Robust approximation of CSPs
Universal algebra meets optimization

Yuichi Yoshida

National Institute of Informatics, and
Preferred Infrastructure, Inc

March 19, 2013

Constraint Satisfaction Problems (CSPs)

Given a set of variables V and a set of constraints $\mathcal{C}$, find an assignment that satisfies all the constraints.

Example (SAT)

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee w) \wedge (y \vee \overline{z})$$

$(x, y, z, w) = (1, 1, 0, 1)$ satisfies the SAT instance.

Constraint Satisfaction Problems (CSPs)

Definition (CSP)

A CSP (denoted $\mathsf{CSP}_q(\Gamma)$), specified by

- finite domain $[q] = \{1, \ldots, q\}$.
- constraint language Γ: a collection of relations over $[q]$.
    - relation: a set of $\mathrm{ar}(R)$-tuples ($\mathrm{ar}(R)$ = arity of R)

E.g.: $R = \{(x, y, z) \in \{0, 1\}^3 \mid (x \wedge y \wedge \overline{z}) = \mathbf{true}\}$.

Constraint Satisfaction Problems (CSPs)

Definition (CSP)

A CSP (denoted $\mathsf{CSP}_q(\Gamma)$), specified by

- finite domain $[q] = \{1, \ldots, q\}$.
- constraint language Γ: a collection of relations over $[q]$.
  - relation: a set of $\mathrm{ar}(R)$-tuples ($\mathrm{ar}(R) =$ arity of R)

E.g.: $R = \{(x, y, z) \in \{0,1\}^3 \mid (x \wedge y \wedge \overline{z}) = \mathbf{true}\}$.

Definition (CSP instance)

A CSP instance (denoted $\mathcal{I} = (V, \mathcal{C})$), specified by

- a variable set V
- a set $\mathcal{C}$ of constraints (R, S), where $R \in \Gamma$; S is set of $\mathrm{ar}(R)$ variables.

Constraint Satisfaction Problems (CSPs)

Definition (CSP)

A CSP (denoted $\mathsf{CSP}_q(\Gamma)$), specified by

- finite domain $[q] = \{1, \ldots, q\}$.
- constraint language Γ: a collection of relations over $[q]$.
  - relation: a set of $\mathrm{ar}(R)$-tuples ($\mathrm{ar}(R)$ = arity of R)

E.g.: $R = \{(x, y, z) \in \{0,1\}^3 \mid (x \wedge y \wedge \overline{z}) = \mathbf{true}\}$.

Definition (CSP instance)

A CSP instance (denoted $\mathcal{I} = (V, \mathcal{C})$), specified by

- a variable set V
- a set $\mathcal{C}$ of constraints (R, S), where $R \in \Gamma$; S is set of $\mathrm{ar}(R)$ variables.

Question: Is there an assignment $\sigma : V \to [q]$ that satisfies all constraints? i.e., $\sigma|_S \in R$ for every $(R, S) \in \mathcal{C}$.

Examples

CSPs can express a lot of problems depending on the choice of Γ!

- k-SAT: $\Gamma =$ all disjunctions of up to k literals.
  E.g. $(u \vee v \vee w), (\overline{u} \vee v)$
- Horn k-SAT: as above, but contain at most one positive literal.
  E.g. $(\overline{u} \vee \overline{v} \vee w)(\Leftrightarrow (u \wedge v \Rightarrow w))$
- k-LIN$_q$: $\Gamma =$ all affine relations (over $\mathbb{Z}_q$) on up to k vars.
  E.g. $(u + v + w = 1) \pmod 2$
- q-Coloring: $\Gamma =$ an inequality relation on two vars over $[q]$.
- k-CSP$_q$: $\Gamma =$ all relations of arity k over $[q]$.

Examples

CSPs can express a lot of problems depending on the choice of Γ!

- k-SAT: $\Gamma =$ all disjunctions of up to k literals.
  E.g. $(u \vee v \vee w), (\overline{u} \vee v)$
- Horn k-SAT: as above, but contain at most one positive literal.
  E.g. $(\overline{u} \vee \overline{v} \vee w)(\Leftrightarrow (u \wedge v \Rightarrow w))$
- k-LIN$_q$: $\Gamma =$ all affine relations (over $\mathbb{Z}_q$) on up to k vars.
  E.g. $(u + v + w = 1) \pmod 2$
- q-Coloring: $\Gamma =$ an inequality relation on two vars over $[q]$.
- k-CSP$_q$: $\Gamma =$ all relations of arity k over $[q]$.

