Knots and the Constraint Satisfaction Problem

Peter Golbus, Robert W. McGrail, Mona Merling, Kenneth Ober, Mary Sharac, and Japheth Wood

The Laboratory for Algebraic and Symbolic Computation
Reem-Kayden Center for Science and Computation
Bard College
Annandale-on-Hudson, NY 12504
{pg237,mcgrail,mm386,ms182,jwood}@bard.edu,
kenneth.ober@gmail.com

Abstract. This work presents a method for associating a class of constraint satisfaction problems to a three-dimensional knot. Given a knot, one can build a knot quandle, which is generally an infinite free algebra. The desired collection of problems is derived from the finite quotients of the knot quandle by applying theory that relates finite algebras to constraint languages. Along the way, notions of tractable and NP-complete quandles and knots are developed. Finally, a partial, computational classification of Rolfsen's Knot Table [25] is described in which many knots in this collection are revealed as NP-complete.

1 Introduction

Since Cook's formulation of NP-completeness [3], computer scientists have labored to unravel the mysteries of nondeterministic polynomial time [28]. Early efforts included the building of a catalogue of individual NP-complete combinatorial problems in the hope that one or more would provide significant insight [18]. In the meantime, more structurally-oriented approaches have emerged that instead focus on subclasses of NP. A notable example is the development of descriptive complexity [6, 12], which considers complexity classes axiomatized by fragments of (existential) second-order logic.

Another promising avenue restricts attention to subclasses of CSP, the class of constraint satisfaction problems [21]. Early on, Schaefer proved that every Boolean constraint satisfaction problem is NP-complete or tractable [26]. Feder and Vardi conjectured that this dichotomy holds for all of CSP [7]. Since then, Bulatov has extended Schaefer's result to three-element domains [1].

More importantly, Feder and Vardi showed that a solution to a constraint satisfaction problem corresponds to a homomorphism between certain finite, first-order structures. This idea was further refined by Jeavons and others [14, 15], and has led to significant insight into the structure of tractable subclasses of CSP [9, 10]. In particular, Jeavons, Cohen, and Pearson explored the relationship between CSP and universal algebra [16].

In [2], Bulatov, Jeavons, and Krokhin used the language of relational clones [29] and tame congruence theory [11, 20] to formulate notions of tractable and

NP-complete algebras. They showed that the process of classifying finite algebras as tractable or NP-complete need only consider the surjective ones, and proved P/NP-complete dichotomy for finite strictly simple surjective algebras. Moreover, they identified the class of idempotent algebras, all of which are surjective, as a prime target for the next round of dichotomy results.

This article introduces a new geometric avenue to the study of the constraint satisfaction problem, tractability, and NP-completeness. In particular, it presents a notion of constraint satisfaction problem over a 3-dimensional knot [4, 25]. This is made possible by the relationship between knots and a variety of algebras [11, 20] known as quandles [17]. Quandles are idempotent, so their study is relevant to the current trajectory of research into algebras and CSP.

2 Knots and Quandles

The basics of knot theory are reviewed in this section. More extensive treatments can be found in [4, 25]. A **knot** $\mathcal{K}$ is a continuous embedding of the unit circle $\mathbb{S}^1$ into $\mathbb{R}^3$. A knot $\mathcal{K}$ is usually identified with its oriented image in $\mathbb{R}^3$. Two knots $\mathcal{K}_1$ and $\mathcal{K}_2$ are **ambient isotopic** if the complement spaces $\mathbb{R}^3 - \mathcal{K}_1$ and $\mathbb{R}^3 - \mathcal{K}_2$ are homeomorphic. This captures the notion of continuous deformation of one knot into another. That is, $\mathcal{K}_1$ can be so transformed into $\mathcal{K}_2$ if and only if the two knots are ambient isotopic.

