Robust Algorithms for Mixed Constraint Satisfaction Problems

Andrei Krokhin
Durham University

Joint work with Víctor Dalmau (UPF Barcelona)

(Disclaimer: These slides contain inaccuracies)

Constraint Satisfaction Problems (CSPs)

Fix a finite set D

Constraint language Γ = finite set of relations on D

Relational structure $\mathcal{B} = (D;\ R_1^{\mathcal{B}}, \ldots, R_k^{\mathcal{B}})$

- CSP(Γ): given constraints $(\mathbf{x}_1, R_1), \ldots, (\mathbf{x}_q, R_q)$ over a set of variables V, and such that each $R_i \in \Gamma$, decide if there is $\varphi : V \to D$ satisfying all constraints
- CSP($\mathcal{B}$): given a (rel) structure $\mathcal{A} = (V;\ R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}})$ is there a homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$?
  $[(a_1, \ldots, a_n) \in R_i^{\mathcal{A}} \Rightarrow (\varphi(a_1), \ldots, \varphi(a_n)) \in R_i^{\mathcal{B}}]$
- Examples: 3Lin-p, k-Sat, k-Col, H-col

The CSP Dichotomy Conjecture

Conjecture 1 (Feder,Vardi'98)
Dichotomy Conjecture: for each CL Γ, the problem $\mathrm{CSP}(\Gamma)$ *is either tractable (i.e., in* **P***) or* **NP***-complete.*

Interesting in itself, also because it generates cool maths.

Proved in many special cases [Schaefer'78, Bulatov'02, etc]

Still open in full generality.

Main tool for attacking it: universal algebra.

Algebraic Approach to CSP, Briefly

- An n-ary operation f on D is a polymorphism of an m-ary relation R on D if R is a subalgebra of $(D, f)^m$.
- $\mathrm{Pol}(\Gamma) = \{f \mid f$ is a polymorphism of each $R \in \Gamma\}$.
- The complexity of $\mathrm{CSP}(\Gamma)$ depends only on the algebra $\mathbb{A}_\Gamma = (D, \mathrm{Pol}(\Gamma))$.
- For any Γ, there is Γ' such that $\mathrm{CSP}(\Gamma) \equiv \mathrm{CSP}(\Gamma')$ and $\mathbb{A}_{\Gamma'}$ is idempotent (i.e. $f(x, \ldots, x) = x$ for all f).
- From now on, all Γ's are such that $\mathbb{A}_\Gamma$ are idempotent.
- The complexity of $\mathrm{CSP}(\Gamma)$ depends only on $\mathrm{var}(\mathbb{A}_\Gamma)$.

More Algebra and CSP

An idempotent operation f is weak near-unanimity (WNU) operation if all op's $f(x,\ldots,x,y,x,\ldots,x)$ are the same.

Conjecture 2 (Bulatov,Jeavons,AK'05)
CSP(Γ) *is tractable if* $\mathbb{A}_\Gamma$ *has a WNU operation, and* **NP**-*complete otherwise.*

Theorem 1 (Barto,Kozik'09; Bulatov'09) *TFAE:*

1. CSP(Γ) *is tractable via the bounded width algorithm,*
2. Γ *cannot express* 3LIN-p *for any prime* p*,*
3. $\mathbb{A}_\Gamma$ *has WNUs of almost all arities [MM'08].*

A Stronger Notion of Tractability

- Call an algorithm for CSP(Γ) robust [Zwick'98] if
  - for any $(1-\epsilon)$-satisfiable instance, it outputs a $(1-g(\epsilon))$-satisfying assignment where loss function g satisfies the conditions (i) $g(0)=0$ and (ii) $g(\epsilon)\to 0$ as $\epsilon\to 0$;
  - running time is polynomial in the size of instance, does not depend on ϵ (which is unknown).
- Robust algorithms:
  - (almost) untouched for 12 years
  - meeting point of universal algebra, combinatorics and approximation algorithms

Which CSPs Have No Robust Algorithms?

