Introduction into Mathematics of Constraint Satisfaction

Andrei Krokhin
Durham University

Part II

Outline of the Course

1. The CSP and its forms
   - Examples of CSPs
2. Complexity issues and computational questions
   - What questions do we ask about CSPs?
3. Mathematical techniques
   - What maths do we use to analyse those questions?

Parameterisation of CSP

With any instance of CSP one can associate two natural parameters reflecting

1. Which variables constrain which others, i.e.,
   - constraint scopes, or
   - query language, or
   - LHS structure $\mathcal{A}$ (as in $\mathcal{A} \to \mathcal{B}$).
2. How values for the variables are constrained, i.e.,
   - constraint relations, or
   - relational database, or
   - RHS structure $\mathcal{B}$ (as in $\mathcal{A} \to \mathcal{B}$).

General restrictions

For classes $\mathfrak{A}, \mathfrak{B}$ of structures, let CSP$(\mathfrak{A}, \mathfrak{B})$ denote the set of all CSP instances $(\mathcal{A}, \mathcal{B})$ with $\mathcal{A} \in \mathfrak{A}$ and $\mathcal{B} \in \mathfrak{B}$.

- Example: CSP$(\mathfrak{A}, \{K_3\})$ is 3-COLOURABILITY for graphs in $\mathfrak{A}$.
- Can study how the complexity depends on $\mathfrak{A}$.
- Example:
  - tractable for locally connected graphs
  - **NP**-complete for planar graphs (of degree ≤ 4)

Restricting LHS

For a class $\mathfrak{A}$ of structures, let CSP($\mathfrak{A}, -$) denote the set of all CSP instances $(\mathcal{A}, \mathcal{B})$ with $\mathcal{A} \in \mathfrak{A}$.

Terminology: structural restriction.

Example: CLIQUE.

For any fixed $\mathcal{A}$, CSP($\{\mathcal{A}\}, -$) is in **P**. Simply check each mapping $A \to B$. If $|A| = k$ then $|B|^k$ is polynomial in $|\mathcal{B}|$. Boring... Well, not quite — one can explore when better (faster) algorithms work.

How does the complexity of CSP($\mathfrak{A}, -$) depend on $\mathfrak{A}$?

Restricting RHS

- Fix a (possibly infinite) relational structure $\mathcal{B}$
- often called a constraint language or template, or Γ.
- CSP($\mathcal{B}$): given a finite structure $\mathcal{A}$ in the same signature as $\mathcal{B}$, is there a homomorphism $h : \mathcal{A} \to \mathcal{B}$?
- Same as evaluating a given $\exists\wedge$-formulas against $\mathcal{B}$.
- In the variable-value formulation, same as fixing a finite set of available constraint relations.
- Same as CSP$(-, \{\mathcal{B}\})$ except $\mathcal{B}$ is not part of input.

How does the complexity of CSP($\mathcal{B}$) depend on $\mathcal{B}$?

Dichotomy for Boolean Relations

Theorem 1 (Schaefer; STOC'78) *Let $\mathcal{B}$ be a Boolean structure. Then* CSP($\mathcal{B}$) *is tractable iff all relations in $\mathcal{B}$ satisfy one of the following conditions:*

1. *0-valid (1-valid),* — *trivial SAT*
2. *Horn (dual Horn),* — *Horn-SAT*
3. *bijunctive,* — *2-SAT*
4. *affine.* — *Linear Eq's*

