Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Resolving Inconsistencies in Disjunctive Temporal Constraints: a Parameterized Complexity Classification

Konrad K. Dabrowski[1] , **Peter Jonsson**[2] , **Sebastian Ordyniak**[3] , **George Osipov**[2,4]
Jorke M. de Vlas[2]
[1]Newcastle University, UK
[2]Linköping University, Sweden
[3]University of Leeds, UK
[4]Royal Holloway, University of London, UK
konrad.dabrowski@newcastle.ac.uk, {peter.jonsson, george.osipov, jorke.de.vlas}@liu.se,
sordyniak@gmail.com

Abstract

The simple temporal problem (STP) and its generalization allowing disjunctive constraints (DTP) are some of the most influential reasoning formalisms for temporal information in AI. We study the problem of resolving inconsistency of data encoded in the DTP, i.e. given a DTP instance, find the minimum number of constraints to remove to make it satisfiable. While this problem is NP-hard in general, it is reasonable to assume that the amount of erroneous data will be small in practical instances. We therefore study the parameterized complexity of this problem parameterized by the number of constraints to be removed to achieve satisfiability, and obtain full P/NP-hard and FPT/W[1]-hard dichotomies for all binary DTP languages.

1 Introduction

The *simple temporal problem* (STP) by Dechter et al. (1991) is one of the most influential AI formalisms: a *long* list of work rooted in the STP can be found in (Hunsberger and Posenato 2021). The STP is a *constraint satisfaction problem* (CSP) with constraints $a \leq x_1 - x_2 \leq b$, where $a, b \in \mathbb{Q} \cup \{-\infty, \infty\}$ and x_1, x_2 are variables with domain $\mathbb{Q}$. A well-known observation is that, by scaling coefficients, we may focus on the case where all numerical bounds are integers (Tsamardinos and Pollack 2003). A direct consequence of this is that we only need to consider integer solutions (Dechter, Meiri, and Pearl 1991, Section 3). We thus define

$$R_{a,b} = \{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 - x_2 \in [a, b]\}$$

where $a, b \in \mathbb{Z} \cup \{-\infty, \infty\}$ and $[a, b]$ is the set $\{a, a + 1, \ldots, b-1, b\}$ where $a \leq b$. We denote the set of such relations by $\mathbf{S}$ and let CSP($\mathbf{S}$) denote the CSP problem restricted to relations in $\mathbf{S}$. The relations in $\mathbf{S}$ appear in all formulations of the STP, but there are more liberal variants containing unary relations $a \leq x \leq b$, the disequality relation, or strict inequalities (see, for instance, (Gerevini and Cristani 1997; Jonsson and Bäckström 1998; Koubarakis 1992)). The CSP for these problems is tractable just like for the basic STP. More general variants using disjunctions are frequently found in the literature, and the corresponding CSPs are typically NP-hard. Examples include *temporal constraint satisfaction problems* (TCSP) (Dechter, Meiri, and Pearl 1991), *disjunctive temporal problems* (DTP) (Tsamardinos and Pollack 2003), and variations proposed by Barber (2000), Oddi & Cesta (2000), Stergiou & Koubarakis (2000) and others. In this context, the following class of relations (that we denote by $\mathbf{T}$) is of particular interest:

$$\{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 - x_2 \in [a_1, b_1] \cup \cdots \cup [a_n, b_n]\}$$

where $n < \infty$ and $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{Z} \cup \{-\infty, \infty\}$. It is no restriction that we work over the integers since these relations are disjunctions of relations in $\mathbf{S}$: merely note that for any satisfiable instance, we can pick any solution and replace each constraint with the STP atom that the solution satisfies. The class $\mathbf{T}$ was introduced by Dechter et al. (1991) and its relations appear in virtually all disjunctive generalizations of the STP. Since the phrase "temporal constraint satisfaction problem" is overloaded with various meanings, we refer to this set of relations as 2-DTP, i.e. DTP restricted to binary relations.

We focus on the problem of resolving inconsistency of data encoded in 2-DTP. Real-world datasets often contain erroneous data due to contradictory sources of information, noisy measurements, human mistakes etc., see (Condotta, Nouaouri, and Sioutis 2016; Chomicki and Marcinkowski 2005; Bertossi and Chomicki 2004). Thus, it is a natural computational problem to identify a maximum-size consistent subset of the data: this is the *maximum constraint satisfaction problem* (MaxCSP). A widespread variant of this problem is MaxSat, where Boolean CNF formulas are considered (Bacchus, Järvisalo, and Martins 2021). MaxCSP (and its generalization *valued* CSP) has also been intensively studied in the finite-domain regime (Krokhin and Zivný 2017). In AI, MaxCSP is often considered in (infinite-domain) spatial and temporal reasoning (Paparrizou and Sioutis 2025; Condotta, Nouaouri, and Sioutis 2016; Condotta et al. 2016). STP and DTP have been studied with respect to *inconsistency measures*, i.e. the degree of conflict or contradiction present in a knowledge base (Condotta and Salhi 2025). Obviously, MaxCSP is an inconsistency measure and it is an example of an *irreducible conflict* measure.

Condotta & Salhi analyze various applications, perform empirical evaluations, and point out that assessing the complexity of such measures is a relevant problem — this paper may be viewed as a contribution to this research program.

It is a consequence of (Dabrowski et al. 2022, Theorem 18) that MAXCSP for the STP is NP-hard for all sets of allowed constraints except for a class of trivially solvable constraints. We will see (in Section 5) that a similar situation holds for DTP. However, the quest for efficient algorithms is not hopeless: under the reasonable assumption that the amount of erroneous data is small, we can employ tools from *parameterized complexity* (see Section 2.3) to design efficient algorithms. Here, the input is equipped with a parameter $k \in \mathbb{N}$, describing some property expected to be small in applications. We want to isolate the non-polynomial runtime to the parameter and aim for a *fixed-parameter tractable* (FPT) algorithm running in $f(k) \cdot n^{O(1)}$ time, where $f : \mathbb{N} \to \mathbb{N}$ is some computable function. Proving that a problem is not in FPT is typically done by showing W[1]-*hardness*: this rules out the existence of an FPT algorithm under standard assumptions.

In our case, the natural parameter is the *number of deleted constraints*, so we make the standard switch to the dual problem MINCSP, where we remove the minimum number of constraints in order to achieve consistency. Our main results are complexity classifications of MINCSP($\mathbf{A}$) for subsets $\mathbf{A} \subseteq \mathbf{T}$. The fact that the relations in $\mathbf{T}$ are based on integers enables us to use results by Bodirsky et al. (2018) concerning *discrete temporal CSPs*, i.e. the CSP problem for languages whose relations are first-order definable in the integers with order $(\mathbb{Z}; <)$. We adapt this result by digging deeper into the algebraic properties of 2-DTP relations, leading to a manageable case distinction: for every finite subset $\mathbf{A} \subseteq \mathbf{T}$, either CSP($\mathbf{A}$) is NP-hard (so MINCSP($\mathbf{A}$) cannot be in FPT unless P = NP) or we are in one of five comprehensible cases. We give an informal description.

(1). A has finite domain. Consider the 2-DTP relation $R(x, y) \equiv x - y \in \{-1, 1\}$. A satisfiable instance of CSP(R) has a solution using values 0 and 1: take a satisfying assignment and replace even values by 0 and odd values by 1. An operation transforming an arbitrary solution into one with finite range is a *finite retraction*. If $\mathbf{A}$ admits a finite retraction, P vs NP dichotomies then follow from (Bulatov 2017) & (Zhuk 2020) (for CSP($\mathbf{A}$)) and (Thapper and Zivný 2016) (for MINCSP($\mathbf{A}$)). The parameterized complexity for finite-domain MINCSP remains a wide open question beyond the scope of this study, and we must leave our results conditional on its eventual resolution. Clearly, a finite domain is against the spirit of classical temporal reasoning on an infinite timeline, and common relations such as $x < y$ and $x - y = 1$ are obstructions to finite retractions.

(2). A is a simple temporal problem. The complexity results follow from (Dabrowski et al. 2022).

(3). A is in the point algebra. The *point algebra* is the set of binary relations $\mathbf{PA} = \{<, \leq, =, \neq\}$ (Vilain and Kautz 1986). Note that $\mathbf{PA} \subseteq \mathbf{T}$ (e.g. $x \neq y$ is equivalent to $x - y \in (-\infty, -1] \cup [1, \infty)$). The complexity results follow from (Osipov, Pilipczuk, and Wahlström 2024).

(4). A is an AP-language. We say that a relation $x - y \in A$ is an *AP-relation* if A is a finite arithmetic progression $\{a, a + d, \ldots, a + \ell \cdot d\}$ for some *length* $\ell \geq 1$ and *difference* d. An *AP-language* is a 2-DTP language comprised of AP-relations with the same difference. We show that an AP-language $\mathbf{A}$ either admits a finite retraction (so case **(1)** applies), every relation is an equation (and both cases **(2)** and **(5)** apply), or MINCSP($\mathbf{A}$) admits a (cost-preserving) reduction from MINCSP(R), where $R(x, y) \equiv x - y \in A$ for some $A \subseteq \mathbb{Z}^{>0}$ with $|A| \geq 2$. We then show that MINCSP(R) is NP-hard and W[1]-hard. Thus, AP-languages do not contain any new islands of tractability.