3-LINp: systems of linear 3-variable equations mod p

- Easy as a CSP, use Gaussian elimination
- Can always satisfy $1/p$-fraction of equations: assign values uniformly at random (without even looking)
- For any $\epsilon, \gamma > 0$, it is **NP**-hard to approximate within $1/p + \gamma$, even on $(1 - \epsilon)$-sat instances [Håstad'01]
- Hence no robust algorithm for 3-LINp

Which CSPs Admit Robust Algorithms?

No bounded width $\Rightarrow$ can express 3LIN-p $\Rightarrow$ no robust algo

Conjecture 3 (Guruswami,Zhou'11)
CSP(Γ) *admits a robust algorithm iff it has bounded width.*

Theorem 2 (Barto,Kozik'12)
If CSP(Γ) *has bounded width (or* $\mathbb{A}_\Gamma$ *has many WNUs) then it admits an SDP-based robust algorithm with loss* $g(\epsilon) = O(\frac{\log\log(1/\epsilon)}{\log(1/\epsilon)})$.

Proof is a combination of deep universal algebra and SDP (semi-definite programming) techniques from combinatorial optimisation.

More Questions about Robust Algorithms

- Which CSPs have robust algorithms with better loss $g(\epsilon)$, modulo complexity-theoretic assumptions?
  - For example, linear loss $g(\epsilon) = O(\epsilon)$.
- What about Mixed CSPs?
  - In each instance, some constraints are hard and can't be violated/removed, while the rest are soft. Hard constraints complicate the situation.
  - Note: The fractions $(1 - \epsilon)$ and $(1 - g(\epsilon))$ in def'n of robust algorithm are then computed of the number of soft constraints.

Example of Mixed CSPs

- A k-uniform hypergraph is a pair $H = (V, E)$ where E is a set of k-element subsets of V.
- A subset of V is an independent set for H if it does not entirely contain any set from E.

  One usually tries to maximise the size of IndSet.
- Consider a Boolean CSP instance which has
  - a variable x_v and soft constraint (x_v) for all $v \in V$
  - a hard constraint $(\overline{x}_{v_1} \vee \ldots \vee \overline{x}_{v_k})$ for each $\{v_1, \ldots, v_k\} \in E$
- Interpret variables sent to 1 as being in IndSet.

Linear Loss and Mixed CSPs

Lemma 1 (Dalmau,AK) *For any Γ, TFAE*

1. CSP(Γ) *has a robust algorithm with linear loss.*
2. MixedCSP(Γ) *has a robust algorithm with linear loss.*

The proof is easy, but works only for linear loss.

Lemma 2 *The existence of a robust algorithm for* MixedCSP(Γ) *is determined by the identities of $\mathbb{A}_\Gamma$. The same holds for linear loss.*

The proof is standard.

A Hardness Result

Theorem 3 (Dinur,Guruswami,Khot,Regev'05)
For any $\delta, \delta' > 0$ and any $k \geq 3$, it is **NP**-*hard to tell whether, for a given k-uniform hypergraph $H = (V, E)$,*

- *some IndSet in H has size at least $(1 - \frac{1}{k-1} - \delta)|V|$, or*
- *every IndSet in H has size at most $\delta'|V|$.*

HORN 3-SAT is CSP(Γ) with $\Gamma = \{\{x \wedge y \rightarrow z\}, \{0\}, \{1\}\}$.
Note: $\mathbb{A}_\Gamma$ is term equivalent to $(\{0, 1\}, \wedge)$, so many WNUs.

Lemma 3 (Bennabas'11) *For any $\epsilon > 0$, it is* **NP**-*hard to tell whether a given instance of* MIXED HORN 3-SAT *has value at least $1 - \epsilon$ or at most ϵ.*

A Generalisation: No NU, No Fun

A near-unanimity (NU) operation f of arity $k \geq 3$ satisfies $f(y, x, \ldots, x) = f(x, y, \ldots, x) = \ldots = f(x, x, \ldots, y) = x$.