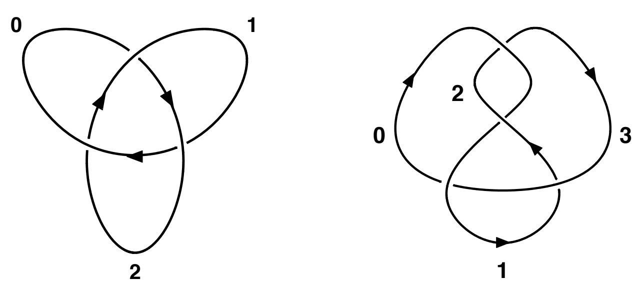

Fig. 1. Trefoil (3_1) and Figure Eight (4_1) Knots

It is often convenient to visualize a knot via 2-dimensional projection. The Trefoil and Figure Eight knots are so presented in Figure 1. The arrows indicate the orientation. Notice that the projection of the Trefoil has three crossings. Any knot that is ambient isotopic to the Trefoil will have at least three crossings in all of its projections. A knot projection that realizes the minimum possible number of crossings is called **reduced**.

Each crossing of a knot projection causes an apparent break in the segment of the strand below the crossing. For the duration of this article, the unbroken segments of the strand are called **arcs**. Each arc is labeled by an integer.

Certain knots $\mathcal{K}$ have an **Alexander-Briggs** representation n_k. In this case, $\mathcal{K}$ has rank k among all represented knots that have a reduced projection with n crossings. The relative rank k is purely nominal. The Alexander-Briggs notation for the Trefoil is 3_1.

2.1 Quandles

Joyce [17] introduced the notion of quandle as an algebraic invariant of knots. Quandles are defined here from the point of view of universal algebra [11, 20], since this perspective is useful to the development of Section 2.3.

Definition 1. *A quandle* $\mathbf{Q} = (Q, \{\triangleright, \blacktriangleright\})$ *is a set* Q *together with binary operations* $\triangleright, \blacktriangleright\colon Q \times Q \to Q$ *satisfying the following axioms:*

Idempotence: $\forall x(x \triangleright x = x)$;
Right Cancellation A: $\forall xy((x \triangleright y) \blacktriangleright y = x)$;
Right Cancellation B: $\forall xy((x \blacktriangleright y) \triangleright y = x)$; *and*
Right Self-Distributivity: $\forall xyz((x \triangleright y) \triangleright z = (x \triangleright z) \triangleright (y \triangleright z))$.

The simplest examples are the **unary quandles $\mathbf{U_n}$** where n is a positive integer. The underlying set of $\mathbf{U_n}$ is $\{0, 1, \ldots, n-1\}$ and the operations $\triangleright$ and $\blacktriangleright$ simply project the first argument: $x \triangleright y = x \blacktriangleright y = x$. That $\mathbf{U_n}$ satisfies the quandle identities is immediately obvious. The **dihedral quandle $\mathbf{D_n}$** has the

$\triangleright, \blacktriangleright$	0	1
0	0	0
1	1	1

$\triangleright, \blacktriangleright$	0	1	2
0	0	2	1
1	2	1	0
2	1	0	2

Fig. 2. $\mathbf{U_2}$ and $\mathbf{D_3}$

same underlying set as $\mathbf{U_n}$, but its operations are defined by $x \triangleright y = x \blacktriangleright y = 2y - x$, where the arithmetic occurs modulo n. Figure 2 displays the operation tables for $\mathbf{U_2}$ and $\mathbf{D_3}$.

2.2 The Knot Quandle

Given a projection for a knot $\mathcal{K}$, one can construct a **quandle presentation** as follows: To each crossing, assign a simple identity using the relevant arc labels and one of two binary operations, $\triangleright$ or $\blacktriangleright$. Figure 3 illustrates the two cases. In the left diagram, arc a passes to arc c at the point where arc b crosses to the left. This translates to the equation $c = a \triangleright b$. The right diagram of Figure 3 has b crossing to the right instead, which corresponds to $c = a \blacktriangleright b$.

For example, the Trefoil projection of Figure 1 has the following presentation:

$$\mathcal{Q}(3_1) = \langle 0, 1, 2 | 1 = 0 \triangleright 2, 2 = 1 \triangleright 0, 0 = 2 \triangleright 1 \rangle.$$

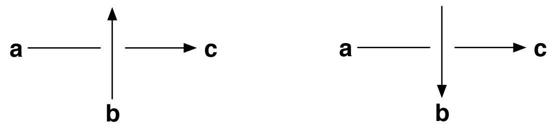

Fig. 3. Left and Right Crossings

2.3 The Reidemeister Moves

The relevance of quandle structure to knots can be inferred from the **Reidemeister Moves**. Reidemeister [24] proved that ambient isotopy can proceed through successive applications of three types of transformations along with planar deformations. In each move, some portion of the knot is the focus. If that part resembles one of the two diagrams of the move, it may be transformed to resemble the other diagram. It is assumed that the rest of the knot remains unchanged during this deformation. This results in a knot that is ambient isotopic to the previous one.

