Promises, constraint satisfaction, and problems

Beyond universal algebra (part II)

Jakub Opršal

AAA 100, 7 Feb 2021

Durham University

overview

Part I (yesterday)

- algebraic approach to (promise) constraint satisfaction.

Part II (today)

- beyond algebraic approach
- open problems

previously on this tutorial…

Theorem. [Barto, Bulín, Krokhin, **O**, '19]

The following are equivalent for all pairs of similar relational structures $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{B}_1, \mathbf{B}_2$:

1. there is a gadget reduction from $\mathrm{PCSP}(\mathbf{B}_1, \mathbf{B}_2)$ to $\mathrm{PCSP}(\mathbf{A}_1, \mathbf{A}_2)$;
2. $(\mathbf{B}_1, \mathbf{B}_2)$ is a homomorphic relaxation a pp-power of $(\mathbf{A}_1, \mathbf{A}_2)$;
3. there is a minion homomorphism from $\mathrm{pol}(\mathbf{A}_1, \mathbf{A}_2)$ to $\mathrm{pol}(\mathbf{B}_1, \mathbf{B}_2)$.

previously on this tutorial...

$$\mathrm{PCSP}(\mathbf{B}_1, \mathbf{B}_2) \xrightarrow{\Sigma_{\mathbf{B}_1}} \mathrm{PCSP}(\mathscr{P}, \mathscr{B}) \xrightarrow{\mathrm{id}} \mathrm{PCSP}(\mathscr{P}, \mathscr{A}) \xrightarrow{\mathbf{I}_{\mathbf{A}_1}} \mathrm{PCSP}(\mathbf{A}_1, \mathbf{A}_2)$$

$\mathscr{A} = \mathrm{pol}(\mathbf{A}_1, \mathbf{A}_2)$, $\mathscr{B} = \mathrm{pol}(\mathbf{B}_1, \mathbf{B}_2)$
Generalised loop conditions $\mathbf{C} \mapsto \Sigma(\mathbf{A}, \mathbf{C})$;
Free structure $\mathscr{M} \mapsto \mathbf{F}_{\mathscr{M}}(\mathbf{A})$;
Indicator structure $\Sigma \mapsto \mathbf{I}_{\mathbf{A}}(\Sigma)$,
Polymorphisms $\mathbf{C} \mapsto \mathrm{pol}(\mathbf{A}, \mathbf{C})$

$$\Sigma(\mathbf{A}, \mathbf{B}) \to \mathscr{M} \quad \text{iff} \quad \mathbf{B} \to \mathbf{F}_{\mathscr{M}}(\mathbf{A})$$

$$\mathbf{I}_{\mathbf{A}}(\Sigma) \to \mathbf{B} \quad \text{iff} \quad \Sigma \to \mathrm{pol}(\mathbf{A}, \mathbf{B})$$

application of part i

Theorem. [Dinur, Regev, Smyth, '05]
For all $k \geq 2$, $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ is NP-hard.

$\mathbf{H}_k$ is the structure with domain $H_k = [k]$ and one ternary relation $\mathrm{nae}_k = [k]^3 \setminus \{(a, a, a) \mid a \in [k]\}$.

Goal. a reduction from $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ to $\mathrm{PCSP}(K_3, K_5)$.

$$\mathrm{PCSP}(\mathbf{H}_2, \mathbf{F}_{\mathscr{K}_{3,5}}(\mathbf{H}_2)) \overset{\Sigma_{\mathbf{H}_2}}{\to} \mathrm{PCSP}(\mathscr{P}, \mathscr{K}_{3,5}) \overset{\mathbf{I}_{K_3}}{\to} \mathrm{PCSP}(K_3, K_5)$$

where $\mathscr{K}_{3,5} = \mathrm{pol}(K_3, K_5)$.

Need. $\mathbf{F}_{\mathscr{K}_{3,5}}(\mathbf{H}_2) \to \mathbf{H}_n$ for some n.