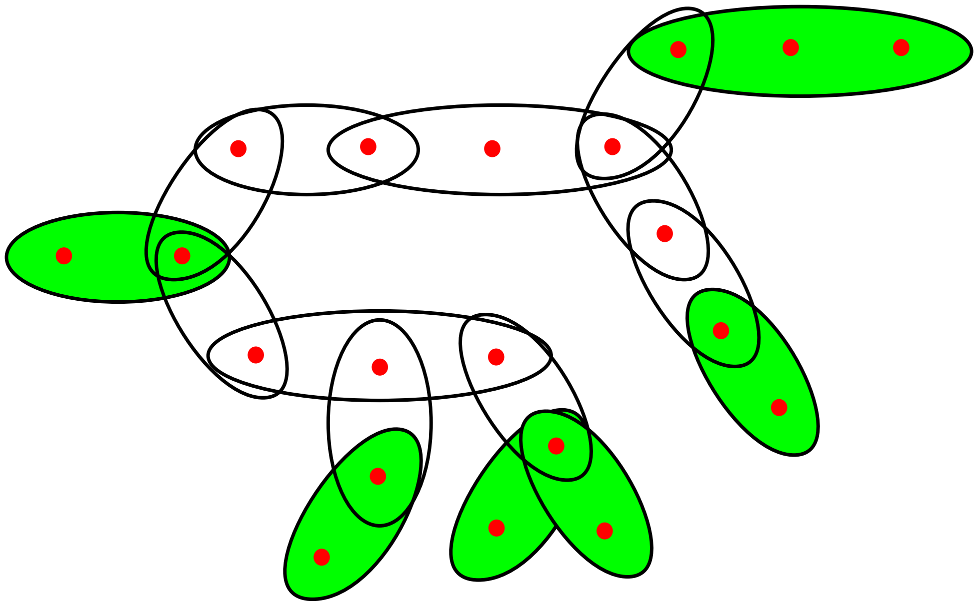
Theorem 4 (Dalmau,AK'13)
If $\mathbb{A}_\Gamma$ *has no NU operation then* MixedCSP(Γ) *has no robust algorithm (unless* $\mathbf{P} = \mathbf{NP}$*).*

Proof idea. Show: if $\mathbb{A}$ has WNUs of almost all arities, but no NU then there exist $a, b \in D$ and, for each $n \geq 3$, $R_n \leq \mathbb{A}^n$ such that $R_n \cap \{a, b\}^n = \{a, b\}^n \setminus \{(a, \ldots, a)\}$.
Then use the result about independent sets in hypergraphs.

Tree Hypergraphs and Pendant Edges

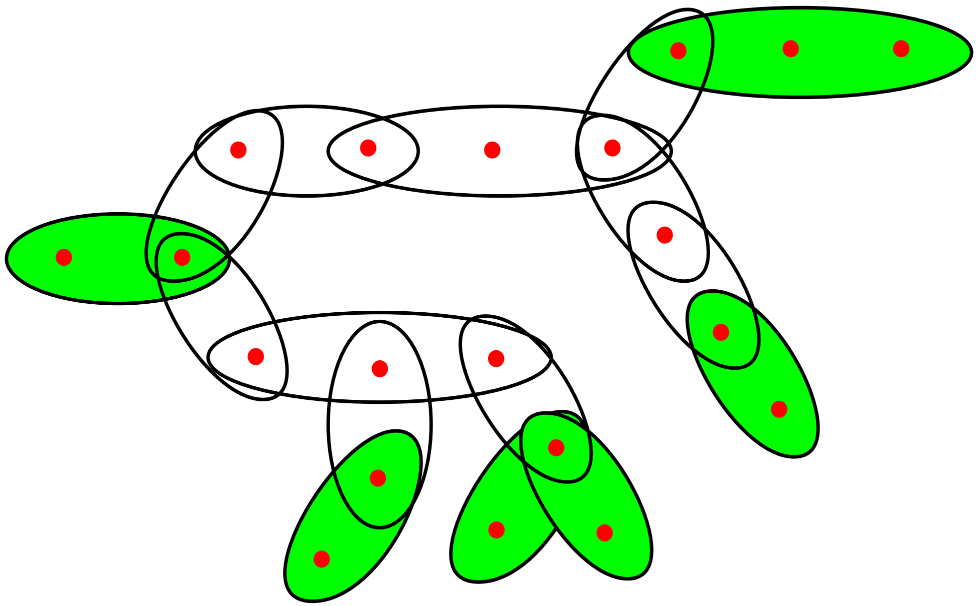
A structure is a tree if its relations are disjoint and its hypergraph is connected and (Berge-)acyclic.

