# Periodic Constraint Satisfaction Problems: Polynomial-Time Algorithms

Hubie Chen

Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
hubes@cs.cornell.edu

**Abstract.** We study a generalization of the constraint satisfaction problem (CSP), the *periodic constraint satisfaction problem*. An input instance of the periodic CSP is a finite set of "generating" constraints over a structured variable set that implicitly specifies a larger, possibly infinite set of constraints; the problem is to decide whether or not the larger set of constraints has a satisfying assignment. This model is natural for studying constraint networks consisting of constraints obeying a high degree of regularity or symmetry. Our main contribution is the identification of two broad polynomial-time tractable subclasses of the periodic CSP.

## 1 Introduction

The *constraint satisfaction problem* (CSP) is a general framework in which many combinatorial search problems can be expressed. An instance of a CSP consists of a network of constraints over a set of variables; the problem is to decide whether or not there is an assignment to the variable set satisfying all of the constraints.

*Periodic constraint satisfaction problems.* This paper investigates a strict generalization of the CSP, the *periodic constraint satisfaction problem*. In this generalization, the full constraint network – of which satisfiability is to be decided – is specified *implicitly*. An instance of the periodic CSP is a finite set of "generating" constraints over a large, structured variable set, for instance, the set of lattice points in $k$-dimensional space. A corresponding full set of constraints is obtained by repeating the generating set periodically, and the problem is to decide whether or not the full set of constraints has a satisfying assignment. This model is natural for studying large, possibly infinite constraint networks consisting of constraints obeying a high degree of regularity or symmetry. Such constraint networks arise naturally in many domains, such as scheduling, planning, and hardware design [20, 19].

So that our results may enjoy maximal applicability, we study a very general formulation of the periodic CSP where the underlying variable set of an instance is a group $G$ (that is, the algebraic structure), and the input constraint network is expanded into the full constraint network according to the action of a subgroup of $G$. This formulation is essentially that studied by Freedman in [12], and encompasses the model studied by Orlin in [21] and subsequently in [20, 19]. Our main contribution is the identification of two broad polynomial-time tractable subclasses of the periodic CSP. (In this context, "polynomial-time" is measured with respect to the input constraint network.) These tractability results encompass the periodic variants of 2-SAT and Horn SAT, and we believe the results to be quite remarkable in light of the fact that the periodic CSP is undecidable in general [12].

*Tractability.* The *identification of tractable subclasses* is a classic approach to coping with computationally hard problems. There are multiple ways to restrict the CSP in hopes of obtaining such tractable subclasses [22]. One way is to restrict the *constraint language*, or the types of constraints that are permitted in instances. This is the form of restriction which was studied by Schaefer in his now classic dichotomy theorem [23], and since then has been studied in several other contexts [7].

Schaefer's theorem showed that *every* constraint language over a two-element domain gives rise to a CSP subclass which is either in P or is NP-complete. This dichotomy theorem and the many others that have followed [7] are valuable because they identify *all* of the ways tractability can arise from restrictions on the constraint language. In addition, knowledge of intractable subclasses can be quite useful in developing further proofs of intractability. Consequently, much attention has been directed towards extending Schaefer's theorem to constraint languages over *any* finite domain. This has resulted in the identification of many new polynomial-time algorithms for subclasses of the CSP [18, 11, 15, 8, 1, 9, 5, 3, 10], some of which are quite sophisticated, as well as a dichotomy theorem on CSP complexity classifying all constraint languages over a *three*-element domain, due to Bulatov [2].

This paper studies the complexity of the periodic CSP from the standpoint of restricting the constraint language, and hence takes the first steps towards the long-term research goal of establishing a dichotomy theorem on the complexity of the periodic CSP over finite domains.

*The closure properties approach.* An algebraic approach to studying the relative expressiveness of constraint languages was introduced in [18] and further studied in [14–16, 4, 1, 5, 2, 3, 9]. In this approach, a dual perspective on constraint languages is given by studying the set of functions under which a constraint language is closed, or the *closure properties* of a constraint language. Many of the tractability results cited above demonstrate that closure under an operation of a particular type is sufficient for a constraint language to be tractable, in the context of the CSP. The tractability results for the periodic CSP given in this paper are also phrased in this form: we show that any constraint language closed under a *semilattice* operation or a *dual discriminator* operation gives rise to a tractable subclass of the periodic CSP.

*Related work.* Orlin [21] demonstrated the PSPACE-completeness of a number of periodic variants of NP-complete languages, including a problem which he called dynamic satisfiability; in our framework, dynamic satisfiability is obtained as a particular case by setting the group to be $\mathbb{Z}$ (the integers), that is, the lattice points in one-dimensional space.

Marathe, Hunt, and Stearns, in joint work with Radhakrishnan in [20] and Rosenkrantz in [19], studied problems arising from a variety of periodic specifications, focusing on approximation algorithms.

Freedman [12] showed that for group $\mathbb{Z}^2$ and $H$ a subgroup of finite index, the periodic version of 3-SAT is undecidable, while the periodic version of 2-SAT is decidable; however, he gave no running time analysis of his procedure for periodic 2-SAT. He also demonstrated that the periodic version of 3-colorability is undecidable.

## 2 Preliminaries

This section presents the definitions and notation used throughout the paper. We will assume that the reader has familiarity with basic notions of group theory.

### 2.1 Constraint Satisfaction Problems

We first introduce the basic terminology of constraint satisfaction problems.

**Definition 1.** *A* relation *over a set $D$ is a subset of $D^k$ (with $k \geq 1$), and is said to have arity $k$. For any tuple $t \in D^k$, we denote the $i$th coordinate of $t$ by $t[i]$. When $R$ is a relation of arity $k$ and $i \in \{1, \ldots, k\}$, let $\pi_i(R)$ denote the relation $\{t[i] : t \in R\}$. A* constraint language *is a finite set of relations which are all over the same set (and may be of different arities).*

**Definition 2.** *A constraint over variable set $V$ and domain $D$ is an expression of the form $R(x_1, \ldots, x_k)$, where $R$ is a relation over $D$ with arity $k$ and $x_1, \ldots, x_k$ are variables in $V$.*

A collection of constraints yields a *constraint satisfaction problem (CSP).* Note that we consider only CSPs where all variables have the same domain, although our results are easily generalized to the more general formulation where different variables may have different domains.