(5). A contains equations and disequations only. The *successor* $succ$ and *not-successor* $\neg succ$ relations are defined as $x - y = 1$ and $x - y \neq 1$, respectively. Generalizations of them are *equations* and *disequations* $succ^p$ and $\neg succ^p$ defined as $x - y = p$ and $x - y \neq p$, respectively. Let $\mathbf{A} \subseteq \{succ^p, \neg succ^p : p \in \mathbb{N}\}$. We note that MINCSP($\{succ^0\}$) and MINCSP($\{succ^0, \neg succ^0\}$) encode well-known NP-hard graph problems GROUP FEEDBACK EDGE SET (Guillemot 2011) and EDGE MULTICUT (Costa, Létocart, and Roupin 2005), respectively. With this in mind, one can show that either MINCSP($\mathbf{A}$) is in P or is NP-hard. On the other hand, both graph problems are in FPT and the known algorithms use different techniques (cf. LP-based branching (Guillemot 2011; Cygan, Pilipczuk, and Pilipczuk 2016; Iwata, Wahlström, and Yoshida 2016) vs. shadow removal (Marx and Razgon 2011; Bousquet, Daligault, and Thomassé 2011)). Our main algorithmic contribution shows that MINCSP($\mathbf{A}$) is in FPT, via a unified algorithm. Our algorithm begins with a preprocessing step that combines iterative compression, homogenization and a branching step to turn the problem into a simpler variant. We then continue with a second, more involved branching step which allows us to reduce the remaining problem to PAIR PARTITION CUT which was recently shown to be FPT (Dabrowski et al. 2025).

The paper is structured as follows. Section 2 provides preliminaries on constraint satisfaction and parameterized complexity. We present the earlier results from (Dabrowski et al. 2022) and (Osipov, Pilipczuk, and Wahlström 2024) together with an overview of our generalization in Section 3. Our technical contributions are collected in Sections 4–6 where we first prove the case distinction. Based on this, we present classifications of classical and parameterized complexity in Sections 5 and 6, respectively. We conclude the paper in Section 7, where we discuss future research directions and vexatious barriers for further progress.

2 Preliminaries

2.1 Constraint Satisfaction

We begin with some terminology. A *(relational) signature* τ is a set of symbols, each with an associated natural number called *arity*. A *(relational)* τ*-structure* $\mathbf{A}$ is a set D (the *domain*) and relations $R^{\mathbf{A}} \subseteq D^k$ for each k-ary symbol $R \in \tau$. We say that $\mathbf{A}$ has arity a if every relation in $\mathbf{A}$ has arity at most a. 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 structures as sets and, for instance, write expressions like $R \in \mathbf{A}$.

Let $\mathbf{A}$ be a τ-structure. First-order formulas over $\mathbf{A}$ are defined in the standard way with relation symbols from τ. Such formulas 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 parameter-free, i.e. we do not allow the use of domain elements within them. We say that a structure $\mathbf{A}$ is a *(first-order) reduct* of a structure $\mathbf{B}$ if it has the same domain as $\mathbf{B}$ and every relation $R^{\mathbf{A}}$ is first-order definable in $\mathbf{B}$.

Let $\mathbf{A}$ be a τ-structure defined on a set D of values. We define the *constraint satisfaction problem* over $\mathbf{A}$ (CSP($\mathbf{A}$)).

CSP($\mathbf{A}$)	
Input:	A tuple (V, C), where V is a set of variables and C is a multiset of constraints of the form $R(v_1, \ldots, v_a)$, where a is the arity of R, $v_1, \ldots, v_a \in V$, and $R \in \mathbf{A}$.
Question:	Is there a function $f : V \to D$ such that $(f(v_1), \ldots, f(v_a)) \in R$ for every $R(v_1, \ldots, v_a) \in C$?

The structure $\mathbf{A}$ is often referred to as a *constraint language* and the function f is a *satisfying assignment* or a *solution*. CSPs are frequently viewed as homomorphism problems. Let $\mathbf{A}, \mathbf{B}$ be two τ-structures with domains A, B, respectively. A *homomorphism* $h : A \to B$ is a function such that if $R \in \tau$ has arity k and $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$, then $(h(a_1), \ldots, h(a_k)) \in R^{\mathbf{B}}$. Thus, CSP($\mathbf{B}$) is the computational problem of deciding whether a given finite τ-structure $\mathbf{A}$ maps homomorphically to $\mathbf{B}$, and we can view CSP($\mathbf{B}$) as the class of finite τ-structures that map homomorphically to $\mathbf{B}$. We say that CSP($\mathbf{A}$) *equals* CSP($\mathbf{B}$) if CSP($\mathbf{A}$)=CSP($\mathbf{B}$) (when the problems are viewed as classes).

Let us now consider the case when constraints can be soft (i.e. deletable at unit cost) and crisp (i.e. undeletable). We study the following computational problem.

MinCSP($\mathbf{A}$)	
Input:	An instance (I, k) where $I = (V, C)$ is an instance of CSP($\mathbf{A}$) and the constraints in C are each either *soft* or *crisp*, and an integer k.
Parameter:	The integer k.
Question:	Is there a set $Z \subseteq C$ of at most k soft constraints such that the instance $I' := (V, C \setminus Z)$ is satisfiable?

We use crisp constraints merely for convenience since they can be modelled by $k+1$ copies of the same soft constraints (assuming that the constraints form a multiset). Given an instance $((V, C), k)$ of MinCSP($\mathbf{A}$), the deletion set Z can be computed with $|C|$ calls to an algorithm for MinCSP($\mathbf{A}$). We can thus safely view MinCSP($\mathbf{A}$) as a decision problem. We use crisp constraints merely for convenience since they can be modelled by $k+1$ copies of the same soft constraints (assuming that the constraints form a multiset). Given an instance $((V, C), k)$ of MinCSP($\mathbf{A}$), the deletion set Z can be computed with $|C|$ calls to an algorithm for MinCSP($\mathbf{A}$). We can thus safely view MinCSP($\mathbf{A}$) as a decision problem.

2.2 Relations

To aid readability, we may denote the constraint $R_{a,b}(x, y)$ by $a \leq x - y \leq b$. Furthermore, if $a = -\infty$, we may instead write $x - y \leq b$, if $b = \infty$, we may write $x - y \geq a$, and if $a = b$, we may write $x - y = a$. We adopt the convention that $\pm\infty + c = \pm\infty$ and $\pm\infty \cdot c = \pm\infty$ for all $c \in \mathbb{Q}^{>0}$. Given a 2-DTP relation

$$R = \{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 - x_2 \in [a_1, b_1] \cup \cdots \cup [a_n, b_n]\},$$

we let the set of *allowed distances* of R be the set $A(R) = [a_1, b_1] \cup \cdots \cup [a_n, b_n]$. 2-DTP relations can be exactly characterized by definitions in $(\mathbb{Z}; <)$.

Lemma 1. *A relation R is a 2-DTP relation if and only if it is binary and first-order definable in $(\mathbb{Z}; <)$.*

Proof. The forward direction is straightforward to verify. Two examples are

- $x - y \in \{-1, 1\} \equiv (x < y \wedge \neg\exists z\,(x < z \wedge z < y)) \vee (y < x \wedge \neg\exists z(y < z \wedge z < x))$ and
- $x - y \leq 3 \equiv \neg\exists z_1, z_2, z_3(y < z_1 \wedge z_1 < z_2 \wedge z_2 < z_3 \wedge z_3 < x)$.

For the other direction, let $\mathbf{E}$ be the structure with domain $\mathbb{Z}$ and relations $R_c = \{(x, y) \in \mathbb{Z}^2 \mid y \leq x + c\}$ for $c \in \mathbb{Z}$. The structure $(\mathbb{Z}; <)$ admits quantifier elimination in the first-order language over $\mathbf{E}$ (Bodirsky, Martin, and Mottet 2018, pages 9:4–9:5), i.e. if a relation is first-order definable in $(\mathbb{Z}; <)$, then it is first-order definable in $\mathbf{E}$ without using any quantifiers (and thus no fresh variables can be introduced). Thus, consider a binary relation R that is first-order definable in $(\mathbb{Z}; <)$. It is definable by a quantifier-free formula $\phi(x, y)$ that we (without loss of generality) may assume is in disjunctive normal form (i.e. a logical formula consisting of a disjunction of conjunctions). If we look at a disjunct $R_{c_1}(x, y) \wedge R_{c_2}(x, y) \wedge \cdots \wedge R_{c_q}(x, y)$ in ϕ, then it is easy to verify that this conjunction of constraints defines an interval $[a, b]$ with $a, b \in \mathbb{Z} \cup \{-\infty, \infty\}$. Since the formula ϕ is finite, it follows that R is defined by a finite union of intervals so it is a 2-DTP relation. □

We use *polymorphisms* as a tool for studying structural features of relations and constraint languages. They are, for instance, one of the building blocks in the *algebraic approach* for analyzing the complexity of CSPs (Barto, Krokhin, and Willard 2017; Bodirsky 2021). We say that a function $f : A^n \to A$ *preserves* a relation $R \subseteq A^m$ if for all $t_1, \ldots, t_n \in R$ the tuple $f(t_1, \ldots, t_n)$ obtained by applying f componentwise to the tuples $t_1, \ldots, t_n$ is also in R. The function f is referred to as a *polymorphism* of R. The definition extends naturally to structures: a polymorphism of a structure $\mathbf{A}$ with domain D is a function from D^n to D which preserves all relations in $\mathbf{A}$. We exemplify this by the equation relation $R_{a,a}$ for some $a \in \mathbb{Z}$. Define $g : \mathbb{Z}^3 \to \mathbb{Z}$ as $g(x, y, z) = x - y + z$. Arbitrarily choose three tuples $(c_1, d_1), (c_2, d_2), (c_3, d_3) \in R_{a,a}$ and apply the function g.

$$\begin{array}{lrcll} & c_1 & - & d_1 & = a \\ & c_2 & - & d_2 & = a \\ g & c_3 & - & d_3 & = a \\ \hline & (c_1 - c_2 + c_3) & - & (d_1 - d_2 + d_3) & = a \end{array}$$

The final equality follows from reordering the terms:

$$(c_1 - c_2 + c_3) - (d_1 + d_2 + d_3) =$$
$$(c_1 - d_1) - (c_2 - d_2) + (c_3 - d_3) = a - a + a = a.$$

Since the tuples $(c_1, d_1), (c_2, d_2), (c_3, d_3)$ were picked arbitrarily, it follows that f preserves $R_{a,a}$.

