Parameterized Complexity Classification for Interval Constraints

Konrad K. Dabrowski
School of Computing, Newcastle University, UK

Peter Jonsson
Department of Computer and Information Science, Linköping University, Sweden

Sebastian Ordyniak
School of Computing, University of Leeds, UK

George Osipov
Department of Computer and Information Science, Linköping University, Sweden

Marcin Pilipczuk
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
IT University Copenhagen, Denmark

Roohani Sharma
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

Constraint satisfaction problems form a nicely behaved class of problems that lends itself to complexity classification results. From the point of view of parameterized complexity, a natural task is to classify the parameterized complexity of MinCSP problems parameterized by the number of unsatisfied constraints. In other words, we ask whether we can delete at most k constraints, where k is the parameter, to get a satisfiable instance. In this work, we take a step towards classifying the parameterized complexity for an important infinite-domain CSP: *Allen's interval algebra* (IA). This CSP has closed intervals with rational endpoints as domain values and employs a set $\mathcal{A}$ of 13 basic comparison relations such as "precedes" or "during" for relating intervals. IA is a highly influential and well-studied formalism within AI and qualitative reasoning that has numerous applications in, for instance, planning, natural language processing and molecular biology. We provide an FPT vs. W[1]-hard dichotomy for MinCSP(Γ) for all $\Gamma \subseteq \mathcal{A}$. IA is sometimes extended with unions of the relations in $\mathcal{A}$ or first-order definable relations over $\mathcal{A}$, but extending our results to these cases would require first solving the parameterized complexity of Directed Symmetric Multicut, which is a notorious open problem. Already in this limited setting, we uncover connections to new variants of graph cut and separation problems. This includes hardness proofs for simultaneous cuts or feedback arc set problems in directed graphs, as well as new tractable cases with algorithms based on the recently introduced *flow augmentation* technique. Given the intractability of MinCSP($\mathcal{A}$) in general, we then consider (parameterized) approximation algorithms. We first show that MinCSP($\mathcal{A}$) cannot be polynomial-time approximated within any constant factor and continue by presenting a factor-2 fpt-approximation algorithm. Once again, this algorithm has its roots in flow augmentation.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases (minimum) constraint satisfaction problem, Allen's interval algebra, parameterized complexity, cut problems

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.11

Related Version *Full Version*: arXiv:2305.13889

Funding *Peter Jonsson*: Partially supported by the Swedish Research Council (VR) under grant 2021-0437 and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 11; pp. 11:1–11:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Sebastian Ordyniak: Supported by the Engineering and Physical Sciences Research Council (EPSRC, project EP/V00252X/1).
George Osipov: Supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
Marcin Pilipczuk: During this research Marcin was part of BARC, supported by the VILLUM Foundation grant 16582.

1 Introduction

Background. The *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)) is the problem of deciding whether there is a variable assignment which satisfies a set of constraints, where each constraint is constructed from a relation in Γ. CSPs over different constraint languages form a nicely behaved class of problems that lends itself to complexity classification results. Such results are an important testbed for studying the power of algorithmic techniques and proving their limitations – A prime example is the dichotomy theorem for finite-domain CSPs that was conjectured by Feder and Vardi [28] and independently proved by Bulatov [14] and Zhuk [54]. Here, all hardness results were known since the work of Bulatov, Jeavons and Krokhin [15] and the algorithms of [14] and [54] completed the proof of the conjecture. In between, *lots* of work went into studying the problem from various algorithmic and algebraic angles, and many ideas emerging from this project have been re-used in different contexts (such as infinite-domain CSPs [8] or *promise CSPs* [42]). Optimization versions of the CSP such as MaxCSP and MinCSP (where the goal is to find an assignment that maximises the number of satisfied constraints (MaxCSP) or minimises the number of unsatisfied constraints (MinCSP)) and the generalization Valued CSP (VCSP) have also been intensively studied. Some notable results include the proof of that every finite-domain VCSP is either polynomial-time solvable or NP-complete [40], and the optimal approximability result for finite-domain MaxCSP under the Unique Games Conjecture [50]. One should note that even if the P/NP borderline for finite-domain VCSPs is fully known, there are big gaps in our understanding of the corresponding FPT/W[1] borderline (with parameter solution weight). The situation is even worse if we consider infinite-domain optimization versions of the CSPs, since we cannot expect to get a full picture of the P/NP borderline even for the basic CSP problem [9].

In the parameterized complexity world, MinCSP is a natural problem to study. Subproblems that have gained attraction include Boolean constraint languages [13, 37, 51], Dechter et al.'s [25] simple temporal problem (STP) [22], linear inequalities [5] and linear equations [20, 24], to list a few. Highly interesting results have emerged from studying the parameterized complexity of problems like these. For instance, the recent dichotomy for MinCSPs over the Boolean domain by Kim et al. [37] was obtained using a novel technique called *directed flow augmentation*. Recent work has indicated *temporal* CSPs as a possible next step [27, 38]. Temporal CSPs are CSPs where the relations underlying the constraints are first-order definable in $(\mathbb{Q}; <)$. The computational complexity of temporal CSPs where we fix the set of allowed constraints exhibits a dichotomy: every such problem is either polynomial-time solvable or NP-complete [11]. The MinCSP problem for temporal CSPs is closely related to a number of graph separation problems. For example, if we take the rationals as the domain and allow constraints $\leq$ and $<$, the MinCSP problem is equivalent to Directed Subset Feedback Arc Set [39], a problem known to be fixed-parameter tractable for two different, but both quite involved reasons [17, 39]. If we allow the relations $\leq$ and $\neq$, we obtain a problem equivalent to Directed Symmetric Multicut, whose parameterized complexity is identified as the main open problem in the area of directed graph separation problems [27, 39]. Another related way forward is to analyse the MinCSP for *Allen's interval algebra*. Allen's interval algebra is a highly influential formalism within AI and qualitative reasoning that has numerous applications, e.g. in planning [4, 48, 49], natural language processing [26, 52] and molecular biology [31]. This CSP uses closed intervals with rational endpoints as domain values and employs a set $\mathcal{A}$ of 13 basic comparison relations such as "precedes" (one interval finishes before the other starts) or "during" (one interval is a strict subset of the other); see Table 1. Formally speaking, the CSP for the interval algebra is not a temporal CSP, since the underlying domain is based on intervals instead of points. This difference is important: complexity classifications for the interval algebra have been harder to obtain than for temporal constraints. There are full classifications for binary relations [41] and for first-order definable constraint languages containing all basic relations [10]; a classification for all first-order definable constraint languages appears remote.

Our contributions. The aim in this paper is to initiate a study of MinCSP in the context of Allen's interval algebra. Obtaining a full parameterized complexity classification for Allen's interval algebra would entail resolving the status of Directed Symmetric Multicut and we do not aim at this very ambitious task. Instead, we restrict ourselves to languages that are subsets of $\mathcal{A}$ and do not consider more involved expressions (say, first-order logic) built on top of $\mathcal{A}$. Even in this limited quest, we are able to uncover new relations to graph separation problems, and new areas of both tractability and intractability. One of the main ingredients for both our tractability and intractability results is a particular characterization of unsatisfiable instances of CSP($\mathcal{A}$). A combinatorial analysis of the relations in $\mathcal{A}$ allows us to identify the minimal obstructions given by certain arc-labelled mixed cycles of the instance. That is, for certain key subsets $\Gamma \subseteq \mathcal{A}$, we provide a complete description of *bad* cycles such that an instance of CSP(Γ) is satisfiable if and only if it does not contain a bad cycle. This allows us to show that MinCSP(Γ) is equivalent to the problem of finding a minimum set of arcs that hit every bad cycle in an arc-labelled mixed graph. We prove that there are seven inclusion-wise maximal subsets Γ of $\mathcal{A}$ such that MinCSP(Γ) is in FPT, and that MinCSP(Γ) is W[1]-hard in all other cases. We show that MinCSP($\mathcal{A}$) is not approximable in polynomial time within *any* constant under the UGC. In fact, we prove this to hold for MinCSP(r) whenever $\mathsf{r} \in \mathcal{A} \setminus \{\equiv\}$. As a response to this, we suggest the use of *fixed-parameter approximation algorithms*. We show that MinCSP($\mathcal{A}$) admits such an algorithm with approximation ratio 2 and a substantially faster algorithm with approximation ratio 4. We describe the results in greater detail below.

