Algebraic Methods for Deciding Complexity of Constraint Satisfaction Problems

William DeMeo
williamdemeo@gmail.com

Hawaii Logic Seminar

29 September 2016

What is a CSP?

Informally, a **C**onstraint **S**atisfaction **P**roblem consists of

- a list of variables ranging over a finite domain and
- a set of constraints on those variables.

Problem: can we assign values to all the variables so that all of the constraints are satisfied?

Examples

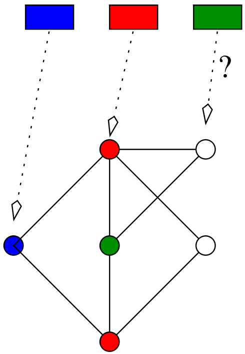
A system of linear equations is a CSP

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
&\vdots\\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m
\end{aligned}$$

Also, a system of nonlinear equations is a CSP

$$
\begin{aligned}
a_{11}x_1^2x_3 + a_{12}x_2x_3x_7 &+ \cdots + a_{1n}x_4x_n^3 = b_1\\
a_{21}x_2x_5 + a_{22}x_2 \qquad &+ \cdots + a_{2n}x_4^3 \quad = b_2\\
&\vdots\\
a_{m1}x_3x_5x_8 + a_{m2}x_2 \qquad &+ \cdots + a_{mn}x_n \quad = b_m
\end{aligned}
$$

For a fixed k, determining whether a graph is k-colorable is a CSP



Given a propositional formula $\varphi(x_1, \ldots, x_n)$, determine whether φ is satisfiable

$$\varphi(x, y, z) = (x \vee y \vee z') \wedge (x' \vee y \vee z')$$

then

$$\varphi(0, 0, 1) = 1$$

Algorithms

There is an efficient algorithm (Gaussian elimination) for solving any linear system. That is

> There is an algorithm that accepts as input a linear system and decides whether that system has a solution.
>
> The running time of the algorithm is bounded above by $f(s)$ where f is a *polynomial* and s is the size of the system.

Algorithms

There is an efficient algorithm (Gaussian elimination) for solving any linear system. That is

> There is an algorithm that accepts as input a linear system and decides whether that system has a solution.
>
> The running time of the algorithm is bounded above by $f(s)$ where f is a *polynomial* and s is the size of the system.

The input, a particular system, is an instance of the problem LINEAR SYSTEM.

Similarly

There is an algorithm that accepts as input a graph and decides whether the graph is 2-colorable.

Running time bounded by $f(s)$, a *polynomial* in size s.

The input, a particular graph, is an instance of the problem 2-COLORABILITY.

There is an algorithm that accepts as input a formula, $\varphi = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_k$ (each φ_i bijunctive) and decides whether φ is satisfiable.
Running time bounded by $f(k)$, a *polynomial* in size k.

The intput formula φ is an instance of the problem 2-SAT.

We say that all these algorithms run in polynomial time.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

However, any candidate solution to either of these problems can be checked in polynomial-time.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

However, any candidate solution to either of these problems can be checked in polynomial-time.

Thus these problems are solvable in nondeterministic polynomial time.

Let X and Y be two problems. We write $X \leq_{\mathsf{p}} Y$ to indicate that Y is at least as hard as X.

Let X and Y be two problems. We write $X \leq_{\mathsf{p}} Y$ to indicate that Y is at least as hard as X.

Somewhat more precisely: any algorithm for solving Y can be transformed into an algorithm for X without drastically increasing its running time.

Let X and Y be two problems. We write $X \leq_{\mathsf{p}} Y$ to indicate that Y is at least as hard as X.

Somewhat more precisely: any algorithm for solving Y can be transformed into an algorithm for X without drastically increasing its running time.

It is possible for $X \leq_{\mathsf{p}} Y \leq_{\mathsf{p}} X$. In that case, write $X \equiv_{\mathsf{p}} Y$.

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

- $\mathbb{P} \subseteq \mathbb{NP}$
- Both $\mathbb{P}$ and $\mathbb{NP}$ are downsets, i.e.,
  $Y \in \mathbb{P} \;\&\; X \leq_{\mathsf{p}} Y \implies X \in \mathbb{P}$

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

- $\mathbb{P} \subseteq \mathbb{NP}$
- Both $\mathbb{P}$ and $\mathbb{NP}$ are downsets, i.e.,
  $Y \in \mathbb{P} \;\&\; X \leq_{\mathsf{p}} Y \implies X \in \mathbb{P}$

The maximal members of $\mathbb{NP}$ are called $\mathbb{NP}$-complete.

3-COLORABILITY, NONLINEAR SYSTEM, and 3-SAT are known to be $\mathbb{NP}$-complete.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away. Almost every problem that mathematicians actually care about can be solved efficiently. Just build bigger computers.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away. Almost every problem that mathematicians actually care about can be solved efficiently. Just build bigger computers.

In particular, this talk becomes pointless. So assume $\mathbb{P} \neq \mathbb{NP}$.