2.3 Parameterized Complexity

In parameterized complexity (Downey and Fellows 1999; Flum and Grohe 2006) the running time of algorithms is measured in terms of a parameter $k \in \mathbb{N}$ as well as the input size n. Thus, a *parameterized* problem is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is the input alphabet. The most favourable parameterized complexity class is FPT, which consists of all *fixed-parameter tractable* problems, i.e. problems decidable in $f(k) \cdot n^{O(1)}$ time, where f is a computable function. Observe that if MINCSP($\mathbf{A}$) is FPT, then CSP($\mathbf{A}$) must be polynomial-time solvable: an instance I of CSP($\mathbf{A}$) is satisfiable if and only if the instance $(I, 0)$ of MINCSP($\mathbf{A}$) is a yes-instance, and instances of the form $(I, 0)$ are solvable in polynomial time if MINCSP($\mathbf{A}$) is in FPT.

Reductions between parameterized problems need to take the parameter into account. To this end, we use FPT-*reductions*. Consider two parameterized problems $L_1, L_2 \subseteq \Sigma^* \times \mathbb{N}$. A mapping $P : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ is an FPT-reduction from L_1 to L_2 if

(1) $(x, k) \in L_1$ if and only if $P((x, k)) \in L_2$,

(2) the mapping can be computed in $f(k) \cdot n^{O(1)}$ time for some computable function f, and

(3) there is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$, if $(x', k') = P((x, k))$, then $k' \leq g(k)$.

The class W[1] contains all problems that are fpt-reducible to INDEPENDENT SET parameterized by the number of vertices in the independent set. Showing W[1]-hardness (by an fpt-reduction) for a problem rules out the existence of an fpt algorithm under the standard assumption that FPT $\neq$ W[1].

A gadget reduction that is useful when studying MINCSPs is *implementations* (Khanna et al. 2000). A constraint language $\mathbf{A}$ implements a binary relation R if there exists an instance I of CSP($\mathbf{A}$) with distinguished variables x_1, x_2 such that (1) for every tuple $(a_1, a_2) \in R$, there is a satisfying assignment to I that maps $(x_1, x_2) \mapsto (a_1, a_2)$, and (2) for every pair of values $(a_1, a_2) \notin R$, the minimum number of constraints violated by an assignment to I that maps $(x_1, x_2) \mapsto (a_1, a_2)$ is exactly 1. For instance, if $\mathbf{A}$ contains a relation $x_1 - x_2 = 1$, then it implements the relation $x_1 - x_2 = 2$ via the instance $\{x_1 - y = 1, y - x_2 = 1\}$ where y is a fresh variable. This exemplifies the *composition* $R \circ S$ of two relations R, S, defined on variables x, y by $\{R(x, z), S(z, y)\}$ where z is a fresh variable. It follows by definition that if $\mathbf{A}$ implements R, then MINCSP($\mathbf{A} \cup \{R\}$) FPT-reduces to MINCSP($\mathbf{A}$) by replacing every R-constraint with its implementation using fresh variables.

3 Summary of Classification Results

We will generalize two classification results from the literature: one by Dabrowski et al. (2022) and one by Osipov et al. (2024). We first state the result by Dabrowski et al. Let $\mathbf{S}_0 = \{R_{a,b} \in \mathbf{S} \mid a \leq 0 \text{ and } b \geq 0\}$, i.e. the set of relations that contain the tuple $(0, 0)$. Every instance of CSP($\mathbf{S}_0$) is satisfiable by setting all variables to 0, so all instances of MINCSP($\mathbf{S}_0$) are yes-instances. The *one-sided* relations are $\mathbf{S}_\pm = \{R_{a,\infty}, R_{-\infty,-a} \mid a \in \mathbb{Z}^{\geq 0}\} \cup \{R_{0,0}\}$ and the *equation relations* are $\mathbf{S}_= = \{R_{a,a} \mid a \in \mathbb{Z}\}$.

Theorem 2 (Dabrowski et al. (2022))**.** *Let $\boldsymbol{A} \subseteq \boldsymbol{S}$.*

1. *If $\boldsymbol{A} \subseteq \boldsymbol{S}_0$, then* MINCSP($\boldsymbol{A}$) *is in* P.
2. *If $\boldsymbol{A} \not\subseteq \boldsymbol{S}_0$ and $\boldsymbol{A} \subseteq \boldsymbol{S}_\pm$, then* MINCSP($\boldsymbol{A}$) *is* NP*-hard and it is in* FPT.
3. *If $\boldsymbol{A} \not\subseteq \boldsymbol{S}_0$ and $\boldsymbol{A} \subseteq \boldsymbol{S}_=$, then* MINCSP($\boldsymbol{A}$) *is* NP*-hard and it is in* FPT.
4. *Otherwise,* MINCSP($\boldsymbol{A}$) *is* NP*-hard and* W[1]*-hard.*

We continue with the result by Osipov et al. The *point algebra* **PA** (Vilain and Kautz 1986) is the structure of binary relations first-order definable in $(\mathbb{Q}; <)$, i.e. **PA** contains the ordering relations $\{<, \leq, =, \neq\}$ and their inverses. The CSP(**PA**) problem is in P (ibid.) and a solvable instance always has an integer solution. We can thus view the relations in **PA** as members of **T**, and we see that **PA** and **S** are incomparable sets: separating relations are, for instance, $\neq$ and $x + y = 1$. For an infinite-domain language $\mathbf{A}$, we say that CSP($\mathbf{A}$) is *trivial* if all instances of CSP($\mathbf{A}$) are satisfied by mapping all variables to the same value or mapping all variables to distinct values. For example, CSP($\mathbf{S}_0$) is trivial.

Theorem 3 (Osipov et al. (2024))**.** *Let $\boldsymbol{A} \subseteq$ **PA**. If $\boldsymbol{A} \subseteq \{=, \leq\}$ or $\boldsymbol{A} \subseteq \{\neq\}$, then* MINCSP($\boldsymbol{A}$) *is trivial. Otherwise,* MINCSP($\boldsymbol{A}$) *is* NP*-hard and the following holds.*

1. *If $\boldsymbol{A} \subseteq \{<, \leq, =\}$, then* MINCSP($\boldsymbol{A}$) *is in* FPT.
2. *If $\boldsymbol{A} \subseteq \{<, =, \neq\}$, then* MINCSP($\boldsymbol{A}$) *is in* FPT.
3. *Otherwise,* MINCSP($\boldsymbol{A}$) *is* W[1]*-hard.*

We now describe our generalization of Theorems 2 and 3. The dividing lines in the dichotomies of **S** and **PA** are easy to define, but it is more difficult for **T**. Let $\mathbf{A}$ be a structure with domain A and pick $B \subseteq A$. The *substructure* $\mathbf{A}[B]$ induced by B has domain B and a relation $R^{\mathbf{A}[B]}$ for every $R^{\mathbf{A}}$ in $\mathbf{A}$ obtained as $R^{\mathbf{A}[B]} = R^{\mathbf{A}} \cap B^r$, where r is the arity of R. A mapping $h : A \to B$ is a *finite retraction* if B is finite, h is a homomorphism from $\mathbf{A}$ to $\mathbf{A}[B]$, and $h(b) = b$ for all $b \in B$. If $\mathbf{A}$ admits a finite retraction to $\mathbf{A}[B]$, then CSP($\mathbf{A}$)=CSP($\mathbf{A}[B]$): indeed, every structure that homomorphically maps to $\mathbf{A}$ also maps to $\mathbf{A}[B]$ via h, and every structure that homomorphically maps to $\mathbf{A}[B]$ also maps to $\mathbf{A}$ via the inclusion map. One example is the structure $(\mathbb{Z}; R)$ with $R(x, y) \equiv x - y \in \{-1, 1\}$ that we discussed in Section 1: here, the finite domain can be chosen to be $\{0, 1\}$.

By combining the forthcoming Theorems 14 and 21, we arrive at the following classification result. The various cases are described in greater detail in the underlying results.

