Towards Single Exponential Time for Temporal and Spatial Reasoning: A Study via Redundancy and Dynamic Programming

Victor Lagerkvist, Johanna Groven, Leif Eriksson

Linköping University, 581 83 Linköping, Sweden
victor.lagerkvist@liu.se, johanna.groven@liu.se, leif.eriksson@liu.se

Abstract

The *region connection calculus* (RCC) and *Allen's interval algebra* (IA) are two well-known NP-hard spatial-temporal qualitative reasoning problems. They are solvable in $2^{\mathcal{O}(n \log n)}$ time, where n is the number of variables, and IA is additionally known to be solvable in $o(n)^n$ time. However, no improvement over exhaustive search is known for RCC, and if they are also solvable in single exponential time $2^{\mathcal{O}(n)}$ is unknown. We investigate multiple avenues towards reaching such bounds. First, we show that branching is insufficient since there are too many *non-redundant* constraints. Concretely, we classify the maximum number of non-redundant constraints in RCC and IA. Algorithmically, we make two significant contributions based on dynamic programming (DP). The first algorithm runs in 4^n time and is applicable to a non-trivial, NP-hard fragment of IA, which includes the well-known *interval graph sandwich problem* of (Golumbic and Shamir 1993). For the richer RCC problem with 8 basic relations we use a more sophisticated approach which asymptotically matches the $o(n)^n$ bound for IA.

1 Introduction

Infinite structures are a cornerstone in the current artificial intelligence (AI) revolution, ranging from neural networks (which typically approximate functions over continuous structures) to classical formalisms such as *qualitative reasoning*. Here, the basic objects are regions in e.g. space/-time and the task is to determine whether a given configuration (with uncertainties) is consistent. These problems have rich expressive power, are generally NP-complete, and can be formulated as infinite-domain *constraint satisfaction problems* (CSPs). While they admit large tractable classes (Dylla et al. 2017) their precise exponential complexity is poorly understood. Many finite-domain problems can be solved in $2^{\mathcal{O}(n)}$ time by enumerating functions from n objects (e.g., variables) to the fixed domain. This can often be matched with asymptotically tight lower bounds $2^{o(n)}$ (unless the *exponential-time hypothesis* fails (Impagliazzo and Paturi 2001)). For infinite domains exhaustive enumeration is more complicated, and for qualitative reasoning this only gives $2^{\mathcal{O}(n^2)}$ time (n is the number of variables) which is often improvable to $2^{\mathcal{O}(n \log n)}$ time. Clearly, this is *much* worse than $2^{\mathcal{O}(n)}$, and the best lower bounds only rule out $2^{o(n)}$ algorithms (Jonsson, Lagerkvist, and Osipov 2021).

In this paper we are interested in narrowing the gap between infinite and finite structures: when are infinite-domain CSPs solvable in $2^{\mathcal{O}(n)}$ time? We study this via qualitative reasoning problems since they are well understood from the angle of classical complexity theory, model theory, and algebra (Bodirsky and Jonsson 2017), and thus form a suitable microcosmos to tackle the $2^{\mathcal{O}(n)}$ versus $2^{\mathcal{O}(n \log n)}$ question. We concentrate on the two most well-known formalisms: *Allen's interval algebra* (IA) (Allen 1983) and the *region connection calculus* (RCC) (Randell, Cui, and Cohn 1992). Beating exhaustive search is possible for IA with the first breakthrough being a $\mathcal{O}((1.0615n)^n)$ time dynamic programming (DP) algorithm (Eriksson and Lagerkvist 2021), subsequently improved to $o(n)^n$ (Eriksson and Lagerkvist 2023). For RCC-5 and RCC-8 the situation is worse: no improvement is known over enumerating certain orders, which solves RCC-8 in $\mathcal{O}((0.531n)^n)$ time and RCC-5 in $\mathcal{O}((0.368n)^n)$ time (Jonsson, Lagerkvist, and Osipov 2021).

We consider two orthogonal approaches. First, if we could reduce the number of constraints m from $\mathcal{O}(n^2)$ towards $\mathcal{O}(n)$ then $2^{\mathcal{O}(m)}$ branching would suddenly be theoretically interesting. Thus, we want to identify constraints that are *redundant* in the sense that their removal does not affect the solution space (constructing *prime* instances). This is a classical problem in qualitative reasoning (Dylla et al. 2017) and prime instances can often be computed quickly (Sioutis, Li, and Condotta 2015; Hu et al. 2021). Curiously, while there are complexity results for identifying redundant constraints (Li et al. 2015), nothing rigorous is known about the maximal number of non-redundant constraints. To address this we describe the maximal n-variable prime instance of a given relation R ($\text{NRD}_{\{R\}}(n)$), initially studied due to its connection to VC-dimension and learnability (Bessiere, Carbonnel, and Katsirelos 2020). We provide a complete classification of $\text{NRD}_{\{R\}}(n)$ for all basic relations, and several natural macro relations in IA and RCC: $\text{NRD}_{\{R\}}(n) \in \Theta(n^2)$ for most relations, while $\text{NRD}_{\{R\}}(n) \in \Theta(n)$ is only reached when R is the equality or the *meets* relation in IA. Thus, improvements to n^2 can be made but prime instances with only $\mathcal{O}(n)$ constraints are in general impossible.

Our second contribution is algorithmic and here we primarily explore how DP, which is not sensitive to the number of constraints, can be exploited to get closer to $2^{\mathcal{O}(n)}$. First, we consider a non-trivial (but NP-hard) fragment of IA based around strict partial orders p, p^{-1} and an incomparability relation $\cap$. In particular this problem captures the well-known, NP-hard *interval graph sandwich problem* (IGSP) with applications in e.g. molecular biology (Golumbic and Shamir 1993). We present a DP algorithm that solves IGSP in $\mathcal{O}(4^n)$ time (ignoring polynomial factors). The algorithm is based on partitioning the endpoints of intervals into two classes $V_<, V_>$ where endpoints gradually are moved from the right to the left without introducing inconsistencies. This is inspired by the DP algorithm by (Eriksson and Lagerkvist 2021) for the CSP problem with first-order definable relations over $(\mathbb{Q}, <)$ of arity at most 3, but the generalization to interval relations is not without complications. This algorithm cannot (to the best of our knowledge) be generalized to RCC-8 while preserving the single exponential running time, and, given that *no* improvement is known over exhaustive search we first concentrate on matching the $o(n)^n$ bound for IA (Eriksson and Lagerkvist 2023). A straightforward DP approach for RCC-8, such as iteratively building orderings, quickly runs into issues because it is unclear which intermediate states to build upon when multiple valid ones exist. Unlike our algorithm for IGSP , early choices in RCC-8 can significantly impact later stages. We resolve this by storing all necessary states and the novel idea of comparing them based on what is *not yet solved*.