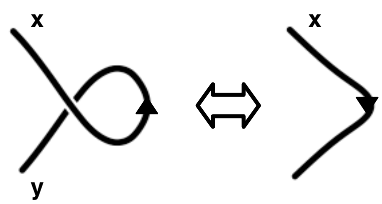

Fig. 4. Type I Reidemeister Move

An example of a **Type I** move appears in Figure 4. The left hand diagram has a segment of the knot looping behind itself. The crossing forms two arcs, x and y, and thus corresponds to the equation $y = x \triangleright x$. A simple twist of the loop yields the right hand diagram, reducing this part to one arc x. Here the role of y within the rest of the knot is now fulfilled by x. Hence $x = y = x \triangleright x$, so from ambient isotopy, one may infer that $\triangleright$ is idempotent.

Figure 5 presents an instance of a **Type II** move. The diagram on the left has arc y crossing over two points of the knot in succession, while on the right, y has been placed so that these two crossings do not occur. The point on the left hand diagram labeled by w is equated with its analogous location in the other diagram. Thus, $x = w = z \blacktriangleright y = (x \triangleright y) \blacktriangleright y$. Reversing the orientation on arc y leads to the companion right cancellation identity.

Lastly, right self distributivity can be gleaned from a **Type III** move (Figure 6). In this scenario, there are two segments that form one crossing in the center of both diagrams, and a third, single-arc segment z that crosses over the other

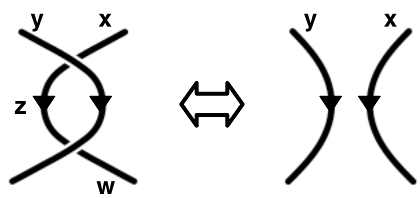

Fig. 5. Type II Reidemeister Move

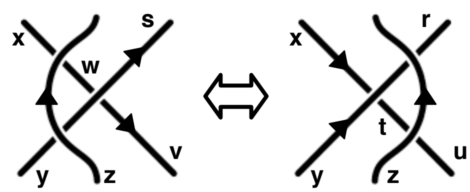

Fig. 6. Type III Reidemeister Move

two segments. The diagrams differ as to whether z crosses to the left or right of the central crossing. From this move, one can infer that $v = u$. Analyses of the crossings in both diagrams yield

$$(x \triangleright y) \triangleright z = t \triangleright z = u = v = w \triangleright s = (x \triangleright z) \triangleright (y \triangleright z).$$

The quandle axioms guarantee that ambient isotopic knots, as well as different projections of the same knot, have isomorphic knot quandles. Hence, the functorial notation $\mathcal{Q}(\mathcal{K})$ of Section 2.2 is well defined.

Since right cancellation ensures that the equation $x \triangleright y = z$ is provably equivalent to $x = z \blacktriangleright y$, the operation $\blacktriangleright$ is uniquely determined by $\triangleright$. One may dispense entirely with $\blacktriangleright$. For example, $\mathbf{Q}'$ is a subquandle of a quandle $\mathbf{Q}$ if it is closed under $\triangleright$, and a function $h : \mathbf{Q} \to \mathbf{Q}''$ is a quandle homomorphism if it preserves $\triangleright$. Henceforth, finite quandles will be presented via the Cayley table for $\triangleright$ alone. Elimination of $\blacktriangleright$ also extends to knot quandles: The knot quandle of 4_1 (Figure 2), which has both types of crossings, can be expressed as

$$\mathcal{Q}(4_1) = \langle 0, 1, 2, 3 | 0 = 1 \triangleright 2, 2 = 1 \triangleright 3, 2 = 3 \triangleright 0, 0 = 3 \triangleright 1 \rangle.$$

2.4 Tricolorable Knots and Finite Images

A precursor to Joyce's concept of quandle is **tricolorability** [23]. A tricoloring of a knot $\mathcal{K}$ is an assignment of one of three colors $\{0, 1, 2\}$ to each arc of $\mathcal{K}$ in such a way that every crossing either has three arcs of the same color or one arc of each color, and such that at least two distinct colors are employed.