(Probably) only two possibilities for its complexity...

Schaefer's Dichotomy Theorem for Boolean CSPs

Theorem ([Sch78])

Every Boolean CSP is either in P or NP-complete. Specifically, $\mathsf{CSP}_2(\Gamma)$ is polynomial-time solvable if every $R \in \Gamma$ is

- 0-valid / 1-valid
- a conjunction of Horn clauses / conjunction of dual Horn clauses
- a 2CNF formula, or
- a conjunction of affine equations

and is NP-complete otherwise.

Schaefer's Dichotomy Theorem for Boolean CSPs

Theorem ([Sch78])

Every Boolean CSP is either in P or NP-complete. Specifically, $\mathsf{CSP}_2(\Gamma)$ is polynomial-time solvable if every $R \in \Gamma$ is

- 0-valid / 1-valid
- a conjunction of Horn clauses / conjunction of dual Horn clauses
- a 2CNF formula, or
- a conjunction of affine equations

and is NP-complete otherwise.

Dichotomy conjectured for every q [FV98]
Proved for $q = 3$ [Bul02].

Max CSPs

Definition (Max-CSP)

$\mathsf{MaxCSP}_q(\Gamma)$: Given an instance of $\mathsf{CSP}_q(\Gamma)$, find an assignment maximizing the fraction of satisfied constraints.

Max CSPs

Definition (Max-CSP)

$\mathsf{MaxCSP}_q(\Gamma)$: Given an instance of $\mathsf{CSP}_q(\Gamma)$, find an assignment maximizing the fraction of satisfied constraints.

Most Max CSPs are NP-Hard...

Definition (Approximation)

Let $c, s \in [0, 1]$. A poly-time algorithm is a (c, s)-approximation algorithm for $\mathsf{MaxCSP}_q(\Gamma)$ if it

finds an assignment β with $\mathbf{val}(\mathcal{I}, \beta) \geq s$

assuming $\mathbf{opt}(\mathcal{I}) \geq c$,

where $\mathbf{opt}(\mathcal{I})$ is the optimal value of $\mathcal{I}$ and $\mathbf{val}(\mathcal{I}, \beta)$ is the fraction of constraints satisfied by β.

Question: For which c and s, can we (c, s)-approximate $\mathsf{MaxCSP}_q(\Gamma)$?

Raghavendra's Theorem

Definition (Unique Games Conjecture (UGC), informal)

For every $\epsilon > 0$, there exists some q and (simple) Γ such that it is NP-Hard to $(1-\epsilon, \epsilon)$-approximate $\mathsf{MaxCSP}(\Gamma)$.

Raghavendra's Theorem

Definition (Unique Games Conjecture (UGC), informal)

For every $\epsilon > 0$, there exists some q and (simple) Γ such that it is NP-Hard to $(1-\epsilon, \epsilon)$-approximate $\mathsf{MaxCSP}(\Gamma)$.

Theorem ([Rag08])

Assuming the UGC, some canonical SDP relaxation gives optimal approximation guarantee for every Max CSP!

Analytical Tools for CSP and MaxCSP

- CSP and MaxCSP were studied in almost different communities.
- Analytical tools and words are different.
  E.g.: essentially unary operations $\leftrightarrow$ dictators
- However, it seems there are some connections...

CSP	MaxCSP
Universal algebra	Harmonic analysis
Polymorphism	Rounding function
Essentially unary	Dictator
Weak near-unanimity	Pseudorandom
Width	LP / SDP

Robust Approximation

General question: what can we do by using universal algebra and harmonic analysis interchangeably.

Robust Approximation

General question: what can we do by using universal algebra and harmonic analysis interchangeably.

Idea: Universal algebra is a tool to study the case $c = 1$. Probably, it is also useful when $c = 1 - \epsilon$.

Robust Approximation

General question: what can we do by using universal algebra and harmonic analysis interchangeably.

Idea: Universal algebra is a tool to study the case $c = 1$. Probably, it is also useful when $c = 1 - \epsilon$.