**Definition 3.** *A* constraint satisfaction problem (CSP) *is a set $\mathcal{C}$ of constraints, each of which is over the same variable set $V$ and domain $D$; an assignment to $\mathcal{C}$ is a function $f : V \to D$. The assignment $f$ satisfies a constraint $R(x_1, \ldots, x_k) \in \mathcal{C}$ if $(f(x_1), \ldots, f(x_k)) \in R$, and satisfies $\mathcal{C}$ if it satisfies every constraint $C \in \mathcal{C}$.*

Having defined a CSP and an associated notion of satisfiability, we can now formally define the decision problem for CSPs over a particular constraint language.

**Definition 4.** *Let $\Gamma$ be a constraint language over a finite domain $D$. The $\mathsf{CSP}(\Gamma)$ decision problem is to decide, given a CSP $\mathcal{C}$ (of finite size) over domain $D$, where each constraint in $\mathcal{C}$ has relation from $\Gamma$, whether or not $\mathcal{C}$ is satisfiable.*

*Example 1.* Let $D = \{T, F\}$, $V = \{-1, 2, 4, 5\}$, and $R_0 = D^2 \setminus \{(F, F)\}$, $R_1 = D^2 \setminus \{(F, T)\}$, $R_2 = D^2 \setminus \{(T, T)\}$. The CSP $\mathcal{C} = \{R_0(2, 5), R_2(2, 5), R_1(-1, 4)\}$ over $V$ and $D$ is satisfiable, for example, by the assignment $f : V \to D$ such that $f(2) = F$ and $f(x) = T$ for all $x \in V \setminus \{2\}$.

Every instance of the $\mathsf{CSP}(\{R_0, R_1, R_2\})$ problem can be reduced in a simple and assignment-preserving manner to the standard boolean 2-satisfiability problem, and vice-versa. For example, the CSP $\mathcal{C}$ is equivalent to the 2-satisfiability formula $(2 \vee 5) \wedge (\neg 2 \vee \neg 5) \wedge (-1 \vee \neg 4)$. □

We now define formally the periodic constraint satisfaction problem.

**Definition 5.** *Let $\Gamma$ be a constraint language over a finite domain $D$, $G$ be a (possibly infinite) group, and $H$ be a subgroup of $G$. The $\mathsf{PeriodicCSP}(\Gamma; G, H)$ decision problem is to decide, given a CSP $\mathcal{C}$ (of finite size) over variable set $G$ and domain $D$, where each constraint in $\mathcal{C}$ has relation from $\Gamma$, whether or not the (possibly infinite) CSP*

$$H\mathcal{C} \overset{\text{def}}{=} \{R(hg_1, \ldots, hg_k) : h \in H, R(g_1, \ldots, g_k) \in \mathcal{C}\}$$

*is satisfiable.*

The following are examples of periodic CSPs.

*Example 2.* Let $D = \{T, F\}$, let $G$ be $(\mathbb{Z}, +)$, let $H$ be $(3\mathbb{Z}, +)$, and let $\mathcal{C}$ be the formula

$$\mathcal{C} = \{R_2(2, 5), R_0(2, 5), R'(-1, 4, 3)\}$$

where $R' = D^3 \setminus \{(F, F, F)\}$ and $R_0, R_2$ are defined as in Example 1.

We have

$$H\mathcal{C} = \ldots \cup \{R_2(-1, 2), R_0(-1, 2), R'(-4, 1, 0)\} \cup$$

$$\{R_2(2, 5), R_0(2, 5), R'(-1, 4, 3)\} \cup \{R_2(5, 8), R_0(5, 8), R'(2, 7, 6)\} \cup \ldots .$$

The infinite set of constraints $H\mathcal{C}$ is satisfied by the assignment $f : \mathbb{Z} \to \{T, F\}$ where $f(x)$ is defined to be $F$ if $x$ is congruent to 2 modulo 6, and $T$ otherwise. □

*Example 3.* Let $D = \{T, F\}$, let $G$ be $(\mathbb{Z}, +)$, let $H$ be $(5\mathbb{Z}, +)$, and let $\mathcal{C}$ be the formula

$$\{R_1(1, 5), N(-4), Y(10), R_2(2, 6)\}$$

where $Y = \{(T)\}, N = \{(F)\}$ and $R_1, R_2$ are defined as in Example 1.

We have

$$H\mathcal{C} = \ldots \cup \{R_1(-4, 0), N(-9), Y(5), R_2(-3, 1)\} \cup$$

$$\{R_1(1, 5), N(-4), Y(10), R_2(2, 6)\} \cup \{R_1(6, 10), N(1), Y(15), R_2(7, 11)\} \cup \ldots .$$

It can be seen that the infinite set of constraints $H\mathcal{C}$ contains the constraints $Y(5)$, $R_1(1, 5)$, and $N(1)$. Since no assignment can satisfy these three constraints, we conclude that $H\mathcal{C}$ is unsatisfiable. □

Observe that when $\mathcal{C}$ is an instance of the $\mathsf{PeriodicCSP}(\Gamma; G, H)$ problem, the containment $\mathcal{C} \subseteq H\mathcal{C}$ holds, so any satisfying assignment for $H\mathcal{C}$ is also a satisfying assignment for $\mathcal{C}$. Thus, if $H\mathcal{C}$ is satisfiable, then $\mathcal{C}$ is satisfiable. Example 3 demonstrates that the converse does not hold: the example CSP $\mathcal{C}$ is easily seen to be satisfiable – no variable appears in more than one constraint – but $H\mathcal{C}$ is not satisfiable.

### 2.2 Closure properties

We now define the notion of the closure properties of a constraint language, which gives a dual perspective (on constraint languages) relevant to considerations of complexity. A number of papers have studied this dual viewpoint [18, 16, 14, 4, 5], and we refer the reader to these papers for more information.