For example, the integer labels in Figure 1 constitute a tricoloring of 3_1. One may also view these labels as elements of $\mathbf{D_3}$ (Figure 2). In fact, the equations of $\mathcal{Q}(3_1)$ hold in $\mathbf{D_3}$. In general, a tricoloring of $\mathcal{K}$ corresponds to a surjective quandle homomorphism $h : \mathcal{Q}(\mathcal{K}) \to \mathbf{D_3}$. In other words, $\mathcal{K}$ is tricolorable if and only if $\mathbf{D_3}$ is a quotient of $\mathcal{Q}(\mathcal{K})$.

Important to the development of Section 4 are situations in which there is a knot $\mathcal{K}$, a finite quandle $\mathbf{Q}$, and a surjective quandle homomorphism $h : \mathcal{Q}(\mathcal{K}) \to \mathbf{Q}$. For the sake of notational convenience, such a quotient $\mathbf{Q}$ of $\mathcal{Q}(\mathcal{K})$ will be called a **$\mathcal{K}$-quandle**. For example, it was shown above that $\mathbf{D_3}$ is a 3_1-quandle.

3 Constraint Satisfaction Problems over Finite Quandles

The basic elements of a **constraint satisfaction problem** [21] include a finite domain A, a countable collection of variables $V = \{v_1, v_2, \ldots, v_n, \ldots\}$, and a **constraint language** Γ, which is a collection of relations $R \subseteq A^n$ for various positive integers n. In this context, a **constraint over Γ** is a pair $\langle (v_{i_1}, v_{i_2}, \ldots, v_{i_m}), R \rangle$, where R is a relation in Γ of arity m.

3.1 Example: Boolean Satisfiability

The main concepts of constraint satisfaction are illustrated through **Boolean Satisfiability** of propositional formulas in conjunctive normal form (CNF), a well known NP-complete constraint satisfaction problem [8, 18]. In general, a proposition ϕ is satisfiable if there is a truth assignment of the variables of ϕ that renders the formula true.

A clause ψ is a disjunction of literals, which are propositions of the form v or $\neg v$ for a variable v. A proposition is in CNF if it can be written as a conjunction of clauses. Boolean satisfiability (Sat) is the problem of determining whether a CNF formula ϕ is satisfiable. It is one of the earliest recognized NP-complete problems [18]. For example, the formulas $\alpha = (\neg v_1 \vee v_2) \wedge (v_3 \vee \neg v_2)$ and $\beta = \neg v_1 \wedge v_1$ are in CNF. Clearly, α is satisfiable while β is not.

One can recast Sat within the realm of formal constraints. Let the domain A be $\{0, 1\}$. Consider a clause $\psi = \bigvee_{j=1}^{k} L_j$, where each literal L_j takes the form v_{i_j} or $\neg v_{i_j}$. In the case where L_j is v_{i_j}, let the relation R_j be the set of k-tuples over $\{0, 1\}$ that have value 1 at the jth coordinate. Otherwise, R_j is the set of k-tuples that have value 0 at the jth coordinate. Notice that if one identifies 1 to true and 0 to false, R_j represents the collection of truth assignments to the variable vector $(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$ under which L_j is true. Furthermore, the relation $R_\psi = \bigcup_{j=1}^{k} R_j$ excludes the only k-vector that makes each of the L_j false. The constraint associated to ψ is $C_\psi = \langle (v_{i_1}, v_{i_2}, \ldots, v_{i_k}), R_\psi \rangle$. This constraint is satisfied under all variable assignments that satisfy ψ.

For instance, the first clause $\psi_1 = \neg v_1 \vee v_2$ of the proposition α introduced above corresponds to the constraint $C_{\psi_1} = \langle (v_1, v_2), \{(0, 0), (0, 1), (1, 1)\} \rangle$. Note that the only ordered pair not included, $(1, 0)$, makes ψ_1, and hence α, false.