Intractability results. Our intractability results are based on novel W[1]-hardness results for a variety of natural *paired* and *simultaneous* cut and separation problems, which we believe to be of independent interest. Here, the input consists of two (directed or undirected) graphs and the task is to find a "generalized" cut that extends to both graphs. The two input graphs share some arcs/edges that can be deleted simultaneously at unit cost, and the goal is to compute a set of k arcs/edges that is a cut in both graphs. Both paired and simultaneous problems have recently received attention from the parameterized complexity community [1, 2, 37]. In the FPT/W[1]-hardness dichotomy for the Boolean domain [37], the fundamental difficult problem is Paired Cut (proven to be W[1]-hard by Marx and Razgon [47]): given an integer k and a directed graph G with two terminals $s, t \in V(G)$ and some arcs grouped into *pairs*, delete at most k pairs to cut all paths from s to t. An intuitive reason why Paired Cut is difficult can be seen as follows. Assume G contains two long

st-paths P and Q and the arcs of P are arbitrarily paired with the arcs of Q. Then, one cuts both paths with a cost of only one pair, but the arbitrary pairing of the arcs allow us to encode an arbitrary permutation – which is very powerful for encoding edge-choice gadgets when reducing from Multicoloured Clique. Our strategy for proving W[1]-hardness of paired problems elaborates upon this idea. In this case, the needed gadgets are quite succinct and their construction is simplified by the fact that we can recycle ideas from [24]. Our hardness proofs for simultaneous problems also use the same underlying idea, but they are much more complicated. The simultaneous setting is obviously not as versatile as the arbitrary pairing of Paired Cut. However, the possibility of choosing common arcs/edges while deleting them at unit costs still leaves enough freedom to encode arbitrary permutations. Such permutations enable us to construct edge-choice gadgets by "simulating" the low-level features that are available for free in paired problems. The resulting construction is complex and appears to contain ideas that may be useful for proving hardness of other kinds of simultaneous problems.

Altogether, we obtain novel W[1]-hardness results for Paired Cut Feedback Arc Set, Simultaneous st-Separator, Simultaneous Directed st-Cut, and Simultaneous Directed Feedback Arc Set. This allows us to identify six intractable fragments that are subsets of basic interval relations: $\{\mathsf{m}, \mathsf{r}_1\}$ for $\mathsf{r}_1 \in \{\equiv, \mathsf{s}, \mathsf{f}\}$, $\{\mathsf{d}, \mathsf{r}_2\}$ for $\mathsf{r}_2 \in \{\mathsf{o}, \mathsf{p}\}$ and $\{\mathsf{p}, \mathsf{o}\}$. The hardness reduction for $\{\mathsf{m}, \mathsf{r}_1\}$, is based on the hardness of Paired Cut and Paired Cut Feedback Arc Set. The other hardness results are based on reductions from Simultaneous Directed Feedback Arc Set, whose W[1]-hardness is shown by a reduction from Simultaneous st-Separator. The reductions from Simultaneous Directed Feedback Arc Set are non-trivial but both their presentation and their correctness proofs are highly simplified by our concrete descriptions of bad cycles.

Tractability results. We identify seven maximal tractable sets of basic interval relations: $\{\mathsf{m}, \mathsf{p}\}$ and $\{r_1, r_2, \equiv\}$ for $r_1 \in \{\mathsf{s}, \mathsf{f}\}$ and $r_2 \in \{\mathsf{p}, \mathsf{d}, \mathsf{o}\}$. All problems are handled by reductions to variants of Directed Feedback Arc Set (DFAS). DFAS and variations are extensively studied in parameterized complexity [6, 12, 16, 17, 29, 30, 45, 46]. DFAS is equivalent to MinCSP$(<)$, and a variant particularly important in our work is Sub-DFAS equivalent to MinCSP$(<, \leq)$, where the goal is to destroy only directed cycles that have a $<$-arc. To show that MinCSP(m, p) is in FPT, we use a straightforward reduction to MinCSP$(<, =)$ which, in turn, reduces to MinCSP$(<, \leq)$.

For the remaining six tractable cases, we reduce them all to a new variant of DFAS called Mixed Feedback Arc Set with Short and Long Arcs (LS-MFAS). The input is a mixed graph with edges and long and short arcs. Forbidden cycles in this graph are of two types: (1) cycles with at least one short arc, no long arcs, and all short arcs in the same direction, and (2) cycles with at least one long arc, all long arcs in the same direction, but the short arcs can be traversed in arbitrary direction. One intuitive way to think about the problem is to observe that a graph G has no forbidden cycle if there exists a placement of the vertices on the number line with certain distance constraints represented by the edges and arcs. Vertices connected by edges (which correspond to $\equiv$-constraints) should be placed at the same point. If there is an arc (u, v), we need to place u before v. Moreover, if the arc is long, then the distance from u to v should be big (say, greater than twice the number of vertices), while is the arc is short, then the distance from u to v should be small (say, at most 1). The reduction from MinCSP$(\mathsf{r}_1, \mathsf{r}_2, \equiv)$ to LS-MFAS creates a mixed graph with edges for $\equiv$-constraints, short arcs for r_1-constraints and long arcs for r_2-constraint. For correctness, consider for example the case with $\mathsf{r}_1 = \mathsf{s}$ and $\mathsf{r}_2 = \mathsf{p}$. Forbidden cycles of the first and the second kind imply an unsatisfiable order on the right and left endpoints, respectively. On the other hand, if bad cycles are absent, we can assign intervals as follows: if two variables are $\equiv$-connected, they are assigned the same interval, if they are s-connected, their intervals have the same left endpoints, left endpoints are ordered according to p-constraints, and the right endpoints of s-connected intervals are ordered according to the s-constraints.

Our algorithm for LS-MFAS builds upon the algorithm of [39] for Sub-DFAS. By iterative compression (see e.g. Chapter 4 in [21]) and branching, we may assume access to $k+1$ vertices that intersect all forbidden cycles, and we know their relative positions in the graph obtained after deleting a hypothetical optimal solution. The aforementioned "placing on a line" way of phrasing the lack of forbidden cycles is the main reason why this leads to a complete algorithm. In the next step, we try to place all remaining vertices relative to the terminals while breaking at most k distance constraints. Note that the number of ways in which a vertex can relate to a terminal is constant: it may be placed at a short/long distance before/after the terminal, or in the same position. Thus, we can define $O(k)$ types for each vertex, and the types determine whether distance constraints are satisfied or not. The optimal type assignment is then obtained by a reduction to Bundled Cut with pairwise-linked deletable arcs, a workhorse problem shown to be fixed-parameter tractable in [37]. We remark that our algorithms also handle the weighted versions of the problems.

Approximation results. In response to the negative complexity results for MinCSP($\mathcal{A}$), we consider approximation algorithms. We show that MinCSP($\mathcal{A}$) is not approximable in polynomial time within any constant under the UGC. We relax the restrictions even more by allowing our approximation algorithms to run in fixed-parameter tractable time. We show that MinCSP($\mathcal{A}$) admits such an algorithm with $c = 2$ and a substantially faster algorithm with $c = 4$. Hence, fpt-approximation is *much* more powerful than ordinary polynomial-time approximation in this case. These results are based on the observation that every relation in $\mathcal{A}$ can be defined as a conjunction of $\{<, =\}$-constraints on the endpoints. In the relaxation, we disregard conjunctions and view all $\{<, =\}$-constraints as an instance of MinCSP($<, =$), which is then reduced to Sub-DFAS. By invoking the Sub-DFAS algorithm of [39], one obtains a 2-approximation algorithm for the weighted variant of the problem.

Roadmap. We present the necessary preliminaries in Section 2. Section 3 is a bird's eye view of our results for the parameterized complexity and approximability of MinCSP(Γ) and the technical details are collected in the following sections. We describe the minimal obstructions to satisfiability for certain subsets of $\mathcal{A}$ in Section 4. These results are essential for connecting the MinCSP($\mathcal{A}$) problem with the graph-oriented view that we use. We complete our dichotomy result by a number of fixed-parameter algorithms in Section 5 and a collection of W[1]-hardness results in Section 6. We conclude the paper with a discussion of our results and future research directions in Section 7. This is a shortened version of the full paper, which can be found on arXiv [23].