Hence, we not only explain *why* branching algorithms cannot give $2^{\mathcal{O}(n)}$ bounds for qualitative reasoning problems but also identify the first fragment solvable in $2^{\mathcal{O}(n)}$ time as well as giving the first significant improvement for RCC-8.

Many proofs have been omitted due to space constraints.

2 Preliminaries

We sometimes use the notation $2^{\mathcal{O}(n)}$ to express running times $2^{f(n)}$ for $f \in \mathcal{O}(n)$. This notation allows one to hide factors polynomial in the input size but it is sometimes also convenient to express running times dominated by $f(n)^n$ for reasonably small f, where the constants may be important. To suppress polynomial factors in the input size we write $\mathcal{O}^*(f(n)^n)$. We assume that the reader is familiar with basic notions such as (strict) partial/total orders, reflexivity, symmetry, and transitivity. For a strict total or partial order $O = (S, <_O)$ we write $\leq_O$ for the non-strict variant of $<_O$, and vice versa if we are given a non-strict order $(S, \leq_O)$. Similarly, we write $=_O$ for the equality relation induced by a given order O. For a set S, we call a partition $(S_1, \ldots, S_K)$ that partitions S into k different sets $S_1, \ldots, S_k \subseteq S$ a k-partition of S.

2.1 Constraint Satisfaction Problems

In this paper we are exclusively concerned with binary relations over a fixed (but infinite) domain $\mathcal{D}$. A set of binary relations Γ over $\mathcal{D}$ is called a *constraint language*. If $\Gamma = \{R_1, \ldots, R_m\}$ is finite and (1) $\bigcup \Gamma = \mathcal{D}^2$ (*jointly exhaustive*), (2) $R_i \cap R_j = \emptyset$ for all $i \neq j$ (*pairwise disjoint*), (3) $\mathsf{Eq}_\mathcal{D} = \{(x, x) \mid x \in \mathcal{D}\} \in \Gamma$, (4) if $R_i \in \Gamma$ then Γ also contains the *inverse* $R_i^{-1} = \{(x, y) \mid (y, x) \in R_i\}$, then Γ is said to be a *partition scheme*. Partition schemes are the predominant way to define qualitative reasoning problems (cf. (Dylla et al. 2017)). For a set $\Gamma' \subseteq \Gamma$ of relations we define its *macro-relation* as $R^U_{\Gamma'} \coloneqq \bigcup_{R \in \Gamma'} R$. We define the closure of Γ under macro-relations as $\Gamma^U \coloneqq \bigcup_{\Gamma' \subseteq \Gamma} \{R^U_{\Gamma'}\}$.

In the *constraint satisfaction problem* over a constraint language Γ ($\mathrm{CSP}(\Gamma)$) over a domain $\mathcal{D}$ we are given a set of variables V and a set of constraints C of the form $R(x, y)$ where $R \in \Gamma$ and $x, y \in V$, and want to know whether there exists a *satisfying assignment* $f \colon V \to \mathcal{D}$ such that $(f(x), f(y)) \in R$ for every constraint $R(x, y) \in C$. We are specifically interested in the case when Γ is a partition scheme where $\mathrm{CSP}(\Gamma)$ is in P, and then want to solve the richer (and generally NP-complete) problem $\mathrm{CSP}(\Gamma^U)$. For these problems, if for $x, y \in V$, there are multiple constraints $R^U_S(x, y) \in C$ and $R^U_{S'}(x, y) \in C$, we can always polynomially reduce to an instance (V, C') where we replace $R^U_S(x, y)$ and $R^U_{S'}(x, y)$ by $R^U_{S \cap S'}(x, y)$ in C', which is still a $\mathrm{CSP}(\Gamma^U)$ constraint by definition. We thus often assume that for each pair $x, y \in V$, there is at most one constraint $R^U_S(x, y) \in C$. This allows us to also use the *relational network* representation where we represent an instance (V, C) of $\mathrm{CSP}(\Gamma^U)$ by a total function f defined by $f(x, y) = S$ if $R^U_S(x, y) \in C$ and $f(x, y) = \Gamma$ otherwise. Note that if $|f(x, y)| = 1$ for each pair $x, y \in V$ where $f(x, y) \neq \Gamma$ then the instance can be viewed as a $\mathrm{CSP}(\Gamma)$ instance. When this distinction is important then we sometimes refer to the former as a *multi relational network* and the latter as a *simple relational network*, or an *instantiation*. A simple relational network f is said to be *consistent* with a $\mathrm{CSP}(\Gamma^U)$ instance (V, C) if $f(x, y) \in \{R_1, \ldots, R_m\}$ for each constraint $R^U_{\{R_1, \ldots, R_m\}}(x, y)$. We sometimes view a relational network as a graph (V, E_C) with labeled edges $E_C = \{(x, y, f(x, y)) \mid x, y \in V, f(x, y) \neq \Gamma\}$.

2.2 Qualitative Reasoning Problems

Let $\mathbb{I}_\mathbb{Q} = \{[a, b] \mid a, b \in \mathbb{Q}, a < b\}$ be the set of all nonempty closed intervals on the rational line. (Allen 1983) introduced a partition scheme $\mathfrak{A}_{13} = \{p, m, o, s, f, d, e, d^{-1}, f^{-1}, s^{-1}, o^{-1}, m^{-1}, p^{-1}\}$ over $\mathbb{I}_\mathbb{Q}$, where p, m, o, s, f, d, e stands for *precedes*, *meets*, *overlaps*, *starts*, *finishes*, *during*, and *equals*. All these relations can be formally defined by their endpoints, e.g., for two intervals $x = [x^-, x^+]$ and $y = [y^-, y^+]$ then xpy iff $x^+ < y^-$, $x^- < x^+$, and $y^- < y^+$.

Fragments using macro-relations are also often studied. For this purpose, we here consider $\alpha \coloneqq R^U_{\{m,o\}}, \alpha^{-1} \coloneqq R^U_{\{m^{-1},o^{-1}\}}, \subset \coloneqq R^U_{\{s,f,d\}}, \subset^{-1} \coloneqq R^U_{\{s^{-1},f^{-1},d^{-1}\}}$ and $\cap \coloneqq R^U_{\mathfrak{A}_{13} \setminus \{p,p^{-1}\}}$. These relations induce the language $\mathfrak{A}_3 \coloneqq \{p, \cap, p^{-1}\}$ and $\mathfrak{A}_7 \coloneqq \{p, p^{-1}, \alpha, \alpha^{-1}, \subset, \subset^{-1}, e\}$ which induces NP-hard problems $\mathrm{CSP}(\mathfrak{A}_3^U)$ and $\mathrm{CSP}(\mathfrak{A}_7^U)$ (Golumbic and Shamir 1993). We remark that all known maximal tractable subclasses of $\mathfrak{A}_{13}^U$ have been identified (Dylla et al. 2017).