A hyperedge in a tree hypergraph is pendant if at most one of its nodes belongs to another hyperedge.



Degree of Monstrosity

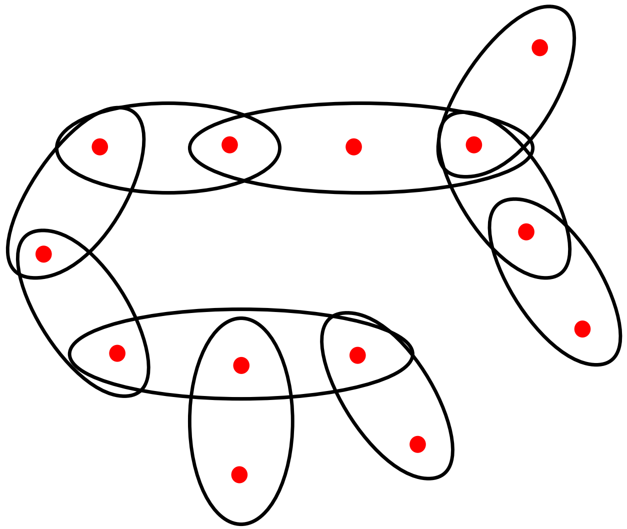
A hyperedge in a tree hypergraph is pendant if at most one of its nodes belongs to another hyperedge.



Degree of monstrosity of a tree hypergraph: number of pendant hyperedges after removing pendant hyperedges.

Degree of Monstrosity

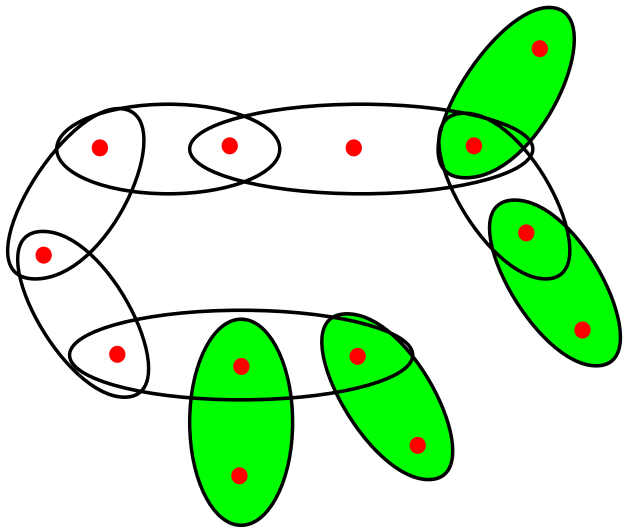
A hyperedge in a tree hypergraph is pendant if at most one of its nodes belongs to another hyperedge.



Degree of monstrosity of a tree hypergraph: number of pendant hyperedges after removing pendant hyperedges.

Degree of Monstrosity

A hyperedge in a tree hypergraph is pendant if at most one of its nodes belongs to another hyperedge.



Degree of monstrosity of a tree hypergraph: number of pendant hyperedges after removing pendant hyperedges.

Linear Loss

Switch to the homomorphism formulation: CSP($\mathcal{B}$).

Obvious fact: if $\mathcal{A}' \to \mathcal{A}$ and $\mathcal{A}' \not\to \mathcal{B}$ then $\mathcal{A} \not\to \mathcal{B}$.

An obstruction set for a structure $\mathcal{B}$ is a class $O_{\mathcal{B}}$ of structures such that, for all structures $\mathcal{A}$,

$$\mathcal{A} \to \mathcal{B} \text{ if and only if } \mathcal{A}' \not\to \mathcal{A} \text{ for all } \mathcal{A}' \in O_{\mathcal{B}}.$$

Theorem 5 (Dalmau, AK'12)
If $\mathcal{B}$ has an obstruction set consisting of trees of bounded degree of monstrosity then MixedCSP($\mathcal{B}$) *admits an LP-based robust algorithm with linear loss.*

Basic LP Relaxation for CSP(Γ)

The basic LP relaxation for instance I with constraints $\mathcal{C}$.
The variables are

- $p_v(a) \in [0,1]$ for each $v \in V, a \in D$;
- $p_C(\mathbf{t}) \in [0,1]$ for each constraint C in I and $\mathbf{t} \in D^{ar(C)}$.