**Definition 6.** *Let $\otimes : D^m \to D$ be a $m$-ary operation on $D$ and let $R$ be a $k$-ary relation over $D$. Suppose $t_1, \ldots, t_m \in R$. The expression $\otimes(t_1, \ldots, t_m)$ is defined to be a $k$-tuple via a coordinate-wise action:*

$$\otimes(t_1, \ldots, t_m) \stackrel{\text{def}}{=} (\otimes(t_1[1], \ldots, t_m[1]), \otimes(t_1[2], \ldots, t_m[2]), \ldots, \otimes(t_1[k], \ldots, t_m[k])).$$

*The relation $\otimes(R)$ is defined to be the set $\{\otimes(t_1, \ldots, t_m) : t_1, \ldots, t_m \in R\}$.*

**Definition 7.** *A relation $R$ is closed under $\otimes$ if $\otimes(R) \subseteq R$. A constraint language $\Gamma$ is closed under $\otimes$ if all relations in $\Gamma$ are closed under $\otimes$. A constraint $R(x_1, \ldots, x_k)$ is said to be closed under $\otimes$ if its relation $R$ is closed under $\otimes$. A set of constraints $\mathcal{C}$ is said to be closed under $\otimes$ if all constraints in $\mathcal{C}$ are closed under $\otimes$.*

## 3 *H*-Arc Consistency

In this section, we introduce a strengthened version of arc consistency for periodic CSPs, called *$H$-arc consistency*. Under a mild assumption, an instance of the $\mathsf{PeriodicCSP}(\Gamma; G, H)$ problem can be transformed into one that is $H$-arc consistent in polynomial time; the procedure for doing this will be used as a subroutine in both of the algorithms we give for $\mathsf{PeriodicCSP}(\Gamma; G, H)$, in the next two sections.

**Definition 8.** *Let $D$ be a finite set. A CSP $\mathcal{C}$ over variable set $V$ and domain $D$ is* arc consistent *if for all pairs of constraints $C_1(v_1, \ldots, v_k)$, $C_2(v'_1, \ldots, v'_l) \in \mathcal{C}$, $v_i = v'_j$ (with $1 \le i \le k, 1 \le j \le l$) implies that $\pi_i(C_1) = \pi_j(C_2)$.*

**Definition 9.** *Let $D$ be a finite set, $G$ be a group, and $H$ be a subgroup of $G$. A CSP $\mathcal{C}$ over variable set $G$ and domain $D$ is* $H$-arc consistent *if for all pairs of constraints $C_1(g_1, \ldots, g_k)$, $C_2(g'_1, \ldots, g'_l) \in \mathcal{C}$, $Hg_i = Hg'_j$ (with $1 \le i \le k, 1 \le j \le l$) implies that $\pi_i(C_1) = \pi_j(C_2)$.*

It is easy to see that $H$-arc consistency implies arc consistency.

We now identify a computational problem which, when solvable, makes establishing $H$-arc consistency possible.

**Definition 10.** *Let $G$ be a group and $H$ be a subgroup of $G$. The* same coset problem *for $(G, H)$ is to decide, given two group elements $g_1, g_2 \in G$, whether or not $Hg_1 = Hg_2$ (that is, whether $g_1$ and $g_2$ are in the same right coset).*

The following is the main theorem of this section, which identifies some key properties of $H$-arc consistency.

**Theorem 1.** *Let $\Gamma$ be a constraint language over domain $D$, $G$ be a group, and $H$ be a subgroup of $G$. Suppose $\mathcal{C}$ is an instance of the problem $\mathsf{PeriodicCSP}(\Gamma; G, H)$. Then there exists a CSP $\mathcal{C}'$ (over variable set $G$ and domain $D$) with the following four properties:*

1. *$\mathcal{C}'$ is $H$-arc consistent.*
2. *$H\mathcal{C}'$ has the same satisfying assignments as $H\mathcal{C}$.*
3. *If the same coset problem for $(G, H)$ is decidable in polynomial time, then $\mathcal{C}'$ can be computed in polynomial time from $\mathcal{C}$.*
4. *For any operation $\otimes$ under which $\Gamma$ is closed, every relation of $\mathcal{C}'$ is closed under $\otimes$.*

**Proof**. The following algorithm computes $\mathcal{C}'$ from $\mathcal{C}$:

While there are two (not necessarily distinct) constraints

$$C_1(g_1, \ldots, g_k), C_2(g'_1, \ldots, g'_l) \in \mathcal{C}$$

where $g_i$ and $g'_j$ (for some $1 \le i \le k, 1 \le j \le l$) are in the same right coset and $\pi_i(C_1) \ne \pi_j(C_2)$, replace the constraints $C_1(g_1, \ldots, g_k), C_2(g'_1, \ldots, g'_l)$ by $D_1(g_1, \ldots, g_k), D_2(g'_1, \ldots, g'_l)$, where

$$D_1 = \{t \in C_1 : t[i] \in \pi_j(C_2)\},$$
$$D_2 = \{t \in C_2 : t[j] \in \pi_i(C_1)\}.$$

Let $\mathcal{C}'$ denote the set of constraints after the algorithm terminates.

The loop iterates at most polynomially many times, since there are at most polynomially many pairs $g_i$ and $g'_j$ in the same coset, and a constant number of possibilities for the sets $\pi_i(C_1)$, $\pi_j(C_2)$. The loop condition takes polynomial time to check, assuming that the same coset problem for $(G, H)$ can be solved in polynomial time, so property (3) is satisfied.

By the loop condition, property (1) will be satisfied by the set of constraints which results when the algorithm terminates.

It is straightforward to verify that if $C_1$ and $C_2$ are closed under $\otimes$, then $D_1$ and $D_2$ are closed under $\otimes$. From this, property (4) follows by induction.

It remains to verify property (2). Suppose the loop terminates after $s$ iterations. Let $\mathcal{C} = \mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_s = \mathcal{C}'$ denote the different sets of constraints encountered during the execution of the algorithm.

