Robust algorithms for CSPs

Libor Barto

joint work with Marcin Kozik

McMaster University
and
Charles University in Prague

AAA 83 Novi Sad, March 15, 2012

(Part 1) Outline

- (Part 2) Introduction
- (Part 3) Problem
- (Part 4) Problem solved
- (Part 5) Proof of a different result
- (Part 6) Proof of one more different result

(Part 2)
Introduction

Constraint Satisfaction Problem (CSP)

Definition (Instance of the CSP)

Instance of the CSP consists of:

- V ... a set of **variables**
- A ... a **domain**
- list of **constraints** of the form $R(x_1, \dots, x_k)$, where
  - $x_1, \dots, x_k \in V$
  - R is a k-ary relation on A (i.e. $R \subseteq A^k$) **constraint relation**

Constraint Satisfaction Problem (CSP)

Definition (Instance of the CSP)

Instance of the CSP consists of:

- V ... a set of **variables**
- A ... a **domain**
- list of **constraints** of the form $R(x_1, \ldots, x_k)$, where
  - $x_1, \ldots, x_k \in V$
  - R is a k-ary relation on A (i.e. $R \subseteq A^k$) **constraint relation**

An assignment $f : V \to A$ satisfies $R(x_1, \ldots, x_k)$, if $(f(x_1), \ldots, f(x_k)) \in R$

$f : V \to A$ is a solution if it satisfies all the constraints

Some questions we can ask

- **Decision CSP:** Does a solution exist?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** Find a map satisfying at least $0.7 \times \textit{Optimum}$ constraints

Some questions we can ask

- **Decision CSP:** Does a solution exist?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** Find a map satisfying at least $0.7 \times$ *Optimum* constraints

Definition

An algorithm (α, β)-approximates CSP $(0 \leq \alpha \leq \beta \leq 1)$ if it returns an assignment satisfying α-fraction of the constraints given a β-satisfiable instance.

Some questions we can ask

- **Decision CSP:** Does a solution exist?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** Find a map satisfying at least $0.7 \times$ *Optimum* constraints

Definition

An algorithm (α, β)-approximates CSP ($0 \leq \alpha \leq \beta \leq 1$) if it returns an assignment satisfying α-fraction of the constraints given a β-satisfiable instance.

Example

$(0.7\beta, \beta)$-approximating algorithm returns a map satisfying at least $0.7 \times$ *Optimum* constraints.

Constraint language

Mentioned problems are computationally hard

One possible restriction (widely studied) — fix a set of possible constraint relations:

Definition

A constraint language Γ is a finite set of relations on a finite set A.

An instance of $\mathrm{CSP}(\Gamma)$ is a CSP instance such that every constraint relation is from Γ.

Example: 2-coloring

$A = \{0, 1\}$, $\Gamma = \{R\}$, $R = \{(0, 1), (1, 0)\}$ (inequality)

Instance: $R(x_1, x_2), R(x_1, x_3), R(x_2, x_4), \ldots$
(can be drawn as a graph)

Solution = 2-coloring (bipartition)

- **Decision** $CSP(\Gamma)$: Is a given graph bipartite? (easy)
- **Max-**$CSP(\Gamma)$: also called Max-Cut (hard)
- **Approx. Max-**$CSP(\Gamma)$
  - $(0.5\beta, \beta)$-approx easy
  - $(0.878\beta, \beta)$-approx easy Goemans and Williamson'95
  - $(16/17\beta, \beta)$-approx hard
    Trevisan, Sorkin, Sudan, Williamson'00, Hastad'01
  - $((0.878 + \varepsilon)\beta, \beta)$ - approx UGC-hard
    Khot, Kindler, Mossel, O'Donnel'07

Example: 3-SAT

$A = \{0,1\}$, $\Gamma = \{R_{000}, R_{001}, R_{011}, R_{111}\}$, $R_{ijk} = \{0,1\}^3 \ \{(i,j,k)\}$