Theorem 4. *Let $\boldsymbol{A} \subseteq \boldsymbol{T}$ be finite. Then* MINCSP($\boldsymbol{A}$) *is either in* P *or* NP*-hard. If $\boldsymbol{A}$ does not admit a finite retraction, then* MINCSP($\boldsymbol{A}$) *is either in* FPT *or* W[1]*-hard.*

4 P vs NP-hard Dichotomy for CSP

Bodirsky et al. (2018) prove that CSP($\mathbf{A}$) exhibits a P vs. NP-hard dichotomy for finite reducts $\mathbf{A}$ of $(\mathbb{Z}; <)$. Finite 2-DTP languages are finite reducts of $(\mathbb{Z}; <)$ (as discussed in Section 2.2) and also exhibit a dichotomy. This section is divided into two subsections, where we first state Bodirsky et al.'s result and then refine it for 2-DTP languages.

4.1 Reducts of $(\mathbb{Z}; <)$

We begin with some definitions. Let $succ$ denote the *successor* relation $\{(x, y) \in \mathbb{Z} \mid y = x+1\}$. We write $succ^p$ to denote the relation $\{(x, y) \in \mathbb{Z} \mid y = x + p\}$. Note that $succ^p$ is implementable in $\{succ\}$ by composing $succ$ with itself $p - 1$ times. A quantifier-free formula in conjunctive normal form over $succ$ is *Horn* if each clause contains at most one literal of the form $succ^p(x, y)$ while every other literal is of the form $\neg succ^p(x, y)$. A relation is *Horn-definable* in $(\mathbb{Z}; succ)$ if there exists a Horn formula that defines the relation in $(\mathbb{Z}; succ)$. Recall the set $\mathbf{S}_= = \{R_{a,a} \mid a \in \mathbb{Z}\}$ from Section 3. Note that $x - y = -p \Leftrightarrow y - x = p$, so we may assume that $\mathbf{S}_= = \{succ^p \mid p \geq 0\}$. Analogously, we define $\mathbf{S}_{=,\neq} = \{succ^p, \neg succ^p \mid p \geq 0\}$ and note that the relations in $\mathbf{S}_{=,\neq}$ are Horn-definable in $(\mathbb{Z}; succ)$.

We finally define a family of polymorphisms that is important in what follows. Let d be a positive integer. The *d-modular* max operation $\max_d : \mathbb{Z}^2 \to \mathbb{Z}$ is defined as

$$\max_d(x, y) = \begin{cases} \max(x, y) & \text{if } x \equiv y \bmod d \\ x & \text{otherwise.} \end{cases}$$

The *d-modular* min operation is defined analogously. Observe that $\max_1 = \max$ and $\min_1 = \min$.

The following result follows from combining Theorem 4.2 with Proposition 6.1 in Bodirsky et al. (2018).

Theorem 5. *Let $\boldsymbol{A}$ be a finite reduct of $(\mathbb{Z}; <)$. Then there exists a structure $\boldsymbol{B}$ such that CSP($\boldsymbol{A}$)=CSP($\boldsymbol{B}$) and at least one of the following cases applies:*

1. *$\boldsymbol{B}$ has a finite domain.*
2. *$\boldsymbol{B}$ is a reduct of $(\mathbb{Q}; <)$.*
3. *$\boldsymbol{B}$ is a reduct of $(\mathbb{Z}; <)$ and $\boldsymbol{B}$ is preserved by* max *or by* min*. In this case, CSP($\boldsymbol{B}$) is in* P*.*
4. *$\boldsymbol{B}$ is a reduct of $(\mathbb{Z}; succ)$ and $\boldsymbol{B}$ is preserved by $\max_d$ or $\min_d$ with $d \geq 2$. In this case, CSP($\boldsymbol{B}$) is in* P*.*
5. *Every relation of $\boldsymbol{B}$ is Horn-definable in $(\mathbb{Z}; succ)$. In this case, CSP($\boldsymbol{B}$) is in* P*.*
6. *CSP($\boldsymbol{B}$) is* NP*-complete.*

4.2 2-DTP Languages

We can extract more information when we only consider 2-DTP relations, and this improved classification is the foundation of our MINCSP analysis. The improved result is partly based on a better understanding of relations preserved by (modular) max or min. We first show that STP relations are characterized by max and/or min.

Lemma 6. *Let $R \in \boldsymbol{T}$. The following are equivalent: (1) R is preserved by* min*, (2) R is preserved by* max*, and (3) R is an* STP *relation.*

Proof. (1) $\Rightarrow$ (3) and (2) $\Rightarrow$ (3). Write A for $A(R)$. Let $a = \min(A)$ and $b = \max(A)$. We show that if R is invariant under min or max, then $A = [a, b]$. Equivalently, if $c \in A$ and $c + r \in A$ for some $r \geq 1$, then $[c, c + r] \subseteq A$. We do induction on r. If $r = 1$, then we are done. If $r \geq 2$, consider tuples $(c + r, 0), (c + 1, 1) \in R$. If R is invariant under min, then applying it to the pair of tuples yields $(c + 1, 0) \in R$ and $c+1 \in A$. Applying the inductive hypothesis to $c + 1$ and $c + r$ yields the result. If R is invariant under max, then $(c + r, 1) \in R$ and $c + r - 1 \in A$, and we apply the inductive hypothesis to c and $c + r - 1$.

(3) $\Rightarrow$ (1) and (3) $\Rightarrow$ (2). Suppose R is an STP relation. Then, $A(R) = [a, b]$ with $a, b \in \mathbb{Z} \cup \{-\infty, \infty\}$. Arbitrarily pick two tuples $s = (s_1, s_2)$ and $t = (t_1, t_2)$ in R. We show that $\max(s, t) \in R$ — the proof for $\min(s, t)$ is analogous. We have $s_1 - s_2 = u_s$ and $t_1 - t_2 = u_t$ for some $u_s, u_t \in A(R)$. Let $u = \max(s_1, t_1) - \max(s_2, t_2)$. We verify that $\min(u_s, u_t) \leq u \leq \max(u_s, u_t)$ and thus that the tuple $\max(s, t)$ is in R. If $\max(s_1, t_1) = s_1$ and $\max(s_2, t_2) = s_2$, then $u = s_1 - s_2 = u_s$. If $\max(s_1, t_1) = s_1$ and $\max(s_2, t_2) = t_2$, then, $u = s_1 - t_2$ and $u_t = t_1 - t_2 \leq s_1 - t_2 \leq s_1 - s_2 = u_s$. The case when $\max(s_1, t_1) = t_1$ is analogous. □

We continue with relations that are preserved by $\max_d$ or $\min_d$ for $d \geq 2$. Let $A^{a,d}_{b,c}$ denote the set of integers $\{a + i \cdot d \mid b \leq i \leq c\}$ and let the 2-DTP relation $R^{a,d}_{b,c}$ satisfy $A(R^{a,d}_{b,c}) = A^{a,d}_{b,c}$. It is easy to verify that $R^{a,d}_{b,c}$ is preserved by both $\max_d$ and $\min_d$. We show a partial converse.

Lemma 7. *Let R be a nontrivial* 2-DTP *relation preserved by $\max_d$ or $\min_d$ for some $d \geq 2$. Then there exist integers a and $\ell \geq 0$ such that $R = R^{a,d}_{0,\ell}$.*

Proof. Let $A = A(R)$, $a = \min(A)$ and $b = \max(A)$. We continue with a series of claims.

Claim (1). *A is finite, i.e. $a > -\infty$ and $b < \infty$.*

Proof of claim: Suppose to the contrary that $b = \infty$ and let $c = \max(\mathbb{Z} \setminus A)$ be the largest value that does not belong to A. R is non-trivial, so $c < \infty$. Note that all integers greater than c are in A. Consider the tuples $(c + d, 0), (c + d + 1, d) \in A$. Applying $\max_d$ to them yields $(c + d, d) \notin R$, so R is not invariant under $\max_d$. Now consider the tuples $(c + 2d, d), (c + d, 1) \in R$. Applying $\min_d$ to them yields $(c + d, d) \notin R$, so R is not invariant under $\min_d$, which is a contradiction. The case when $a = -\infty$ is analogous. ◇

Claim (2). *For all $c, c' \in A$, d divides their difference $c-c'$.*

Proof of claim: We assume the opposite. Suppose R is invariant under $\max_d$. For every integer i, we have $(c,0) \in R$ and $(c+i\cdot d, c+i\cdot d-c') \in R$. By applying $\max_d$ to the tuples, we get $(c+i\cdot d, 0) \in R$ because $c = c+i\cdot d \bmod d$ and $0 \neq c-c'+i\cdot d \bmod d$. But then $c+i\cdot d \in A$ for all i, implying that A is infinite and contradicting Claim 1. If R is invariant under $\min_d$, similar reasoning applies starting from the tuples $(c,0)$ and $(c-i\cdot d, c-i\cdot d-c')$. ⋄

Claim 2 implies that $b-a = \ell\cdot d$ for some $\ell \geq 0$. It remains to show that $a+i\cdot d \in A$ for all $0 < i < \ell$. Suppose R is preserved by $\max_d$. We show that if $a+i\cdot d \in A$, then $a+(i-1)\cdot d \in A$. Indeed, $(a+i\cdot d, 0) \in R$ and $(a+d, d) \in R$, so $(a+i\cdot d, d) \in R$, and $a+(i-1)\cdot d \in A$. The argument in the case when R is preserved by $\min_d$ is similar, starting with the tuples $(a+i\cdot d, 0)$ and $(a+(i-1)\cdot d, (i-1)\cdot d)$. □

We now have all ingredients for a sharper dichotomy result.