Otherwise CSP($\mathcal{B}$) *is* **NP***-complete.*

Dichotomy for Graph H-coloring

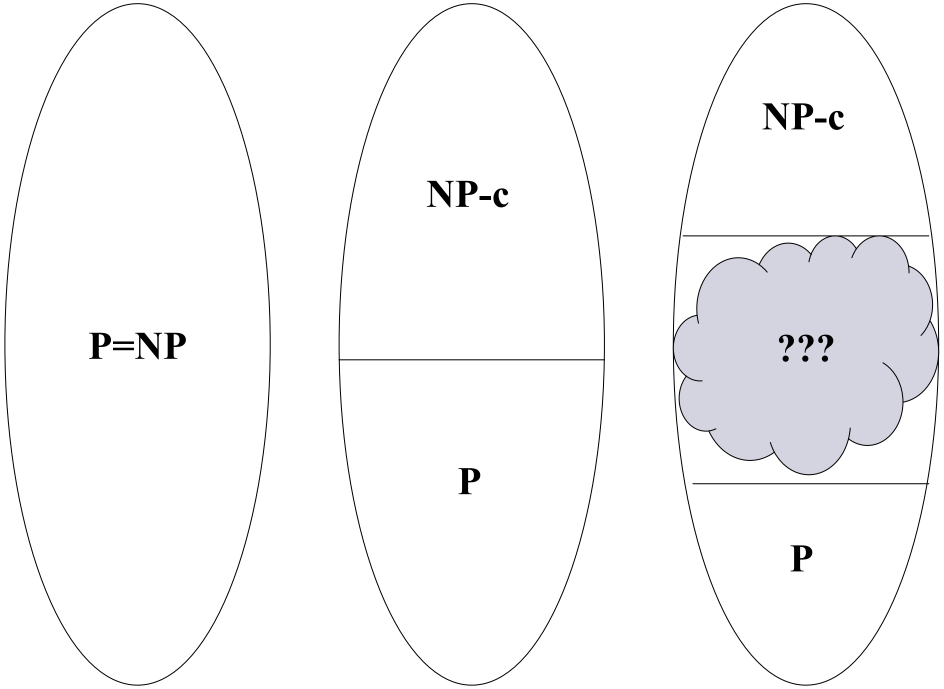
Theorem 2 (Hell,Nešetřil; JCombTh'1990) *Let* $\mathcal{H}$ *be an undirected graph. If it is bipartite or has a loop then* CSP($\mathcal{H}$) *is tractable, otherwise* CSP($\mathcal{H}$) *is* **NP**-*complete.*

Dichotomy Conjecture

Conjecture 1 (Feder,Vardi; SICOMP'1998)
Dichotomy Conjecture: For each fixed finite $\mathcal{B}$, the problem CSP($\mathcal{B}$) *is either tractable (i.e., in* **P***) or* **NP***-complete.*

Why is This Dichotomy Interesting?

Which of these diagrams is a true picture of **NP**?



Theorem 3 (Ladner'74) *If* **P** $\neq$ **NP** *then there are infinitely many complexity classes between* **P** *and* **NP**.

Non-Dichotomy Results

Theorem 4 (Bauslaugh'94) *For every computational problem L, there is an infinite digraph $\mathcal{B}$ such that L and* $\mathrm{CSP}(\mathcal{B})$ *are polynomial-time equivalent.*

Theorem 5 (Bodirsky,Grohe'08)

1. *There exist "nice" infinite relational structures $\mathcal{B}$ with* **coNP**-*intermediate* $\mathrm{CSP}(\mathcal{B})$.
2. *There is an efficiently decidable class $\mathfrak{A}$ of undirected graphs such that* $\mathrm{CSP}(\mathfrak{A}, -)$ *is* **NP**-*intermediate.*

The Three Approaches

The three main approaches to our classification problems are:

- via Combinatorics (Graphs & Posets)
  - Jarik's lectures, also my 3rd lecture
- via Logic and Games
  - LICS was last week; maybe some in 3rd lecture
- via Algebra
  - Ross' lectures

Combinatorics: Encoding CSP($\mathcal{B}$)

Theorem 6 (FV'98) *For every structure $\mathcal{B}$ there exist*

- *a poset $P_{\mathcal{B}}$;*
- *a bipartite graph $G_{\mathcal{B}}$;*
- *a digraph $H_{\mathcal{B}}$*

such that these problems are polynomially equivalent:

- CSP($\mathcal{B}$),
- *poset-retraction*$(P_{\mathcal{B}}) = \mathrm{CSP}(P_{\mathcal{B}} \cup \{\{b_i\} \mid b_i \in P_{\mathcal{B}}\})$,
- *bipartite graph-retraction*$(G_{\mathcal{B}})$,
- CSP($H_{\mathcal{B}}$).

Logic and Games Approach CSP($\mathcal{B}$)

One can view CSP($\mathcal{B}$) as the membership problem for the class of structures $\mathcal{A}$ such that $\mathcal{A} \to \mathcal{B}$.

Typical result describes the class CSP($\mathcal{B}$)

- by a logical specification (e.g., formula in a nice logic) that can be checked easily against a given structure, or
- as a class of structures $\mathcal{A}$ for which there exists an (easily detectable) winning strategy in a certain game on $\mathcal{A}$ and $\mathcal{B}$.