$\mathbf{F}_{\mathrm{pol}(K_3,K_5)}(\mathbf{H}_2)$

- vertices: $F = \mathrm{pol}^{(2)}(K_3, K_5)$,
- hyperedges: $(f_1, f_2, f_3) \in R^{\mathbf{F}}$ if $\exists g \in \mathrm{pol}^{(6)}(K_3, K_5)$ with

$$\begin{aligned} f_1(x,y) &\approx g(x,x,y,y,y,x) \\ f_2(x,y) &\approx g(x,y,x,y,x,y) \\ f_3(x,y) &\approx g(y,x,x,x,y,y). \end{aligned}$$

Claim. $\mathbf{F}_{\mathrm{pol}(K_3,K_5)}(\mathbf{H}_2) \to \mathbf{H}_n$ for some n.

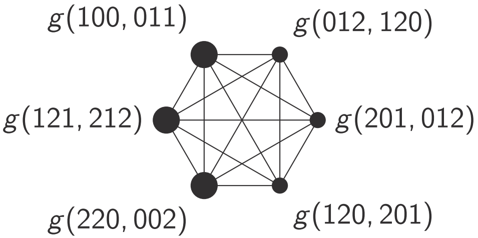
Since $\mathbf{F}$ is finite, it is enough to show that $\mathbf{F}$ does not have a 'hyperloop' (f, f, f). Such a hyperloop would give

$$g(x,x,y,y,y,x) \approx g(x,y,x,y,x,y) \approx g(y,x,x,x,y,y)$$

a.k.a. an Olšák polymorphism.

without Olšák things are hard

Proof. $\mathbf{I}_{\mathbf{K}_3}$(Olšák) contains:



■

Corollary [Bulín, Krokhin, Opršal, '19]
For all $d \geq 3$, $\mathrm{PCSP}(K_d, K_{2d-1})$ is NP-hard.

Corollary
If pol$(\mathbf{A}, \mathbf{B})$ *contains no Olšák function, then* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ *is NP-hard.*

previously on this tutorial...

$$\mathrm{PCSP}(\mathbf{B}_1, \mathbf{B}_2) \xrightarrow{\Sigma_{\mathbf{B}_1}} \mathrm{PCSP}(\mathscr{P}, \mathscr{B}) \xrightarrow{\mathrm{id}} \mathrm{PCSP}(\mathscr{P}, \mathscr{A}) \xrightarrow{\mathbf{I}_{\mathbf{A}_1}} \mathrm{PCSP}(\mathbf{A}_1, \mathbf{A}_2)$$

$\mathscr{A} = \mathrm{pol}(\mathbf{A}_1, \mathbf{A}_2)$, $\mathscr{B} = \mathrm{pol}(\mathbf{B}_1, \mathbf{B}_2)$
Generalised loop conditions $\mathbf{C} \mapsto \Sigma(\mathbf{A}, \mathbf{C})$;
Free structure $\mathscr{M} \mapsto \mathbf{F}_{\mathscr{M}}(\mathbf{A})$;
Indicator structure $\Sigma \mapsto \mathbf{I}_{\mathbf{A}}(\Sigma)$,
Polymorphisms $\mathbf{C} \mapsto \mathrm{pol}(\mathbf{A}, \mathbf{C})$

$$\Sigma(\mathbf{A}, \mathbf{B}) \to \mathscr{M} \quad \text{iff} \quad \mathbf{B} \to \mathbf{F}_{\mathscr{M}}(\mathbf{A})$$

$$\mathbf{I}_{\mathbf{A}}(\Sigma) \to \mathbf{B} \quad \text{iff} \quad \Sigma \to \mathrm{pol}(\mathbf{A}, \mathbf{B})$$

beyond gadget reductions

history of promises

Austrin, Guruswami, Håstad. $(2+\epsilon)$-Sat *is NP-hard*, SICOMP 2017.

Theorem. [Austrin, Guruswami, Håstad, '17]
PCSP($(2k+1)$-Sat, $(k, 2k+1)$-Sat) is NP-hard.