Definition (Robust Approximation [Zwi98])

$\mathsf{CSP}_q(\Gamma)$ admits $f(\epsilon)$-robust approximation if there is a $(1-\epsilon, 1-f(\epsilon))$-approximation algorithm for every $\epsilon \geq 0$, where $f(0) = 0$ and $\lim_{\epsilon \to 0} f(\epsilon) = 0$.

Question: For which Γ, how much can we robustly approximate?

Robust Approximation

General question: what can we do by using universal algebra and harmonic analysis interchangeably.
Idea: Universal algebra is a tool to study the case $c = 1$. Probably, it is also useful when $c = 1 - \epsilon$.

Definition (Robust Approximation [Zwi98])

$\mathsf{CSP}_q(\Gamma)$ admits $f(\epsilon)$-robust approximation if there is a $(1-\epsilon, 1-f(\epsilon))$-approximation algorithm for every $\epsilon \geq 0$, where $f(0) = 0$ and $\lim_{\epsilon \to 0} f(\epsilon) = 0$.

Question: For which Γ, how much can we robustly approximate?
Motivation with hindsight: Real-world instances are corrupted by noise ☺

Robust Approximation for Boolean CSPs

Horn k-SAT

- $O(\frac{1}{\log 1/\epsilon})$-robust approximation via LP [Zwi98].
- $o(\frac{1}{\log 1/\epsilon})$-robust approximation is UG-Hard [GZ11].

2-SAT

- $O(\sqrt{\epsilon})$-robust approximation via SDP [CMM09].
- $o(\sqrt{\epsilon})$-robust approximation is UG-Hard [KKMO07, MOO10].

3-LIN$_2$

$(1-\epsilon, \frac{1}{2}+\epsilon)$-approximation is NP-Hard for any $\epsilon > 0$ [Hås01].

Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

$\mathsf{CSP}(\Gamma)$ has width k if it can be solved by iteratively checking consistency of k vars.

Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

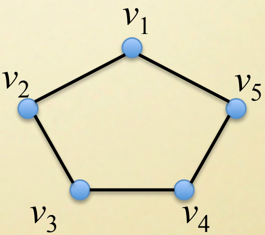
$(v_1, v_2) : \{{\color{blue}\bullet\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet\bullet}\}$

$(v_2, v_3) : \{{\color{blue}\bullet\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet\bullet}\}$

$(v_1, v_3) : \{{\color{blue}\bullet\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet\bullet}\}$

$(v_1, v_4) : \{{\color{blue}\bullet\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet\bullet}\}$

$(v_1, v_5) : \{{\color{blue}\bullet\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet\bullet}\}$



Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

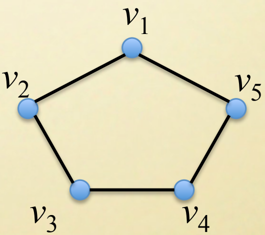
$(v_1, v_2) : \{ {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet} \}$

$(v_2, v_3) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$

$(v_1, v_3) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$

$(v_1, v_4) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$

$(v_1, v_5) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$



Characterizing Robustly Approximable CSPs

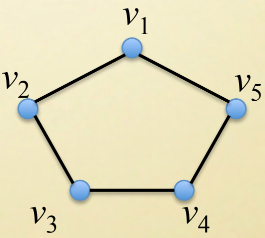
If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

$$(v_1, v_2) : \{ {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet} \}$$
$$(v_2, v_3) : \{ {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet} \}$$
$$(v_1, v_3) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$$
$$(v_1, v_4) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$$
$$(v_1, v_5) : \{ {\color{blue}\bullet}{\color{blue}\bullet}, {\color{blue}\bullet}{\color{red}\bullet}, {\color{red}\bullet}{\color{blue}\bullet}, {\color{red}\bullet}{\color{red}\bullet} \}$$



Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

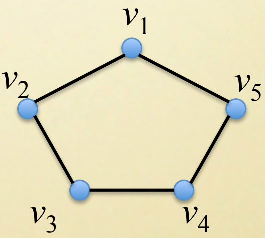
(v_1, v_2) : { ●●, ●● }

(v_2, v_3) : { ●●, ●● }

(v_1, v_3) : {●●, ●●}

(v_1, v_4) : {●●, ●●, ●●, ●●}

(v_1, v_5) : {●●, ●●, ●●, ●●}



Characterizing Robustly Approximable CSPs

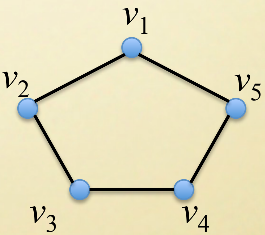
If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