(Randell, Cui, and Cohn 1992) introduced a

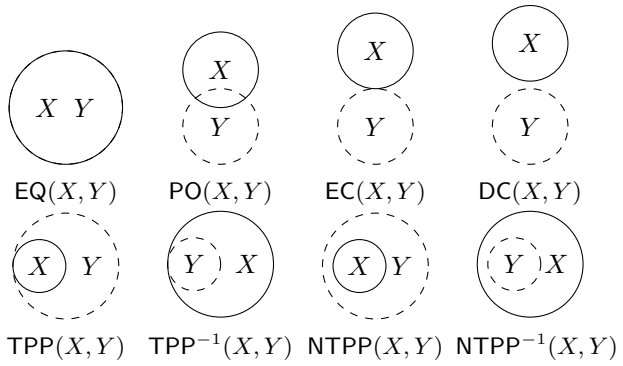

Figure 1: Illustration of the basic relations of RCC-8 with two-dimensional disks.

set of eight relations RCC-8 $= \{EQ, PO, EC, DC, TPP, TPP^{-1}, NTPP, NTPP^{-1}\}$ (see Figure 1). We also consider the macro-relations $PP \coloneqq R^U_{\{TPP,NTPP\}}$, $PP^{-1} \coloneqq R^U_{\{TPP^{-1},NTPP^{-1}\}}$ and $DR \coloneqq R^U_{\{EC,DC\}}$. These relations induce the fragment RCC-5 $= \{EQ, PO, DR, PP, PP^{-1}\}$ of RCC that also forms a partition scheme. $\mathrm{CSP}(\text{RCC-5}^U)$ and $\mathrm{CSP}(\text{RCC-8}^U)$ are both NP-hard and all maximal tractable subclasses have been identified (cf. (Dylla et al. 2017)).

3 Redundancy

If Γ is a partition scheme where $\mathrm{CSP}(\Gamma)$ is solvable in polynomial time then an arbitrary instance (V, C), $|V| = n$, $|C| = m$, of $\mathrm{CSP}(\Gamma^U)$ can be solved in $2^{\mathcal{O}(m)} \cdot (n+m)^{\mathcal{O}(1)}$ time by exhaustive branching (Jonsson, Lagerkvist, and Osipov 2021). Specifically, for $\mathrm{CSP}(\mathfrak{A}^U_{13})$ we obtain the bound $2^{m \cdot \log_2 12} \cdot (n+m)^{\mathcal{O}(1)}$, and for $\mathrm{CSP}(\text{RCC-8}^U)$ the moderately better bound $2^{m \cdot \log_2 7} \cdot (n+m)^{\mathcal{O}(1)}$. Unfortunately, since we may have a quadratic number of constraints, these bounds are asymptotically *worse* than baseline upper bounds of $2^{\mathcal{O}(n \log n)} \cdot (n+m)^{\mathcal{O}(1)}$ which can be obtained by enumerating orders (Jonsson and Lagerkvist 2017; Jonsson, Lagerkvist, and Osipov 2021). Hence, in this section we investigate methods for decreasing m, with a particular focus on identification of *redundant* constraints. We give strong evidence that this method is not likely to bring $2^{\mathcal{O}(m)}$ closer to $2^{\mathcal{O}(n)}$. We stress that analyzing expected running times of branching algorithms equipped with sophisticated heuristics on such instances is a significantly harder problem.

Let Γ be a finite constraint language over a domain $\mathcal{D}$, and let (V, C) be a $\mathrm{CSP}(\Gamma)$ instance (where C is a set of constraints, i.e., not the network representation). We say that $c \in C$ is *non-redundant* if there exists a satisfying assignment $I : V \to \mathcal{D}$ to $(V, C \setminus \{c\})$ that does not satisfy (V, C), and we say that (V, C) is *non-redundant*, or *prime*, if every constraint in C is non-redundant. For each $n \geq 1$ define $\mathrm{NRD}_\Gamma(n)$ over the set of n-variable, non-redundant instances $\mathcal{I}_\Gamma(n)$ of $\mathrm{CSP}(\Gamma)$ as $\mathrm{NRD}_\Gamma(n) \coloneqq \max\{|C| \mid (V, C) \in \mathcal{I}_\Gamma(n)\}$. This function is well-defined since there exists a finite number of

$\mathrm{NRD}_{\{R\}}(n)$	IA			RCC	
	$\mathfrak{A}_3$	$\mathfrak{A}_7$	$\mathfrak{A}_{13}$	RCC-5	RCC-8
$\frac{n(n-1)}{2}$	–	–	–	DR, PO	EC, DC, PO
$\Omega(\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor)$	$\cap$	α	o	–	TPP
$\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$	p	$p, \subset$	$p, s,$ f, d	PP	$NTPP$
$2n-4$	–	–	m	–	–
$n-1$	–	e	e	EQ	EQ

Table 1: The non-redundancy $\mathrm{NRD}_{\{R\}}(n)$ for $n \geq 3$ for all IA relations $R \in \mathfrak{A}_i$ for $i \in \{3, 7, 13\}$ and for all RCC relations $R \in$ RCC-i over $\mathbb{R}^d$ for $i \in \{5, 8\}$. The leftmost column shows the $\mathrm{NRD}_{\{R\}}(n)$ value for all relations R in that the corresponding rows. Each other column classifies $\mathrm{NRD}_{\{R\}}(n)$ for all relations of one of these algebras. Converse relations are omitted as they have the same $\mathrm{NRD}_{\{R\}}(n)$ as their converse. The symbol '-' marks brackets where an algebra has no applicable relation R.

n-variable $\mathrm{CSP}(\Gamma)$ instances, and, furthermore, we have $\mathrm{NRD}_\Gamma(n) \in \mathcal{O}(n^2)$, as well as $\max_{R \in \Gamma} \mathrm{NRD}_{\{R\}}(n) \leq \mathrm{NRD}_\Gamma(n) \leq \Sigma_{R \in \Gamma} \mathrm{NRD}_{\{R\}}(n)$ (Bessiere, Carbonnel, and Katsirelos 2020).

We now continue classifying the non-redundancy by identifying n-variable non-redundant instances of all relations in $\mathfrak{A}_i$ and RCC-j for $i \in \{3, 7, 13\}$ and $j \in \{5, 8\}$. In most cases we can show that these are indeed maximal.