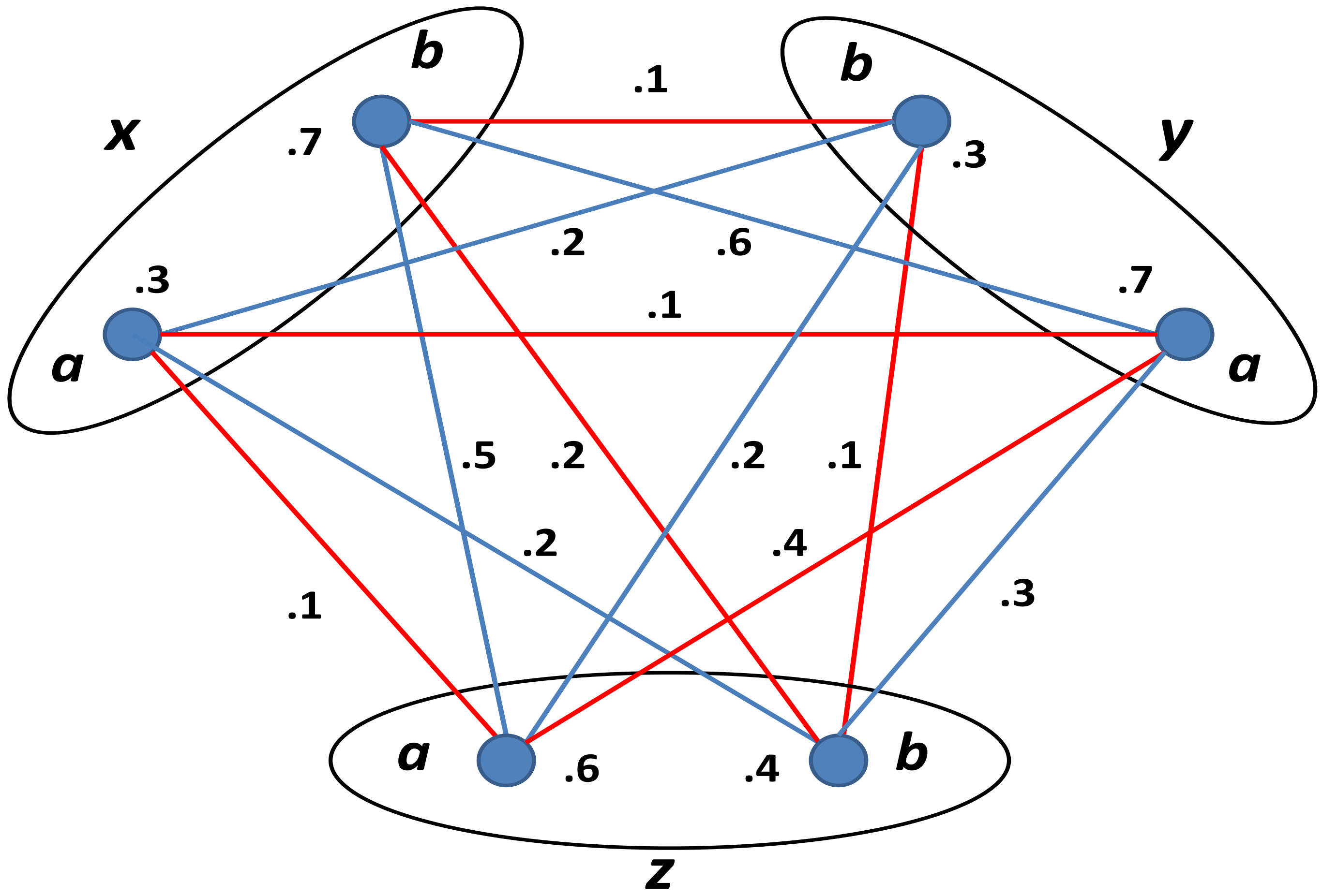
$$\text{maximize } \frac{1}{|\mathcal{C}|} \sum_{C=(\mathbf{x},R)\in\mathcal{C}} \sum_{\mathbf{t}\in R} p_C(\mathbf{t}) \qquad \text{subject to:}$$