(v_1, v_2) : { ●●, ●● }
(v_2, v_3) : { ●●, ●● }
(v_1, v_3) : {●●, ●●}
(v_1, v_4) : { ●●, ●● }
(v_1, v_5) : {●●, ●●, ●●, ●●}



Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has **width** k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

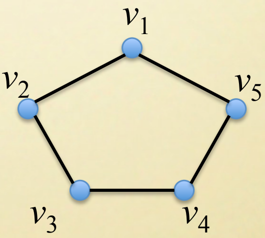
$(v_1, v_2) : \{$ ●●, ●● $\}$

$(v_2, v_3) : \{$ ●●, ●● $\}$

$(v_1, v_3) : \{$●●, ●●$\}$

$(v_1, v_4) : \{$ ●●, ●● $\}$

$(v_1, v_5) : \{$●●, ●●$\}$



Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

CSP(Γ) has width k if it can be solved by iteratively checking consistency of k vars.

E.g.: 2-Coloring has width 2.

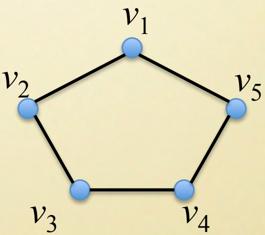
(v_1, v_2) : { ●●, ●● }

(v_2, v_3) : { ●●, ●● }

(v_1, v_3) : {●●, ●●}

(v_1, v_4) : { ●●, ●● }

(v_1, v_5) : {●●, ●●}



Contradiction!

Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

$\mathsf{CSP}(\Gamma)$ has width k if it can be solved by iteratively checking consistency of k vars.

CSP	Robust Approximability	Width
Horn k-SAT	Possible via LP	1
2-SAT	Possible via SDP	bounded width $(= 2)$
3-LIN_2	NP-Hard	∞

Characterizing Robustly Approximable CSPs

If you are familiar with universal algebra...

Definition (Width, informal)

$\mathsf{CSP}(\Gamma)$ has width k if it can be solved by iteratively checking consistency of k vars.

CSP	Robust Approximability	Width
Horn k-SAT	Possible via LP	1
2-SAT	Possible via SDP	bounded width ($= 2$)
3-LIN_2	NP-Hard	∞

Is this a coincidence? Of course not ☺.

Characterizing Robustly Approximable CSPs

Theorem

CSP(Γ) admits robust approximation

- via LP iff Γ has width 1 [KOT+12].
- via SDP iff Γ has bounded width [BK12].

Characterizing Robustly Approximable CSPs

Theorem

CSP(Γ) admits robust approximation

- via LP iff Γ has width 1 [KOT+12].
- via SDP iff Γ has bounded width [BK12].

Theorem

If Γ does not have bounded width, CSP(Γ) can "simulate" 3-LIN$_2$ and hence CSP(Γ) does not admit robust approximation.

Characterizing Robustly Approximable CSPs

Theorem

CSP(Γ) admits robust approximation

- via LP iff Γ has width 1 [KOT$^+$12].
- via SDP iff Γ has bounded width [BK12].

Theorem

If Γ does not have bounded width, CSP(Γ) can "simulate" 3-LIN$_2$ and hence CSP(Γ) does not admit robust approximation.

Corollary

CSP(Γ) admits robust approximation iff Γ has bounded width.

Width 1

$$\Updownarrow$$

Robust approximation via LP

Polymorphism

Definition (Polymorphism)

A function $f : [q]^k \to [q]$ is called a polymorphism of Γ if for any $R \in \Gamma$ of arity r,

$$\begin{array}{c} (x_1^1, \ldots, x_r^1) \in R \\ \vdots \\ (x_1^k, \ldots, x_r^k) \in R \\ \downarrow f \\ (z_1, \ldots, z_r) \in R \end{array}$$

$\mathsf{Pol}(\Gamma)$: set of polymorphisms of Γ.

Polymorphism

Example

- min is a polymorphism of Horn k-SAT for any k.
  Consider $R = \{(u, v, w) \mid u \wedge v \Rightarrow w\}$.

$$\begin{gathered}(1,0,1) \in R \\ (0,1,1) \in R \\ \downarrow \min \\ (0,0,1) \in R\end{gathered}$$

- majority is a polymorphism of 2-SAT.
- $x - y + z \pmod 2$ is a polymorphism of 3-LIN_2.
- Essentially, the only polymorphism 3-SAT has is $f(x) = x_i$ (dictator).