3.2 CSP(Γ)

Definition 2. *Given a domain A, a collection of variables V, and a constraint language Γ over A,* CSP(Γ) *is the combinatorial decision problem with the following components:*

Instance: *An instance of* CSP(Γ) *is a triple $\mathcal{I} = (V', A, \mathcal{C})$ where $\mathcal{C}$ is a finite set of constraints over Γ and V' is a finite subset of V.*

Solution: *A solution to an instance $\mathcal{I}$ of* CSP(Γ) *is a function $\theta : V' \to A$ such that for every constraint $\langle (v_1, v_2, \ldots, v_m), R \rangle \in \mathcal{C}$,*

$$(\theta(v_1), \theta(v_2), \ldots, \theta(v_m)) \in R.$$

For Sat, let $A = \{0, 1\}$, and let Γ be the collection of all relations over A that exclude exactly one tuple. Given a CNF proposition $\phi = \bigwedge_{l=1}^{m} \psi_j$, the associated CSP instance is $\mathcal{I}_\phi = (V_\phi, A, \mathcal{C}_\phi)$, where $\mathcal{C}_\phi = \{C_{\psi_1}, C_{\psi_2}, \ldots, C_{\psi_m}\}$ and V_ϕ is the set of variables appearing in $\mathcal{C}_\phi$. Applying this construction to the running example $\alpha = (\neg v_1 \vee v_2) \wedge (v_3 \vee \neg v_2)$ yields the instance

$$\mathcal{I}_\alpha = (V_\alpha, A, \{C_{\psi_1}, C_{\psi_2}\})$$

where $V_\alpha = \{v_1, v_2, v_3\}$,

$$C_{\psi_1} = \langle (v_1, v_2), \{(0,0), (0,1), (1,1)\} \rangle,$$

and

$$C_{\psi_2} = \langle (v_3, v_2), \{(0,0), (1,0), (1,1)\} \rangle.$$

For a finite constraint language Γ, consider any algorithm which, when given an instance $\mathcal{I} = (V', A, \mathcal{C})$ of CSP(Γ), produces a solution $\theta : V' \to A$ whenever one exists, and otherwise indicates that none exists. Every finite Γ has such an algorithm. In many cases this algorithm succeeds in time polynomial in the length of the description of $\mathcal{I}$. However, CSP(Γ) is generally NP-hard.

Definition 3. *The constraint language Γ is* **tractable** *if for every finite $\Gamma' \subseteq \Gamma$ there exists a polynomial time algorithm to decide* CSP(Γ')*. The constraint language Γ is* **NP-complete** *if* CSP(Γ') *is NP-complete for some finite $\Gamma' \subseteq \Gamma$.*

For example, k-satisfiability (k-Sat) considers the satisfiability of CNF formulas constructed from clauses with no more than k literals. The constraint language for k-Sat is finite. For $k = 2$ this problem is tractable. However, it has long been known that 3-Sat is NP-complete [18].

3.3 CSP and Quandles

According to the template of Definition 2, one need only find suitable representatives for the domain A and the constraint language Γ in order to construct a notion of constraint satisfaction problem over a finite quandle $\mathbf{Q}$. The former is straightforward; let the domain A be the underlying set Q. For the latter, one may appeal to the subquandle structure of $\mathbf{Q}$. A **finite subpower** of $\mathbf{Q}$ is a subquandle of the direct product $\mathbf{Q^n}$ for some nonnegative integer n. Denote the set of finite subpowers of $\mathbf{Q}$ by $\mathrm{SP}_{fin}(\mathbf{Q})$. Notice that each $\mathbf{Q'} \in \mathrm{SP}_{fin}(\mathbf{Q})$ is certainly a relation over the set Q. Let the constraint language Γ be $\mathrm{SP}_{fin}(\mathbf{Q})$.

Definition 4. *A quandle* $\mathbf{Q}$ *is* **tractable** *if* $\mathrm{SP}_{fin}(\mathbf{Q})$ *is tractable. A quandle* $\mathbf{Q}$ *is* **NP-complete** *if* $\mathrm{SP}_{fin}(\mathbf{Q})$ *is NP-complete. Let* $\mathrm{CSP}(\mathbf{Q})$ *stand for* $\mathrm{CSP}(\Gamma)$ *where* $\Gamma = \mathrm{SP}_{fin}(\mathbf{Q})$.