2 Preliminaries

In this section, we briefly present the rudiments of parameterized complexity, define the CSP and MinCSP problems, and provide some basics concerning interval relations. Before we begin, we need some terminology and notation for graphs. Let G be a (directed or undirected) graph; we allow graphs to contain loops. We denote the set of vertices in G by $V(G)$. If G is undirected, then $E(G)$ denotes the set of edges in G. If G is directed, then $A(G)$ denotes the

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

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

set of arcs in G, and $E(G)$ denotes the set of edges in the underlying undirected graph of G. We use uv to denote an undirected edge with end-vertices u and v. We use (u, v) to denote a directed arc from u to v; u is the *tail* and v is the *head*. For $X \subseteq E(G)$, we write $G - X$ to denote the directed graph obtained by removing all edges/arcs corresponding to X from G if G is undirected and $A(G - X) = A(G) \setminus \{(u, v), (v, u) \mid \{u, v\} \in X\}$ if G is directed. If $X \subseteq V(G)$, then we let $G - X = G[V(G) \setminus X]$ be the subgraph induced in G by $V(G) \setminus X$. An st-cut in G is a set of edges/arcs X such that the vertices s and t are separated in $G - X$.

A *parameterized* problem is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is the input alphabet. The parameterized complexity class FPT contains the problems decidable in $f(k) \cdot n^{O(1)}$ time, where f is a computable function and n is the instance size. Reductions between parameterized problems need to take the parameter into account. To this end, we use *parameterized reductions* (or 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 a parameterized 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 $k' \leq g(k)$. We will sometimes prove that certain problems are not in FPT. 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].

We continue by defining CSPs. A *constraint language* Γ is a set of relations over a domain D. Each relation $R \in \Gamma$ has an associated *arity* $r \in \mathbb{N}$ and $R \subseteq D^r$. All relations considered in this paper are binary and all constraint languages are finite. An instance $\mathcal{I}$ of CSP(Γ) consists of a set of variables $V(\mathcal{I})$ and a set of constraints $C(\mathcal{I})$ of the form $R(x, y)$, where $R \in \Gamma$ and $x, y \in V(\mathcal{I})$. To simplify notation, we may write $R(x, y)$ as xRy. An assignment $\varphi : V(\mathcal{I}) \to D$ *satisfies* a constraint $R(x, y)$ if $(\varphi(x), \varphi(y)) \in R$ and *violates* $R(x, y)$ if $(\varphi(x), \varphi(y)) \notin R$. The assignment φ is a *satisfying assignment* (or a *solution*) if it satisfies every constraint in $C(\mathcal{I})$.

CSP(Γ)

INSTANCE:	An instance $\mathcal{I}$ of CSP(Γ).
QUESTION:	Does $\mathcal{I}$ admit a satisfying assignment?

The *value* of an assignment φ for $\mathcal{I}$ is the number of constraints in $C(\mathcal{I})$ satisfied by φ. For any subset of constraints $X \subseteq C(\mathcal{I})$, let $\mathcal{I} - X$ denote the instance with $V(\mathcal{I} - X) = V(\mathcal{I})$ and $C(\mathcal{I} - X) = C(\mathcal{I}) \setminus X$. The (parameterized) *almost constraint satisfaction problem* (MINCSP(Γ)) is defined as follows:

MINCSP(Γ)

INSTANCE:	An instance $\mathcal{I}$ of CSP(Γ) and an integer k.
PARAMETER:	k.
QUESTION:	Is there a set $X \subseteq C(\mathcal{I})$ such that $\lvert X\rvert \leq k$ and $\mathcal{I} - X$ is satisfiable?

Next, we review the basics of *Allen's interval algebra* [3] (IA). Its domain is the set $\mathbb{I}$ of all pairs $(x, y) \in \mathbb{Q}^2$ such that $x < y$, i.e. $\mathbb{I}$ can be viewed as the set of all closed intervals $[a, b]$ of rational numbers. If $I = [a, b] \in \mathbb{I}$, then we write I^- for a and I^+ for b. Let $\mathcal{A}$ denote the set of 13 *basic* relations that are presented in Table 1, and let $2^{\mathcal{A}}$ denote the 8192 binary relations that can be formed by taking unions of relations in $\mathcal{A}$. The complexity of CSP(Γ) is known for every $\Gamma \subseteq 2^{\mathcal{A}}$ [41] and in each case CSP(X) is either polynomial-time solvable or NP-complete. In particular, CSP($\mathcal{A}$) is in P. When considering subsets $\Gamma \subseteq \mathcal{A}$, note that any constraint $x\mathsf{r}\mathsf{i}y$ is equivalent to $y\mathsf{r}x$ for $\mathsf{r} \in \{\mathsf{p}, \mathsf{m}, \mathsf{o}, \mathsf{d}, \mathsf{s}, \mathsf{f}\}$, so we may assume that $\mathsf{r} \in \Gamma$ if and only if $\mathsf{ri} \in \Gamma$. Furthermore, for the remainder of the paper, we may assume $\mathcal{A} = \{\mathsf{p}, \mathsf{m}, \mathsf{o}, \mathsf{d}, \mathsf{s}, \mathsf{f}, \equiv\}$.

When studying MINCSP and its parameterized complexity, it is convenient to allow *crisp* constraints, i.e. constraints that cannot be deleted. Formally, for a language $\Gamma \subseteq \mathcal{A}$ and a relation $\mathsf{r} \in \Gamma$, we say that Γ *supports crisp* r*-constraints* if, for every value of the parameter $k \in \mathbb{N}$, we can construct an instance $\mathcal{I}_{\mathsf{r}}$ of MINCSP(Γ) with variables $x, y \in V(\mathcal{I}_{\mathsf{r}})$ (and possibly some auxiliary variables) such that the constraint $x\mathsf{r}y$ is equivalent to $\mathcal{I}_{\mathsf{r}} - X$ for all $X \subseteq C(\mathcal{I})$ such that $\lvert X\rvert \leq k$. Then, if we want to enforce a constraint $x\mathsf{r}y$ in an instance of MINCSP(Γ), we can use $\mathcal{I}_{\mathsf{r}}$ with fresh variables $V(\mathcal{I}) \setminus \{x, y\}$ in its place. Straightforward reasoning about interval constraints readily shows that every $\mathsf{r} \in \mathcal{A}$ supports crisp constraints, and this also holds for the constraint language $\{<, =\}$.

3 Overview

In this section we prove the dichotomy theorem for the parameterized complexity of MINCSP(Γ) for every subset $\Gamma \subseteq \mathcal{A}$ of interval relations. We also discuss constant-factor approximation algorithms for MINCSP($\mathcal{A}$). Some observations reduce the number of subsets of relations that we need to consider in the classification. For the first one, we need a simplified definition of *implementations*. More general definitions are used in e.g. [33] and [35].

▶ **Definition 1.** *Let Γ be a constraint language and* r *be a binary relation over the same domain. A* (simple) implementation *of a relation* r *in* Γ *is an instance* $\mathcal{C}_{\mathsf{r}}$ *of CSP(Γ) with* primary *variables* x_1, x_2 *and, possibly,* auxiliary *variables* $y_1, \ldots, y_\ell$ *such that:*

- *if an assignment φ satisfies $\mathcal{C}_{\mathsf{r}}$, then it satisfies the constraint $x_1\mathsf{r}x_2$;*
- *if an assignment φ' does not satisfy $x_1\mathsf{r}x_2$, then it cannot be extended to the auxiliary variables $y_1, \ldots, y_\ell$ so that it satisfies $\mathcal{C}_{\mathsf{r}}$.*
- *if an assignment φ' does not satisfy $x_1\mathsf{r}x_2$, then it can be extended to the auxiliary variables $y_1, \ldots, y_\ell$ so that all but one constraint in $\mathcal{C}_{\mathsf{r}}$ are satisfied.*

In this case we say that Γ implements r.

Intuitively, we can replace every occurrence of a constraint xry with its implementation in Γ while preserving the cost of making the instance satisfiable. This intuition is made precise in the following lemma, and identifying the two implementations in Lemma 3 is left to the reader.

▶ **Lemma 2** (Proposition 5.2 in [35]). *Let Γ be a constraint language that implements a relation* r. *If* MinCSP(Γ) *is in* FPT*, then so is* MinCSP$(\Gamma \cup \{\mathsf{r}\})$. *If* MinCSP$(\Gamma \cup \{\mathsf{r}\})$ *is* W*[1]-hard, then so is* MinCSP(Γ).

▶ **Lemma 3** (Implementations). *Let $\Gamma \subseteq \mathcal{A}$ be a subset of interval relations. If Γ contains* m*, then Γ implements* p*, and if Γ contains* f *and* s*, then Γ implements* d *and* o.