Details and examples: in my 3rd lecture (maybe)

A Semigroup Approach: Encoding CSP

Theorem 7 (Klíma, Tesson, Thérien '07)
For every finite structure $\mathcal{B}$, there is a finite semigroup S satisfying $x^2 = x$ and $xyz = yxz$ and such that $\mathrm{CSP}(\mathcal{B})$ *is poly-time equivalent to* SYSTEMS OF EQUATIONS *over S.*

There's a full classification result for monoids, though ...

An Algebraic Approach: Scheme of Things

Constraint language $\mathcal{B}$

$\Downarrow$

Set of relations $\langle \mathcal{B} \rangle$

$\Updownarrow$ ←-- crucial

Set of operations $\mathrm{Pol}(\mathcal{B})$

$\Updownarrow$

Finite algebra $\mathbf{A} = (B, \mathrm{Pol}(\mathcal{B}))$

$\Updownarrow$

Hard CSP $\longleftarrow$ Variety generated by $\mathbf{A}$ $\longrightarrow$ Easy CSP

CSP-related Problems

- Satisfiability with additional features
- Counting/Enumerating solutions
- Equivalence/Isomorphism of solution spaces
- Unique solution
- Quantified CSP
- Connectivity of the solution graph
- Optimization
  - Max CSP and Min CSP
  - MinCost CSP and Soft/Valued CSP

Bounded Occurrence CSP

- A structure is said to have degree at most k if each element appears $\leq k$ times in its relations.
- Let $\mathfrak{D}_k$ be the class of all structures of degree $\leq k$.
- For any $\mathcal{B}$, can consider the problem CSP$(\mathfrak{D}_k, \{\mathcal{B}\})$.
- Equivalently, restriction to instances where each variable appears $\leq k$ times.

Theorem 8 (Kratochvil,Stavicky,Tuza'93) *There is an exponential function $f(k) \geq k$ such that each instance of k-*SAT *with $\leq f(k)$ variable occurrences (and one per constraint) is satisfiable, but k-*SAT *with $\leq f(k)+1$ variable occurrences is* **NP***-complete.*

Bounded Occurrence Boolean CSP

- Fix Boolean structure $\mathcal{B}$, consider CSP$(\mathfrak{D}_k, \{\mathcal{B}\})$.
- Assume can use constants 0,1 in instances.
- If $k \geq 3$, same as unbounded case [Dalmau,Ford'03]
- Let $k = 2$. Same as unbounded case when some relation in $\mathcal{B}$ is not a Δ-matroid [Feder'01]
  - Δ-matroid = 2-step condition
- Some types of Δ-matroids - easy [Dalmau,Ford'03]
- Full classification is open
  - closely related to Δ-matroid parity problem

CSP with global constraints

- Global constraint, given additionally — need to check the entire assignment to ensure satisfiability
- Very popular in AI, ≥ 350 different kinds of GCs
- Example: surjectivity — each value must be used
  - Recent: **NP**-hardness for C_4^r [Martin,Paulusma'11]
  - Open: complexity for No-Rainbow-Colouring
- Example: global cardinality constraint — prescribes how many times each value can be used (seen before?)
  - Full classification for CSP($\mathcal{B}$) with GCC [Bulatov,Marx'10]

Counting CSP

- Goal: count the number of solutions to an instance
- Boolean case done [Creignou,Hermann'95]
  - Linear equations are easy (why?)
  - The rest is hard
- All CSP($\mathcal{B}$) done [Bulatov,Grohe'05, Bulatov'07]
  - Algebraic approach works
  - Recent simplification [Dyer,Richerby'10]
- All CSP($\mathfrak{A}, -$) done [Dalmau,Jonsson'04]
- Recent: approximate counting [Goldberg et al.'10]

Listing/Enumeration CSP

- Goal: list all solutions
- All CSP($\mathfrak{A}, -$) done [Atserias,Grohe,Marx'08]
- If too many solutions: list with polynomial delay
  - Done for Boolean CSP($\mathcal{B}$) [Creignou et al'97]
  - Partial classif. for CSP($\mathfrak{A}, -$) [Bulatov et al'09]
  - Full classif. for CSP($\mathfrak{A}, -$) $\Rightarrow$ Dichotomy cracked

Equivalence/Isomorphism

- Equivalence: given two instances with the same variables, do they have the same solution sets?
- Isomorphism: can variables be permuted/renamed to make solution sets equal?
- Boolean case done [Böhler et al'02+'04]
- Equations over groups [Nordh'05]
- Strong classification results [Bova,Chen,Valeriote'11]
  - possibly full classification, depending on resolution of one purely universal-algebraic conjecture
- No work for CSP($\mathfrak{A}$, −) ???