3.4 NP-Complete Quandles

The unary quandle $\mathbf{U_2}$ of Figure 2 plays a central role in this article. Every relation over $\{0, 1\}$ is a finite subpower of $\mathbf{U_2}$. Subsequently, the constraint language associated to 3-Sat is a finite subset of $\mathrm{SP}_{fin}(\mathbf{U_2})$, so by Definitions 3 and 4, $\mathbf{U_2}$ is NP-complete.

Suppose $\mathbf{Q'} \in \mathrm{SP}_{fin}(\mathbf{Q})$. Then $\mathrm{SP}_{fin}(\mathbf{Q'}) \subseteq \mathrm{SP}_{fin}(\mathbf{Q})$. If $\mathbf{Q'}$ is also NP-complete, then there exists a finite $\Gamma \subseteq \mathrm{SP}_{fin}(\mathbf{Q'})$ with $\mathrm{CSP}(\Gamma)$ NP-complete. Clearly, $\Gamma \subseteq \mathrm{SP}_{fin}(\mathbf{Q})$, as well, so $\mathbf{Q}$ is NP-complete.

Proposition 1. *If* $\mathbf{Q'} \in \mathrm{SP}_{fin}(\mathbf{Q})$ *is NP-complete, then so is* $\mathbf{Q}$.

Idempotence and right cancellation alone dictate that $\mathbf{U_2}$ is the only quandle of size 2, up to isomorphism. This proves the following:

Corollary 1. *Suppose* $\mathbf{Q}$ *has a subquandle of size 2. Then* $\mathbf{Q}$ *is NP-complete.*

$\triangleright$	0	1	2	3	4	5
0	0	3	1	4	2	0
1	2	1	5	0	1	3
2	4	0	2	2	5	1
3	1	5	3	3	0	4
4	3	4	0	5	4	2
5	5	2	4	1	3	5

Fig. 7. $\mathbf{Wood_6}$

There appears to be no shortage of NP-complete quandles. Included in this class is the quandle $\mathbf{Wood_6}$ of Figure 7. Notice that it has $\{0, 5\}$ as a subquandle so Corollary 1 applies. Also, $\mathbf{Wood_6}$ is a 3_1-quandle since it is a homomorphic image of $\mathcal{Q}(3_1)$. This quandle is used in Section 4.1 for the computational classification of certain knots.

3.5 Strictly Simple Quandles and Tractability

Let $\mathbf{A} = (A, F)$ be an algebra, so that A is a set and F a set of operations on A. A **term operation** is a constant-free expression $e(x_1, x_2, \ldots, x_k)$ over F regarded as a function $e : A^k \to A$. An algebra $\mathbf{A}$ is **surjective** if all of its term operations are surjective. For example, idempotence ensures that all quandles are surjective. An algebra $\mathbf{A}$ is **strictly simple** if it has no nontrivial subalgebras or quotients.

In [2, 5], Schaefer's dichotomy result is extended to finite strictly simple surjective algebras; the NP-complete, strictly simple surjective algebras are precisely those that have only **essentially unary** operations. A binary operation $* : A^2 \to A$ is essentially unary if there exists a unary operator $u : A \to A$ and an index i such that $x_1 * x_2 = u(x_i)$ for all values of $x_1, x_2 \in A$. The only strictly simple quandle with an essentially unary $\rhd$ is the unary quandle $\mathbf{U_2}$.

Thus, any other strictly simple quandle must be tractable. It can be shown that for a prime $p > 2$, the quandle $\mathbf{D_p}$ is strictly simple. Hence, there are infinitely many tractable quandles.

3.6 Notes for Section 3

The formulation of the CSP has experienced some evolution since its introduction by Montanari [19, 21]. The presentation of CSP(Γ) here closely follows the notational conventions of [2, 5]. In fact, Definition 2 is lifted from [5] virtually unaltered.