Any solution to $\mathcal{C}'$ is a solution to $\mathcal{C}$, since when a replacement step is performed, $D_1 \subseteq C_1$ and $D_2 \subseteq C_2$; thus, any solution to $H\mathcal{C}'$ is a solution to $H\mathcal{C}$.

Now let $f$ be a satisfying assignment for $H\mathcal{C}$. We show that if $f$ satisfies $H\mathcal{C}_i$, then $f$ satisfies $H\mathcal{C}_{i+1}$. (By induction, it will follow that $f$ is a satisfying assignment for $H\mathcal{C}'$.) Let $C_1(g_1,\ldots,g_k), C_2(g'_1,\ldots,g'_l)$ denote the constraints in $\mathcal{C}_i$ that were replaced; $g_i$ and $g'_j$ denote the variables that were identified to be in the same coset; and $D_1(g_1,\ldots,g_k), D_2(g'_1,\ldots,g'_l)$ denote the new constraints in $\mathcal{C}_{i+1}$. It suffices to show that if $f$ satisfies $C_1(hg_1,\ldots,hg_k)$ and $C_2(hg'_1,\ldots,hg'_l)$ for all $h \in H$, then $f$ satisfies $D_1(hg_1,\ldots,hg_k)$ and $D_2(hg'_1,\ldots,hg'_l)$ for all $h \in H$.

Suppose $h \in H$. Since $g_i$ and $g'_j$ are in the same coset, there exists $h'$ such that $g_i = h'g'_j$. Since $C_2(hh'g'_1,\ldots,hh'g'_l)$ is satisfied by assumption, $f(hg_i) = f(hh'g'_j) \in \pi_j(C_2)$. By assumption, $C_1(hg_1,\ldots,hg_k)$ is satisfied by $f$; since $f(hg_i) \in \pi_j(C_2)$, $D_1(hg_1,\ldots,hg_k)$ is also satisfied by $f$. (The proof that all constraints $D_2(hg'_1,\ldots,hg'_l)$ are satisfied is symmetric.) $\square$

## 4 Semilattice operations

In this section, we show that instances of the $\mathsf{PeriodicCSP}(\Gamma; G, H)$ problem closed under a semilattice operation are tractable in polynomial time (under a mild assumption on the "computability" of $G$ and $H$).

**Definition 11.** *A* semilattice operation *is a binary operation $\otimes : D^2 \to D$ that is associative, commutative and idempotent. Note that when $\otimes$ is an semilattice operation, we can unambiguously define $\otimes$ on a finite subset of $D$ as follows: for $S = \{s_1,\ldots,s_k\}$, $\otimes(S) = \otimes(s_1, \otimes(s_2, \ldots \otimes(s_{k-1}, s_k)))$.*

In previous work, it was shown that instances of the $\mathsf{CSP}(\Gamma)$ problem closed under a semilattice operation are tractable. Note that Horn clauses are examples of constraints closed under a semilattice operation [18], so this tractability result implies the tractability of Horn SAT.

**Theorem 2.** *[18] Let $\Gamma$ be a constraint language. If $\Gamma$ is closed under a semilattice operation, then $\mathsf{CSP}(\Gamma)$ is decidable in polynomial time.*

We have the following tractability result for the periodic constraint satisfaction problem. Note that this result strictly generalizes Theorem 2; this is seen by setting $H$ to be the subgroup containing only the identity element and $G$ to be any sufficiently large finite group.

**Theorem 3.** *Let $\Gamma$ be a constraint language, $G$ be a group, and $H$ be a subgroup of $G$. If $\Gamma$ is closed under an semilattice operation and the same coset problem for $(G, H)$ is decidable in polynomial time, then $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is decidable in polynomial time. Moreover, if an instance of $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is satisfiable, it has an $H$-invariant satisfying assignment.*

By an $H$*-invariant satisfying assignment*, we mean a satisfying assignment $f : G \to D$ such that for all $g \in G$ and $h \in H$, the equality $f(g) = f(hg)$ holds.
**Proof**. Suppose that $\otimes$ is an semilattice operation under which $\Gamma$ is closed. The following is a decision procedure for $\mathsf{PeriodicCSP}(\Gamma; G, H)$: given a problem instance $\mathcal{C}$, apply the $H$-arc consistency algorithm of Theorem 1 to $\mathcal{C}$, and let $\mathcal{C}'$ denote the resulting CSP. If there is a constraint in $\mathcal{C}'$ such that its associated relation is the empty set, *reject*; otherwise, *accept*.

If the algorithm rejects, then $H\mathcal{C}'$ has no satisfying assignment, since no assignment can satisfy the constraint $C \in \mathcal{C}' \subseteq H\mathcal{C}'$ with the empty set as associated relation. Since $H\mathcal{C}$ and $H\mathcal{C}'$ have the same set of satisfying assignments, $H\mathcal{C}$ is not satisfiable.

If the algorithm accepts, then let $f : G \to D$ be such that for all constraints $C(g_1, \ldots, g_k) \in \mathcal{C}'$, $f$ maps all elements of $Hg_i$ to $\otimes(\pi_i(C))$. This definition is unambiguous because $\mathcal{C}'$ is $H$-arc consistent. We claim that $f$ is a satisfying assignment for $H\mathcal{C}$; it suffices to show that $f$ satisfies every constraint in $H\mathcal{C}'$.

Let $C(g_1, \ldots, g_k)$ be a constraint in $\mathcal{C}'$, let $t_1, \ldots, t_m$ be the tuples in $C$ and suppose $h \in H$. Observe that $(f(hg_1), \ldots, f(hg_k)) = (f(g_1), \ldots, f(g_k)) = (\otimes(\pi_1(C)), \ldots, \otimes(\pi_k(C))) = \otimes(t_1, \ldots, t_m)$. The tuple $\otimes(t_1, \ldots, t_m)$ is in $C$ since $\mathcal{C}'$ is closed under $\otimes$. Thus $(f(hg_1), \ldots, f(hg_k)) \in C$, and we conclude that $f$ satisfies $C(hg_1, \ldots, hg_k)$. $\Box$

This theorem can be readily generalized to constraint languages closed under a *set function*; for the definition of this notion and more information, we refer the reader to [8].