Instance: $R_{000}(x_1, x_2, x_3), R_{001}(x_1, x_3, x_5), R_{011}(x_3, x_2, x_6)$

or: $(x_1 \vee x_2 \vee x_3)$ & $(x_1 \vee \neg x_3 \vee \neg x_5)$ & $(x_3 \vee \neg x_2 \vee \neg x_6)$

- **Decision** $CSP(\Gamma)$: 3-SAT (hard)
- **Max-**$CSP(\Gamma)$: Max-3-SAT (hard)
- **Approx. Max-**$CSP(\Gamma)$:
  - $(7/8\beta, \beta)$-approx easy Karloff, Zwick'96
  - $(\delta, 1)$-approx hard for some $\delta < 1$
    (=PCP theorem, Arora, Lund, Motwani, Sudan, Szegedy'98)
  - $(7/8 + \varepsilon, 1)$-approx hard Hastad'01

Example: 3-Lin-2

$A = \{0,1\}$, $\Gamma =$\{affine subspaces of Z_2^3\}

Instance: system of linear equation over Z_2
(each equation contains at most 3 variables)

- **Decision** $CSP(\Gamma)$: easy (Gaussian elimination)
- **Max-**$CSP(\Gamma)$: hard
- **Approx. Max-**$CSP(\Gamma)$:
  - $(1/2\beta, \beta)$-approx easy
  - $(1/2+\varepsilon, 1-\varepsilon)$-approx hard Hastad'01

(Part 3)
Problem

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0) = 0$.

Motivation: Instances close to satisfiable (e.g. corrupted by noise), we want to find an "almost solution".

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0) = 0$.

- 2-SAT, HORN-SAT have robust algorithms Zwick'98

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0) = 0$.

- 2-SAT, HORN-SAT have robust algorithms Zwick'98
  - $(1 - O(\varepsilon^{1/3}), 1 - \varepsilon)$-approx algorithm for 2-SAT
  - $(1 - O(1/(\log(1/\varepsilon))), 1 - \varepsilon)$-approx algorithm for HORN-SAT

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0)=0$.

- 2-SAT, HORN-SAT have robust algorithms Zwick'98
- If the decision problem for $\mathrm{CSP}(\Gamma)$ is NP-complete, then $\mathrm{CSP}(\Gamma)$ has no robust algorithm (PCP,
  for $|A|=2$ Khanna,Sudan,Trevisan, Williamson'00
  for larger Jonsson, Krokhin, Kuivinen'09)

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0) = 0$.

- 2-SAT, HORN-SAT have robust algorithms Zwick'98
- If the decision problem for $\mathrm{CSP}(\Gamma)$ is NP-complete, then $\mathrm{CSP}(\Gamma)$ has no robust algorithm (PCP,
  for $|A| = 2$ Khanna,Sudan,Trevisan, Williamson'00
  for larger Jonsson, Krokhin, Kuivinen'09)
- LIN-p has no robust algorithm Hastad'01

Between decision and approximation

Definition (Zwick'98)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm, if there is a polynomial time algorithm which
$(1-g(\varepsilon), 1-\varepsilon)$-approximates $\mathrm{CSP}(\Gamma)$ (for every ε),
where $g(\varepsilon) \to 0$ when $\varepsilon \to 0$, and $g(0) = 0$.

- 2-SAT, HORN-SAT have robust algorithms Zwick'98
- If the decision problem for $\mathrm{CSP}(\Gamma)$ is NP-complete, then $\mathrm{CSP}(\Gamma)$ has no robust algorithm (PCP,
  for $|A| = 2$ Khanna,Sudan,Trevisan, Williamson'00
  for larger Jonsson, Krokhin, Kuivinen'09)
- LIN-p has no robust algorithm Hastad'01

What distinguishes between LIN-p, 3-SAT and 2-SAT, HORN-SAT?