Theorem 1. *The non-redundancy* $\mathrm{NRD}_{\{R\}}(n)$ *of all relations* $R \in \mathfrak{A}_i$ *for* $i \in \{3, 7, 13\}$ *and of all* RCC *relations* $R \in$ RCC-i *for* $i \in \{5, 8\}$ *can be classified as seen in Table 1 for* $n \geq 3$.

Proof. (Sketch) For relations R with $\mathrm{NRD}_{\{R\}}(n) = \frac{n(n-1)}{2}$, any instance without reflexive constraints $R(x, x)$ and symmetric constraints $R(x, y), R(y, x)$ is non-redundant and the maximal size of such an instance corresponds to a complete directed graph. For relations R with $\mathrm{NRD}_{\{R\}}(n) \geq \lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$, a complete binary graph corresponds to a prime instance. For $m \in \mathfrak{A}_{13}$, we have a start- and endpoint s and t and $n-2$ paths (s, x_i, t) that form a prime instance. Finally, for e and EQ, each constraint merges one of the n equivalence classes of variables, meaning we can have $n-1$ non-redundant constraints. □

4 Towards Single Exponential Time

We have proven that branching, in the worst-case, is unlikely to improve upon $2^{\mathcal{O}(n^2)}$ for RCC-i ($i \in \{5, 8\}$), $\mathfrak{A}_3$, and the full algebra $\mathfrak{A}_{13}$. This does not rule out other algorithmic ideas, however, and we now use DP to obtain $2^{\mathcal{O}(n)}$ for $\mathrm{CSP}(\mathfrak{A}^U_3)$ (Section 4.1) and $o(n)^n$ for $\mathrm{CSP}(\text{RCC-8}^U)$ (Section 4.2). We remark that the former is the first *unconditional* (cf. (Eriksson and Lagerkvist 2022)) single exponential algorithm for an NP-hard spatiotemporal reasoning problem and the latter the first major improvement for RCC-8.

Algorithm 1 Solving $\mathrm{CSP}(\mathfrak{A}_3^U)$ in $\mathcal{O}^*(4^n)$ time.

Input: Variables V, Network $C : V^2 \to \mathfrak{A}_3^U$
Output: $True$ exactly if the network is satisfiable

1: $V_e \leftarrow \{x^- \mid x \in V\} \cup \{x^+ \mid x \in V\}$
2: $S \leftarrow \mathrm{Queue}((\emptyset, V_e))$
3: $M \leftarrow \{(\emptyset, V_e)\}$
4: **while** $S \neq \emptyset$ **do**
5: $\quad (V_<, V_>) \leftarrow S.\mathrm{pop}()$
6: $\quad$ **if** $V_< = V_e$ **then**
7: $\quad\quad$ **return** $True$
8: $\quad$ **end if**
9: $\quad$ **for all** $x \in V_>$ **do**
10: $\quad\quad s' \leftarrow (V_< \cup \{x\}, V_> \setminus \{x\})$
11: $\quad\quad$ **if** $(V_e, <_{s'})$ is consistent and $s' \notin M$ **then**
12: $\quad\quad\quad S.\mathrm{append}(s')$
13: $\quad\quad\quad M \leftarrow M \cup \{s'\}$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end while**
17: **return** $False$

4.1 $\mathfrak{A}_3$ and the Graph Interval Sandwich Problem

It is known that every solution to an IA instance (V, C) corresponds to an ordering of the $2n$ interval endpoints x^-, x^+ of intervals $x = (x^-, x^+) \in V$ on the rational line. In the case of $\mathfrak{A}_3$, all endpoints can moreover w.l.o.g. be assumed to be distinct and every relation between intervals can be uniquely determined by only considering relations between three of the four endpoints, albeit this does not hold for an arbitrary selection of three endpoints. E.g. if we have $y^- < y^+ < x^+$, we might either have $y \cap x$ or $y\ p\ x$, but if we have $y^- < x^- < y^+$, the relation of the intervals is $y \cap x$. Here, we construct all endpoint orderings in a DP fashion by moving endpoints from the right to the left one endpoint at a time, i.e. constructing the ordering linearly, appending one endpoint at a time to it. To ensure that each ordering still corresponds to a solution, we fix the relations for each endpoint x moved by considering all triples of endpoints (x, y^-, y^+), for all other intervals (y^-, y^+). Still, constructing all possible endpoints orderings like this would take too long, as there are $2|V|!$ different orderings in the worst case. To work around this issue, we only consider one total order for each subset of V. When constructing the total order of endpoints, we partition V into a set $V_<$ of endpoints that are already totally ordered by the so far constructed order and another set $V_>$ that is not yet ordered. E.g. if we have so far constructed the total order $x_1 < x_2 < x_3$, then $V_< = \{x_1, x_2, x_3\}$. For each 2-partitions $(V_<, V_>)$, we then only save one total ordering giving us a runtime of $\mathcal{O}^*(4^n)$. We show later that if there exists a total order on V, this approach suffices to find it. If there exists no total order corresponding to a satisfying instance, the instance must be unsatisfiable. Then, we will not find such a total order under this restriction either and rightfully reject the instance. Note also that for all $x \in V_<$, we have $x < y$ for all $y \in V_>$ in the final total order of all endpoints that extends the one corresponding to $(V_<, V_>)$.

Formally, let (V, C) be a $\mathrm{CSP}(\mathfrak{A}_3^U)$ instance and $V^- = \{x^- \mid x \in V\}, V^+ = \{x^+ \mid x \in V\}$ the sets of endpoints. We relate to each 2-partition $s = (V_<, V_>)$ of $V^- \cup V^+$ created by moving endpoints $x_1, \ldots, x_\ell$ from $V^- \cup V^+$ to $\emptyset$ sequentially the partial order $(V^- \cup V^+, <_s)$ corresponding to the sequence $x_1, \ldots, x_\ell$ of endpoints used to construct it. As we consider one such sequence for each 2-partition $s = (V_<, V_>)$, $<_s$ is well-defined. We call s *consistent* exactly if for all variables $x, y \in V$ and the relations $R_e(x, y)$ imposed by $(V^- \cup V^+, <_s)$, we have $R_e(x, y) \in \{R_1, \ldots, R_m\} \cup \{\bot\}$ for each $R^U_{\{R_1,\ldots,R_m\}}(x, y) \in C$, where $R_e(x, y) = \bot$ if the order of endpoints of x, y in $(V^- \cup V^+, <_s)$ does not induce a relation in $\mathfrak{A}_3$.

Theorem 2. *Algorithm 1 solves* $\mathrm{CSP}(\mathfrak{A}_3)$ *in time* $\mathcal{O}^*(4^n)$.