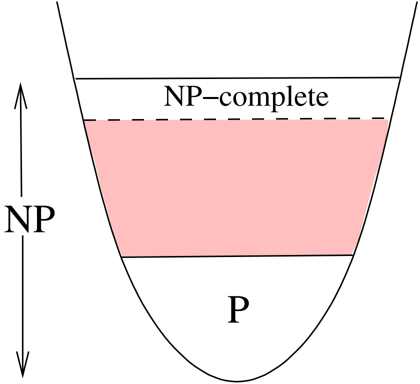
\$$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away. Almost every problem that mathematicians actually care about can be solved efficiently. Just build bigger computers.

In particular, this talk becomes pointless. So assume $\mathbb{P} \neq \mathbb{NP}$.

Theorem (Ladner, 1975)

If $\mathbb{P} \neq \mathbb{NP}$ then there are problems in $\mathbb{NP} - \mathbb{P}$ that are not $\mathbb{NP}$-complete.



If $\mathbb{P} \neq \mathbb{NP}$ then the pink area is nonempty.

Formal Definition of CSP

Let D be a set, n a positive integer
An n-*ary relation on* D is a subset of D^n

Formal Definition of CSP

Let D be a set, n a positive integer
An n*-ary relation on* D is a subset of D^n

$\mathrm{Rel}_n(D)$ denotes the set of all n-ary relations on D

$$\mathrm{Rel}(D) = \bigcup_{n>0} \mathrm{Rel}_n(D)$$

Let D be a finite set and $\Delta \subseteq \mathrm{Rel}(D)$

$\mathrm{CSP}(\langle D, \Delta \rangle)$ is the following decision problem:

Instance. A finite set $V = \{ v_1, \dots, v_n \}$ of variables and a finite set $\{ C_1, \dots, C_m \}$ of constraints;

each constraint C_i is a pair $(\langle x_{i1}, \dots, x_{i\,p_i} \rangle, \delta_i)$ in which $x_{i1}, \dots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

Let D be a finite set and $\Delta \subseteq \mathrm{Rel}(D)$

$\mathrm{CSP}(\langle D, \Delta\rangle)$ is the following decision problem:

Instance. A finite set $V = \{ v_1, \dots, v_n \}$ of variables and a finite set $\{ C_1, \dots, C_m \}$ of constraints;

each constraint C_i is a pair $(\langle x_{i1}, \dots, x_{i\,p_i} \rangle\,,\, \delta_i)$ in which $x_{i1}, \dots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

Question. Does there exist a solution, that is, a "context" $\rho \colon V \to D$, such that for all $i \leq m$, $\langle \rho(x_{i1}), \dots, \rho(x_{i\,p})\rangle \in \delta_i$?

Let D be a finite set and $\Delta \subseteq \mathrm{Rel}(D)$

$\mathrm{CSP}(\langle D, \Delta\rangle)$ is the following decision problem:

Instance. A finite set $V = \{ v_1, \dots, v_n \}$ of variables and a finite set $\{ C_1, \dots, C_m \}$ of constraints;

each constraint C_i is a pair $(\langle x_{i1}, \dots, x_{i\,p_i} \rangle\, , \, \delta_i)$ in which $x_{i1}, \dots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

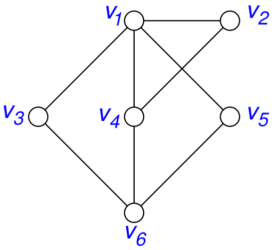
Question. Does there exist a solution, that is, a “context” $\rho\colon V \to D$, such that for all $i \leq m$, $\langle \rho(x_{i1}), \dots, \rho(x_{i\,p})\rangle \in \delta_i$?

$\mathrm{CSP}(\langle D, \Delta\rangle)$ always lies in $\mathbb{NP}$.

Example: 3-colorability

$$D = \{r, g, b\}, \quad \Delta = \{\kappa_3\}$$
$$\kappa_3 = \{\, (x, y) \in D : x \neq y \,\}$$

Then $\mathrm{CSP}(\langle D, \Delta \rangle)$ is the 3-colorability problem



$$V = \{v_1, \ldots, v_6\}$$
$$\langle v_1, v_2 \rangle \in \kappa$$
$$\langle v_1, v_3 \rangle \in \kappa$$
$$\langle v_1, v_4 \rangle \in \kappa$$
$$\langle v_2, v_4 \rangle \in \kappa$$
$$\vdots$$
$$\langle v_5, v_6 \rangle \in \kappa$$

Two Motivating Questions

1. Dichotomy Conjecture
   Every $\mathrm{CSP}(\langle D, \Delta\rangle)$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.

Two Motivating Questions

1. Dichotomy Conjecture
   Every $\mathrm{CSP}(\langle D, \Delta \rangle)$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.
2. Tractability Problem
   Characterize those CSPs that lie in $\mathbb{P}$.

Two Motivating Questions

1. Dichotomy Conjecture
   Every $\mathrm{CSP}(\langle D, \Delta\rangle)$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.
2. Tractability Problem
   Characterize those CSPs that lie in $\mathbb{P}$.

What would a characterization look like? What language could we use?