**Theorem 4.** *Let $\Gamma$ be a constraint language, $G$ be a group, and $H$ be a subgroup of $G$. If $\Gamma$ is closed under a set function and the same coset problem for $(G, H)$ is decidable in polynomial time, then $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is decidable in polynomial time. Moreover, if an instance of $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is satisfiable, it has an $H$-invariant satisfying assignment.*

It is easy to see that the same coset problem for group $G = (\mathbb{Z}, +)^k$ and subgroup $H = (d_1\mathbb{Z}, +) \times \cdots \times (d_k\mathbb{Z}, +)$ for any $k \geq 1$ and $d_1, \ldots, d_k \geq 0$ is solvable in polynomial time, yielding the following corollary.

**Corollary 1.** *Suppose that $\Gamma$ is a constraint language closed under a set function, $G$ is the group $(\mathbb{Z}, +)^k$ and $H$ is the $G$-subgroup $(d_1\mathbb{Z}, +) \times \cdots \times (d_k\mathbb{Z}, +)$ for some $k \geq 1$ and $d_1, \ldots, d_k \geq 0$. Then, $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is decidable in polynomial time.*

In fact, using an algorithm of [13] it can be shown that the same coset problem is solvable for $(\mathbb{Z}, +)^k$ paired with any subgroup definable by a finite set of equations of the form $w = w'$, where $w, w'$ are elements of $\mathbb{Z}^k$.

## 5 Dual discriminator operations

In this section, we show that instances of the $\mathsf{PeriodicCSP}(\Gamma; G, H)$ problem closed under a dual discriminator operation are tractable in polynomial time.

The tractability result of the previous section required an assumption on the "computability" of $G$ and $H$; we will need such assumptions for this section's result, as well.

**Definition 12.** *For any set $D$, let $\Delta : D^3 \to D$ be the ternary operation defined as $\Delta(x, y, z) = y$ if $y = z$; $x$ otherwise. We call $\Delta$ the* dual discriminator *on $D$.*

As with semilattice operations, closure under dual discriminators was previously shown to ensure tractability in the context of $\mathsf{CSP}(\Gamma)$. This tractability result strictly generalizes the tractability of 2-SAT – it is easily verified that the relations $R_0$, $R_1$, and $R_2$ given by Example 1 are closed under $\Delta$.

**Theorem 5.** *[17, Theorem 13] Let $\Gamma$ be a constraint language. If $\Gamma$ is closed under a dual discriminator operation, then $\mathsf{CSP}(\Gamma)$ is decidable in polynomial time.*

In previous work, a structure theorem was proved on constraints closed under a dual discriminator operation – namely, that any such constraint can be decomposed into binary constraints of a few particular types.

**Definition 13.** *A* binary constraint *is a constraint which has a relation of arity two. A* complete constraint *is a binary constraint $C$ where the relation of $C$ is equal to $\pi_1(C) \times \pi_2(C)$. A* permutation constraint *is a binary constraint $C$ where the relation of $C$ is equal to $\{(d, \sigma(d)) : d \in \pi_1(C)\}$ for some bijection $\sigma : \pi_1(C) \to \pi_2(C)$. A* two-fan constraint *is a binary constraint $C = R(v, w)$ where the relation $R$ is equal to $(\{x\} \times \pi_2(C)) \cup (\pi_1(C) \times \{y\})$ for some $x, y \in D$. The element $x$ is called the* $v$-fanout*, and the element $y$ is called the* $w$-fanout.

**Lemma 1.** *[17, Propositions 10 and 12], [6, Lemma 3.4] If $C$ is a constraint over domain $D$ closed under the dual discriminator on $D$, then $C$ is logically equivalent to a conjunction of complete constraints, permutation constraints, and two-fan constraints.*

In light of Lemma 1, when studying constraints closed under a dual discriminator, we are justified in focusing on the three types of binary constraints given by Definition 13. This observation motivates the following definitions.

**Definition 14.** *Say that a CSP $\mathcal{C}$ is* implicative *if it is finite, arc consistent, contains only permutation constraints and two-fan constraints, and contains no constraint with empty set as the associated relation.*

*Let $G$ be a group and $H$ be a subgroup of $G$. Say that a CSP $\mathcal{C}$ over variable set $G$ is* $H$-implicative *if it is implicative and $H$-arc consistent.*

We give a graph-theoretic characterization of implicative CSPs in the next definition and subsequent lemmas.

**Definition 15.** *Suppose that the CSP $\mathcal{C}$ is implicative.*

*Define the* implication graph of $\mathcal{C}$ *to be the directed graph $\mathcal{G}$ where:*

- *The vertex set of $\mathcal{G}$ consists of all expressions of the form $(v = d)$ and $(v \neq d)$, with $v \in V$ and $d \in D$.*
- *The edge set of $\mathcal{G}$ is the union of the set $\{(v = d, v \neq d') : d, d' \in D, v \in V, d \neq d'\}$ and the edge sets of the constraints in $\mathcal{C}$, where we define the edge set of a constraint as follows.*
  - *If $R(v_1, v_2)$ is a permutation constraint with relation $R = \{(d, \sigma(d)) : d \in \pi_1(C)\}$, define the edge set of $C$ to be the union of the sets*

$$\{(v_1 = d, v_2 = \sigma(d)) : d \in \pi_1(C)\} \qquad \{(v_2 \neq \sigma(d), v_1 \neq d) : d \in \pi_1(C)\}$$

$$\{(v_2 = \sigma(d), v_1 = d) : d \in \pi_1(C)\} \qquad \{(v_1 \neq d, v_2 \neq \sigma(d)) : d \in \pi_1(C)\}$$

  - *If $R(v_1, v_2)$ is a two-fan constraint with relation $R = (\{x\} \times \pi_2(C)) \cup (\pi_1(C) \times \{y\})$, define the edge set of $C$ to be the union of the sets*

$$\{(v_1 = d, v_2 = y) : d \neq x, d \in \pi_1(C)\}$$

$$\{(v_2 \neq y, v_1 \neq d) : d \neq x, d \in \pi_1(C)\}$$

$$\{(v_2 = d, v_1 = x) : d \neq y, d \in \pi_2(C)\}$$

$$\{(v_1 \neq x, v_2 \neq d) : d \neq y, d \in \pi_2(C)\}$$

A feature of the implication graph of a CSP is the following lemma, whose proof is straightforward given the above definition, and omitted.