Proof. (Sketch) We already argued soundness above. It remains to prove that Algorithm 1 always finds a satisfying total order if it exists. Consider two different consistent partial orders $(V^- \cup V^+, <_s), (V^- \cup V^+, <_{s'})$ corresponding to $s = s' = (V_<, V_>)$ such that $(V^- \cup V^+, <_s)$ can be expanded to total order corresponding to a solution, while $(V^- \cup V^+, <_{s'})$ cannot. All relations including intervals $x \in V$ with $x^-, x^+ \in V_<$ are decided (i.e. fixed by $<_s, <_{s'}$) and as the orders are consistent, these cannot cause any inconsistencies in the future. Thus there must be some *open* intervals $x, y \in V$ with $x^-, y^- \in V_<$ but $x^+, y^+ \in V_>$ such that $x^- <_s y^-$ but $y^- <_{s'} x^-$. But in either case, we already fix $x \cap y$ under $<_s, <_{s'}$ and x, y have the same relations with all $z \in V$ with $z^-, z^+ \in V_>$ in the same expansions (i.e. if we append the same remaining endpoint orderings to $<_s, <_{s'}$). So $(V^- \cup V^+, <_s), (V^- \cup V^+, <_{s'})$ must either both be expandable to a satisfying total order or not. We conclude that if (V, C) is satisfiable, at least one sequence $x_1, \ldots, x_{2n} \in V^- \cup V^+$ corresponding to a satisfying assignment for (V, C) is *found* (as a series of 2-partitions) by Algorithm 1. Finally, as the set S contains at most every two partition of $V^- \cup V^+$, we have a bound of at most $2^{|V^- \cup V^+|} = 4^n$ iterations of the outer loop in line 4. □

We can extend this as follows. Call a graph $G = (V, E)$ an *interval graph* if and only if (V, C_E) is a satisfiable $\mathrm{CSP}(\mathfrak{A}_3^U)$ instance, where $\cap(v, w) \in C_E(v, w)$ if $(v, w) \in E$ and $R^U_{\{p,p^{-1}\}}(v, w) \in C_E$ otherwise. In the *interval graph sandwich problem* (IGSP) we are then given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with $E_1 \cap E_2 = \emptyset$, and want to decide if there is an interval graph $G = (V, E)$ with $E_1 \subseteq E \subseteq E_1 \cup E_2$. IGSP is known to be a special case of $\mathrm{CSP}(\mathfrak{A}_3^U)$ (Golumbic and Shamir 1993), which, together with Theorem 2, yields the following corollary.

Corollary 3. IGSP *is solvable in* $\mathcal{O}^*(4^n)$ *time where* $n = |V|$ *is the number of vertices.*

4.2 An Improved Algorithm for RCC-8

As the final result we present an improved DP algorithm for $\mathrm{CSP}(\text{RCC-8}^U)$. We recall that $\mathrm{CSP}(\text{RCC-8}^U)$ can be solved in $\mathcal{O}((0.531n)^n)$ time by enumerating so-called *ordered partitions*, or total orders (Jonsson, Lagerkvist, and Osipov 2021). Here, the intuition is that for a given total order $(V, <_T)$, if $x <_T y$, then the relation between x and

y in our instance is neither EQ, TPP^{-1}, nor $NTPP^{-1}$, and the resulting instance can be solved in polynomial time. I.e., total orders function as *certificate*s. We thus need to significantly decrease how many total orders we need to handle. We work bottom-up towards defining a recurrence relation $R_f(S)$, by focusing on how we compare different total orders to each other. However, if one attempts to apply standard DP with total orders as states we quickly run into problems. Normally there are methods to quickly choose between two possible states, but this is not the case here since choices we make early can have a significant impact later. Hence, for each set $S \subseteq V$, $R_f(S)$ thus has to represent a set of total orders. Our goal is hence to minimize the size of these sets without losing correctness. We resolve this by comparing the total orders using *what is not yet solved* which then allows us to only keep a subset of all possible total orders.

As a road map, we first define how an instance behaves under an assumed total order followed by showing that a sub-instance can be solved in polynomial time and that the solution in some sense is as general as possible. We then introduce the crucial concept of *inconsistency paths* (IPs): a potential path of transitivity beginning in the part of the instance that has not yet been considered, but such that the path ends in the part that has already been handled. Using IPs, we then see how we can compare total orders and more importantly, how we can minimize how many of these total orders $R_f(S)$ needs to contain. This works since if we have two total orders $T = (V_T, \leq_T)$ and $T' = (V_T, \leq'_T)$ and we know that all IPs relevant for T also exist in T', then T is at least as good as T'. Hence, by minimizing $R_f(S)$ according to this, we can guarantee that we keep enough to ensure that we can construct a solution to the problem (if one exists). Last, we analyze the sizes of these minimal $R_f(S)$ in order to achieve the desired complexity.

We start by defining how we apply a partial order to a given relational network.

Definition 4. *Given a* $\mathrm{CSP}(\text{RCC-}8^U)$ *instance* $I = (V, C)$ *with multi relational network* f *and a partial order* $P = (V, \leq_P)$*, we for all* $x, y \in V$ *define the instance* $(P \circ f)(x, y) =$

$$\begin{cases} f(x,y) \setminus \{TPP^{-1}, NTPP^{-1}, EQ\}, & \text{if } x <_P y, \\ f(x,y) \setminus \{TPP, NTPP, EQ\}, & \text{if } y <_P x, \\ f(x,y) \cap \{EQ\}, & \text{if } x =_P y, \\ f(x,y), & \text{otherwise.} \end{cases}$$

Additionally, given two relational networks f *and* g*, let* $(f \cap g)(x, y) = f(x, y) \cap g(x, y)$.

Given this we can now show that if an arbitrary instance I is a yes-instance (i.e. there exists an assignment satisfying I), there will be a total order T, such that I under T is also a yes-instance.

Lemma 5. *If a* $\mathrm{CSP}(\text{RCC-}8^U)$ *instance* I *with multi relational network* f *is a yes-instance then there is a total order* T *such that* $T \circ f$ *is a yes-instance.*

Proof. (Sketch) Any relational network satisfying I describes a partial order over V. Topologically sorting this partial order yields the desired properties. □

If $P = (V_P, \leq_P)$ is a partial order then we write $P < S$ for the partial order $(V_P \cup S, \leq_{P'})$ where $\leq_{P'} = \leq_P \cup \{(x, y) \mid x \in V_P \text{ and } y \in S\}$. Hence, we may write $(P < S) \circ f$ if $V_P \cup S = V$.