Polymorphisms Determine Complexity

Theorem ([BJK05])

Let Γ and Γ' be constraint languages with $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(\Gamma')$. Then, $\mathsf{CSP}(\Gamma')$ is log-space reducible to $\mathsf{CSP}(\Gamma)$.

To study computational complexity of $\mathsf{CSP}(\Gamma)$, we only have to study its polymorphisms!

Polymorphisms Determine Complexity

Theorem ([BJK05])

Let Γ and Γ' be constraint languages with $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(\Gamma')$. Then, $\mathsf{CSP}(\Gamma')$ is log-space reducible to $\mathsf{CSP}(\Gamma)$.

To study computational complexity of $\mathsf{CSP}(\Gamma)$, we only have to study its polymorphisms!

Theorem ([DK12])

Let Γ and Γ' be constraint languages with $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(\Gamma')$. If $\mathsf{CSP}(\Gamma)$ is robustly approximable, then $\mathsf{CSP}(\Gamma')$ is also robustly approximable.

To study robust approximability of $\mathsf{CSP}(\Gamma)$, we only have to study its polymorphisms!

Width 1 $\Leftrightarrow$ Robust Approximation via LP

Theorem ([KOT$^+$12])

TFAE.

1. Γ has width 1.
2. Pol(Γ) has a set operation.
3. BasicLP solves CSP(Γ).
4. BasicLP robustly approximates CSP(Γ).

Set operation: $f(x_1, \ldots, x_k)$ only depends on the (not multi-)set $\{x_1, \ldots, x_k\}$.
E.g. $\min(x_1, \ldots, x_k), \max(x_1, \ldots, x_k)$.

We will see (2) $\Rightarrow$ (3) (and (2) $\Rightarrow$ (4)).

Proof Idea

- De-combinatorialize the local propagation algorithm for solving $\mathsf{CSP}(\Gamma)$ when $c = 1$.
- Specifically, solve LP and use the set operation as a rounding procedure!
- Hope it works when $c = 1 - \epsilon$.

Proof Idea

- De-combinatorialize the local propagation algorithm for solving $\mathsf{CSP}(\Gamma)$ when $c = 1$.
- Specifically, solve LP and use the set operation as a rounding procedure!
- Hope it works when $c = 1 - \epsilon$.

Question: How can we use set operations? We don't have satisfying assignments beforehand.

Proof Idea

- De-combinatorialize the local propagation algorithm for solving $\mathsf{CSP}(\Gamma)$ when $c = 1$.
- Specifically, solve LP and use the set operation as a rounding procedure!
- Hope it works when $c = 1 - \epsilon$.

Question: How can we use set operations? We don't have satisfying assignments beforehand.

Answer: Set operations cannot distinguish satisfying LP solutions from satisfying assignments!

A Canonical LP Relaxation

BasicLP

$$\max \quad \mathop{\mathbf{E}}_{(R,S)\in\mathcal{C}} \Pr_{\beta\sim\boldsymbol{\mu}_S}[\beta \in R]$$

s.t. $\boldsymbol{\mu}_S$ is a probability distribution over $[q]^S$.

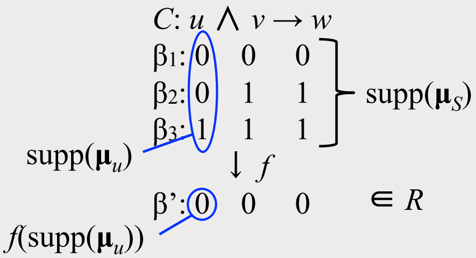
$\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_{S'}$ have the same marginal dist. $\boldsymbol{\mu}_u$ on every $u \in S \cap S'$.

Theorem

If Γ has width 1, then BasicLP solves CSP(Γ).

Proof.

Suppose $\mathbf{lp}(\mathcal{I}) = 1$.
$\beta(u) = f(\mathrm{supp}(\boldsymbol{\mu}_u))$ is a solution:
Fix a constraint $C = (R, S)$.