**Lemma 2.** *Suppose that $\mathcal{C}$ is an implicative CSP (over variable set $V$ and domain $D$), and that $f : V \to D$ is a satisfying assignment. For every pair of vertices $c, c'$ in the implication graph of $\mathcal{C}$ such that there is a path from $c$ to $c'$, if $f$ satisfies $c$, then $f$ satisfies $c'$.*

We are now able to establish a *sufficient condition* for such an implicative CSP to be satisfiable.

**Lemma 3.** *Suppose that $\mathcal{C}$ is an implicative CSP. If there is no path in the implication graph of $\mathcal{C}$ from a vertex $v = d$ to the vertex of its "negation" $v \neq d$, then the CSP $\mathcal{C}$ is satisfiable.*

**Proof**. We prove this by induction on the number of variables of $\mathcal{C}$. We define a partial function $f$ as follows. Pick a variable $v$ and and value $d \in D$, and set $f(v) = d$. For all vertices of the form $v' = d'$ such that there exists a path in the implication graph of $\mathcal{C}$ from $v = d$ to $v' \neq d'$, set $f(v') = d'$. Note that this is well-defined, because if there were paths from $v = d$ to both $v' = d_1$ and $v' = d_2$ (for distinct $d_1, d_2$), then there would be a path from $v = d$ to $v' \neq d_1$ (as there is an edge from $v' = d_2$ to $v' \neq d_1$). By the symmetry of the graph, there would be a path from $v' = d_1$ to $v \neq d$, implying that there is a path from $v = d$ to $v \neq d$, a contradiction to Lemma 2.

Let $\mathcal{C}'$ be the subset of $\mathcal{C}$ containing those constraints where neither variable has been defined by $f$.

We claim that any extension $f'$ of $f$ satisfying $\mathcal{C}'$ will satisfy $\mathcal{C}$. It suffices to show that any such extension $f'$ will satisfy $\mathcal{C} \setminus \mathcal{C}'$. For constraints in $\mathcal{C} \setminus \mathcal{C}'$ containing two variables on which $f$ is defined, this is clear from the definition of the implication graph. Any constraint $C$ in $\mathcal{C} \setminus \mathcal{C}'$ containing a variable $v$ on which $f$ is defined and a variable $w$ on which $f$ is not defined must be a two-fan constraint, where $f(v)$ is the $v$-fanout; consequently, any value of $f(w)$ for which $f$ satisfying $\mathcal{C}'$ will result in $f$ satisfying $C$, as $\mathcal{C}$ is arc consistent.

The implication graph of $\mathcal{C}'$ is a subgraph of the implication graph of $\mathcal{C}$, so $\mathcal{C}'$ is satisfiable by induction. □

We now give an analog of implication graph (Definition 15) for periodic CSPs. Because one can naively extend Definition 15 to infinite constraint sets by allowing the implication graph to be infinite, it is worth noting that our second notion of implication graph is a *finite* graph, albeit one with edge weights.

**Definition 16.** (Implication graph of $H\mathcal{C}$) *Let $G$ be a group and $H$ a subgroup of $G$ with finite index. Let $S$ be a subset of $G$ containing exactly one element of $G$ from each right $H$-coset. Let $r : G \to S$ map a group element $g$ to the element of $S$ which is in the same right coset as $g$.*

*Suppose that $\mathcal{C}$ is an $H$-implicative CSP over variable set $G$.*

*Define the* implication graph *of $H\mathcal{C}$ relative to $S$ to be the directed graph $\mathcal{G}$ with edge labels where:*

- *The vertex set of $\mathcal{G}$ consists of all expressions of the form $(g = d)$ and $(g \neq d)$, with $g \in S$ and $d \in D$.*
- *The edge set of $\mathcal{G}$ consists of all edges of the form $r'(e)$, where $e$ is an edge from the implication graph of $\mathcal{C}$ and $r'$ acts on an edge $e$ by mapping the two variables of $e$ under the map $r$. (For example, $r'(g_1 = d_2, g_2 \neq d_2) \stackrel{\text{def}}{=} (r(g_1) = d_2, r(g_2) \neq d_2)$.)*
  *The label of the edge $r'(e)$ is equal to the group element $r(g_1)g_1^{-1}g_2r(g_2)^{-1}$, where $g_1$ and $g_2$ are the two variables of $e$ and the edge is directed from the vertex with variable $g_1$ to the vertex with variable $g_2$.*

*Define the* weight *of a path in $\mathcal{G}$ with edges $e_1, \ldots, e_s$ to be the group product $l_1 \cdots l_s$, where $l_i$ is the label of $e_i$ (for all $1 \leq i \leq s$).*

The following lemma is analogous to Lemma 2, and describes how to derive implications concerning satisfying assignments from the implication graph.

**Lemma 4.** *Suppose that $\mathcal{C}$ is an $H$-implicative CSP (over variable set $G$ and domain $D$), and that $f : G \to D$ is a satisfying assignment. For every pair of vertices $g \sim d, g' \sim' d'$ in the implication graph of $H\mathcal{C}$ (with $\sim, \sim' \in \{=, \neq\}$) such that there is a path of weight $w$ from $g \sim d$ to $g' \sim' d'$, if $f(g) \sim d$, then $f(wg') \sim d'$.*

Next, we present a parallel of Lemma 3 – a sufficient condition for an $H$-implicative periodic CSP to be satisfiable, based on properties of the implication graph.