Lemma 8. *Let $\mathbf{A} \subseteq \mathbf{T}$ be finite. Then there is a structure $\mathbf{B}$ such that CSP($\mathbf{A}$)=CSP($\mathbf{B}$) and one of the following cases applies:*

1. *$\mathbf{B}$ has a finite domain.*
2. *$\mathbf{B} \subseteq \mathbf{PA}$, and CSP($\mathbf{B}$) is in* P.
3. *$\mathbf{B} \subseteq \mathbf{S}$ and CSP($\mathbf{B}$) is in* P.
4. *$\mathbf{B} \subseteq \mathbf{T}$ is preserved by $\min_d$ and $\max_d$ for some $d \geq 2$ and CSP($\mathbf{B}$) is in* P.
5. *$\mathbf{B} \subseteq \mathbf{S}_{=,\neq}$, and CSP($\mathbf{B}$) is in* P.
6. *CSP($\mathbf{B}$) is* NP*-complete.*

Proof. $\mathbf{A}$ is a first-order reduct of $(\mathbb{Z}; <)$ with finite signature by Lemma 1. Apply Theorem 5 to obtain $\mathbf{B}$. The requirement CSP($\mathbf{A}$)=CSP($\mathbf{B}$) shows that all relations in $\mathbf{B}$ are binary. Note that if $\mathbf{B}$ is a reduct of $(\mathbb{Z}; <)$, then $\mathbf{B} \subseteq \mathbf{T}$ by Lemma 1: the same is true if $\mathbf{B}$ is a reduct of $(\mathbb{Z}; succ)$ since this structure is first-order definable in $(\mathbb{Z}; <)$. We consider the six cases from Theorem 5.

1. $\mathbf{B}$ has a finite domain, so we are in Case 1.
2. $\mathbf{B}$ is a reduct of $(\mathbb{Q}; <)$ and only contains binary relations. Hence, $\mathbf{B} \subseteq \mathbf{PA}$ and we are in Case 2.
3. $\mathbf{B} \subseteq \mathbf{T}$ and is preserved by max or by min. Lemma 6 shows that $\mathbf{B} \subseteq \mathbf{S}$ (since $\mathbf{B} \subseteq \mathbf{T}$) and we are in Case 3.
4. $\mathbf{B} \subseteq \mathbf{T}$ and preserved by $\max_d$ or $\min_d$ for some $d \geq 2$. Arbitrarily choose a relation $R \in \mathbf{B}$. Lemma 7 implies that there exist integers a and $\ell \geq 0$ such that $R = R^{a,d}_{0,\ell}$ and this relation is preserved by $\max_d$ and $\min_d$ (as pointed out before Lemma 7). We conclude that $\mathbf{B}$ is preserved by both $\max_d$ and $\min_d$ and we are in Case 4.
5. Every relation in $\mathbf{B}$ is binary and Horn-definable in $(\mathbb{Z}; succ)$. A Horn-clause over two variables is of the form $x-y = a_0 \vee x-y \neq a_1 \vee \cdots \vee x-y \neq a_r$ or $x-y \neq a_1 \vee \cdots \vee x-y \neq a_r$ where r is allowed to equal 0. It is sufficient to consider the case when $r \leq 1$, since using two or more distinct $\neq$-relations defines a relation that equals $\mathbb{Z}^2$. Note that a relation $(x-y = a_0) \vee (x-y \neq a_1)$ is equal to one of $x-y = a_0$, $x-y \neq a_1$, or $\mathbb{Z}^2$. The only nontrivial types of clauses are thus $x-y = a_0$ and $x-y \neq a_1$. Furthermore, a conjunction of such clauses is either trivial or equal to $x-y = a_0$ or $x-y \neq a_1$, so we are in Case 5.
6. CSP($\mathbf{B}$) is NP-complete, so we are in Case 6. □

5 P vs NP-hard Dichotomy for MinCSP

We turn our attention to the computational complexity of MinCSP over 2-DTP languages. The proof is based on Lemma 8, and we start with the $\max_d$ / $\min_d$ cases. Define $m_d : \mathbb{Z} \to \{0, \ldots, d-1\}$ such that $m_d(x) = (x \bmod d)$, i.e. $m_d(x)$ is the unique integer in $\{0, \ldots, d-1\}$ congruent to x modulo d.

Lemma 9. *Consider $A := A^{a,d}_{0,\ell}$ for $a, \ell, d \in \mathbb{Z}$ with $\ell \geq 0$, $d \geq 1$. If A contains 0 or both negative and positive values, then $R := R^{a,d}_{0,\ell}$ is preserved by the finite retraction m_d.*

Proof. If $0 \in A$, then $a = 0 \bmod d$ and $x-y = 0 \bmod d$ for every $(x,y) \in R$. Then $m_d(x) - m_d(y) = 0 \bmod d$, and since $m_d(x), m_d(y) \in \{0, \ldots, d-1\}$, we have $m_d(x) - m_d(y) = 0$. Suppose $0 \notin A$ and A contains negative and positive values. Let $a' = a \bmod d$. Then, $a' \neq 0$ so A contains $a'-d$ and a'. We claim that $(x,y) \in R$ implies $m_d(x) - m_d(y) \in \{a'-d, a'\} \subseteq A$. Indeed, $(x,y) \in R$ implies that $x-y = a \bmod d$, so $m_d(x) - m_d(y) = a' \bmod d$. Moreover, $m_d(x) - m_d(y) \in [-(d-1), d-1]$, and the only values congruent to a' modulo d in this interval are $a'-d$ and a'. □

Lemma 10. *Let $\mathbf{A} \subseteq \mathbf{T}$ be finite and invariant under $\max_d$ or $\min_d$ for some $d \geq 2$. Then, one of the following applies: (1) $\mathbf{A}$ admits a finite retraction, (2) every relation in $\mathbf{A}$ is an equation, or (3) $\mathbf{A}$ implements a relation $x-y \in A$ such that $|A| \geq 2$ and all values in A are positive.*

Proof. Suppose $\mathbf{A} = (\mathbb{Z}; R_1, R_2, \ldots, R_\ell)$ and let $A_i = A(R_i)$. If for every i, we have $0 \in A_i$ or A_i contains positive and negative values, then each A_i (and $\mathbf{A}$) is preserved by the finite retraction m_d by Lemma 9. If $|A_1| = \cdots = |A_\ell| = 1$, then all relations of $\mathbf{A}$ are equations. Otherwise, we have a set A_i that only contains positive values or only contains negative values, and a set A_j with $|A_j| \geq 2$. Without loss of generality, assume all values in A_i are positive. If $|A_i| > 1$, then we are done. Otherwise, by composing R_i with itself sufficiently many times, we obtain a relation $x-y = b$ such that $b > \max(|a| : a \in A_j)$. Then, composing the latter relation with R_j, we obtain a relation $x-y \in \{a+b : a \in A_j\}$ where $\{a+b : a \in A_j\} \subseteq \mathbb{Z}^{>0}$ has size at least 2. □

We continue with the case when $\mathbf{A} \subseteq \mathbf{S}_{=,\neq}$.

Lemma 11 (Lemma 5 in (Dabrowski et al. 2022))**.** *The problem* MinCSP($R_{a,a}$) *is* NP*-hard for all $a \neq 0$.*

Lemma 12. *Let R be a* 2-DTP *relation such that $A(R)$ is a finite set of positive values. Then* MinCSP(R) *is* NP*-hard.*

Proof. NP-hardness follows by Lemma 11 if $|A| = 1$. If not, let $a = \min(A)$ and $b = \max(A)$. By composing R with itself a times, we get a relation $x-y \in B$ with $\max(B) = ab$. By composing R with itself b times, we get a relation $x-y \in C$ with $\min(C) = ab$. Then

$(x-y \in B) \wedge (x-y \in C) \equiv (x-y = ab)$ and Lemma 11 applies because $ab > 0$. □

Lemma 13. *Let $\mathbf{A} \subseteq \mathbf{S}_{=,\neq}$ be finite. Then either CSP($\mathbf{A}$) is trivial or* MINCSP($\mathbf{A}$) *is* NP*-hard.*

Proof. We consider three cases. **(1).** If $\mathbf{A}$ contains $succ^p$ for some $p > 0$, then MINCSP($\mathbf{A}$) is NP-hard by Lemma 11. **(2).** If $\mathbf{A}$ contains $succ^0$ and $\neg succ^0$, then MINCSP($\mathbf{A}$) is NP-hard (Osipov and Wahlström 2023, Section 3.2). **(3).** Otherwise, $\mathbf{A} \subseteq (\mathbb{Z}; \{=\} \cup \{\neg succ^p\}_{p>0}\})$. Every instance of CSP($\mathbf{A}$) is satisfiable. First, while the instance contains a constraint $x = y$, replace y with x everywhere and delete the constraint. Then pick a large integer $N > p$ for all p such that $\neg succ^p \in \mathbf{A}$, enumerate the variables as $x_1, \dots, x_n$ and let $\alpha(x_i) = i \cdot N$ for all $i \in [n]$. This assignment satisfies every constraint $x_i - x_j \neq p$ since $\alpha(x_i) - \alpha(x_j) = N(i-j)$ is a multiple of N, i.e. a number in $\{\dots, -2N, -N, 0, N, 2N, \dots\}$, and p does not belong to this set by the choice of N. □

We are now ready to prove the classification result.