Another observation utilizes the symmetry of interval relations. By switching the left and the right endpoints of all intervals in an instance $\mathcal{I}$ of MinCSP$(\mathcal{A})$ and then negating their values, we obtain a *reversed instance* $\mathcal{I}^R$. Formally, instance $\mathcal{I}^R$ of CSP$(\mathcal{A})$ has the same set of variables as $\mathcal{I}$, and contains a constraint $uf(\mathsf{r})v$ for every $u\mathsf{r}v$ in $C(\mathcal{I})$, where $f : \mathcal{A} \to \mathcal{A}$ is defined as $f(\mathsf{r}) = \mathsf{ri}$ for $\mathsf{r} \in \{\mathsf{m}, \mathsf{p}, \mathsf{o}\}$, $f(\equiv) = \equiv$, $f(\mathsf{d}) = \mathsf{d}$, $f(\mathsf{s}) = \mathsf{f}$ and $f(\mathsf{f}) = \mathsf{s}$.

▶ **Lemma 4** (Lemma 4.2 of [41]). *An instance $\mathcal{I}$ of CSP$(\mathcal{A})$ is satisfiable if and only if the reversed instance $\mathcal{I}^R$ is satisfiable.*

To obtain our results, we use combinatorial tools and represent an instance $\mathcal{I}$ of CSP$(\mathcal{A})$ as an *arc-labelled mixed graph* $G_{\mathcal{I}}$, i.e. a graph that contains edges for symmetric constraints and labelled arcs for asymmetric ones. More precisely, the graph $G_{\mathcal{I}}$ is obtained by introducing all variables of $\mathcal{I}$ as vertices, adding directed arcs (u, v) labelled with $\mathsf{r} \in \mathcal{A} \setminus \{\equiv\}$ for every constraint $u\mathsf{r}v$ in $C(\mathcal{I})$, and undirected edges uv for every constraint $u \equiv v$ in $\mathcal{I}$. Note that $G_{\mathcal{I}}$ may have parallel arcs with different labels and may contain loops. The undirected graph underlying $G_{\mathcal{I}}$ is called the *primal graph* of $\mathcal{I}$; we allow the primal graph to contain loops and parallel edges (in both cases, this will mean the primal graph contains a cycle). The advantage of the graph representation is supported by the following lemma:

▶ **Lemma 5** (Cycles). *Let $\mathcal{I}$ be an inclusion-wise minimal unsatisfiable instance of CSP$(\mathcal{A})$ (i.e. removing any constraint of $\mathcal{I}$ results in a satisfiable instance). Then the primal graph of $\mathcal{I}$ is a cycle.*

The proof of the lemma is deferred to Section 4. All cycles discussed in the rest of the section are cycles of the primal graph. From the combinatorial point of view, minimal unsatisfiable instances are *bad cycles* in the labelled graph. For example, in MinCSP(p), the bad cycles correspond to the directed cycles. For MinCSP$(\mathsf{p}, \equiv)$, the bad cycles contain at least one p-arc and all p-arcs in the same direction. Thus, MinCSP(Γ) can now be cast as a certain feedback edge set problem – our goal is to find a set of k edges in the primal graph that intersects all bad cycles. We present such a characterization for several cases in Section 4.

Our algorithmic results can be summarized as follows.

▶ **Lemma 6.** MinCSP(m, p) *and* MinCSP$(\mathsf{r}_1, \mathsf{r}_2, \equiv)$ *are in* FPT *for* $\mathsf{r}_1 \in \{\mathsf{s}, \mathsf{f}\}$ *and* $\mathsf{r}_2 \in \{\mathsf{p}, \mathsf{o}, \mathsf{d}\}$.

The algorithm for MinCSP(m, p) is obtained using a simple reduction to Subset Directed Feedback Arc Set.

Subset Directed Feedback Arc Set (Sub-DFAS)	
Instance:	A directed graph G, a subset of red arcs $R \subseteq A(G)$, and an integer k.
Parameter:	k.
Question:	Is there a subset $Z \subseteq A(G)$ of size at most k such that $G - Z$ contains no directed cycles with at least one red arc?

Chitnis et al. [17] have proved that Sub-DFAS is solvable in $O^*(2^{O(k^3)})$ time. The algorithm for the remaining cases is more complicated and relies on the bad cycle characterization in Section 4 and a sophisticated modification of the algorithm for Sub-DFAS from [39] in Section 5.

For the negative results, we start by proving W[1]-hardness for certain paired and simultaneous graph cut problems, and we identify $\Gamma \subseteq \mathcal{A}$ such that paired or simultaneous problems reduce to MinCSP(Γ). For intuition, consider a constraint $x \equiv y$. If we consider the left and the right endpoints separately, then $\equiv$ implies two equalities: $x^- = y^-$ and $x^+ = y^+$. Together with another relation (e.g. m), this double-equality relation can be used to encode the pairing of the edges of two graphs (namely, the left-endpoint graph and the right-endpoint graph). We note that the double-equality relation is also the cornerstone of all hardness results in the parameterized complexity classification of Boolean MinCSP [37]. Lemma 7 is based on paired problems and Lemma 8 is based on simultaneous problems.

▶ **Lemma 7.** *MinCSP*$(\mathsf{m}, \equiv)$*, MinCSP*(m, s) *and MinCSP*(m, f) *are* W[1]*-hard.*

▶ **Lemma 8.** *MinCSP*(d, o)*, MinCSP*(p, o) *and MinCSP*(d, p) *are* W[1]*-hard.*

Combining all results above, we are ready to present the full classification.

▶ **Theorem 9** (Full classification). *Let $\Gamma \subseteq \mathcal{A}$ be a subset of interval relations. Then MinCSP(Γ) is in* FPT *if $\Gamma \subseteq \{\mathsf{m}, \mathsf{p}\}$ or $\Gamma \subseteq \{\mathsf{r}_1, \mathsf{r}_2, \equiv\}$ for any $\mathsf{r}_1 \in \{\mathsf{s}, \mathsf{f}\}$ and $\mathsf{r}_2 \in \{\mathsf{p}, \mathsf{o}, \mathsf{d}\}$, and* W[1]*-hard otherwise.*

W[1]-hardness of MinCSP($\mathcal{A}$) motivates us to look at approximation algorithms for this problem. Our first observation is that MinCSP(r) for any $\mathsf{r} \in \mathcal{A} \setminus \{\equiv\}$ is NP-hard to approximate within any constant under the Unique Games Conjecture (UGC) of Khot [34]. This follows by combining two facts: Lemma 11, which implies that an instance $\mathcal{I}$ of CSP(r) is satisfiable if and only if the arc-labelled graph $G_{\mathcal{I}}$ is acyclic, and Corollary 1.2 in [32], which states that under the UGC, Directed Feedback Arc Set (DFAS) is NP-hard to approximate within any constant [32]. If we allow the approximation algorithm to run in fpt time, then we obtain the following result.

▶ **Theorem 10.** *MinCSP($\mathcal{A}$) is 2-approximable in $O^*(2^{O(k^3)})$ time and 4-approximable in $O^*(2^{O(k)})$ time.*

Proof sketch. We obtain the algorithms by reducing the problem to Sub-DFAS and invoking the exact algorithm of [21] and the faster $O^*(2^{O(k)})$ time 2-approximation algorithm of [44], respectively. There are straightforward reductions from MinCSP($<, =$) to MinCSP($\leq, =$) to Sub-DFAS, so we focus on the reduction from MinCSP($\mathcal{A}$) to MinCSP($<, =$). Let $(\mathcal{I}, k)$ be an instance of MinCSP($\mathcal{A}$). Replace every constraint $x\{\mathsf{o}\}y$ by its implementation in $\{\mathsf{s}, \mathsf{f}\}$ according to Lemma 3. By Lemma 2, this does not change the cost of the instance. Using Table 1, we can rewrite all constraints of $\mathcal{I}'$ as conjunctions of two atomic constraints of the form $x < y$ or $x = y$. Disregarding the pairing, let S be the set of all atomic constraints. Apply one of the MinCSP($<, =$) algorithms to $(S, 2k)$. On the one hand, deleting k constraints from $\mathcal{I}'$ corresponds to deleting at most $2k$ constraints in S. On the other hand, if there is $X \subseteq S$, $|X| \leq 2k$, such that $S - X$ is satisfiable, define the set of interval constraints X' such that at least one of the defining $\{<, =\}$-constraints is in X. Noting that $\mathcal{I} - X'$ is satisfiable and $|X'| \leq |X| \leq 2k$ completes the proof. ◀

4 Bad Cycles

In this section, we sketch the proof of Lemma 5 and describe the minimal obstructions to satisfiability for certain subsets of $\mathcal{A}$, along with a brief sketch of why these are the minimal obstructions.