Definition 6. *Given a* $\mathrm{CSP}(\text{RCC-}8^U)$ *instance* $I = (V, C)$ *with multi relational network* f *and a total order* $T = (V_T, \leq_T)$ *with* $V_T \subseteq V$*, we for all* $x, y \in V$ *define* $((T < V \setminus V_T) \odot f)(x, y) =$

$$\begin{cases} f(x,y) \setminus \{TPP^{-1}, NTPP^{-1}, EQ\}, & \text{if } x <_T y, \\ f(x,y) \setminus \{TPP, NTPP, EQ\}, & \text{if } y <_T x, \\ f(x,y) \cap \{EQ\}, & \text{if } x =_T y, \\ \text{RCC-8}, & \text{otherwise.} \end{cases}$$

Note that we assume that $x <_T y$ *for all* $x \in V_T$ *and* $y \in V \setminus V_T$ *to simplify the notation.*

Additionally we say that $(T < V \setminus V_T) \odot f$ *is* reduced locally k-consistent *if for every set* $S \subseteq V$ *with* $|S| \leq k$ *then for every pair* $x, y \in S$ *and for every relation* $r \in ((T < V \setminus V_T) \odot f)(x, y)$ *then the instance* g *over* S *defined as* $g(u, v) = ((T < V \setminus V_T) \odot f)(u, v)$ *for all* $u, v \in S$*,* $u, v \neq x, y$ *and* $g(u, v) = r$*, is a yes-instance. Any relation* $r \in ((T < V \setminus V_T) \odot f)(x, y)$ *not satisfying the above criteria is* k-excessive.

While the two definitions of $(P < V \setminus V_P) \circ f$ and $(T < V \setminus V_T) \odot f$ may seem very similar, there are some key differences between them: $(P < V \setminus V_P) \circ f$ can be hard to solve since if $P = (\emptyset, \emptyset)$ then $(P < V \setminus V_T) \circ f = f$, and $(P < V \setminus V_P) \circ f \Rightarrow f$. In contrast, as we will soon prove, $(T < V \setminus V_T) \odot f$ is tractable, but $(T < V \setminus V_T) \odot f \not\Rightarrow f$. Additionally, note that if $V_T = V$ then $T \odot f = T \circ f$. This means that $(T < V \setminus V_T) \odot f$ will be easy to work with and preferable to use, but we will not be sure we are actually on the correct track until $V_T = V$. Given an arbitrary $\mathrm{CSP}(\text{RCC-}8^U)$ instance $I = (V, C)$ with multi relational network f and a total order $T = (V_T, <_T)$, let $RLC_k((T < V \setminus V_T) \odot f)$ be a function outputting $(T < V \setminus V_T) \odot f$ with all k-excessive relations removed.

Lemma 7. *The function* $RLC_k((T < V \setminus V_T) \odot f)$ *can be calculated in* $\mathcal{O}(poly(|V|))$ *time for any constant* k.

Proof. (Sketch) For any subset of V of size k any k-excessive relations can be removed in polynomial time by brute-force checking. As there are $|V|^k$ of these subsets it is all in $\mathcal{O}(poly(|V|))$. □

We order relations in the following way: $DC = \diamond_1$, $EC = \diamond_2$, $PO = \diamond_3$, $TPP = \diamond_4$, and $NTPP = \diamond_5$. Given $\diamond \subseteq \{\diamond_1, \ldots, \diamond_5\}$ we let $\diamond_{min} \in \diamond$ be the smallest element under this ordering.

Observation 8. *By taking the sequence* $DC = \diamond_1$*,* $EC = \diamond_2$*,* $PO = \diamond_3$*,* $TPP = \diamond_4$*, and* $NTPP = \diamond_5$*, and studying the transitivity of the relations of* RCC-8 *under some total order* T *we can observe the following important properties:*

Algorithm 2 Solving $(T < V \setminus V_T) \odot f$ in $\mathcal{O}(poly(||I||))$

Input: Multi relational network $g = (T < V \setminus V_T) \odot f$

1: $g' \leftarrow RLC_3(g)$
2: **for** $x_i \in x_1 <_T \ldots <_T x_{|V_T|}$ **do**
3: **for** $x_j \in x_{i-1} >_T \ldots >_T x_1$ **do**
4: $g'(x_j, x_i) \leftarrow \{g'(x_j, x_i)_{min}\}$
5: $g' \leftarrow RLC_3(g')$
6: **end for**
7: **end for**
8: **return** g'

- *For every combination of $i \in [2, \ldots, 5]$ and $j \in [5]$, the set of allowed relations for $x \diamond z$ given $x \diamond_i y$ and $y \diamond_j z$ is a subset of the allowed relations for $x \diamond_{i-1} y$ and $y \diamond_j z$.*
- *For $i \in [5]$ and every pair $j, j' \in [5]$ with $j < j'$, let $\diamond_{i,j}$ and $\diamond_{i,j'}$ be the sets of relations allowed between x and z given $x \diamond_i y$ and $y \diamond_j z$, and given $x \diamond_i y$ and $y \diamond_{j'} z$ respectively. For every $\diamond_k \in \diamond_{i,j'}$ then either $\diamond_k \in \diamond_{i,j}$, or for all $\diamond_{k'} \in \diamond_{i,j}$ we have $k' < k$.*

We now establish correctness of Algorithm 2 which shows that $(T < V \setminus V_T) \odot f$ can be solved in polynomial time.

Lemma 9. *Given a* CSP(RCC-8^U) *network $(T < V \setminus V_T) \odot f$ for some total order $T = (V_T, \leq_T)$ and network f, Algorithm 2 returns $g(x, y) = \emptyset$ for all $x, v \in V_T$ if $(T < V \setminus V_T) \odot f$ is a no-instance. Otherwise Algorithm 2 returns a relational network g such that*

- *g satisfies $(T < V \setminus V_T) \odot f$, and*
- *there is no relational network $g' \neq g$ satisfying $(T < V \setminus V_T) \odot f$ such that there is a pair $x, y \in V$ where if $g'(x, y) = \{\diamond_i\}$ and $g(x, y) = \{\diamond_j\}$ then $i < j$.*

Proof. RLC_k is correct by definition as it only removes relations that cannot be used to satisfy the original instance. Additionally, if the relational network ever becomes inconsistent, every relation is $\emptyset$. For the remainder of the algorithm it comes down to three things: assigning $g(x_i, x_j)$, $g(x_j, x_k)$ and $g(x_i, x_k)$ for $i < j < k$. By Line 4 and the ordering of the looping, we ensure $g(x_i, x_j)$ is always assigned first, and assigned according to the first of DC, EC, PO, TPP, and $NTPP$, as desired by our stated properties. By Observation 8, we can see that this ensures maximum allowed relations for $g(x_i, x_k)$, as well as allowing $g(x_j, x_k)$ to be as early relations from DC, EC, PO, TPP, and $NTPP$ as possible. Hence, assigning relations in this way fulfills the properties stated in the lemma. □

Now we are finally ready to introduce inconsistency paths, i.e., our representation of what is not yet solved.