Decision CSPs and bounded width

- $\operatorname{Pol}\Gamma =$ clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ Bulatov, Jeavons, Krokhin 00

Decision CSPs and bounded width

- $\mathrm{Pol}\,\Gamma =$ clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathrm{HSP}(\mathrm{Pol}\,\Gamma)$ Bulatov, Jeavons, Krokhin 00
- $\mathrm{CSP}(\Gamma)$ has bounded width iff it can be solved by local consistency checking

Decision CSPs and bounded width

- $\operatorname{Pol}\Gamma =$ clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ Bulatov, Jeavons, Krokhin 00
- $\mathrm{CSP}(\Gamma)$ has bounded width iff it can be solved by local consistency checking
- $\mathrm{CSP}(\Gamma)$ has bounded width iff Γ "cannot encode linear equations", more precisely, $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ does not contain a reduct of a module (for core Γ) Barto, Kozik'09 Bulatov'09

Decision CSPs and bounded width

- $\operatorname{Pol}\Gamma$ = clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ Bulatov, Jeavons, Krokhin 00
- $\mathrm{CSP}(\Gamma)$ has bounded width iff it can be solved by local consistency checking
- $\mathrm{CSP}(\Gamma)$ has bounded width iff Γ "cannot encode linear equations", more precisely, $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ does not contain a reduct of a module (for core Γ) Barto, Kozik'09 Bulatov'09
- Lin-p, 3-SAT do not have bounded width,
  2-SAT, HORN-SAT have bounded width

Decision CSPs and bounded width

- $\operatorname{Pol}\Gamma$ = clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ Bulatov, Jeavons, Krokhin 00
- $\mathrm{CSP}(\Gamma)$ has bounded width iff it can be solved by local consistency checking
- $\mathrm{CSP}(\Gamma)$ has bounded width iff Γ "cannot encode linear equations", more precisely, $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ does not contain a reduct of a module (for core Γ) Barto, Kozik'09 Bulatov'09
- **Lin-p, 3-SAT do not have bounded width, 2-SAT, HORN-SAT have bounded width !!!**

Decision CSPs and bounded width

- $\operatorname{Pol}\Gamma$ = clone of polymorphisms (operations compatible with all relations in Γ)
- Complexity of the decision problem for $\mathrm{CSP}(\Gamma)$ controlled by $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ Bulatov, Jeavons, Krokhin 00
- $\mathrm{CSP}(\Gamma)$ has bounded width iff it can be solved by local consistency checking
- $\mathrm{CSP}(\Gamma)$ has bounded width iff Γ "cannot encode linear equations", more precisely, $\mathsf{HSP}(\operatorname{Pol}\Gamma)$ does not contain a reduct of a module (for core Γ) Barto, Kozik'09 Bulatov'09
- **Lin-p, 3-SAT do not have bounded width, 2-SAT, HORN-SAT have bounded width**

Conjecture (Guruswami-Zhou 11)

$\mathrm{CSP}(\Gamma)$ admits a robust algorithm iff $\mathrm{CSP}(\Gamma)$ has bounded width.

Universal algebra attacks robust approximation

- robust approximation also (+-) controlled by polymorphisms Dalmau, Krokhin'11
- $\Rightarrow$ one direction of the Guruswami-Zhou conjecture is true

Universal algebra attacks robust approximation

- robust approximation also (+-) controlled by polymorphisms
  Dalmau, Krokhin'11
- $\Rightarrow$ one direction of the Guruswami-Zhou conjecture is true
- Conjecture confirmed for width 1 CSPs
  Kun, O'Donell, Tamaki, Yoshida, Zhou'11,
  Dalmau, Krokhin'11.
  width 1 iff linear programming relaxation can be used.

Universal algebra attacks robust approximation

- robust approximation also (+-) controlled by polymorphisms
  Dalmau, Krokhin'11
- ⇒ one direction of the Guruswami-Zhou conjecture is true
- Conjecture confirmed for width 1 CSPs
  Kun, O'Donell, Tamaki, Yoshida, Zhou'11,
  Dalmau, Krokhin'11.
  width 1 iff linear programming relaxation can be used.
- Conjecture confirmed
  Barto, Kozik'11.