Theorem 14. *Let $\mathbf{A}$ be a finite subset of $\mathbf{T}$. Then,* MINCSP($\mathbf{A}$) *is in* P *or* NP*-hard. In particular, one of the following holds: (1) $\mathbf{A}$ admits a finite retraction, (2)* CSP($\mathbf{A}$) *is trivial, or (3)* MINCSP($\mathbf{A}$) *is* NP*-hard.*

Proof. By Lemma 8, there is a structure $\mathbf{B}$ such that CSP($\mathbf{A}$)=CSP($\mathbf{B}$) and one of the following six cases holds.

1. $\mathbf{B}$ has a finite domain, i.e. $\mathbf{A}$ admits a finite retraction.
2. $\mathbf{B} \subseteq \mathbf{PA}$ and MINCSP($\mathbf{B}$) is trivial or NP-hard by Theorem 3.
3. $\mathbf{B} \subseteq \mathbf{S}$. Then MINCSP($\mathbf{A}$) is trivial if and only if $\mathbf{B} \subseteq \mathbf{S}_0$ by Theorem 2 and NP-hard otherwise.
4. $\mathbf{B} \subseteq \mathbf{T}$ is preserved by $\min_d$ and $\max_d$ for some $d \geq 2$. By Lemma 10, if $\mathbf{B}$ does not admit a finite retraction (implying that we are in Case 1) or not all relations of $\mathbf{B}$ are equations (and we are in Case 3), then $\mathbf{B}$ implements a relation $x - y \in A$ such that $|A| \geq 2$ and all values in A are positive. By Lemma 12, MINCSP($\mathbf{B}$) is NP-hard.
5. $\mathbf{B} \subseteq \mathbf{S}_{=,\neq}$. Then CSP($\mathbf{B}$) is trivial if $\mathbf{B} \subseteq \mathbf{S}_0$, and otherwise MINCSP($\mathbf{B}$) is NP-hard by Lemma 13.
6. CSP($\mathbf{B}$) (and thus MINCSP($\mathbf{B}$)) is NP-complete. □

If $\mathbf{A}$ admits a finite retraction, then the retracted structure $\mathbf{B}$ has finite domain and MINCSP($\mathbf{B}$) is either in P or NP-hard (Thapper and Zivný 2016). The borderline is described in the article.

6 FPT vs W[1]-hard Dichotomy for MINCSP

We prove the result in two steps. We first prove that MINCSP($\mathbf{S}_{=,\neq}$) is in FPT and then obtain the classification result by, once again, invoking Lemma 8.

6.1 FPT Algorithm for MINCSP($\mathbf{S}_{=,\neq}$)

We first reduce MINCSP($\mathbf{S}_{=,\neq}$) to the following problem.

HOMOGENISED MINCSP($\mathbf{S}_{=,\neq}$) COMPRESSION (HMC)	
Input:	A tuple $((V,C),k,X)$ where $((V,C),k)$ is an instance of MINCSP($\mathbf{S}_{=,\neq}$) without soft $\neg succ^p$ constraints and $X \subseteq C$ is a set such that $\lvert X\rvert \leq k+1$, $I - X$ is satisfiable, and all soft constraints in $I - X$ are $succ^0$ constraints.
Parameter:	k.
Question:	Is there a $Z \subseteq C$ of at most k soft constraints such that $I - Z$ is satisfiable and $X \cap Z = \emptyset$?

Lemma 15. *If HMC is* FPT*, then* MINCSP($\mathbf{S}_{=,\neq}$) *is* FPT*.*

Proof. Arbitrarily pick an instance $I = ((V,C),k)$ of MINCSP($\mathbf{S}_{=,\neq}$). We use *iterative compression* combined with some preprocessing to reduce I to $O(2^{k+1}|C|)$ instances of HMC. We first assume without loss of generality that all $\neg succ^p$ constraints are crisp. To achieve this, we replace every constraint $c = \neg succ^p(x,y)$ by a crisp constraint $\neg succ^p(x,z_c)$ and a soft constraint $succ^0(z_c,y)$, where z_c is a fresh variable.

We now describe our iterative compression step. We begin with an instance without any constraints. We then add the constraints, one by one, and keep track of a solution X with $|X| \leq k$ such that removing X results in a satisfiable instance. When given a new constraint, we first add it to X to maintain the property that removing X results in a satisfiable instance. We then "compress" X (with the help of a reduction to HMC): we use the existence of a solution of size $k+1$ to create a new solution of size k. If this is at some point impossible, then there is no solution to the original instance. Otherwise, we find a solution after we have added all constraints.

Let $X \subseteq C$ with $|X| \leq k+1$ be the set to be compressed and $U = V(X)$ the set of all variables appearing in constraints from X. Note that $|U| \leq 2k+2$. Since removing X results in a satisfiable instance, there is a solution $f_X : V \to \mathbb{Z}$ to $(V, C \setminus X)$ that can be computed in polynomial time (Gerevini and Cristani 1997, Theorem 5). Now let $Z \subseteq C$ be a hypothetical optimal solution; that is, $|Z| \leq k$ and removing Z results in a satisfiable instance. Let $f_Z : V \to \mathbb{Z}$ be a satisfying assignment to $(V, C \setminus Z)$.

We continue with a *homogenization step*. Instead of computing f_Z, we find the difference $f'_Z = f_Z - f_X$ and update all constraints accordingly. That is, we replace every $succ^p$ constraint $u - v = p$ with $(u - f_X(u)) - (v - f_X(v)) = p$ or equivalently $u - v = p + f_X(u) - f_X(v)$ and we replace every $\neg succ^p$ constraint $u - v \neq p$ with $u - v \neq p + f_X(u) - f_X(v)$. As f_X is a solution of $(V, C \setminus X)$, every $succ^p$ constraint $u - v = p$ that is not contained in X becomes $u - v = 0$, and all $succ^p$ constraints outside of X therefore become $succ^0$ constraints.

Finally, we try all options for the intersection $X \cap Z$ and remove these constraints, decreasing the parameter k by $|X \cap Z|$. At the cost of the factor 2^{k+1} in the running time, we can assume without loss of generality that X and Z are

disjoint. For each guess X_I, i.e. a subset of X, we have that $(I', k - |X_I|, X \setminus X_I)$ with $I' = (V, C \setminus X_I)$ is an instance of HMC. Moreover, I has a solution of size at most k if and only if there is an $X_I \subseteq X$ such that $(I', k - |X_I|, X \setminus X_I)$ with $I' = (V, C \setminus X_I)$ has a solution. Therefore, we can solve the compression problem by solving at most 2^{k+1} instances of HMC. □

Next, we provide an FPT algorithm for HMC. Our algorithm is based on a reduction to PAIR PARTITION CUT. A *partition* $\mathcal{P}$ of a finite set N is a family of pairwise disjoint subsets $B_1, \ldots, B_m$ of N such that $\bigcup_{i=1}^{m} B_i = N$. Let G be an undirected graph, T be a subset of its vertices called *terminals*, and $\mathcal{P}$ be a partition of T. A subset of edges X in G is a *$\mathcal{P}$-cut* if no component of $G - X$ contains terminals from more than one subset of $\mathcal{P}$. Given a graph G with a set of terminals $T \subseteq V(G)$, a *pair cut request* is a tuple $(\{s, u\}, \{t, v\})$ where $s, t \in T$ and $u, v \in V(G)$. We say that a path P is an *uv-path* if its endpoints are the vertices u and v. A cut $X \subseteq E(G)$ *fulfils* $(\{s, u\}, \{t, v\})$ if $G - X$ does not contain an su-path or a tv-path.

PAIR PARTITION CUT (PPC)	
Input:	An undirected graph G with positive integer edge weights $w_G : E(G) \to \mathbb{N}^+$, a set of vertices (terminals) $T \subseteq V(G)$, a partition $\mathcal{P}$ of T, a set $\mathcal{F}$ of pair cut requests, and an integer k.
Parameter:	k.
Question:	Is there a $\mathcal{P}$-cut $X \subseteq E(G)$ of total weight at most k that fulfils every pair cut request in $\mathcal{F}$?

Theorem 16 (Theorem 13 in (Dabrowski et al. 2025))**.** PAIR PARTITION CUT *is in* FPT.

We are now ready to show that HMC is in FPT.

Lemma 17. *HMC is in* FPT.

Proof. Let (I, k, X) with $I = (V, C)$ be an instance of HMC and U the variables appearing in X. Suppose that Z is a hypothetical solution for (I, k, X) witnessed by the assignment $f_Z : V \to \mathbb{Z}$.

We first "guess" how the function f_Z partitions U into different sets, i.e. we try all options for a partition $\mathcal{P}_U$ of U, and assume variables in U have the same value w.r.t. f_Z if and only if they are contained in the same part of the guessed partition $\mathcal{P}_U$. This incurs a factor of at most $(2k+2)^{2k+2}$ in the running time, since $|U| \leq 2k + 2$.

We now check whether the guessed partition $\mathcal{P}_U$ is compatible with the constraints in X as follows. Since $\neg succ^p$ constraints are crisp, X only contains $succ^p$ constraints. We partition U into *small components*, components of the graph with vertex set U having an edge uv if X contains a constraint on u and v or if u and v belong to the same part of $\mathcal{P}_U$. We know that

1. the guess implies that f_Z is constant on variables belonging to the same part of $\mathcal{P}_U$, and
2. since $X \cap Z = \emptyset$, f_Z satisfies all constraints in X,

so there is at most one option for f_Z on each small component. If there is no option for some small component, we reach a contradiction in this branch.