Definition 10. *Given a* CSP(RCC-8) *network f over V and a partial order P over V such that $x_1 <_P \cdots <_P x_i$, let $R_t(x_1, \ldots, x_i)$ be the set of relations allowed between x_1 and x_i given transitivity when going from x_j to x_{j-1} for all $j \in [2, \ldots, i]$ and the assumption that P is applied to f, i.e. that the network is now $P \circ f$. An* inconsistency path *(IP) for a relational network g satisfying $(T < V \setminus V_T) \odot f$ for some relational network f and total order $T = (V_T, \leq_T)$, is a pair (h, t) such that there is a sequence $x_1 <_T \cdots <_T x_k < t$, $\{x_1, \ldots x_k\} \subseteq V_T$, and a sequence $y_1, \ldots, y_i$, $t = y_1$, $h = y_i$, $\{y_1, \ldots y_i\} \subseteq V \setminus V_T$ such that $g(x_1, h) \cap R_t(x_1, \ldots x_k, t, \ldots, h) = \emptyset$.*

Let $\mathcal{I}_f(g)$ be the set of all IPs in g. Finding all IPs, and hence everything that is yet to be solved, turns out to be easy.

Lemma 11. *$\mathcal{I}_f(g)$ can be calculated in $\mathcal{O}(poly(V))$ time.*

Proof. There are $|V \setminus V_T|^2$ possible IPs and checking them can be done by e.g. breadth-first search. □

We also need to compare different relational networks.

Definition 12. *We say $g_1 \prec_f g_2$ if $\mathcal{I}_f(g_1) \subset \mathcal{I}_f(g_2)$, $g_1 \equiv_f g_2$ if $\mathcal{I}_f(g_1) = \mathcal{I}_f(g_2)$, and $g_1 \sqcap_f g_2$ if neither $g_1 \prec_f g_2$, $g_2 \prec_f g_1$ nor $g_1 \equiv_f g_2$, i.e. g_1 and g_2 are incomparable.*

Note that g_1 and g_2 are not necessarily over the same total orders $T_1 = (V_T, \leq_{T_1})$ and $T_2 = (V_T, \leq_{T_2})$. Now, by Lemma 9 we know that we can get the best relational network for each T.

Lemma 13. *The relational network g over T returned by Algorithm 2 is $\prec_f$-minimal, i.e. there is no other relational network g' over T that also satisfies $(T < V \setminus V_T) \odot f$ and such that $g' \prec_f g$.*

Proof. (Sketch) Follows by Observation 8 and the fact that Algorithm 2 assigns according to the ordering DC, EC, PO, TPP, $NTPP$. □

With Lemma 13 we can now use T in place of any relational network g satisfying $(T < V \setminus V_T) \odot f$, i.e., we can assume that $\mathcal{I}_f(T) = \mathcal{I}_f(g')$, where g' is the output of Algorithm 2.

Definition 14. *Given a set of total orders $\mathbf{T}$ over a set of variables $V_T \subseteq V$ and a multi relational network f over V, let $\sqcap_f^{min}(\mathbf{T})$ be a maximal subset of $\mathbf{T}$ such that (1) for no $T, T' \in \sqcap_f^{min}(\mathbf{T})$ we have $T \equiv_f T'$ and (2) for every $T \in \sqcap_f^{min}(\mathbf{T})$ there is no $T' \in \mathbf{T}$ such that $T' \prec_f T$. Additionally, if $(T < V \setminus V_T) \odot f$ is a no-instance, then $T \notin \sqcap_f^{min}(\mathbf{T})$.*

Note here the last property, as this will allow us to keep only total orders that are actually of interest to us. We can achieve, and keep, this property by running Algorithm 2 on every total order. With the minimal sets defined, we also want to know what the cost is to compute them.

Lemma 15. *Given a set $\mathbf{T}$ of total orders over a set of variables $V_T \subseteq V$ and a multi relational network f over V, we can compute $\sqcap_f^{min}(\mathbf{T})$ in $\mathcal{O}^*(|\mathbf{T}|^2)$ time.*

Proof. We can brute-force compare all orders in $\mathbf{T}$. Comparing two orders can be done in polynomial time by Lemma 11, and the last property of Definition 14 is done by Algorithm 2, which is polynomial in $|V|$. □

While the factor $|\mathbf{T}|^2$ does not seem too costly, we will later see that this is actually the main bottleneck of the final algorithm, and any improvement here would have a significant impact. Before turning to the the main result of this section we introduce two simplifying notations.

Definition 16. *For any set S, let $S^=$ be the total order $(S, \{x = y \mid x, y \in S\})$, i.e. the total order where all variables are equivalent. For a total order $T = (V_T, \leq_T)$ (assuming $V_T \cap S = \emptyset$) we then by $T < S^=$ mean the total order constructed by first taking T, and then adding the set S as a new partition that comes after all the ones in T.*

We are now ready to define the main recurrence relation $R_f(S)$ (which constitutes our algorithm) as

$$\begin{cases} \{(\emptyset, \emptyset)\} \text{ if } S = \emptyset, \text{ and otherwise} \\ \sqcap_f^{min}(\bigcup_{V_T \subseteq S} \{(T < (S \setminus V_T)^=) \mid T \in R_f(V_T)\}) \end{cases}$$

First we establish correctness.

Lemma 17. *If, and only if, $R_f(V) \neq \emptyset$ then f, and hence $I = (V, C)$, is a yes-instance.*

Proof. By Lemma 5 we know that T exists, and if $R_f(V) \neq \emptyset$, then, since $\sqcap_f^{min}$ calls Algorithm 2, we must have a yes-instance by Lemma 9. For the other direction the only thing that can introduce non-correctness is going from $\mathbf{T}$ to $\sqcap_f^{min}(\mathbf{T})$. Assume $(T' < V \setminus V_T) \circ f$ is a yes-instance while $(T < V \setminus V_T) \circ f$ is a no-instance, with $T \prec_f T'$, $T, T' \in \mathbf{T}$, and $T' \notin \sqcap_f^{min}(\mathbf{T})$. Hence, there must be a function $h \colon (V \setminus V_T)^2 \to \{DC, EC, PO, TPP, NTPP, TPP^{-1}, NTPP^{-1}, EQ\}$ such that $((T' < V \setminus V_T) \circ f) \cap h$ is a yes-instance. Since $\mathcal{I}_f(T) \subset \mathcal{I}_f(T')$ then $((T < V \setminus V_T) \circ f) \cap h$ must also be a yes-instance. Hence, $(T' < V \setminus V_T) \circ f \Rightarrow (T < V \setminus V_T) \circ f$. □

What remains to prove is the bound on the size of $\sqcap_f^{min}(\mathbf{T})$, since this is extremely relevant for the complexity analysis of the algorithm.