□

Theorem

If Γ has width 1, then $\mathsf{CSP}(\Gamma)$ is $O(\frac{1}{\log 1/\epsilon})$-robustly approximable via $\mathsf{BasicLP}$.

Proof sketch.

1. Pick θ from a certain distribution.
2. Define $R_u = \{a \in [q] \mid \boldsymbol{\mu}_u(a) \geq \theta\}$ for each $u \in V$.
3. Assign each u the value $f(R_u)$.

Theorem

If Γ has width 1, then $\mathsf{CSP}(\Gamma)$ is $O(\frac{1}{\log 1/\epsilon})$-robustly approximable via BasicLP.

Proof sketch.

1. Pick θ from a certain distribution.
2. Define $R_u = \{a \in [q] \mid \boldsymbol{\mu}_u(a) \geq \theta\}$ for each $u \in V$.
3. Assign each u the value $f(R_u)$.

For each constraint $C = (R, S)$, R_u plays the role of $\mathrm{supp}(\boldsymbol{\mu}_u)$ and $R \cap \prod_{u \in S} R_u$ plays the role of $\mathrm{supp}(\boldsymbol{\mu}_S)$. □

Bounded Width

$$\Updownarrow$$

Robust Approximation via SDP

Bounded Width

Theorem ([BK12, KS09])

TFAE.

1. Γ has bounded width.
2. $\mathsf{Pol}(\Gamma)$ has pseudorandom operations.
3. $\mathsf{BasicSDP}$ solves $\mathsf{CSP}(\Gamma)$.
4. $\mathsf{BasicSDP}$ robustly approximates $\mathsf{CSP}(\Gamma)$.

Proof Idea

- De-combinatorialize the local propagation algorithm for solving $\mathsf{CSP}(\Gamma)$ when $c = 1$.
- Specifically, solve SDP and use the pseudorandom operation as a rounding procedure!
- Hope it works when $c = 1 - \epsilon$.

Proof Idea

- De-combinatorialize the local propagation algorithm for solving CSP(Γ) when $c = 1$.
- Specifically, solve SDP and use the pseudorandom operation as a rounding procedure!
- Hope it works when $c = 1 - \epsilon$.

[BK12] Instead of pseudorandom operations, we would use more sophisticated universal algebraic tool!

A Canonical SDP Relaxation

BasicSDP

$$\max \quad \mathop{\mathbf{E}}_{(R,S)\in\mathcal{C}} \Pr_{\beta\sim\boldsymbol{\mu}_S}[\beta \in R]$$

s.t. $\boldsymbol{\mu}_S$ is a probability distribution over $[q]^S$.

$\boldsymbol{\mu}_S, \boldsymbol{\mu}_{S'}$ have consistent marginal dist. $\boldsymbol{\mu}_{uv}$ on every $\{u, v\} \subseteq S \cap S'$.

Theorem

If Γ has bounded width, then BasicSDP solves $\mathsf{CSP}(\Gamma)$.

- Can assume every constraint is binary.
- Make an instance with
  - a constraint $R_u = (\mathrm{supp}(\boldsymbol{\mu}_u), u)$ for each u.
  - a constraint $R_{uv} = (\mathrm{supp}(\boldsymbol{\mu}_{uv}), \{u, v\})$ for each $\{u, v\}$.
- If the new instance has a solution, then the old one has a solution.

Theorem

If Γ has bounded width, then BasicSDP solves CSP(Γ).

- Can assume every constraint is binary.
- Make an instance with
  - a constraint $R_u = (\mathrm{supp}(\boldsymbol{\mu}_u), u)$ for each u.
  - a constraint $R_{uv} = (\mathrm{supp}(\boldsymbol{\mu}_{uv}), \{u, v\})$ for each $\{u, v\}$.
- If the new instance has a solution, then the old one has a solution.
  - Why?
  - $(a, b) \in \mathrm{supp}(\boldsymbol{\mu}_{uv})$ implies (a, b) is a satisfying tuple from $\mathbf{sdp}(\mathcal{I}) = 1$.
  - $a \in \mathrm{supp}(\boldsymbol{\mu}_u)$ from consistency.

Theorem

If Γ has bounded width, then BasicSDP solves CSP(Γ).