▶ **Lemma 5** (Cycles). *Let $\mathcal{I}$ be an inclusion-wise minimal unsatisfiable instance of CSP($\mathcal{A}$) (i.e. removing any constraint of $\mathcal{I}$ results in a satisfiable instance). Then the primal graph of $\mathcal{I}$ is a cycle.*

The proof of Lemma 5 starts by taking a minimal unsatisfiable instance $\mathcal{I}$. Using Table 1, we write $\mathcal{I}$ as an instance $\mathcal{I}'$ of the *point algebra* (PA) [53] CSP, which takes rationals $\mathbb{Q}$ as the variable domain and we use only the basic constraint language $\{<, =\}$, where the relations are interpreted in the obvious way. This instance $\mathcal{I}'$ must contain a minimal unsatisfiable sub-instance $\mathcal{I}''$ of the point algebra, which has a cycle as its primal graph. We then map the constraints in $\mathcal{I}''$ back to the constraints in $\mathcal{I}$ that implied them, and find that $\mathcal{I}$ must also have a cycle as its primal graph.

▶ **Lemma 11** (Bad Cycles). *Let $\mathcal{I}$ be an instance of CSP($\mathsf{r}_1, \mathsf{r}_2$) for some $\mathsf{r}_1, \mathsf{r}_2 \in \mathcal{A}$, and consider the arc-labelled mixed graph $G_{\mathcal{I}}$. Then $\mathcal{I}$ is satisfiable if and only if $G_{\mathcal{I}}$ does not contain any bad cycles described below.*

1. *If $\mathsf{r}_1 = \mathsf{d}$ and $\mathsf{r}_2 = \mathsf{p}$, then the bad cycles are cycles with p-arcs in the same direction and no d-arcs meeting head-to-head.*
2. *If $\mathsf{r}_1 = \mathsf{d}$ and $\mathsf{r}_2 = \mathsf{o}$, then the bad cycles are cycles with all d-arcs in the same direction and all o-arcs in the same direction (the direction of the d-arcs may differ from that of the o-arcs).*
3. *If $\mathsf{r}_1 = \mathsf{o}$ and $\mathsf{r}_2 = \mathsf{p}$, then the bad cycles are (a) directed cycles of o-arcs and (b) cycles with all p-arcs in the forward direction, with every consecutive pair of o-arcs in the reverse direction separated by a p-arc (this case includes directed cycles of p-arcs).*
4. *If $\mathsf{r}_1 \in \{\mathsf{f}, \mathsf{s}\}$ and $\mathsf{r}_2 \in \{\mathsf{d}, \mathsf{o}, \mathsf{p}\}$, then the bad cycles are (a) directed cycles of r_1-arcs and (b) cycles with at least one r_2-arc and all r_2-arcs in the same direction (and r_1-arcs directed arbitrarily).*

5 FPT Algorithms

We prove Lemma 6 in this section. The fpt algorithm for $\{\mathsf{m}, \mathsf{p}\}$ is simple and we omit the details: it works by first reducing the problem to MinCSP($<, =$) and then to Sub-DFAS. The remaining six cases are handled by reducing them to a fairly natural generalization of Directed Feedback Arc Set problem, and showing that this problem is in FPT.

Lemma 11.4 suggests that all six remaining fragments allow uniform treatment. Indeed, to check whether an instance of CSP($\mathsf{r}_1, \mathsf{r}_2, \equiv$) is satisfiable, one can identify all variables constrained to be equal. This corresponds exactly to contracting all edges in the graph $G_{\mathcal{I}}$. Then $\mathcal{I}$ becomes an instance of CSP($\mathsf{r}_1, \mathsf{r}_2$), and the criterion of Lemma 11.4 applies. This observation allows us to formulate MinCSP($\mathsf{r}_1, \mathsf{r}_2, \equiv$) as a variant of feedback arc set on mixed graphs.

▶ **Definition 12.** *Consider a mixed graph G with arcs of two types – short and long – and a walk W in G from u to v that may ignore direction of the arcs. The walk W is* undirected *if it only contains edges, it is* short *if it contains a short arc but no long arcs, and it is* long *if it contains a long arc. The walk W is* directed *if it is either short and all short arcs are directed from u to v or if it is long and all long arcs are directed from u to v. If W is short or long, but not directed, it is* mixed.

Note that short arcs on a long-directed walk may be directed arbitrarily.

Mixed Feedback Arc Set with Short and Long Arcs (LS-MFAS)	
Instance:	A mixed graph G with the arc set $A(G)$ partitioned into *short* A_s and *long* A_ℓ, and an integer k.
Parameter:	k.
Question:	Is there a set $Z \subseteq E(G) \cup A(G)$ with $\|Z\| \leq k$ such that $G - Z$ contains neither short-directed cycles nor long-directed cycles?

The main result of this section is the following theorem.

▶ **Theorem 13.** *LS-MFAS can be solved in $O^*(2^{O(k^8 \log k)})$ time.*

We see that Lemma 11.4 and Theorem 13 imply $\textsc{MinCSP}(\mathsf{r}_1, \mathsf{r}_2, \equiv)$ being in FPT whenever $\mathsf{r}_1 \in \{\mathsf{s}, \mathsf{f}\}$ and $\mathsf{r}_2 \in \{\mathsf{p}, \mathsf{o}, \mathsf{d}\}$. It is informative to understand the structure of mixed graphs without bad cycles in the sense of LS-MFAS. The proof of the following lemma is fairly easy with the placing-vertices-on-the-number-line intuition from the introduction.

▶ **Lemma 14.** *Let G be a mixed graph with long and short arcs. Then G contains no long-directed cycles nor short-directed cycles if and only if there exists a pair of mappings $\sigma_1, \sigma_2 : V(G) \to \mathbb{N}$ such that*

1. *for every $u, v \in V(G)$, u and v are connected by an undirected walk if and only if $(\sigma_1, \sigma_2)(u) = (\sigma_1, \sigma_2)(v)$;*
2. *for every $u, v \in V(G)$, there exists a short (u, v)-walk in G if and only if $\sigma_1(u) = \sigma_1(v)$;*
3. *for every $u, v \in V(G)$, if there exists a short-directed (u, v)-walk in G, then $\sigma_2(u) < \sigma_2(v)$;*
4. *for every $u, v \in V(G)$, if there exists a long-directed (u, v)-walk in G, then $\sigma_1(u) < \sigma_1(v)$.*

We now introduce the technical machinery used in our algorithm for LS-MFAS. We start by using iterative compression, a standard method in parameterized algorithms (see e.g. Chapter 4 in [21]). This allows us to assume access to a set of $k+1$ edges and arcs intersecting every bad cycle. The problem resulting from iterative compression reduces to Bundled Cut with pairwise-linked deletable edges, defined in [37] and solved using the flow-augmentation technique of [36]. To describe Bundled Cut, let G be a directed graph with two distinguished vertices $s, t \in V(G)$. Let $\mathcal{B}$ be a family of pairwise disjoint subsets of $E(G)$, which we call *bundles*. The edges of $\bigcup \mathcal{B}$ are *soft* and the edges of $E(G) \setminus \bigcup \mathcal{B}$ are *crisp*. A set $Z \subseteq \bigcup \mathcal{B}$ *violates* a bundle $B \in \mathcal{B}$ if $Z \cap B \neq \emptyset$ and *satisfies* B otherwise.

Bundled Cut	
Instance:	A directed graph G, two distinguished vertices $s, t \in V(G)$, a family $\mathcal{B}$ of pairwise disjoint subsets of $E(G)$, and an integer k.
Parameter:	k.
Question:	Is there an st-cut $Z \subseteq \bigcup \mathcal{B}$ that violates at most k bundles?