**Lemma 5.** *(Sufficient condition for satisfiability of $H\mathcal{C}$) Let $G$ be a group and $H$ a subgroup of $G$ with finite index. Let $S$ be any subset of $G$ containing exactly one element of $G$ from each right $H$-coset. Suppose that $\mathcal{C}$ is an $H$-implicative CSP over variable set $G$.*

*If there is no path with identity weight in the implication graph of $H\mathcal{C}$ (relative to $S$) from a vertex $v = d$ to the vertex of its "negation" $v \neq d$, then the CSP $H\mathcal{C}$ is satisfiable.*

**Proof**. We show that for any finite subset $\mathcal{C}'$ of $H\mathcal{C}$, there is no path from a vertex $v = d$ to its negation $v \neq d$ in the implication graph of $\mathcal{C}'$. Every such finite subset $\mathcal{C}'$ is then satisfiable by Lemma 3, and that $H\mathcal{C}$ is satisfiable follows by a standard compactness argument.

Suppose there is a path from in the implication graph of a finite subset $\mathcal{C}'$ of $H\mathcal{C}$. Let $e_1, \dots, e_s$ be the edges of the path, and let $g_1, \dots, g_{s+1}$ be the elements of $G$ from the vertices on the path. Letting $r$ and $r'$ be defined as in Definition 16, $r'(e_1), \dots, r'(e_s)$ is a path in the implication graph of $H\mathcal{C}$ with path weight equal to

$$(r(g_1)g_1^{-1}g_2r(g_2)^{-1})(r(g_2)g_2^{-1}g_3r(g_3)^{-1}) \cdots (r(g_s)g_s^{-1}g_{s+1}r(g_{s+1})^{-1}).$$

After cancellations, this is equal to $r(g_1)g_1^{-1}g_{s+1}r(g_{s+1})^{-1}$ which is the identity element, because $g_1 = g_{s+1}$. $\square$

We are almost in position to state and prove the tractability result of this section. Before doing so, however, we identify the computability assumptions on a group-subgroup pair $(G, H)$ needed for $\mathsf{PeriodicCSP}(\Gamma; G, H)$ to be tractable.

**Definition 17.** *Let $G$ be a group and $H$ be a subgroup of $G$. Say that $(G, H)$ is $\Delta$-helpful if*

- *$H$ is of finite index,*
- *the same coset problem for $(G, H)$ is decidable in polynomial time,*
- *products and inverses can be computed in polynomial time, and*
- *given a directed graph with edge weights from $G$ and two specified vertices $w_1, w_2$, it can be checked whether or not there is a path from $w_1$ to $w_2$ with edge weight equal to the identity of $G$ in polynomial time. (The weight of a path is defined as in Definition 16.)*

The following is our second tractability result – namely, that closure under the dual discriminator implies efficient decidability.

**Theorem 6.** *Let $\Gamma$ be a constraint language, $G$ be a group, and $H$ be a subgroup of $G$. If $\Gamma$ is closed under the dual discriminator operation $\Delta$ and $(G, H)$ is $\Delta$-helpful, then $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is decidable in polynomial time.*

**Proof**. Fix $S$ to be a subset of $G$ containing exactly one element of $G$ from each right $H$-coset. Let $r : G \to S$ map a group element $g$ to the element of $S$ which is in the same right coset as $g$.

Let $\mathcal{C}$ be a $\mathsf{PeriodicCSP}(\Gamma; G, H)$ problem instance. By Lemma 1, any constraint from $\mathcal{C}$ is equivalent to a set of binary constraints which can be computed from $\mathcal{C}$ in polynomial time. Thus, we can assume without loss of generality that every relation in $\Gamma$ is a binary relation.

The following is an algorithm to decide satisfiability of $H\mathcal{C}$.

1. Perform steps (b), (c), and (d) below.
2. While there is a path in the implication graph of $H\mathcal{C}$ from $g = d$ to $g \neq d$ of identity weight (for some $g \in S$ and $d \in D$), do the following:
   (a) For every constraint $R(g_1, g_2)$ where $g_i$ (for either $i = 1$ or 2) is in the same coset as $g$, replace $R$ with $R' = \{t \in R : t[i] \neq d\}$
   (b) Apply the algorithm of Theorem 1 to $\mathcal{C}$ (replacing $\mathcal{C}$ with the new set of constraints $\mathcal{C}'$)
   (c) If $\mathcal{C}$ contains a constraint with empty relation, *reject*.
   (d) Remove all complete constraints from $\mathcal{C}$.
3. If the loop terminates without rejecting, then *accept*.

*Proof of Correctness.* When the algorithm of Theorem 1 is applied to $\mathcal{C}$, $\mathcal{C}$ remains $\Delta$-closed. It follows that, after step (b), by Lemma 1, every constraint in $\mathcal{C}$ is either a complete constraint, a permutation constraint, or a two-fan constraint. We will argue that every step of the algorithm preserves satisfiability of $H\mathcal{C}$, implying that if the algorithm rejects in step (c), it does so correctly. Removing complete constraints as in (d) does not affect whether or not $H\mathcal{C}$ is satisfiable, because of the $H$-consistency of $\mathcal{C}$. Suppose that step (a) of the algorithm is executed; this means that a path of identity weight was found from $g = d$ to $g \neq d$. By Lemma 4, no satisfying assignment of $H\mathcal{C}$ could map $g$ to $d$, and hence no satisfying assignment could map any element of the coset $Hg$ to $d$, and so the replacement of step (a) does not change the satisfiability of $H\mathcal{C}$. Moreover, the replacement of step (a) preserves the $\Delta$-closure of $\mathcal{C}$.

If the algorithm accepts, then $H\mathcal{C}$ is satisfiable by Lemma 5.

*Polynomiality of Running Time.* Observe that the implication graph $\mathcal{G}$ of $H\mathcal{C}$ can be constructed in polynomial time (by the $\Delta$-helpfulness of $H\mathcal{C}$): $r$ can be computed in polynomial time (as the $(G, H)$ same coset problem is decidable in polynomial time), and thus the labels of the edges of $\mathcal{G}$ can be computed in polynomial time (as products and inverses in $G$ can be computed in polynomial time). The loop of step (2) is executed at most a constant number of times, since there are only a constant number of pairs $(g, d)$ with $g \in S$ and $d \in D$. □

Using an algorithm of [13] it can be shown that the pair consisting of the group $G = (\mathbb{Z}, +)^k$ and a subgroup of the form $H = (d_1\mathbb{Z}, +) \times \cdots \times (d_k\mathbb{Z}, +)$ (for any $k \geq 1$ and $d_1, \ldots, d_k \geq 0$) is $\Delta$-helpful; hence, we obtain the following corollary.