- $\sum_{a\in D} p_v(a) = 1$ for each $v \in V$
- $\sum_{\mathbf{t}\in D^{ar(C)},\mathbf{t}[i]=a} p_C(\mathbf{t}) = p_{\mathbf{x}[i]}(a)$ for all $C = (\mathbf{x}, R), i, a$

Since Γ is fixed, this relaxation has polynomial size (in I).

Basic Linear Program: Example

$V = \{x, y, z\}$, $D = \{a, b\}$, $\mathcal{C} = \{x \neq y, y \neq z, x \neq z\}$.



How General Robust Algorithms Work

Given an $(1-\epsilon)$-satisfiable instance I,

- Solve LP (SDP) relaxation of the instance I.
- Use LP (SDP) solution as guidance to remove $g(\epsilon)$ constraints so that the remaining instance I' is "consistent enough" (i.e. receives no homomorphisms from obstructions).
- Instance I' has a satisfying assignment.
- This assignment is $(1-g(\epsilon))$-satisfying for I.

Algebra for Linear Loss

A totally symmetric (TS) operation is an operation f such that $f(a_1, \ldots, a_n)$ depends only on $\{a_1, \ldots, a_n\}$.

Theorem 6 (Feder,Vardi'98) *TFAE:*

1. *$\mathcal{B}$ has an obstruction set consisting of trees,*
2. *$\mathcal{B}$ has TS polymorphisms of all arities.*

Problem 1 (Dalmau,AK'13) *TFAE?*

1. CSP($\mathcal{B}$) *admits a robust algorithm with linear loss,*
2. *$\mathcal{B}$ has an obstruction set consisting of trees of b.d.m.,*
3. *$\mathcal{B}$ has TS polymorphisms of all arities and an NU polymorphism of some arity.*

More Robust Algorithms for Mixed CSP

Theorem 7 (Khot'02) *If* Γ *consists of graphs of permutations then* MixedCSP(Γ) *has a robust algorithm.*

Khot's proof works if $\mathbb{A}_\Gamma$ has a discriminator operation.

$$t(x,y,z) = \begin{cases} z, & \text{if } x = y; \\ x, & \text{if } x \neq y. \end{cases}$$

The argument can be generalised to the case when $\mathbb{A}_\Gamma$ has

- both a majority (ternary NU) operation
- and a conservative Mal'tsev operation, i.e. $p(x,y,z) \in \{x,y,z\}$ and $p(x,y,y) = p(y,y,x) = x$.

Boolean Mixed CSPs

- What do the above results say for the Boolean case?
- If a two-element algebra $\mathbb{A}_\Gamma$ has an NU operation then one of the following conditions holds:
  1. Γ satisfies the bounded monstrosity condition,
     – then MixedCSP(Γ) has a robust algorithm
  2. $\mathbb{A}_\Gamma$ has discriminator operation,
     – then MixedCSP(Γ) has a robust algorithm
  3. $\mathbb{A}_\Gamma$ is term equivalent to $(\{0,1\}, maj)$.
     – CSP(Γ) is essentially 2-SAT.