Universal algebra attacks robust approximation

- robust approximation also (+-) controlled by polymorphisms Dalmau, Krokhin'11
- $\Rightarrow$ one direction of the Guruswami-Zhou conjecture is true
- Conjecture confirmed for width 1 CSPs Kun, O'Donell, Tamaki, Yoshida, Zhou'11, Dalmau, Krokhin'11.
  width 1 iff linear programming relaxation can be used.
- Conjecture confirmed Barto, Kozik'11. Using a semidefinite programming relaxation and Prague strategies.
  - Randomized $(1 - O(\log\log(1/\varepsilon)/\log(1/\varepsilon)), 1 - \varepsilon)$-approx algorithm
  - Deterministic $(1 - O(\log\log(1/\varepsilon)/\sqrt{\log(1/\varepsilon)}), 1 - \varepsilon)$-approx algorithm

Universal algebra attacks robust approximation

- robust approximation also (+-) controlled by polymorphisms Dalmau, Krokhin'11
- $\Rightarrow$ one direction of the Guruswami-Zhou conjecture is true
- Conjecture confirmed for width 1 CSPs
  Kun, O'Donell, Tamaki, Yoshida, Zhou'11,
  Dalmau, Krokhin'11.
  width 1 iff linear programming relaxation can be used.
- Conjecture confirmed Barto, Kozik'11. Using a semidefinite programming relaxation and Prague strategies.
  - Randomized $(1 - O(\log\log(1/\varepsilon)/\log(1/\varepsilon)), 1-\varepsilon)$-approx algorithm
  - Deterministic $(1 - O(\log\log(1/\varepsilon)/\sqrt{\log(1/\varepsilon)}), 1-\varepsilon)$-approx algorithm
- Bonus Krokhin'11: even the quantitative dependence on ε is +- controlled by polymorphisms.

This was (Part 4) Problem solved

Now (Part 5)
Proof of a different result

MAX-CUT Goemans and Williamson'95

$A = \{-1, 1\}$, $\Gamma = \{R\}$, $R = \{(-1,1),(1,-1)\}$ (inequality)

Instance $\mathcal{I}$: $V = \{x_1, x_2, \ldots,\}$, $\mathcal{C} = R(x_2, x_1), R(x_1, x_4), \ldots$

MAX-CUT Goemans and Williamson'95

$A = \{-1, 1\}$, $\Gamma = \{R\}$, $R = \{(-1, 1), (1, -1)\}$ (inequality)

Instance $\mathcal{I}$: $V = \{x_1, x_2, \ldots,\}$, $\mathcal{C} = R(x_2, x_1), R(x_1, x_4), \ldots$

Max-CSP – hard:

Find **numbers** $f(x), x \in V$, $f(x) \in \{-1, 1\}$ which maximize

$$\mathrm{Opt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - f(x)f(y)}{2}$$

MAX-CUT Goemans and Williamson'95

$A = \{-1, 1\}$, $\Gamma = \{R\}$, $R = \{(-1, 1), (1, -1)\}$ (inequality)

Instance $\mathcal{I}$: $V = \{x_1, x_2, \ldots,\}$, $\mathcal{C} = R(x_2, x_1), R(x_1, x_4), \ldots$

Max-CSP – hard:

Find **numbers** $f(x), x \in V,\ f(x) \in \{-1, 1\}$ which maximize

$$\mathrm{Opt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - f(x)f(y)}{2}$$

SDP (semidefinite programming) relaxation – easy:

Find **vectors** $g(x), x \in V,\ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

MAX-CUT cont'd

Find vectors $g(x), x \in V, \ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

MAX-CUT cont'd

Find vectors $g(x), x \in V, \ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

- $\mathrm{SDPOpt}(\mathcal{I}) \geq \mathrm{Opt}(\mathcal{I})$, if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathrm{Opt}(\mathcal{I}) = 1$.