Let $U_1, \ldots, U_m$ be the small components of U, let $u_1, \ldots, u_m$ be any variable from each component, and let M be a sufficiently large integer. We claim that, without loss of generality, we may assume that $f_Z(u_i) = iM$, and by extension, determine f_Z on U. To see this, we partition V into hypothetical *big components*, components that are connected by $succ^p$ constraints (outside of Z) or by variables belonging to the same guessed partition group. Each small component is a subset of a big component. Moreover, different small components belong to different big components: if a big component contains a path connecting two small components, then f_Z must be constant on this path since all $succ^p$ constraints outside of X are $succ^0$ constraints, which contradicts the guessed partition. Hence, if we increase f_Z by a constant value on each big component, we still satisfy all $succ^p$ constraints (outside of Z) and all $\neg succ^p$ constraints within the component. By setting $f_Z(u_i) = iM$, we ensure that all $\neg succ^p$ constraints between different components are also still satisfied since M is larger than p.

For the final step of our algorithm, we compute Z by reducing the remaining problem to an instance of PAIR PARTITION CUT, which is solvable in FPT time by Theorem 16. The reduction starts by letting G be the constraint graph of all $succ^0$ constraints outside of X, i.e. G has a vertex for every variable and an edge between two variables if they occur in a $succ^0$ constraint outside of X. We let the set of terminals T be the set U. The partition of $\mathcal{P}$ is the guessed partition $\mathcal{P}_U$ of U. This way, any $\mathcal{P}$-cut is a solution that ignores all $\neg succ^p$ constraints. Now consider some $\neg succ^p$ constraint $u - v \neq p$. Let A_u (resp. A_v) be the set of terminals reachable from u (resp. v) on the graph G. For each pair $s \in A_u$, $t \in A_v$ with $f_Z(s) - f_Z(t) = p$, we add the pair cut request $\{\{s, u\}, \{t, v\}\}$. This completes the reduction.

It is clear that if the instance (I, k, X) of HMC has a solution Z, then the instance of PPC is satisfiable. We now show that if the instance of PPC has a solution Z, then Z is also a solution to (I, k, X) by showing that we can compute an assignment $f_Z : V \to \mathbb{Z}$ that satisfies $I - Z$.

Recall that we know the value for f_Z on U, and - as we now know Z - that the big components are no longer hypothetical, so we can extend f_Z to a value for all big components that contain a small component. Let $V_1, \ldots, V_{m'}$ be the remaining big components. For each V_i, we now set f_Z to $(m + i + 1)M$ on the entire component. Since V_i is disconnected from X, it only contains equality constraints, and these are satisfied by f_Z.

Now consider a $\neg succ^p$ constraint $u - v \neq p$. If u and v are in different big components, then $f_Z(u) - f_Z(v)$ is larger than p since M is large enough. If u and v are in one big component that does not extend a small component, then the inequality must be satisfied, since it would otherwise not be satisfied on $C \setminus X$. Finally, if u and v are in the same big component that does extend a small component, then u and v each reach some vertex from this small component using $succ^0$ constraints, say s and t. If $f_Z(s) - f_Z(t) = p$, then we fail the added pair cut request, and otherwise we satisfy the

$\neg succ^p$ constraint. This completes our construction of f_Z. Since PPC is solvable in FPT time and our pre-processing only adds an $f(k)$-factor, we conclude that HMC is in FPT. □

Lemmas 15 and 17 together show our main result.

Theorem 18. MINCSP($\boldsymbol{S}_{=,\neq}$) *is in* FPT.

6.2 Classification Result

We begin by presenting a hardness result for certain $R^{a,d}_{b,c}$ relations. We combine relation composition with a scaling technique and obtain the hardness result in Lemma 20.

Lemma 19 (Corollary 2 in (Dabrowski et al. 2022))**.** *Let I be an instance of* CSP($\boldsymbol{S}$)*. For a constant $c \in \mathbb{Q}^{>0}$, define an instance I_c by multiplying the endpoints in every constraint of I by c. Then, for every k, (I, k) is a yes-instance of* MINCSP($\boldsymbol{S}$) *if and only if (I_c, k) is a yes-instance.*

Lemma 20. *Let R be a relation of the form $R^{a,d}_{b,c}$ with $a > 0, c > b \geq 0$. Then* MINCSP(R) *is* W[1]*-hard.*

Proof. We compose R with itself d times and get the relation $R^{da,d}_{db,dc}$. This reduction preserves solution cost: each chain of compositions can be broken with one constraint deletion. By Lemma 19, it suffices to show hardness for $R^{a,1}_{b,c}$ as scaling by d gives hardness for the other cases. $R^{a,1}_{b,c}$ is equal to $R_{a+b,a+c}$. Since $a > 0$ and $c > b \geq 0$, we have $a + c > a + b > 0$, and W[1]-hardness follows from Theorem 2. □

We can now present our parameterized classification.

Theorem 21. *Let $\boldsymbol{A} \subseteq \boldsymbol{T}$ be finite,* CSP($\boldsymbol{A}$) *in* P*, and* MINCSP($\boldsymbol{A}$) *is* NP*-hard. If $\boldsymbol{A}$ does not admit a finite retraction, then* MINCSP($\boldsymbol{A}$) *is either in* FPT *or* W[1]*-hard. Specifically, there exists a structure $\boldsymbol{B}$ such that CSP($\boldsymbol{A}$)=CSP($\boldsymbol{B}$) and one of the following holds:*

1. *$\boldsymbol{B}$ has a finite domain.*
2. *$\boldsymbol{B} \subseteq \boldsymbol{PA}$ and the dichotomy follows from Theorem 3.*
3. *$\boldsymbol{B} \subseteq \boldsymbol{S}$ and the dichotomy follows from Theorem 2.*
4. *$\boldsymbol{B} \subseteq \boldsymbol{T}$ is preserved by $\min_d$ and $\max_d$ for some $d > 1$ and* MINCSP($\boldsymbol{B}$) *is* W[1]*-hard.*
5. *$\boldsymbol{B} \subseteq \boldsymbol{S}_{=,\neq}$ and* MINCSP($\boldsymbol{B}$) *is in* FPT*.*

Proof. By Lemma 8, there exists a structure $\mathbf{B}$ such that CSP($\mathbf{A}$)=CSP($\mathbf{B}$) and one of the following six cases holds.

1. $\mathbf{B}$ has a finite domain.
2. $\mathbf{B} \subseteq \mathbf{PA}$.
3. $\mathbf{B} \subseteq \mathbf{S}$.
4. $\mathbf{B} \subseteq \mathbf{T}$ is preserved by $\min_d$ and $\max_d$ for some $d > 1$. By Lemma 10, if $\mathbf{B}$ does not admit a finite retraction (implying that we are in Case 1) or not all relations of $\mathbf{B}$ are equations (implying that we are in Case 3), then $\mathbf{B}$ implements a relation $x - y \in A$ such that $|A| \geq 2$ and all values in A are positive. As $\mathbf{B}$ is preserved by $\min_d$ and $\max_d$, Lemma 7 shows that A is of the form $R^{a,d}_{0,\ell}$, with $\ell > 0$. As all values in A are positive, we also have $a > 0$. W[1]-hardness now follows from Lemma 20.
5. $\mathbf{B} \subseteq \mathbf{S}_{=,\neq}$. By Theorem 18, MINCSP($\mathbf{B}$) is in FPT.
6. CSP($\mathbf{B}$) is NP-complete and so is CSP($\mathbf{A}$). □

7 Concluding Remarks

We have analyzed the complexity of the MINCSP($\mathbf{A}$) problem for 2-DTP languages $\mathbf{A}$. This has led to the following results that generalize previous classifications by Dabrowski et al. (2022) and Osipov et al. (2024).

R1 a P vs. NP-hard dichotomy for MINCSP($\mathbf{A}$) when $\mathbf{A}$ is finite, and

R2 an FPT vs. W[1]-hard dichotomy for MINCSP($\mathbf{A}$) when $\mathbf{A}$ is finite and does not admit a finite retraction.

The proof of R1 and R2 show how complexity classifications for CSPs can be transferred to various classifications for MINCSPs with the aid of algebraic techniques. An advantage with this approach is that hardness results are inherited from the CSP and do not need to be reproved in the MINCSP setting. This leads to a more streamlined classification process and focused proofs with increased readability.

When proving R2, we constructed an FPT algorithm for MINCSP($\mathbf{S}_{=,\neq}$) that is based on a novel reduction to PAIR PARTITION CUT. This shows that graph problems such as GROUP FEEDBACK EDGE SET and EDGE MULTICUT can be solved in FPT time by a unified algorithm that is very different compared to previous algorithms. This result and the reduction technique may be of independent interest.

Our work raises several questions. We first note that decidability of the *meta-problem* is unknown, i.e. given a finite set $\mathbf{A}$ of 2-DTP relations, is MINCSP($\mathbf{A}$) in FPT or is it W[1]-hard? This is a consequence of the decidability status for Theorem 5 being unknown. The basic problem is that Theorem 5 requires us to find a "reformulated" structure $\mathbf{B}$ with certain properties, and it is not clear how to do this.

Generalizing R2 to structures with finite retractions is potentially very difficult. The finite-domain MINCSP problem has been intensively studied, but there are very few general classification results – the main exception is the Boolean case (Kim et al. 2025). A thorough overview of this research programme can be found in (Kratsch et al. 2026).