Mixed 2-SAT

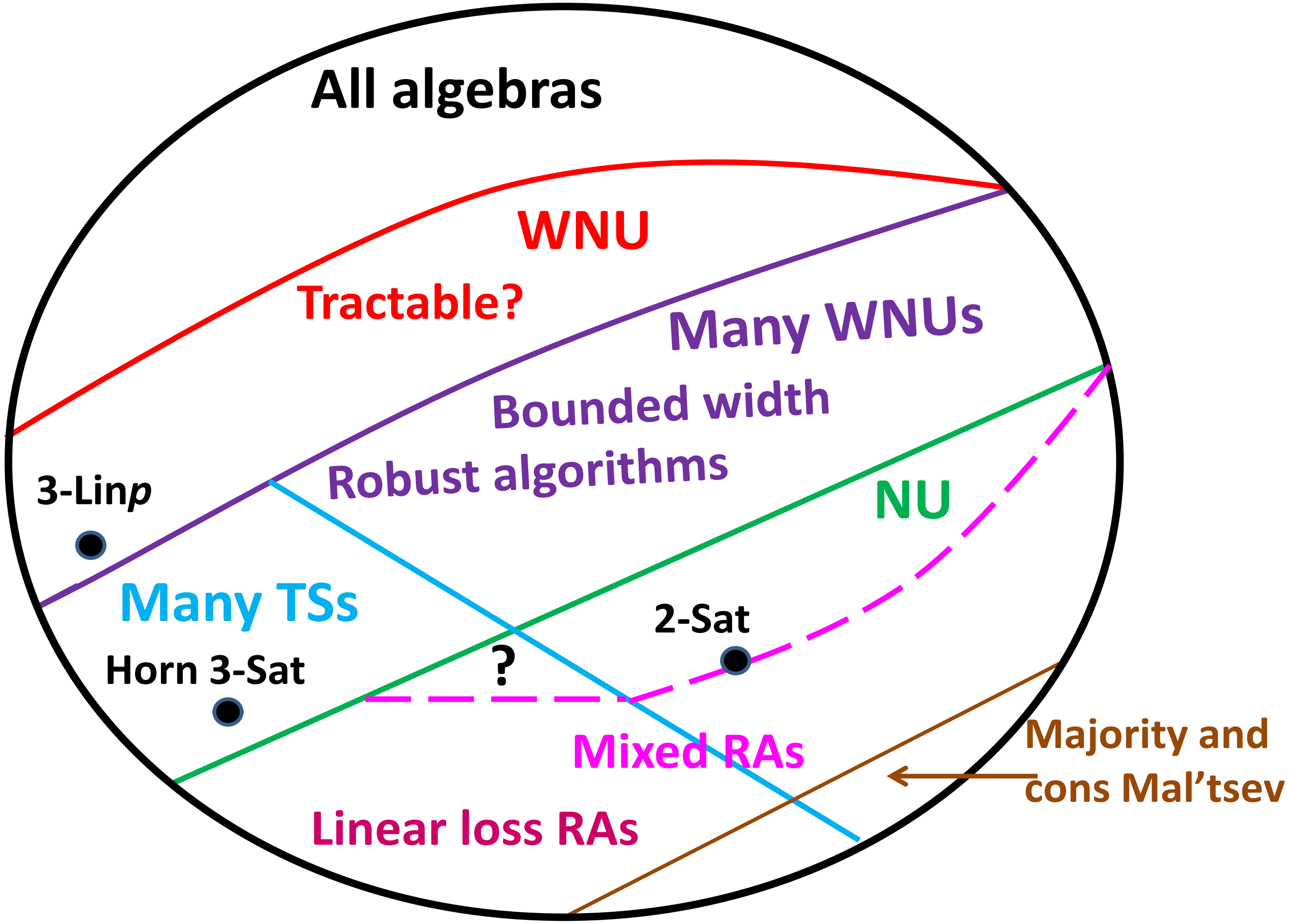
Lemma 4 (Guruswami,Lee'13) *For any $\epsilon > 0$, there is no poly-time algorithm that tells* MIXED 2-SAT *instances with value at least $(1-\epsilon)$ from those with value at most ϵ, unless the Unique Games Conjecture fails.*

The essence of the UGC is that, for many problems, SDP gives the best approximation algorithms.

Lemma 5 *If $\mathbb{A}$ is an NU algebra such that $\mathrm{var}(\mathbb{A})$ does not contain an algebra term equivalent to $(\{0,1\}, maj)$ then $\mathbb{A}$ has ternary terms t_1, t_2 satisfying*

$$t_1(x,x,y) = t_1(x,y,x) = t_2(y,x,y) = t_2(y,y,x).$$

Summary


All algebras
WNU
Tractable?
Many WNUs
Bounded width
Robust algorithms
NU
3-Lin*p*
Many TSs
2-Sat
Horn 3-Sat
?
Mixed RAs
Majority and
cons Mal'tsev
Linear loss RAs