MAX-CUT cont'd

Find vectors $g(x), x \in V, \ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

- $\mathrm{SDPOpt}(\mathcal{I}) \geq \mathrm{Opt}(\mathcal{I})$, if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathrm{Opt}(\mathcal{I}) = 1$.
- We need to round the vector solution g to a reasonably good assignment f

MAX-CUT cont'd

Find vectors $g(x), x \in V, \ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

- $\mathrm{SDPOpt}(\mathcal{I}) \geq \mathrm{Opt}(\mathcal{I})$, if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathrm{Opt}(\mathcal{I}) = 1$.
- We need to round the vector solution g to a reasonably good assignment f
  - Choose a random hyperplane through the origin and choose one side S
  - Put $f(v) = 1$ if $g(v) \in S$ and $f(v) = -1$ otherwise

MAX-CUT cont'd

Find vectors $g(x), x \in V, \ \|g(x)\|^2 = 1$ which maximize

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R(x,y) \in \mathcal{C}} \frac{1 - g(x)g(y)}{2}$$

- $\mathrm{SDPOpt}(\mathcal{I}) \geq \mathrm{Opt}(\mathcal{I})$, if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathrm{Opt}(\mathcal{I}) = 1$.
- We need to round the vector solution g to a reasonably good assignment f
  - Choose a random hyperplane through the origin and choose one side S
  - Put $f(v) = 1$ if $g(v) \in S$ and $f(v) = -1$ otherwise
- This is $(0.878\beta, \beta)$-approx and robust algorithm

(Part 6)
Proof of one more different result

SDP relaxation for general CSP

Notation and simplifying assumptions:

- A – domain
- Γ contains only binary relations, $\mathrm{CSP}(\Gamma)$ has bounded width
- V – variables, $\mathcal{I}$ - instance, $\mathcal{C}$ – constraints

SDP relaxation for general CSP

Notation and simplifying assumptions:

- A – domain
- Γ contains only binary relations, $\mathrm{CSP}(\Gamma)$ has bounded width
- V – variables, $\mathcal{I}$ - instance, $\mathcal{C}$ – constraints
- $\forall \{x, y\} \subseteq V$, $x \neq y$ there is at most one constraint $R_{xy}(x, y) \in \mathcal{C}$

[picture]

SDP relaxation for general CSP

Notation and simplifying assumptions:

- A – domain
- Γ contains only binary relations, $\mathrm{CSP}(\Gamma)$ has bounded width
- V – variables, $\mathcal{I}$ - instance, $\mathcal{C}$ – constraints
- $\forall \{x, y\} \subseteq V$, $x \neq y$ there is at most one constraint $R_{xy}(x, y) \in \mathcal{C}$
- $\mathrm{Opt}(\mathcal{I})$ – optimal fraction of satisfied constraints
- ... and we want to find an assignment satisfying a big fraction of the constraints)

[picture]

SDP relaxation for general CSP

Notation and simplifying assumptions:

- A – domain
- Γ contains only binary relations, $\mathrm{CSP}(\Gamma)$ has bounded width
- V – variables, $\mathcal{I}$ - instance, $\mathcal{C}$ – constraints
- $\forall \{x, y\} \subseteq V$, $x \neq y$ there is at most one constraint $R_{xy}(x, y) \in \mathcal{C}$
- $\mathrm{Opt}(\mathcal{I})$ – optimal fraction of satisfied constraints
- ... and we want to find an assignment satisfying a big fraction of the constraints)

[picture]

Canonical SDP relaxation is strong enough to get optimal approximation constants (assuming UGC) Raghavendra'08

Let's try to use it for our problem.