Polymorphisms

Definition

Let $\delta \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* δ if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in \delta \implies \big(f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})\big) \in \delta$$

Polymorphisms

Definition

Let $\delta \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* δ if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in \delta \implies \big(f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})\big) \in \delta$$

f is an n-*ary operation* on D.

Polymorphisms

Definition

Let $\delta \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* δ if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in \delta \implies \\ \big(f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})\big) \in \delta$$

$$\begin{array}{cccccc}
a_{11} & a_{12} & \dots & a_{1k} & \in & \delta \\
a_{21} & a_{22} & \dots & a_{2k} & \in & \delta \\
\vdots & \vdots & & \vdots & & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nk} & \in & \delta \\
\downarrow f & \downarrow f & & \downarrow f & & \\
\star & \star & \dots & \star & \in & \delta
\end{array}$$

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the *polymorphisms* of Δ.

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the *polymorphisms* of Δ.

Let F be a set of operations on D. Then $\mathrm{Inv}(F)$ denotes the set of all relations preserved by all operations in F.

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the *polymorphisms* of Δ.

Let F be a set of operations on D. Then $\mathrm{Inv}(F)$ denotes the set of all relations preserved by all operations in F.

Important point: $\langle D, \mathrm{Pol}(\Delta)\rangle$ is an algebraic structure

Theorem

Let $\Gamma, \Delta \subseteq \mathrm{Rel}(D)$. *Then*

$$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \implies \mathrm{CSP}(\Delta) \leq_{\mathsf{p}} \mathrm{CSP}(\Gamma).$$

Theorem

Let $\Gamma, \Delta \subseteq \mathrm{Rel}(D)$. *Then*

$$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \implies \mathrm{CSP}(\Delta) \leq_{\mathsf{p}} \mathrm{CSP}(\Gamma).$$

Thus, the richer the algebraic structure, the easier the corresponding CSP

One can go back and forth between relational and algebraic structures

$$\begin{array}{ccc} \textbf{Relational} & & \textbf{Algebraic} \\ \langle D, \Delta \rangle & \longrightarrow & \langle D, \mathrm{Pol}(\Delta) \rangle \\ \langle D, \mathrm{Inv}(F) \rangle & \longleftarrow & \langle D, F \rangle \end{array}$$

$$\mathrm{CSP}\langle D, \Delta \rangle \equiv_{\mathsf{p}} \mathrm{CSP}\langle D, \mathrm{Inv}(\mathrm{Pol}(\Delta)) \rangle$$

One can go back and forth between relational and algebraic structures

$$\begin{array}{ccc} \textbf{Relational} & & \textbf{Algebraic} \\ \langle D, \Delta \rangle & \longrightarrow & \langle D, \mathrm{Pol}(\Delta) \rangle \\ \langle D, \mathrm{Inv}(F) \rangle & \longleftarrow & \langle D, F \rangle \end{array}$$

$$\mathrm{CSP}\langle D, \Delta \rangle \equiv_{\mathsf{p}} \mathrm{CSP}\langle D, \mathrm{Inv}(\mathrm{Pol}(\Delta)) \rangle$$

Perhaps the expressive power of algebra can be used to classify CSPs.

Algebraic Facts

Let $\mathbf{A}$ and $\mathbf{B}$ be algebras

$\mathbf{B}$ a subalgebra of $\mathbf{A} \implies \mathrm{CSP}(\mathbf{B}) \leq_{\mathsf{p}} \mathrm{CSP}(\mathbf{A})$.

$\mathbf{B}$ a homomorphic image of $\mathbf{A} \implies \mathrm{CSP}(\mathbf{B}) \leq_{\mathsf{p}} \mathrm{CSP}(\mathbf{A})$.

$\mathrm{CSP}(\mathbf{A}^n) \equiv_{\mathsf{p}} \mathrm{CSP}(\mathbf{A})$

Theorem (Bulatov, Jeavons, Krokhin, 2000)

If $\langle D, \Delta\rangle$ is a core and every polymorphism is essentially unary, then $\mathrm{CSP}(\Delta)$ is $\mathbb{NP}$-complete.

f is *essentially unary* if $f(x_1, \dots, x_n) = g(x_j)$ for some unary g and some $j \leq n$.

Theorem (Bulatov, Jeavons, Krokhin, 2000)

If $\langle D, \Delta \rangle$ is a core and every polymorphism is essentially unary, then $\mathrm{CSP}(\Delta)$ is $\mathbb{NP}$-complete.

f is *essentially unary* if $f(x_1, \ldots, x_n) = g(x_j)$ for some unary g and some $j \leq n$.

Corollary

3-COLORABILITY, NONLINEAR SYSTEM, and 3-SAT are $\mathbb{NP}$-complete.

Informal reformulation of the dichotomy conjecture
If $\mathbf{A}$ has some kind of decent algebraic structure then $\mathrm{CSP}(\mathbf{A}) \in \mathbb{P}$ otherwise $\mathrm{CSP}(\mathbf{A})$ is $\mathbb{NP}$-complete.