The presentation of CSP($\mathbf{Q}$) of Sections 3.3 through 3.5 is just the tip of the algebra/CSP iceberg [2, 5, 7, 16]. In this tradition, given a finite algebra $\mathbf{A} = (A, F)$, the constraint language of interest is the set Inv($\mathbf{A}$) of relations over A fixed by the non-constant term operations in F. For a quandle $\mathbf{Q}$, it follows that $\mathrm{Inv}(\mathbf{Q}) = \mathrm{SP}_{fin}(\mathbf{Q})$, which significantly simplifies the presentation.

4 Constraint Satisfaction Problems over Knots

Given a knot $\mathcal{K}$, the knot quandle $\mathcal{Q}(\mathcal{K})$ is generally an infinite algebra, and so it does not present an ideal setting for constraint satisfaction problems as formulated in Section 3. A more appropriate context can be had by instead considering a finite, homomorphic image of $\mathcal{Q}(\mathcal{K})$—i.e., a $\mathcal{K}$-quandle.

In this way, a **constraint satisfaction problem over** $\mathcal{K}$ is a constraint satisfaction problem over $\mathbf{Q}$ for some $\mathcal{K}$-quandle $\mathbf{Q}$. The knot $\mathcal{K}$ is **tractable** if $\mathbf{Q}$ is tractable for all $\mathcal{K}$-quandles $\mathbf{Q}$, and is **NP-complete** if $\mathbf{Q}$ is NP-complete for at least one $\mathcal{K}$-quandle $\mathbf{Q}$.

4.1 A Partial Computational Classification of Rolfsen's Knot Table

The Rolfsen Knot Table [25] includes all knots whose reduced forms have 10 or fewer crossings. For each knot $\mathcal{K}$ in this collection it was determined whether

$\mathbf{Wood_6}$, which is shown to be NP-complete in Section 3.4, is a $\mathcal{K}$-quandle. An affirmative answer for $\mathcal{K}$ proved that $\mathcal{K}$ is NP-complete. For example, in Section 3.4, it was also determined that $\mathbf{Wood_6}$ is a 3_1-quandle, so 3_1 is NP-complete.

The reader can verify the existence of a surjective quandle homomorphism $g : \mathbf{Wood_6} \to \mathbf{D_3}$. So if $h : \mathcal{Q}(\mathcal{K}) \to \mathbf{Wood_6}$ is a surjective homomorphism, then so is $g \circ h : \mathcal{Q}(\mathcal{K}) \to \mathbf{D_3}$. Subsequently, only the tricolorable knots in Rolfsen's collection had to be tested.

A program written in SWI-Prolog [30] converted a braid representation of each of these knots to a quandle presentation and then searched for a nontrivial solution for the presentation in $\mathbf{Wood_6}$. For good measure, the latter stage of this process was repeated using alternative quandle presentations computed from KnotPlot [27] images. Each positive result was verified by hand computation. Every knot listed in Figure 8 was proved NP-complete this way.

Crossings	**Rank**
3	1
6	1
7	4, 7
8	5, 10, 11, 15, 18, 19, 20, 21
9	1, 2, 4, 6, 10, 11, 15, 16, 17, 23, 24, 28, 29, 34, 35, 37, 38, 40, 46, 47, 48
10	5, 9, 10, 14, 19, 21, 29, 31, 32, 36, 40, 42, 59, 61, 62, 63, 64, 65, 66, 67, 68, 69, 74, 75, 76, 77, 78, 82, 84, 85, 87, 89, 96, 97, 98, 99, 103, 106, 107, 108, 112, 113, 114, 120, 122, 136, 139, 140, 141, 142, 143, 144, 145, 146, 147, 158, 159, 160, 163, 164, 165

Fig. 8. Some NP-Complete Knots of Rolfsen's Knot Table

4.2 Current Challenges

Conspicuously absent from Section 4.1 is any mention of tractable knots. So far, verifying tractability has proved substantially more challenging than demonstrating NP-completeness. At the submission time of this article, only the Unknot 0_1, which has trivial knot quandle, is known to be tractable. This remains an active area of research for the ASC lab.

Meanwhile, Corollary 1 ensures a plethora of NP-complete quandles, but $\mathbf{Wood_6}$ is the only known NP-complete $\mathcal{K}$-quandle. Further progress in this area will require a larger catalogue of NP-complete $\mathcal{K}$-quandles. Possible avenues of exploration include finite Alexander quandles [13, 22].