- Can assume every constraint is binary.
- Make an instance with
  - a constraint $R_u = (\mathrm{supp}(\boldsymbol{\mu}_u), u)$ for each u.
  - a constraint $R_{uv} = (\mathrm{supp}(\boldsymbol{\mu}_{uv}), \{u, v\})$ for each $\{u, v\}$.
- If the new instance has a solution, then the old one has a solution.
  - Why?
  - $(a, b) \in \mathrm{supp}(\boldsymbol{\mu}_{uv})$ implies (a, b) is a satisfying tuple from $\mathbf{sdp}(\mathcal{I}) = 1$.
  - $a \in \mathrm{supp}(\boldsymbol{\mu}_u)$ from consistency.

We now show several facts about R_u and R_{uv}.

$R_u = \mathrm{supp}(\boldsymbol{\mu}_u)$ and $R_{uv} = \mathrm{supp}(\boldsymbol{\mu}_{uv})$.

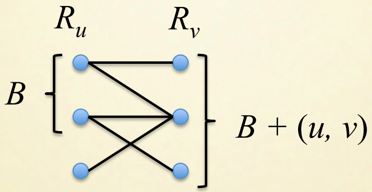
Lemma

R_{uv} is a subdirect subset of $R_u \times R_v$.

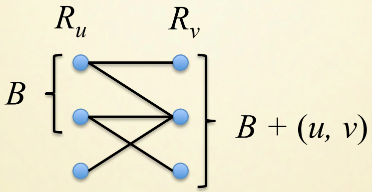
Proof.

- It is a subset: If $(a, b) \in \mathrm{supp}(\boldsymbol{\mu}_{uv})$, then $a \in \mathrm{supp}(\boldsymbol{\mu}_u)$ and $b \in \mathrm{supp}(\boldsymbol{\mu}_b)$.
- It is subdirect: If $a \in \mathrm{supp}(\boldsymbol{\mu}_u)$, then $(a, b) \in \mathrm{supp}(\boldsymbol{\mu}_{uv})$ for some b. $\square$

For $B \subseteq R_u$, let $B + (u, v) = \{c \in [q] \mid \exists b \in B, (b, c) \in R_{uv}\}$.



For $B \subseteq R_u$, let $B + (u, v) = \{c \in [q] \mid \exists b \in B, (b, c) \in R_{uv}\}$.



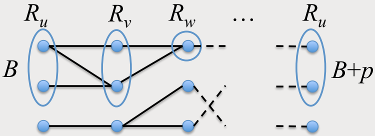
Lemma

For $B \subseteq R_u$, $\boldsymbol{\mu}_v(B + (u, v)) \geq \boldsymbol{\mu}_u(B)$. The equality holds iff $B = B + (u, v) - (u, v)$.

Definition (Pattern)

A (correct) sequence p of variables is called a pattern.

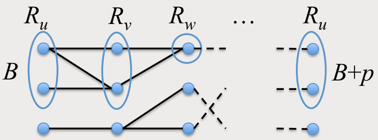
$B + p$, $B - p$ defined in a natural way.



Definition (Pattern)

A (correct) sequence p of variables is called a pattern.

$B + p$, $B - p$ defined in a natural way.



Lemma

For any $B \subseteq R_u$ and patterns p, q from u to u we have

- If $B + p = B$, then $B - p = B$.
- If $B + p + q = B$, then $B + p = B$.

Definition (Weak Prague instance)

An instance with constraints $\{R_u\}$ and $\{R_{uv}\}$ is a weak Prague instance if (for every $u, v \in V$, $B \subseteq R_u$ and patterns p, q from u to u)

- R_{uv} is a subdirect subset of $R_u \times R_v$.
- If $B + p = B$, then $B - p = B$.
- If $B + p + q = B$, then $B + p = B$.

Definition (Weak Prague instance)

An instance with constraints $\{R_u\}$ and $\{R_{uv}\}$ is a weak Prague instance if (for every $u, v \in V$, $B \subseteq R_u$ and patterns p, q from u to u)

- R_{uv} is a subdirect subset of $R_u \times R_v$.
- If $B + p = B$, then $B - p = B$.
- If $B + p + q = B$, then $B + p = B$.

Theorem ([BK12])

Every weak Prague instance has a solution.

$\Rightarrow$ SDP solves bounded-width CSPs.

Definition (Weak Prague instance)

An instance with constraints $\{R_u\}$ and $\{R_{uv}\}$ is a weak Prague instance if (for every $u, v \in V$, $B \subseteq R_u$ and patterns p, q from u to u)

- R_{uv} is a subdirect subset of $R_u \times R_v$.
- If $B + p = B$, then $B - p = B$.
- If $B + p + q = B$, then $B + p = B$.