Unique Solution

- Goal: decide whether there is a unique solution
  - variant: assuming there is at most one solution
- Belongs to complexity class
  $\mathbf{DP} = \{L \setminus L' \mid L, L' \in \mathbf{NP}\}$
- Boolean case - known to be **DP**-complete only under randomized reductions;
  derandomization: long-standing open problem
- Can relativize uniqueness to a subset of variables
  - Full classif. modulo resolution of Feder-Vardi [Jonsson,K'04]

Quantified CSP

- Goal: evaluate a $\forall\exists\wedge$-sentence against a structure $\mathcal{B}$.
- Example: $\forall x \exists y \forall z [(x \vee \neg y) \wedge (y \vee \neg z)]$
- standard problem within **PSPACE**.
- many results on complexity of QCSP$(\mathcal{B})$
- algebraic approach works, but not to full extent
- there is not even a good conjecture about dividing lines

Solution Graph Connectivity

- for a CSP($\mathcal{B}$) instance I, form the solution graph $G(I)$:
  - nodes are solutions
  - two solutions are adjacent if differ in one variable
- Connectivity of $G(I)$ studied
  - Boolean case almost done [Gopalan et al'08]
  - Graph colouring done [Cereceda et al'08]

Why Connectivity?

For a CSP instance I, let $d = \#\text{constraints}/\#\text{variables}$, called the density of I.

Fact 1 (Achlioptas,Coja-Oghlan'08)
There are two thresholds $t_1 < t_2$ such that

- *a random CSP instance with $d > t_2$ has no solution with high probability (w.h.p.)*
- *a random instance with $d < t_2$ has a solution w.h.p.*
- *there is a simple poly-time algorithm for instances with $d < t_1$*
- *no poly-time algorithm known for $t_1 < d < t_2$.*

Why Connectivity?

It is believed that

- $G(I)$ is connected for $d < t_1$
- $G(I)$ shatters into exponentially many components at $d = t_1$
- as $d \rightsquigarrow t_2$, the components become smaller and farther apart
- $G(I)$ is empty for $d > t_2$

Max CSP and Min CSP

- Given a CSP instance, find an assignment satisfying maximum # constraints (failing to satisfy min # constraints)
- Much work both on exact algorithms/complexity and on approximation algorithms/hardness.
- More in Venkat's and Ryan's lectures

MinCost CSP

- Each instance $(\mathcal{A}, \mathcal{B})$ has an associated family of costs $c_a(b) \in \mathbb{N}$ (of mapping $a \in A$ to $b \in B$)
- Goal: to find homomorphism $h : \mathcal{A} \to \mathcal{B}$ of minimum total cost $\sum_{a \in A} c_a(h(a))$
- Can do analysis with a fixed template $\mathcal{B}$
- Lots of papers on the (di)graph case [Gutin et al]
  - full solution [Hell,Rafiey'10]
- Algebraic approach works
  - full general solution [Takhanov'07-'10]

Valued CSP

- Used in AI to express preferences (desirability/cost)
- Each constraint has preferences over satisfying assignments, i.e. has a cost function instead of relation.
- Goal: find an assignment minimising overall cost.

VCSP

VALUED CSP (VCSP)

Instance: A collection $f_1(\mathbf{x}_1), \ldots, f_q(\mathbf{x}_q)$ of expressions over $V = \{x_1, \ldots, x_n\}$, each $f_i : B^{n_i} \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$

Goal: Find an assignment $\phi : V \to B$ that minimises the total cost; in other words, minimise the function $f : B^n \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$, defined by

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{q} f_i(\mathbf{x}_i).$$

VCSP

- All costs in $\{0, \infty\}$ = CSP
- All costs in $\{0, 1\}$ = Max CSP
- What is MinCost CSP?
- Can do analysis with a fixed “template”
  - VCSP day during the Algebra workshop