Generalizing R1 and R2 to infinite constraint languages may also be very difficult, since an infinite-language analogue of Theorem 5 needs to be established. The MAX-ATOMS problem is the problem CSP($\mathbf{M}$) where the infinite language $\mathbf{M}$ contains the relation $M_k = \{(x, y, z) \in \mathbb{Z}^3 \mid \max(x, y) + k \geq z\}$ for all $k \in \mathbb{Z}$. One can verify that every relation in $\mathbf{M}$ is first-order definable in $(\mathbb{Z}; <)$. MAX-ATOMS is polynomial-time equivalent to determining the winner in mean payoff games (Bezem, Nieuwenhuis, and Rodríguez-Carbonell 2008; Möhring, Skutella, and Stork 2004) and this is a problem in NP ∩ coNP, but currently not known to be in P. Thus, generalizing Theorem 5 would settle the long-standing question concerning the complexity of MAX-ATOMS and mean payoff games.

A possible research direction is to generalize R1 and R2 into DTP languages. In such a project, the complexity of MINCSP($\mathbf{A}$) when $\mathbf{A}$ is a reduct of $(\mathbb{Q}; <)$ must be fully understood due to Theorem 5. A P vs. NP-hard

dichotomy was recently established in (Bodirsky, Bonnet, and Semanisinová 2025), but the parameterized complexity is still poorly understood. Although the parameterized dichotomy for MINCSP($\mathbf{A}$) with $\mathbf{A}$ belonging to the point algebra **PA** was recently established (Osipov, Pilipczuk, and Wahlström 2024), this result is already quite challenging despite the simplicity of **PA**. A probably more manageable project is to consider extensions of STP (and in the long run 2-DTP) with at most binary relations. Examples include unary relations $a \leq x \leq b$, the disequality relation, or strict inequalities (see also (Gerevini and Cristani 1997; Koubarakis 1992)). The CSP for these problems is tractable like the basic STP, but full complexity classifications of the corresponding MINCSP problems are currently unknown.

Acknowledgments

We would like to thank Manuel Bodirsky for rapidly answering a series of questions and the anonymous reviewers for providing important comments. The second and the fifth author were partially supported by the Swedish Research Council (VR) under grant 2021-04371. The fourth author was supported by VR under grant 2024-00274.

AI Declaration

The authors have not employed any Generative AI tools.

References

Bacchus, F.; Järvisalo, M.; and Martins, R. 2021. Maximum satisfiability. In Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds., *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press. chapter 24, 929–991.

Barber, F. 2000. Reasoning on interval and point-based disjunctive metric constraints in temporal contexts. *Journal of Artificial Intelligence Research* 12:35–86.

Barto, L.; Krokhin, A. A.; and Willard, R. 2017. Polymorphisms, and how to use them. In *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 1–44.

Bertossi, L., and Chomicki, J. 2004. Query answering in inconsistent databases. In *Logics for Emerging Applications of Databases*. Springer. 43–83.

Bezem, M.; Nieuwenhuis, R.; and Rodríguez-Carbonell, E. 2008. The Max-Atom problem and its relevance. In *Proc. 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-2008)*, 47–61.

Bodirsky, M.; Bonnet, É.; and Semanisinová, Z. 2025. Temporal valued constraint satisfaction problems. In *Proc. 50th International Symposium on Mathematical Foundations of Computer Science (MFCS-2025)*, 24:1–24:19.

Bodirsky, M.; Martin, B.; and Mottet, A. 2018. Discrete temporal constraint satisfaction problems. *Journal of the ACM* 65(2):9:1–9:41.

Bodirsky, M. 2021. *Complexity of Infinite-Domain Constraint Satisfaction*. Cambridge University Press.

Bousquet, N.; Daligault, J.; and Thomassé, S. 2011. Multicut is FPT. In *Proc. 43rd Annual ACM Symposium on Theory of Computing*, 459–468.

Bulatov, A. A. 2017. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS-2017)*, 319–330.

Chomicki, J., and Marcinkowski, J. 2005. Minimal-change integrity maintenance using tuple deletions. *Information and Computation* 197(1-2):90–121.

Condotta, J., and Salhi, Y. 2025. Exploring inconsistency measurement in disjunctive temporal problems. *Information and Computation* 307:105376.

Condotta, J.-F.; Mensi, A.; Nouaouri, I.; Sioutis, M.; and Saïd, L. B. 2016. Local search for maximizing satisfiability in qualitative spatial and temporal constraint networks. In *Proc. 17th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA-2016)*, 247–258. Springer.

Condotta, J.-F.; Nouaouri, I.; and Sioutis, M. 2016. A SAT approach for maximizing satisfiability in qualitative spatial and temporal constraint networks. In *Proc. 15th International Conference on the Principles of Knowledge Representation and Reasoning (KR-2016)*.

Costa, M.; Létocart, L.; and Roupin, F. 2005. Minimal multicut and maximal integer multiflow: A survey. *European Journal of Operational Research* 162(1):55–69.

Cygan, M.; Pilipczuk, M.; and Pilipczuk, M. 2016. On group feedback vertex set parameterized by the size of the cutset. *Algorithmica* 74(2):630–642.

Dabrowski, K. K.; Jonsson, P.; Ordyniak, S.; and Osipov, G. 2022. Resolving inconsistencies in simple temporal problems: A parameterized approach. In *Proc. 36th AAAI Conference on Artificial Intelligence (AAAI-2022)*, 3724–3732.

Dabrowski, K. K.; Jonsson, P.; Ordyniak, S.; Osipov, G.; and Wahlström, M. 2025. Almost consistent systems of linear equations. *ACM Transactions on Algorithms* 21(4):44:1–44:55.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial intelligence* 49(1-3):61–95.

Downey, R., and Fellows, M. 1999. *Parameterized Complexity*. Monographs in Computer Science. Springer.

Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Springer.

Gerevini, A., and Cristani, M. 1997. On finding a solution in temporal constraint satisfaction problems. In *Proc. 15th International Joint Conference on Artificial Intelligence (IJCAI-1997)*, 1460–1465.

Guillemot, S. 2011. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization* 8(1):61–71.

Hunsberger, L., and Posenato, R. 2021. Simple temporal networks: A practical foundation for temporal representation and reasoning. In *Proc. 28th International Symposium on Temporal Representation and Reasoning (TIME-2021)*, 1:1–1:5.

Iwata, Y.; Wahlström, M.; and Yoshida, Y. 2016. Half-integrality, LP-branching, and FPT algorithms. *SIAM Journal on Computing* 45(4):1377–1411.

Jonsson, P., and Bäckström, C. 1998. A unifying approach to temporal constraint reasoning. *Artificial Intelligence* 102(1):143–155.

Khanna, S.; Sudan, M.; Trevisan, L.; and Williamson, D. P. 2000. The approximability of constraint satisfaction problems. *SIAM Journal on Computing* 30(6):1863–1920.

Kim, E. J.; Kratsch, S.; Pilipczuk, M.; and Wahlström, M. 2025. Flow-augmentation III: complexity dichotomy for Boolean CSPs parameterized by the number of unsatisfied constraints. *SIAM Journal on Computing* 54(4):1065–1137.

Koubarakis, M. 1992. Dense time and temporal constraints with $\neq$. In *Proc. 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR-1992)*, 24–35.

Kratsch, S.; Pilipczuk, M.; Sharma, R.; and Wahlström, M. 2026. Applications of flow-augmentation. *Computer Science Review* 60:100869.

Krokhin, A. A., and Zivný, S. 2017. The complexity of valued CSPs. In *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 233–266.

Marx, D., and Razgon, I. 2011. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 469–478.

Möhring, R. H.; Skutella, M.; and Stork, F. 2004. Scheduling with AND/OR precedence constraints. *SIAM Journal on Computing* 33(2):393–415.

Oddi, A., and Cesta, A. 2000. Incremental forward checking for the disjunctive temporal problem. In *Proc. 14th European Conference on Artificial Intelligence (ECAI-2000)*, 108–112.

Osipov, G., and Wahlström, M. 2023. Parameterized complexity of equality MinCSP. *arXiv preprint arXiv:2305.11131*. This is the report version of a paper that appeared at the 31st Annual European Symposium on Algorithms (ESA-2023).

Osipov, G.; Pilipczuk, M.; and Wahlström, M. 2024. Parameterized complexity of MinCSP over the point algebra. In *Proc. 32nd Annual European Symposium on Algorithms (ESA-2024)*, volume 308, 93:1–93:15.

Paparrizou, A., and Sioutis, M. 2025. Learning to resolve inconsistencies in qualitative constraint networks. *Information Systems* 133:102557.

Stergiou, K., and Koubarakis, M. 2000. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence* 120(1):81–117.

Thapper, J., and Zivný, S. 2016. The complexity of finite-valued CSPs. *Journal of the ACM* 63(4):37:1–37:33.

Tsamardinos, I., and Pollack, M. E. 2003. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence* 151(1-2):43–89.

Vilain, M. B., and Kautz, H. A. 1986. Constraint propagation algorithms for temporal reasoning. In *Proc. 5th National Conference on Artificial Intelligence (AAAI-1986)*, 377–382.

Zhuk, D. 2020. A proof of the CSP dichotomy conjecture. *Journal of the ACM* 67(5):30:1–30:78.