Table 2 Correspondence between edges, short arcs and long arcs of the LS-MFAS instance and the arcs introduced in the reduction to Bundled Cut in Theorem 13.

	i odd	i even, j odd	i even, j even
Edge uv	$(i,j) \to (i,j)$ $(i,j) \leftarrow (i,j)$	$(i,j) \to (i,j)$ $(i,j) \leftarrow (i,j)$	$(i,j) \to (i,j)$ $(i,j) \leftarrow (i,j)$
Short (u,v)	$(i,j) \to (i,j)$ $(i,1) \leftarrow (i,j)$	$(i,j) \to (i,j)$ $(i,1) \leftarrow (i,j)$	$(i,j) \to (i,j+1)$ $(i,1) \leftarrow (i,j)$
Long (u,v)	$(i,1) \to (i,1)$	$(i,j) \to (i+1,1)$	$(i,j) \to (i+1,1)$

In general, Bundled Cut is W[1]-hard even if all bundles are of size 2. However, there is a special case of Bundled Cut that is tractable. Let $(G, s, t, \mathcal{B}, k)$ be a Bundled Cut instance. A soft arc e is *deletable* if there is no crisp copy of e in G. An instance $(G, s, t, \mathcal{B}, k)$ has *pairwise-linked deletable arcs* if for every $B \in \mathcal{B}$ and every two deletable arcs $e_1, e_2 \in B$, there exists in G a path from an endpoint of one of the arcs e_1, e_2 to an endpoint of the second of those arcs that does not use any arcs of $\mathcal{B} \setminus \{B\}$. The assumption of pairwise-linked deletable arcs makes Bundled Cut tractable.

▶ **Theorem 15** (Theorem 4.1 of [38]). *Bundled Cut instances with pairwise-linked deletable arcs can be solved in $O^*(2^{O(k^4 d^4 \log(kd))})$ time, where d is the maximum number of deletable arcs in a single bundle.*

Armed with Lemma 14 and Theorem 15, we are ready to prove the main result.

Proof of Theorem 13. Let (G, A_s, A_ℓ, k) be an instance of LS-MFAS. By iterative compression, we may assume that we have access to a set $Y \subseteq V(G)$ of size at most $k+1$ that intersects all bad cycles. We refer to the vertices of Y as *terminals*.

Fix a hypothetical solution $Z \subseteq A(G) \cup E(G)$. Guess which pairs of terminals are connected by undirected paths in $G - Z$ and identify them. Define an *ordering* $\sigma : Y \to \mathbb{N} \times \mathbb{N}$ that maps terminals to

$$\{(1,1), \dots, (1,q_1), \cdots, (i,1), \dots, (i,q_i), \cdots, (p,1), \dots, (p,q_p)\}$$

such that the following hold. For every pair of terminals $y, y' \in Y$, let $\sigma(y) = (i,j)$ and $\sigma(y') = (i',j')$ where (1) $i = i'$ if y and y' are connected by a short path in $G - Z$, (2) $j < j'$ if y reaches y' by a short-directed path in $G - Z$, and (3) $i < i'$ if y reaches y' by a long-directed path in $G - Z$. Note that σ exists by Lemma 14. If an ordering satisfies the conditions above, we say that it is *compatible with* $G - Z$. In what follows, we write $(i,j) < (i',j')$ to denote that (i,j) lexicographically precedes (i',j'), i.e. either $i = i'$ and $j < j'$ or $i < i'$.

For the algorithm, proceed by guessing an ordering σ, creating $2^{O(k \log k)}$ branches in total. For each σ, create an instance $(H := H(G,\sigma), \mathcal{B} := \mathcal{B}(G,\sigma), k)$ of Bundled Cut as follows. Introduce two distinguished vertices s and t in H. For every vertex $v \in V(G)$, create vertices v^i_1 in H for all odd $i \in [2p+1]$ and vertices v^i_j in H for all even $i \in [2p+1]$ and all $j \in [2q_i + 1]$. Connect the vertices created above by *downward* arcs $(v^i_j, v^{i'}_{j'})$ for all $(i,j) > (i',j')$. For every terminal y, let $\sigma(y) = (i,j)$, and add arcs (s, y^{2i}_{2j}) and (y^{2i}_{2j+1}, t) in H. Using the rules below, create a bundle B_e in $\mathcal{B}$ for every $e \in E(G) \cup A(G)$, add the newly created arcs to H.

- For an edge $e = uv$, let B_e consist of the arcs (u^i_j, v^i_j) and (v^i_j, u^i_j) for all (i, j).
- For short arcs $e = (u, v)$, let B_e consist of the arcs (u^i_j, v^i_j) for all i, j such that i or j is odd, the arcs (u^i_j, v^i_{j+1}) for all even i, j, and the arcs (v^i_j, u^i_1) for all i, j.
- For long arcs $e = (u, v)$, let B_e consist of the arcs (u^i_1, v^i_1) for all odd i, and arcs the arcs (u^i_j, v^{i+1}_1) for all even i and all j.

This completes the construction. Bundle construction rules are summarized in Table 2. Observe that the downward arcs ensure that $(H, \mathcal{B}, k)$ has the pairwise-linked deletable arc property. Moreover, the bundle size is $O(k)$, so we can solve $(H, \mathcal{B}, k)$ in $O^*(2^{O(k^8 \log k)})$ time.

We now sketch the correctness argument. Fix a guessed ordering σ. For any candidate solution W in $(H := H(G, \sigma), \mathcal{B} = \mathcal{B}(G, \sigma), k)$, the existence of downward arcs imply that for every $v \in V(G)$ there is a threshold (i_v, j_v) such that v^j_i is reachable from s in $H - W$ if and only if $(i, j) \leq (i_v, j_v)$. This threshold is meant to indicate that v should be placed on the line somewhere around the terminal x for which $\sigma(x) = (\lfloor i_v/2 \rfloor, \lfloor j_v/2 \rfloor)$. A short walk from u to v in G projects, for every even i and even j, to a walk from u^j_i to v^{j+1}_i. A long walk from u to v in G projects, for every odd i, to a walk from u^1_i to v^1_i. Together with the fact that terminals intersect all forbidden cycles in G, this gives a correspondence between forbidden cycles in G and st-paths in H. ◀

6 W[1]-hard Problems

Here, we show Lemmas 7 and 8. As the first and most challenging step, we show W[1]-hardness for variants of paired and simultaneous graph cut problems from which we then reduce to the hard variants of MinCSP(Γ). Our reductions will make use of the following well-known problem, whose W[1]-hardness follows by a simple reduction from Multicoloured Clique (see e.g. Exercise 13.3 in [21]).

Multicoloured Biclique (MC-BiClique)	
Instance:	An undirected graph G with a partition $V(G) = A_1 \uplus \ldots \uplus A_k \uplus B_1 \uplus \ldots \uplus B_k$, where $\lvert A_i \rvert = \lvert B_i \rvert = n$ for each $i \in [k]$ and both $\uplus_{i \in [k]} A_i$ and $\uplus_{i \in [k]} B_i$ form independent sets in G.
Parameter:	k.
Question:	Does G contain $K_{k,k}$ as a subgraph, a.k.a. a *multicoloured biclique*?

6.1 Paired Problems

We consider the problems Paired Cut and Paired Cut Feedback Arc Set (PCFAS) in what follows.

Paired Cut	
Instance:	Undirected graphs G_1 and G_2, vertices $s_i, t_i \in V(G_i)$, a set of disjoint edge pairs $\mathcal{B} \subseteq E(G_1) \times E(G_2)$, and an integer k.
Parameter:	k.
Question:	Is there a subset $X \subseteq \mathcal{B}$ such that $\lvert X \rvert \leq k$ and $X_i = \{e_i \mid \{e_1, e_2\} \in X\}$ is an $\{s_i, t_i\}$-cut in G_i for both $i \in \{1, 2\}$?

The PCFAS problem is similar, but G_2 is directed and X_2 is required to be such that $G_2 - X_2$ is acyclic (instead of being an $\{s_2, t_2\}$-cut). We show W[1]-hardness of both problems. Since both reductions are from MC-BiClique and quite similar, we start to describe the common part of both reductions. Let $I = (G, A_1, \ldots, A_k, B_1, \ldots, B_k, k)$ be an instance of MC-BiClique. We define two directed graphs G_A and G_B as follows. G_A

contains the vertices s_A and t_A. Moreover, for every $i \in [k]$, G_A contains the vertices in $P_i^A = \{v_{i,1}, \dots, v_{i,n-1}\}$. For convenience, we let $v_{i,0} = s_A$ and $v_{i,n} = t_A$ for every $i \in [k]$. Moreover, for every vertex $a_{i,j}$ and every $i' \in [k]$, G_A contains the directed path $P^A_{i,j,i'}$ from $v_{i,j-1}$ to $v_{i,j}$ that has one edge (using fresh auxiliary vertices) for every edge between $a_{i,j}$ and a vertex in $B_{i'}$. Therefore, we may assume in what follows that there is a bijection between the edges of $P^A_{i,j,i'}$ and the edges between $a_{i,j}$ and a vertex in $B_{i'}$. This concludes the description of G_A. G_B is defined very similarly to G_A with the roles of the sets $A_1, \dots, A_k$ and $B_1, \dots, B_k$ being reversed.