Lemma 18. *Given a set $\mathbf{T}$ of total orders over a set of variables $V_T \subseteq V$ and an multi relation network f over V with $n = |V|$ then $|\sqcap_f^{min}(\mathbf{T})| \in \mathcal{O}^*((cn/\log n)^{n/2})$ for some constant c.*

Proof. (Sketch) By assuming that $|\sqcap_f^{min}(\mathbf{T})| \leq g(m, k')^{(n-k')/m}$ were $g(m, k')$ describes the maximum number $\prec$-minimal total orders containing m variables and using k' IPs. This then gives contradictions for anything larger than $g(m, k')^{(n-k')/m} \leq k!^{(n-k(k-1))/k}$, as the maximum number of $\prec$-minimal total orders occurs when every variable in each subset has one unique IP for every other variable in the same subset, and for no two IPs (h, t) and (h', t') used in the same subset should $t = t'$. □

We combine everything and give the first $o(n)^n$ result for CSP(RCC-8^U) (where $c \geq 0$ stems from Lemma 18).

Theorem 19. *An arbitrary* CSP(RCC-8^U) *instance $I = (V, C)$ with $n = |V|$ is solvable in $\mathcal{O}^*((cn/\log n)^n)$ time and $\mathcal{O}^*((cn/\log n)^{n/2})$ space.*

Proof. By Lemma 17 we have a correct algorithm in $R_f(S)$, by Lemma 18 we know that for all $S \subseteq V$ then $|R_f(S)| \in \mathcal{O}^*((cn/\log n)^{n/2})$, using a DP approach to calculate $R_f(S)$ for all $S \subseteq V$ can be done in 3^n-steps by standard methods, and lastly, calculating $\sqcap_f^{min}$ in every step of the recurrence function/DP is in $\mathcal{O}^*((2^{|S|}|R_f(S)|)^2)$ time by Lemma 15. Combining all this gives that I is solvable in $\mathcal{O}^*((cn/\log n)^n)$ time and $\mathcal{O}^*((cn/\log n)^{n/2})$ space. □

While the DP part of the algorithm has been reasonably hidden until this last theorem (or until the recurrence relation for the astute reader) the concept has been very relevant for why we want to minimize $\sqcap_f^{min}(\mathbf{T})$ and why focusing on IPs are relevant in the first place.

5 Concluding Remarks

Our work opens up mathematically and algorithmically motivated questions. First, non-redundancy of finite-domain CSPs has recently shown to coincide with ε-*sparsifiability* (Brakensiek and Guruswami 2025). Is this possible for qualitative reasoning problems (often ω-*categorical*) and other types of infinite-domain CSPs? This would allow one to sparsify instances via coding theory merely by knowing NRD_Γ. Second, are there any other (NP-hard) fragments of qualitative reasoning problems solvable in single-exponential time? Here, one can get reasonable candidates by selecting a maximal tractable class and adding an arbitrary relation which makes the problem hard. For example, can we find $2^{\mathcal{O}(n)}$ fragments of e.g. the *cardinal direction calculus* or the *star calculus* in this way? In the other direction, one may conjecture that NP-hard qualitative reasoning problems are solvable in $o(n)^n$ time but not substantially faster. Can this be formally established under e.g. the *exponential-time hypothesis* (Impagliazzo and Paturi 2001)?

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions on how to improve the presentation. The first author is partially supported by the Swedish research council under grant VR-2022-03214.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26: 832–843.

Bessiere, C.; Carbonnel, C.; and Katsirelos, G. 2020. Chain Length and CSPs Learnable with Few Queries. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*, 1420–1427. AAAI Press.

Bodirsky, M.; and Jonsson, P. 2017. A Model-Theoretic View on Qualitative Constraint Reasoning. *Journal of Artificial Intelligence Research (JAIR)*, 58: 339–385.

Brakensiek, J.; and Guruswami, V. 2025. Redundancy Is All You Need. In Koucký, M.; and Bansal, N., eds., *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC-2025)*, 1614–1625. ACM.

Dylla, F.; Lee, J. H.; Mossakowski, T.; Schneider, T.; Delden, A. V.; Ven, J. V. D.; and Wolter, D. 2017. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Computing Surveys (CSUR)*, 50(1): 7:1–7:39.

Eriksson, L.; and Lagerkvist, V. 2021. Improved Algorithms for Allen's Interval Algebra: a Dynamic Programming Approach. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-2021)*, 1873–1879. ijcai.org.

Eriksson, L.; and Lagerkvist, V. 2022. A Multivariate Complexity Analysis of Qualitative Reasoning Problems. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI-2022)*, 1804–1810. ijcai.org.

Eriksson, L.; and Lagerkvist, V. 2023. Improved Algorithms for Allen's Interval Algebra by Dynamic Programming with Sublinear Partitioning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI-2023)*, 1919–1926. ijcai.org.

Golumbic, M. C.; and Shamir, R. 1993. Complexity and algorithms for reasoning about time: a graph-theoretic approach. *Journal of the ACM*, 40(5): 1108–1133.

Hu, Q.; Gao, Y.; Long, Z.; Wang, H.; Li, T.; and Sioutis, M. 2021. On Large-Scale Qualitative Spatio-Temporal Constraint Redundancy Removal. In Chen, S.; Hu, J.; Li, T.; Martínez, L.; and Liu, J., eds., *Proceedings of the 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE-2021)*, 398–405. IEEE.

Impagliazzo, R.; and Paturi, R. 2001. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2): 367 – 375.

Jonsson, P.; and Lagerkvist, V. 2017. An initial study of time complexity in infinite-domain constraint satisfaction. *Artificial Intelligence*, 245: 115–133.

Jonsson, P.; Lagerkvist, V.; and Osipov, G. 2021. Acyclic orders, partition schemes and CSPs: Unified hardness proofs and improved algorithms. *Artificial Intelligence*, 296: 103505.

Li, S.; Long, Z.; Liu, W.; Duckham, M.; and Both, A. 2015. On redundant topological constraints. *Artificial Intelligence*, 225: 51–76.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A Spatial Logic based on Regions and Connection. In *Proceedings of the 1sth International Conference on Principles of Knowledge Representation and Reasoning (KR-1992)*, 165–176.

Sioutis, M.; Li, S.; and Condotta, J. 2015. Efficiently Characterizing Non-Redundant Constraints in Large Real World Qualitative Spatial Networks. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-2015)*, 3229–3235. AAAI Press.