Canonical SDP relaxation

Find vectors $g(x, a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a \in B} \mathbf{x}_a$)

Canonical SDP relaxation

Find vectors $g(x, a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a \in B} \mathbf{x}_a$) such that for all $x, y \in V$, $a, b \in A$

- (SDP1) $\mathbf{x}_a \mathbf{y}_b \geq 0$
- (SDP2) $\mathbf{x}_a \mathbf{x}_b = 0$ if $a \neq b$
- (SDP3) $\mathbf{x}_A = \mathbf{y}_A$, $\|\mathbf{x}_A\|^2 = 1$

Canonical SDP relaxation

Find vectors $g(x,a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a\in B} \mathbf{x}_a$) such that for all $x, y \in V$, $a, b \in A$

- (SDP1) $\mathbf{x}_a \mathbf{y}_b \geq 0$
- (SDP2) $\mathbf{x}_a \mathbf{x}_b = 0$ if $a \neq b$
- (SDP3) $\mathbf{x}_A = \mathbf{y}_A$, $\|\mathbf{x}_A\|^2 = 1$

maximizing

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R_{xy}(x,y)\in\mathcal{C}} \sum_{(a,b)\in R_{xy}} \mathbf{x}_a \mathbf{y}_b.$$

Intuition:

Canonical SDP relaxation

Find vectors $g(x,a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a \in B} \mathbf{x}_a$)
such that for all $x, y \in V$, $a, b \in A$

- (SDP1) $\mathbf{x}_a \mathbf{y}_b \geq 0$
- (SDP2) $\mathbf{x}_a \mathbf{x}_b = 0$ if $a \neq b$
- (SDP3) $\mathbf{x}_A = \mathbf{y}_A$, $\|\mathbf{x}_A\|^2 = 1$

maximizing

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R_{xy}(x,y) \in \mathcal{C}} \sum_{(a,b) \in R_{xy}} \mathbf{x}_a \mathbf{y}_b.$$

Intuition:

- $\mathbf{x}_a \mathbf{y}_b$ is a weight (nonnegative) of the pair (a,b) between variables x, y

Canonical SDP relaxation

Find vectors $g(x,a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a \in B} \mathbf{x}_a$) such that for all $x, y \in V$, $a, b \in A$

- (SDP1) $\mathbf{x}_a \mathbf{y}_b \geq 0$
- (SDP2) $\mathbf{x}_a \mathbf{x}_b = 0$ if $a \neq b$
- (SDP3) $\mathbf{x}_A = \mathbf{y}_A$, $\|\mathbf{x}_A\|^2 = 1$

maximizing

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R_{xy}(x,y) \in \mathcal{C}} \sum_{(a,b) \in R_{xy}} \mathbf{x}_a \mathbf{y}_b.$$

Intuition:

- $\mathbf{x}_a \mathbf{y}_b$ is a weight (nonnegative) of the pair (a,b) between variables x, y
- Sum of all weights (between x, y) is 1 from (SDP3)

Canonical SDP relaxation

Find vectors $g(x,a) =: \mathbf{x}_a, x \in V, a \in A$ (notation: $\mathbf{x}_B = \sum_{a \in B} \mathbf{x}_a$)
such that for all $x, y \in V$, $a, b \in A$

- (SDP1) $\mathbf{x}_a \mathbf{y}_b \geq 0$
- (SDP2) $\mathbf{x}_a \mathbf{x}_b = 0$ if $a \neq b$
- (SDP3) $\mathbf{x}_A = \mathbf{y}_A$, $\|\mathbf{x}_A\|^2 = 1$

maximizing

$$\mathrm{SDPOpt}(\mathcal{I}) = \frac{1}{|\mathcal{C}|} \sum_{R_{xy}(x,y) \in \mathcal{C}} \sum_{(a,b) \in R_{xy}} \mathbf{x}_a \mathbf{y}_b.$$

Intuition:

- $\mathbf{x}_a \mathbf{y}_b$ is a weight (nonnegative) of the pair (a,b) between variables x, y
- Sum of all weights (between x, y) is 1 from (SDP3)
- We are trying to give small weights to pairs outside R_{xy}

Strategy

- We try to produce a good assignment from the SDP output vectors.