Theorem ([BK12])

Every weak Prague instance has a solution.

$\Rightarrow$ SDP solves bounded-width CSPs.

For the general case, polish the input instance a lot to obtain a weak Prague instance.

Other Topics

Quantitative Characterization of Boolean CSPs

Theorem ([DK12])

Let Γ be a Boolean constraint language.

- If $\mathsf{Pol}(\Gamma)$ contains $x \vee (y \wedge z)$ or $x \wedge (y \vee z)$, we can $O(\epsilon)$-robustly approximate.
- Otherwise, if $\mathsf{Pol}(\Gamma)$ contains a majority, we can $O(\sqrt{\epsilon})$-robustly approximate.
- Otherwise, if $\mathsf{Pol}(\Gamma)$ contains $\min$ or $\max$, we can $O(\frac{1}{\log 1/\epsilon})$-robustly approximate.
- Otherwise, robust approximation is NP-Hard.

All these positive results are (almost) tight under UGC.

Open Problem: Can we generalize to non-Boolean CSPs?

Ordering CSPs

Definition (Ordering CSPs)

- Assignment: ordering of the variables without ties.
- Constraints: allowed relative orderings of k-subsets of variables.

Example: Max Acyclic Subgraph

Ordering CSPs

Definition (Ordering CSPs)

- Assignment: ordering of the variables without ties.
- Constraints: allowed relative orderings of k-subsets of variables.

Example: Max Acyclic Subgraph

Theorem ([GMR08])

Assuming UGC, no (interesting) ordering CSP admits robust approximation.

Temporal CSPs

Temporal CSPs

- Assignment: ordering of the variables possibly with ties.
- Constraints: allowed relative orderings of k-subsets of variables.

Example: Correlation Clustering

Temporal CSPs

Temporal CSPs

- Assignment: ordering of the variables possibly with ties.
- Constraints: allowed relative orderings of k-subsets of variables.

Example: Correlation Clustering

Theorem ([TY13])

$\mathsf{CSP}(\Gamma)$ admits robust approximation iff Γ is Horn $=$-SAT. That is, each constraint is of the form

$$(u_1 = v_1) \wedge (u_2 = v_2) \wedge \cdots \wedge (u_{k-1} = v_{k-1}) \Rightarrow (u_k = v_k).$$

Quantitative version is also available.

Graph Isomorphism

Graph Isomorphism (MaxGI)

Given two graphs $G = (V, E)$ and $H = (V, F)$, find a bijection $\sigma : V \to V$ that maximizes the number of matched edges, i.e., $\{(u, v) \in E \mid (\sigma(u), \sigma(v)) \in F\}$.

Graph Isomorphism

Graph Isomorphism (MaxGI)

Given two graphs $G = (V, E)$ and $H = (V, F)$, find a bijection $\sigma : V \to V$ that maximizes the number of matched edges, i.e., $\{(u, v) \in E \mid (\sigma(u), \sigma(v)) \in F\}$.

Theorem ([WYZV13])

- MaxGI is NP-Hard.
- Trees admit robust approximation.

Graph Isomorphism

Graph Isomorphism (MaxGI)

Given two graphs $G = (V, E)$ and $H = (V, F)$, find a bijection $\sigma : V \to V$ that maximizes the number of matched edges, i.e., $\{(u, v) \in E \mid (\sigma(u), \sigma(v)) \in F\}$.

Theorem ([WYZV13])

- MaxGI is NP-Hard.
- Trees admit robust approximation.

Open Problem:

- Do planar graphs admit robust approximation?
- Sherali-Adams LP relaxation solves GI of planar graphs. Does it give robust approximation?

Summary

Universal algebra is a useful tool to study robust approximation of CSPs.
Standard CSPs and temporal CSPs are basically solved.

Summary

Universal algebra is a useful tool to study robust approximation of CSPs.
Standard CSPs and temporal CSPs are basically solved.

Open Problem: What comes next by unifying universal algebra and optimization?

Summary

Universal algebra is a useful tool to study robust approximation of CSPs.
Standard CSPs and temporal CSPs are basically solved.

Open Problem: What comes next by unifying universal algebra and optimization?

Fin.