(k, g)-Sat requires that in an instance of g-Sat at least k literals are satisfied in each clause.

$$R_{(a_1,\dots,a_g)} = \{(b_1, \dots, b_g) : \#\{i \mid b_i \neq a_i\} \geq k\}$$

Proof.
Invent polymorphisms and reduce from a version of the PCP theorem [Arora, Safra, "98]. ∎

the PCP theorem

PCP stands for ‘probabilistically checkable proofs’, but the theorem can be formulated as an inapproximability of the CSP:

Theorem. [Arora, Safra, ‘98]
There exists a (Boolean) CSP template $\mathbf{D}$ and $\epsilon < 1$ such that given an instance of $\mathrm{CSP}(\mathbf{D})$, it is NP-hard to distinguish between the following two cases:

- accept if the instance is solvable,
- reject if at most ϵ-fraction of constraints can be satisfied.

$$\mathrm{CSP}(K_3) \xrightarrow{\text{PCP}} \mathrm{PCSP}(\mathscr{P}, \mathscr{M}) \xrightarrow{\mathbf{I_A}} \mathrm{PCSP}(\mathbf{A}, \mathbf{B})$$

Corollary [Raz, '98; et al.]

For all $\epsilon > 0$, there exists N such that: Given a minor condition Σ of arity at most N, it is NP-hard to distinguish the following two cases:

- accept if Σ is trivial
- reject if at most ϵ-fraction of identities in Σ can be simultaneously satisfied by projections.

In CS literature, this problem is referred to as label cover.

- most hardness results in PCSP are obtained by reduction from the PCP theorem via some version of label cover.
- to obtain new hardness results, often a new stronger version of hardness of label cover is needed. [DRS'05, BG'18, BWŽ'20]

$$\mathrm{CSP}(K_3) \xrightarrow{\mathrm{PCP}} \mathrm{PCSP}(\mathscr{P}, \mathscr{M}) \xrightarrow{\mathbf{I_A}} \mathrm{PCSP}(\mathbf{A}, \mathbf{B})$$

Corollary [Austrin, Guruswami, Håstad, '17]

If $\mathrm{pol}(\mathbf{A}, \mathbf{B})$ has bounded essential arity then $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is NP-hard.

(A minion $\mathscr{M}$ has bounded essential arity k, if every $f \in \mathscr{M}$ is a minor of a function of arity k.)

Unlike for CSPs,

- no finite set of identities can imply tractability of a PCSP!
- there are many PCSPs whose hardness cannot be explained by the algebraic approach!

This calls for reductions that are better than gadgets reductions!

beyond gadget reductions

[Wrochna, Živný, '20]

- use the arc-graph pp-power as a reduction — this is the other way than you would expect!
- they obtain hardness of $\mathrm{PCSP}(K_k, K_{\binom{k}{\lfloor k/2 \rfloor}-1})$ for all $k \geq 4$.

[Barto, Kozik, '20+] (csp-seminar.org/talks/libor-barto/).

- describe a sufficient condition for reducing one PCSP to another — this condition is given by weakening minion homomorphisms to 'ϵ-homomorphisms' (list homomorphisms).
- this show hardness of PCSP with polymorphisms of bounded essential arity without the PCP theorem!

problems

search vs. decision

Search. Given a finite structure $\mathbf{I}$ that maps homomorphically to $\mathbf{A}$, find a homomorphism $h\colon \mathbf{I} \to \mathbf{B}$.

Decide. Given $\mathbf{I}$ arbitrary structure with the same language,

- accept if $\mathbf{I} \to \mathbf{A}$,
- reject if $\mathbf{I} \not\to \mathbf{B}$.

Problem 1

Does search always belong to the same complexity class as decision?

complexity of concrete templates

Problem 2

Fix $\mathbf{A}$*, classify how the complexity of* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ *depends on* $\mathbf{B}$*.*

- can be sold as approximation variant of $\mathrm{CSP}(\mathbf{A})$,
- very few classification to-date: $\mathbf{A} =$ NAE-Sat [DRS05], some progress on $\mathbf{A} =$ 1-in-3-Sat [Barto, Battistelli, Berg, '21].
- can provide nice conditions for hardness (e.g., [DRS05] shows implies that absence of Olšák implies hardness).
- contains important special cases: $\mathbf{A} = K_3$ is the approximate graph colouring.