Finally, define a set $\mathcal{B} \subseteq E(G_A) \times E(G_B)$ of bundles as follows. For every edge $e = \{a_{i,j}, b_{i',j'}\} \in E(G)$, $\mathcal{B}$ contains the pair (e^A, e^B), where e^A is the edge corresponding to e on the path $P^A_{i,j,i'}$ and e^B is the edge corresponding to e on the path $P^B_{i',j',i}$. This concludes the construction and the following lemma shows its main property.

▶ **Lemma 16.** *$I = (G, A_1, \dots, A_k, B_1, \dots, B_k, k)$ is a yes-instance of MC-BiClique if and only if there is a set $X \subseteq \mathcal{B}$ with $|X| = k^2$ and $X_A = \{e \mid (e, e') \in X\}$ is an (s_A, t_A)-cut in G_A and $X_B = \{e' \mid (e, e') \in X\}$ is an (s_B, t_B)-cut in G_B.*

The lemma above makes it relatively straightforward to show W[1]-hardness for Paired Cut and PCFAS.

▶ **Lemma 17.** *Paired Cut and PCFAS are W[1]-hard.*

6.2 Simultaneous Problems

In this section we prove W[1]-hardness of several simultaneous cut problems. Our basis is the following problem.

Simultaneous Separator (Sim-Separator)	
Instance:	Two directed graphs D_1 and D_2 with $V = V(D_1) = V(D_2)$, vertices $s, t \in V$, and an integer k.
Parameter:	k.
Question:	Is there a subset $X \subseteq V \setminus \{s, t\}$ of size at most k such that neither $D_1 - X$ nor $D_2 - X$ contains a path from s to t?

We begin by proving that this problem is W[1]-hard in Theorem 18. We will then prove that simultaneous variants of Directed st-Cut and Directed Feedback Arc Set are W[1]-hard via reductions from Sim-Separator; these results can be found in Theorems 19 and 20, respectively. It will be convenient to use the term *st-separator* when working with directed graphs: given a directed graph $G = (V, E)$ and two vertices $s, t \in V$, we say that $X \subseteq V \setminus \{s, t\}$ is an *st-separator* if the graph $G - X$ contains no directed path from s to t and no directed path from t to s.

▶ **Theorem 18.** *Sim-Separator is W[1]-hard even if both input digraphs are acyclic.*

We continue by using the W[1]-hardness of Sim-Separator to prove W[1]-hardness of the following two problems.

Simultaneous Directed st-Cut (Sim-Cut)	
Instance:	Two directed graphs D_1 and D_2 with $V = V(D_1) = V(D_2)$, vertices $s, t \in V$, and an integer k.
Parameter:	k.
Question:	Is there a subset $X \subseteq E(D_1) \cup E(D_2)$ of size at most k such that neither $D_1 - X$ nor $D_2 - X$ contains a path from s to t?

Simultaneous Directed Feedback Arc Set (Sim-DFAS)	
Instance:	Directed graphs D_1, D_2 with $V = V(D_1) = V(D_2)$, and an integer k.
Parameter:	k.
Question:	Is there a subset $X \subseteq E(D_1) \cup E(D_2)$ of size at most k such that both $D_1 - X$ and $D_2 - X$ are acyclic?

▶ **Theorem 19.** *Sim-Cut is* W[1]*-hard even if both input digraphs are acyclic.*

▶ **Theorem 20.** *Sim-DFAS is* W[1]*-hard.*

6.3 Intractable Fragments

We begin by proving that MinCSP($\mathsf{m}, \equiv$) and MinCSP(m, s) are W[1]-hard. We introduce two binary relations: let $\equiv^-$ denote the *left-equals* relation and $\equiv^+$ denote the *right-equals* relation, which hold for any pair of intervals with matching left endpoints and right endpoints, respectively. Both relations can be implemented using only m as follows: $\{z\mathsf{m}x, z\mathsf{m}y\}$ implements $x \equiv^- y$ where z is a fresh variable; similarly, $\{x\mathsf{m}z, y\mathsf{m}z\}$ implements $x \equiv^+ y$. Thus, we may assume that the relations $\equiv^-$ and $\equiv^+$ are available whenever we have access to the m relation. We are now ready to present the reduction for MinCSP($\mathsf{m}, \equiv$), which will be from the Paired Cut problem that was shown to be W[1]-hard in Lemma 17.

▶ **Theorem 21.** *MinCSP*($\mathsf{m}, \equiv$) *is* W[1]*-hard.*

We continue by showing that MinCSP(m, s) is W[1]-hard. First note even though we no longer have access to $\equiv$, we can add the constraints $x \equiv^- y$ and $x \equiv^+ y$ which imply $x \equiv y$. As previously, the relations $\equiv^-$ and $\equiv^+$ can be implemented using only m. We remark that $\{x \equiv^- y, x \equiv^+ y\}$ is *not* an implementation of $\equiv$, so we can only use $\equiv$ in crisp constraints. Our reduction is based on the PCFAS problem, which was shown to be W[1]-hard in Lemma 17. While the reduction is quite similar to the reduction for MinCSP($\mathsf{m}, \equiv$), it is non-trivial to replace the role of $\equiv$ with s.

▶ **Theorem 22.** *MinCSP*(m, s) *is* W[1]*-hard.*

We finally show the W[1]-hardness of MinCSP(d, p), MinCSP(d, o), and MinCSP(p, o) via parameterized reductions from Sim-DFAS (which is a W[1]-hard problem by Theorem 20).

▶ **Theorem 23.** *MinCSP*(d, p)*, MinCSP*(d, o)*, and MinCSP*(p, o) *are* W[1]*-hard.*

7 Discussion

We have initiated a study of the parameterized complexity of MinCSP for Allen's interval algebra. We prove that MinCSP restricted to the relations in $\mathcal{A}$ exhibits a dichotomy: MinCSP(Γ) is either fixed-parameter tractable or W[1]-hard when $\Gamma \subseteq \mathcal{A}$. Even though the restriction to the relations in $\mathcal{A}$ may seem severe, one should keep in mind that a CSP instance over $\mathcal{A}$ is sufficient for representing *definite* information about the relative positions of intervals. In other words, such an instance can be viewed as a data set of interval information and the MinCSP problem can be viewed as a way of filtering out erroneous information (that may be the result of contradictory sources of information, noise in the measurements, human mistakes etc.) Various ways of "repairing" unsatisfiable data sets of qualitative information have been thoroughly discussed by many authors; see, for instance, [7, 18, 19] and the references therein.

Proving a full parameterized complexity classification for Allen's interval algebra is hindered by a barrier: such a classification would settle the parameterized complexity of Directed Symmetric Multicut, and this problem is considered to be one of the main open problems in the area of directed graph separation problems [27, 39]. This barrier comes into play even in very restricted cases: as an example, it is not difficult to see that MinCSP for the two Allen relations ($\mathsf{f} \cup \mathsf{fi}$) and ($\mathsf{f} \cup \equiv$) is equivalent to the MinCSP problem for the two PA relations $\neq$ and $\leq$ and thus equivalent to Directed Symmetric Multicut.