**Corollary 2.** *Suppose that $\Gamma$ is a constraint language closed under a dual discriminator operation, $G$ is the group $(\mathbb{Z}, +)^k$ and $H$ is the $G$-subgroup $(d_1\mathbb{Z}, +) \times \cdots \times (d_k\mathbb{Z}, +)$ for some $k \geq 1$ and $d_1, \ldots, d_k \geq 0$. Then, $\mathsf{PeriodicCSP}(\Gamma; G, H)$ is decidable in polynomial time.*

## 6 Conclusions and future work

We identified two broad tractable subclasses of the periodic CSP, which were described using the language of closure properties. One prominent and intriguing open question is whether or not the periodic version of CSPs consisting of linear equations over a finite field (equivalently, constraints closed under an affine operation [18]) are tractable, or even decidable. We conjecture that this subclass of the periodic CSP is decidable in polynomial time; demonstrating that such CSPs have "periodic" solutions and that an upper bound on the size of the period is computable might be intermediate steps to resolving this conjecture. As far as the non-periodic, usual CSP is concerned, constraints of the described form are decidable in polynomial time via Gaussian elimination.

Resolution of this conjecture in the positive would provide a full dichotomy theorem for constraint languages over domain size two, by combining the results in this paper with the undecidability result of Freedman [12]. This dichotomy theorem would give a direct correspondence between CSP complexity and periodic CSP complexity (in domain size two), stating that those constraint languages $\Gamma$ such that $\mathsf{CSP}(\Gamma)$ is tractable give rise to a tractable periodic CSP, while those constraint languages $\Gamma$ such that $\mathsf{CSP}(\Gamma)$ is NP-complete give rise to an undecidable periodic CSP. In addition, such a resolution would also suggest the conjecture that for all constraint languages $\Gamma$, general problems of the form $\mathsf{PeriodicCSP}(\Gamma; (\mathbb{Z}, +)^k, (d_1\mathbb{Z}, +) \times \cdots \times (d_k\mathbb{Z}, +))$ are decidable in polynomial time if and only if $\mathsf{CSP}(\Gamma)$ is. Roughly speaking, this second conjecture states that constraint languages that are tractable in the context of the CSP are well-behaved in that they are also tractable in the context of the periodic CSP! A similar well-behavedness phenomenon seems to take place in the case of the quantified CSP [7], where the tractable constraint languages (in domain size two) are exactly the non-trivial tractable constraint languages in the case of CSP (given by Schaefer's theorem). We believe that further investigation and validation of this phenomenon may prove to be fruitful in understanding the problem structure of CSPs and the variants thereof.

*Acknowledgements.* The author wishes to thank Jon Kleinberg and Dexter Kozen for helpful discussions.

## References

1. Andrei Bulatov and Peter Jeavons. Tractable constraints closed under a binary operation. Technical Report PRG-TR-12-00, Oxford University, 2000.
2. Andrei A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of 43rd IEEE Symposium on Foundations of Computer Science*, pages 649–658, 2002.
3. Andrei A. Bulatov. Malt'sev constraints are tractable. Technical Report PRG-RR-02-05, Oxford University, 2002.

4. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. Constraint satisfaction problems and finite algebras. In *Proceedings 27th International Colloquium on Automata, Languages, and Programming – ICALP'00*, volume 1853 of *Lecture Notes In Computer Science*, pages 272–282, 2000.
5. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. The complexity of maximal constraint languages. In *ACM Symposium on Theory of Computing*, pages 667–674, 2001.
6. M.C. Cooper, D.A. Cohen, and P.G. Jeavons. Characterising tractable constraints. *Artificial Intelligence*, 65(2):347–361, 1994.
7. Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2001.
8. V. Dalmau and J. Pearson. Closure functions and width 1 problems. In *Principles and Practice of Constraint Programming—CP'99*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer-Verlag, 1999.
9. Victor Dalmau. A new tractable class of constraint satisfaction problems. In *6th International Symposium on Artificial Intelligence and Mathematics*, 2000.
10. Victor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *Constraint Programming '02*, Lecture Notes In Computer Science, 2002.
11. Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.
12. Michael Freedman. K-sat on groups and undecidability. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 572–576, 1998.
13. Franz Höfting and Egon Wanke. Polynomial algorithms for minimum cost paths in periodic graphs. In *SODA*, pages 493–499, 1993.
14. Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.
15. Peter Jeavons, David Cohen, and Martin Cooper. Constraints, consistency, and closure. *Articial Intelligence*, 101(1-2):251–265, 1998.
16. Peter Jeavons, David Cohen, and Justin Pearson. Constraints and universal algebra. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):51–67, 1998.
17. P.G. Jeavons, D.A. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings of 1st International Conference on Principles and Practice of Constraint Programming*, pages 276–291. Springer-Verlag, 1995.
18. P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.
19. M.V. Marathe, H.B. Hunt III, D.J. Rosenkrantz, and R.E. Stearns. Theory of periodically specified problems: Complexity and approximability. In *Proc. 13th IEEE Conference on Computational Complexity*, 1998.
20. M.V. Marathe, H.B. Hunt III, R.E. Stearns, and V. Radhakrishnan. Approximation algorithms for pspace-hard hierarchically and periodically specified problems. *SIAM Journal on Computing*, 27(5):1237–1261, 1998.
21. James Orlin. The complexity of dynamic languages and dynamic optimization problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 218–227, 1981.
22. Justin K. Pearson and Peter G. Jeavons. A survey of tractable constraint satisfaction problems. Technical report, Royal Holloway, University of London, 1997.
23. Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.