Conjecture 3 (Brakensiek-Guruswami)

For all non-bipartite loopless graphs G *and* H*,* $\mathrm{PCSP}(G, H)$ *is NP-hard.*

power of algorithms

Problem 4

Characterise applicability of some algorithm in solving PCSPs.

local consistency algorithm

Fix $k \in \mathbb{N}$. Given an instance $\mathbf{I}$ of CSP($\mathbf{A}$):

1. for all subsets $K \subseteq I$ of size at most k:
   let $\mathcal{F}_K$ be the set of all partial homomorphisms $\mathbf{I} \to \mathbf{A}$ defined on K.
2. for all $K \subseteq L$:
   - remove from $\mathcal{F}_L$ all f's s.t. $f|_K \notin \mathcal{F}_K$,
   - remove from $\mathcal{F}_K$ all f's that do not extend to a member of $\mathcal{F}_L$,
3. if $\mathcal{F}_K = \emptyset$ for some K, return **False**,
4. repeat (2) & (3) as long as something changes, else return **True**.

For PCSP($\mathbf{A}$, $\mathbf{B}$), run consistency on $\mathbf{I}$ as an instance of CSP($\mathbf{A}$). We require that any consistent instance $\mathbf{I}$ has a homomorphism to $\mathbf{B}$.

Problem 5

Characterise all finite template PCSPs solvable by local consistency.

affine integer programming

The basic affine integer program for an instance $\mathbf{I}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is the following system of equations over $\mathbb{Z}$:

- variables are $v_{i,a}$ for all $i \in I$, $a \in A$, and $v_{\mathbf{i},\mathbf{a}}$ for all R, $\mathbf{i} \in R^{\mathbf{I}}$, $\mathbf{a} \in R^{\mathbf{A}}$,
- subject to

$$\begin{aligned} \textstyle\sum_{a \in A} v_{i,a} &= 1 && \text{for each } i \in I, \\ \textstyle\sum_{\mathbf{a} \in R^{\mathbf{A}}, \mathbf{a}_j = a} v_{\mathbf{i},\mathbf{a}} &= v_{\mathbf{i}_j, a} && \text{for each } R \text{ and } \mathbf{i} \in R^{\mathbf{I}}. \end{aligned}$$

This gives an algorithm for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$: solve the BAIP of $\mathbf{I}$ over $\mathbf{A}$, and return **True** if it has a solution, else return **False**.

- The same as asking if $\Sigma(\mathbf{A}, \mathbf{I})$ is satisfied by affine functions over $\mathbb{Z}$.
- The applicability of BAIP are characterised via alternating functions.

linear programming

The basic linear program for an instance $\mathbf{I}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is the same as BAIP with the exception that the variables are taking values in $\mathbb{Q} \cap [0, 1]$.

- The same as asking if $\Sigma(\mathbf{A}, \mathbf{I})$ is satisfied by convex combinations over $\mathbb{Q}$.
- Such linear programs are solvable in polynomial time, and therefore give a polynomial time algorithm for PCSPs in a similar way as BAIP.
- The applicability of BLP is characterised by symmetric functions.

Problem 6

Is there a (finite template) $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ which is solvable by some level of Sherali-Adams but it is not solvable by local consistency?

Brakensiek-Guruswami algorithm

Assume $\mathbf{I}$ is an instance of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$.

1. solve the BLP program for $\mathbf{I}$, if no solution return **False**, else pick a solution* v,
2. start with the BAIP for $\mathbf{I}$ with variables w_{-}, and add the equation $w_{-} = 0$ whenever $v_{-} = 0$.
3. solve the resulting AIP, if no solution return **False** else return **True**.

Theorem [Brakensiek, Guruswami, Wrochna, Živný, '20]

The above algorithm solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ iff $\mathrm{pol}(\mathbf{A}, \mathbf{B})$ contains for all k a function f satisfying

$$f(x_1, \dots, x_k, y_1, \dots, y_{k+1}) \approx f(x_{\pi(1)}, \dots, x_{\pi(k)}, y_{\sigma(1)}, \dots, y_{\sigma(k+1)})$$

for all permutations π, σ.

an algorithm

- Every tractable PCSP that I am aware of is either a homomorphic relaxation of a finite template CSP with a Siggers polymorphism, or solvable by Brakensiek-Guruswami algorithm!
- Unfortunately, BG algorithm does not solve all CSPs with Siggers (e.g., $C_2 + C_3$). We need a refinement.

Conjecture 7

When we replace LP with Sherali-Adams in the first step of BG algorithm, the resulting algorithm solves all finite template CSPs with Siggers polymorphism.

Prize for a negative answer. A bottle of fine single malt Scotch whisky.