One way of continuing this work without necessarily settling the parameterized complexity of Directed Symmetric Multicut is to consider fpt approximability: it is known that Directed Symmetric Multicut is 2-approximable in fpt time [27]. Thus, a possible research direction is to analyse the fpt approximability for MinCSP(Γ) when Γ is a subset of $2^{\mathcal{A}}$ or, more ambitiously, when Γ is first-order definable in $\mathcal{A}$. A classification that separates the cases that are constant-factor fpt approximable from those that are not may very well be easier to obtain than mapping the FPT/W[1] borderline. There is at least one technical reason for optimism here, and we introduce some definitions to outline this idea. An n-ary relation R is said to have a *primitive positive definition* (pp-definition) in a structure Γ if it can be first-order defined by only using the relations in Γ together with the equality relation and the operators existential quantification and conjunction. If the equality relation is not needed, then we say that R has an *equality-free primitive positive definition* (efpp-definition) in Γ. Bonnet et al. [13, Lemma 10] have shown that constant-factor fpt approximability is preserved by efpp-definitions [13], i.e. if R is efpp-definable in Γ and MinCSP(Γ) is constant-factor fpt approximable, then MinCSP($\Gamma \cup \{R\}$) is also constant-factor fpt approximable. Bonnet et al. focus on Boolean domains, but it is clear that their Lemma 10 works for problems with arbitrarily large domains. Lagerkvist [43, Lemma 5] has shown that in most cases one can use pp-definitions instead of efpp-definitions. This implies that the standard algebraic approach via polymorphisms (that, for instance, underlies the full complexity classification of finite-domain CSPs [14, 54]) often becomes applicable when analysing constant-factor fpt approximability. One should note that, on the other hand, the exact complexity of MinCSP is only preserved by much more limited constructions such as *proportional implementations* (see Section 5.2. in [35]). We know from the literature that this may be an important difference: it took several years after Bonnet et al.'s classification of approximability before the full classification of exact parameterized complexity was obtained using a much more complex framework [37]. It is also worth noting that parameterized approximation results for MinCSP may have very interesting consequences, e.g. [13] resolved the parameterized complexity of Even Set, which was a long-standing open problem.

References

1. Akanksha Agrawal, Daniel Lokshtanov, Amer E. Mouawad, and Saket Saurabh. Simultaneous feedback vertex set: A parameterized perspective. *ACM Transactions on Computation Theory*, 10(4):1–25, 2018.
2. Akanksha Agrawal, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Simultaneous feedback edge set: a parameterized perspective. *Algorithmica*, 83(2):753–774, 2021.
3. James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
4. James F. Allen and Johannes A. G. M. Koomen. Planning using a temporal world model. In *Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI-1983)*, pages 741–747, 1983.

5 Kristóf Bérczi, Alexander Göke, Lydia Mirabel Mendoza Cadena, and Matthias Mnich. Resolving infeasibility of linear systems: A parameterized approach. *CoRR*, abs/2209.02017, 2022.

6 Benjamin Bergougnoux, Eduard Eiben, Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. Towards a polynomial kernel for directed feedback vertex set. *Algorithmica*, 83(5):1201–1221, 2021.

7 Leopoldo Bertossi and Jan Chomicki. Query answering in inconsistent databases. In *Logics for Emerging Applications of Databases*, pages 43–83. Springer, 2004.

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

9 Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, pages 184–196, 2008.

10 Manuel Bodirsky, Peter Jonsson, Barnaby Martin, Antoine Mottet, and Zaneta Semanisinová. Complexity classification transfer for CSPs via algebraic products. *CoRR*, abs/2211.03340, 2022.

11 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):9:1–9:41, 2010.

12 Marthe Bonamy, Łukasz Kowalik, Jesper Nederlof, Michał Pilipczuk, Arkadiusz Socała, and Marcin Wrochna. On directed feedback vertex set parameterized by treewidth. In *Proc. 44th International Workshop on Graph-Theoretic Concepts in Computer Science (WG-2018)*, volume 11159, pages 65–78, 2018.

13 Édouard Bonnet, László Egri, and Dániel Marx. Fixed-parameter approximability of Boolean MinCSPs. In *Proc. 24th Annual European Symposium on Algorithms (ESA-2016)*, pages 18:1–18:18, 2016.

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

15 Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

16 Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC-2008)*, pages 177–186, 2008.

17 Rajesh Chitnis, Marek Cygan, Mohammataghi Hajiaghayi, and Dániel Marx. Directed subset feedback vertex set is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(4):1–28, 2015.

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

19 Jean-François Condotta, Issam Nouaouri, and Michael Sioutis. 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)*, 2016.

20 Robert Crowston, Gregory Gutin, Mark Jones, and Anders Yeo. Parameterized complexity of satisfying almost all linear equations over $\mathbb{F}_2$. *Theory of Computing Systems*, 52(4):719–728, 2013.

21 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015.

22 Konrad K. Dabrowski, Peter Jonsson, Sebastian Ordyniak, and George Osipov. Resolving inconsistencies in simple temporal problems: A parameterized approach. In *Proc. 36th AAAI Conference on Artificial Intelligence, (AAAI-2022)*, pages 3724–3732, 2022.

23 Konrad K. Dabrowski, Peter Jonsson, Sebastian Ordyniak, George Osipov, Marcin Pilipczuk, and Roohani Sharma. Parameterized complexity classification for interval constraints. *CoRR*, abs/2305.13889, 2023.

24 Konrad K. Dabrowski, Peter Jonsson, Sebastian Ordyniak, George Osipov, and Magnus Wahlström. Almost consistent systems of linear equations. In *Proc. 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-2023)*, pages 3179–3217, 2023.

25 Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95, 1991.

26 Pascal Denis and Philippe Muller. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, pages 1788–1793, 2011.

27 Eduard Eiben, Clément Rambaud, and Magnus Wahlström. On the parameterized complexity of symmetric directed multicut. In *Proc. 17th International Symposium on Parameterized and Exact Computation (IPEC-2022)*, volume 249, pages 11:1–11:17, 2022.

28 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

29 Alexander Göke, Dániel Marx, and Matthias Mnich. Hitting long directed cycles is fixed-parameter tractable. In *Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP-2020)*, volume 168, pages 59:1–59:18, 2020.

30 Alexander Göke, Dániel Marx, and Matthias Mnich. Parameterized algorithms for generalizations of directed feedback vertex set. *Discrete Optimization*, 46:100740, 2022.

31 Martin Charles Golumbic and Ron Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *Journal of the ACM*, 40(5):1108–1133, 1993.

32 Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: inapproximability of maximum acyclic subgraph. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS-2008)*, pages 573–582, 2008.

33 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2000.

34 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proc. 24th Annual ACM Symposium on Theory of Computing (STOC-2002)*, pages 767–775, 2002.

35 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Solving hard cut problems via flow-augmentation. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA-2021)*, pages 149–168, 2021.

36 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Directed flow-augmentation. In *Proc. 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC-2022)*, pages 938–947, 2022.

37 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation III: complexity dichotomy for Boolean CSPs parameterized by the number of unsatisfied constraints. In *Proc. 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-2023)*, pages 3218–3228, 2023.

38 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation I: directed graphs. *CoRR*, abs/2111.03450, 2023.

39 Eun Jung Kim, Tomáš Masařík, Marcin Pilipczuk, Roohani Sharma, and Magnus Wahlström. On weighted graph separation problems and flow-augmentation. *CoRR*, abs/2208.14841, 2022.

40 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. *SIAM Journal on Computing*, 46(3):1087–1110, 2017.

41 Andrei A. Krokhin, Peter Jeavons, and Peter Jonsson. Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra. *Journal of the ACM*, 50(5):591–640, 2003.

42 Andrei A. Krokhin and Jakub Oprsal. An invitation to the promise constraint satisfaction problem. *ACM SIGLOG News*, 9(3):30–59, 2022.

43 Victor Lagerkvist. A new characterization of restriction-closed hyperclones. In *Proc. 50th IEEE International Symposium on Multiple-Valued Logic (ISMVL-2020)*, pages 303–308, 2020.

44 Daniel Lokshtanov, Pranabendu Misra, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Fpt-approximation for fpt problems. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA-2021)*, pages 199–218, 2021.

45 Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. Linear time parameterized algorithms for subset feedback vertex set. *ACM Transactions on Algorithms*, 14(1):7:1–7:37, 2018.

46 Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. When recursion is better than iteration: A linear-time algorithm for acyclicity with few error vertices. In *Proc. 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-2018)*, pages 1916–1933, 2018.

47 Dániel Marx and Igor Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. *Information Processing Letters*, 109(20):1161–1166, 2009.

48 Lenka Mudrová and Nick Hawes. Task scheduling for mobile robots using interval algebra. In *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA-2015)*, pages 383–388, 2015.

49 Richard N. Pelavin and James F. Allen. A model for concurrent actions having temporal extent. In *Proc. 6th National Conference on Artificial Intelligence (AAAI-1987)*, pages 246–250, 1987.

50 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC-2008)*, pages 245–254, 2008.

51 Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.

52 Fei Song and Robin Cohen. The interpretation of temporal relations in narrative. In *Proc. 7th National Conference on Artificial Intelligence (AAAI-1988)*, pages 745–750, 1988.

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

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