Marx, Can you beat treewidth?, Theory Comput. 6 (1) (2010) 85–112.
[62] Rolf H. Möhring, Martin Skutella, Frederik Stork, Scheduling with AND/OR precedence constraints, SIAM J. Comput. 33 (2) (2004) 393–415.
[63] Reinhard Moratz, Representing relative direction as a binary relation of oriented points, in: Proc. 17th European Conference on Artificial Intelligence (ECAI-2006), vol. 141, 2006, pp. 407–411.
[64] Bernhard Nebel, Hans-Jürgen Bürckert, Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra, J. ACM 42 (1) (1995) 43–66.

[65] Meei Pyng Ng, Mike A. Steel, Nicholas C. Wormald, The difficulty of constructing a leaf-labelled tree including or avoiding given subtrees, Discrete Appl. Math. 98 (3) (2000) 227–235.
[66] Rolf Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006.
[67] Angelo Oddi, Amedeo Cesta, Incremental forward checking for the disjunctive temporal problem, in: Proc. 14th European Conference on Artificial Intelligence (ECAI-2000), 2000, pp. 108–112.
[68] Itsik Pe'er, Ron Shamir, Satisfiability problems on intervals and unit intervals, Theor. Comput. Sci. 175 (2) (1997) 349–372.
[69] Marco Ragni, Stefan Wölfl, Branching Allen, in: Proc. Spatial Cognition IV: Reasoning, Action, Interaction, International Conference Spatial Cognition 2004, 2004, pp. 323–343.
[70] David Randell, Zhan Cui, Anthony Cohn, A spatial logic based on regions and connection, in: Proc. 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR-1992), 1992, pp. 165–176.
[71] Jochen Renz, Bernhard Nebel, On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus, Artif. Intell. 108 (1–2) (1999) 69–123.
[72] Neil Robertson, Paul D. Seymour, Graph minors. III. Planar tree-width, J. Comb. Theory, Ser. B 36 (1) (1984) 49–64.
[73] Francesca Rossi, Peter van Beek, Toby Walsh (Eds.), Handbook of Constraint Programming, Elsevier, 2006.
[74] Marko Samer, Stefan Szeider, Constraint satisfaction with bounded treewidth revisited, J. Comput. Syst. Sci. 76 (2) (2010) 103–114.
[75] Robert Sedgewick, Permutation generation methods, ACM Comput. Surv. 9 (2) (1977) 137–164.
[76] Michael Steel, The complexity of reconstructing trees from qualitative characters and subtrees, J. Classif. 9 (1992) 91–116.
[77] Peter Stockman, Upper bounds on the time complexity of temporal CSPs, Master's thesis, Linköping University, Department of Computer and Information Science, 2016.
[78] Marc B. Vilain, Henry A. Kautz, Constraint propagation algorithms for temporal reasoning, in: Proc. 5th National Conference on Artificial Intelligence (AAAI-1986), 1986, pp. 377–382.
[79] Tandy Warnow, Computational Phylogenetics: an Introduction to Designing Methods for Phylogeny Estimation, Cambridge University Press, 2017.
[80] Dmitriy Zhuk, A proof of the CSP dichotomy conjecture, J. ACM 67 (5) (2020) 30:1–30:78.