5 Conclusion

The theory developed in Sections 3 and 4 might provide a useful classifying invariant for knots. However, this was not the original purpose of this work. Rather, the motivation has been to provide a path whereby the tools of knot theory can shed light on the mysteries of nondeterministic polynomial time through the constraint satisfaction problem. The authors of this article continue to seek geometric characteristics of knots, such as tricolorability, that are in some way related to computational phenomena.

5.1 Acknowledgements

Aleksandar Chakarov, Bella Manoim, Tomasz Przytycki, Sphurti Saraph, Georgi Smilyanov, and Adina-Raluca Stoica verifed the NP-complete status of the bulk of the knots listed in Figure 8. Also, Bruce F. Golbus, Gregory Landweber, and Tracey Baldwin McGrail provided many corrections and suggestions toward improving this article.

References

1. Andrei A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *43rd IEEE Symposium on Foundations of Computer Science, FOCS '02*, pages 649–658, Vancouver, Canada, November 2002.
2. 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.
3. Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing (STOC'71)*, pages 151–158, Shaker Heights, Ohio, 1971.
4. Peter Cromwell. *Knots and Links*. Cambridge University Press, 2004.
5. Victor Dalmau. A new tractable class of constraint satisfaction problems. In *Proceedings of the 6th International Symposium on Artificial Intelligence and Mathematics*, 2000.
6. Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Monographs in Mathematics. Springer, 2nd edition, 2005.
7. Tomas 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*, 28:57–104, 1998.
8. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
9. Martin Grohe. The structure of tractable constraint satisfaction problems. In *Proceedings of the 31st Symposium on Mathematical Foundations of Computer Science (MFCS'06)*, volume 4162 of *Lecture Notes in Computer Science*, pages 58–72, 2006.
10. Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1), March 2007.

11. David Hobby and Ralph N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Comtemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.
12. Neil Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, New York, 1999.
13. A. Inoue. Quandle homomorphims of knot quandles to alexander quandles. *Journal of Knot Theory and its Ramifications*, 10:813–821, 2001.
14. Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200(1-2):185–204, 1998.
15. Peter Jeavons, David Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, July 1997.
16. Peter Jeavons, David Cohen, and Justin Pearson. Constraints and universal algebra. *Annals of Mathematics and Artificial Intelligence*, 24:51–67, 1998.
17. David Joyce. A classifying invariant of knots, the knot quandle. *Journal of Pure and Applied Algebra*, 23:37–66, 1982.
18. Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum, New York, 1972.
19. A. K. Mackworth and E. C. Freuder. The complexity of constraint satisfaction revisited. *Artificial Intelligence*, 59:57–62, 1993.
20. Ralph N. McKenzie, G. F. McNulty, and W. F. Taylor. *Algebras, Lattices, and Varieties*, volume I. Wadsworth and Brooks, California, 1987.
21. Ugo Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.
22. Sam Nelson. Classification of finite Alexander quandles. *Topology Proceedings*, 27:245–258, 2003.
23. Jozef H. Przytycki. 3-coloring and other elementary invariants of knots. In *Knot Theory*, volume 42, pages 275–295. Banach Center Publications, 1998.
24. H. Reidemeister. *Knoten und Gruppen*. Abh. Math. Sem. University of Hamburg, 1926.
25. Dale Rolfsen. *Knots and Links*, volume 7 of *Mathematical Lecture Series*. Publish or Perish, 1976.
26. T. J. Schaefer. The complexity of satisfiability problems. In *Proceeding of the 10th ACM Symposium on the Theory of Computing (STOC'78)*, pages 216–226, 1978.
27. Robert Glenn Scharein. *KnotPlot: A Program for Viewing Mathematical Knots*. Centre for Experimental and Constructive Mathematics, Simon Fraser University, October 2002.
28. Michael Sipser. The history and status of the P versus NP question. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC'92)*, pages 603–618, 1992.
29. A. Szendrei. Term minimal algebras. *Algebra Universalis*, 32(4):439–477, 1994.
30. Jan Wielemaker. *SWI-Prolog 5.6 Reference Manual*, April 2006.