Strategy

- We try to produce a good assignment from the SDP output vectors.
- In particular, is it true that if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathcal{I}$ has a solution? This was suggested by Guruswami as the first step to attack the conjecture

Strategy

- We try to produce a good assignment from the SDP output vectors.
- In particular, is it true that if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathcal{I}$ has a solution? This was suggested by Guruswami as the first step to attack the conjecture
- So, assume $\mathrm{SDPOpt}(\mathcal{I}) = 1$.

Strategy

- We try to produce a good assignment from the SDP output vectors.
- In particular, is it true that if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathcal{I}$ has a solution? This was suggested by Guruswami as the first step to attack the conjecture
- So, assume $\mathrm{SDPOpt}(\mathcal{I}) = 1$.
- It follows that $\mathbf{x}_a \mathbf{y}_b = 0$ for every $(a, b) \notin R_{xy}$

Strategy

- We try to produce a good assignment from the SDP output vectors.
- In particular, is it true that if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathcal{I}$ has a solution? This was suggested by Guruswami as the first step to attack the conjecture
- So, assume $\mathrm{SDPOpt}(\mathcal{I}) = 1$.
- It follows that $\mathbf{x}_a \mathbf{y}_b = 0$ for every $(a, b) \notin R_{xy}$
- Define $P_{xy} = \{(a, b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\}$. Replace R_{xy} with P_{xy}. If the new instance has a solution then the old one has a solution.

Strategy

- We try to produce a good assignment from the SDP output vectors.
- In particular, is it true that if $\mathrm{SDPOpt}(\mathcal{I}) = 1$ then $\mathcal{I}$ has a solution? This was suggested by Guruswami as the first step to attack the conjecture
- So, assume $\mathrm{SDPOpt}(\mathcal{I}) = 1$.
- It follows that $\mathbf{x}_a\mathbf{y}_b = 0$ for every $(a, b) \notin R_{xy}$
- Define $P_{xy} = \{(a, b) \in A^2 : \mathbf{x}_a\mathbf{y}_b > 0\}$. Replace R_{xy} with P_{xy}. If the new instance has a solution then the old one has a solution.
- Define $P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$. And let's see what we get

Random facts about P_x, P_{xy}

$$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\},\ P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

Random facts about P_x, P_{xy}

$$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\},\ P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)
  - It is a subset: If $\mathbf{x}_a \mathbf{y}_b > 0$ then $\mathbf{x}_a, \mathbf{y}_b \neq \mathbf{o}$
  - It is subdirect: If $\mathbf{x}_a \neq \mathbf{o}$ then $0 \neq \|\mathbf{x}_a\|^2 = \mathbf{x}_a \mathbf{x}_A = \mathbf{x}_a \mathbf{y}_A$, therefore $\mathbf{x}_a \mathbf{y}_b \neq 0$ for some b

Random facts about P_x, P_{xy}

$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\}$, $P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

For $B \subseteq P_x$ let $B + (x,y) = \{c \in A : (\exists b \in B)\ (b,c) \in P_{xy}\}$

- For $B \subseteq P_x$, we have $\mathbf{y}_{B+(x,y)} = \mathbf{x}_B + \mathbf{w}$,
  where $\mathbf{w}\mathbf{x}_B = 0$, and $\mathbf{w} = \mathbf{o}$ iff $B = B + (x,y) - (x,y)$.

Random facts about P_x, P_{xy}

$$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\},\ P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

For $B \subseteq P_x$ let $B + (x,y) = \{c \in A : (\exists b \in B)\ (b,c) \in P_{xy}\}$

- For $B \subseteq P_x$, we have $\mathbf{y}_{B+(x,y)} = \mathbf{x}_B + \mathbf{w}$, where $\mathbf{w}\mathbf{x}_B = 0$, and $\mathbf{w} = \mathbf{o}$ iff $B = B + (x,y) - (x,y)$.
    - $\mathbf{w}\mathbf{x}_B = (\mathbf{y}_{B+(x,y)} - \mathbf{x}_B)\mathbf{x}_B = \mathbf{y}_{B+(x,y)}\mathbf{x}_B - \mathbf{x}_B\mathbf{x}_B = \mathbf{y}_{B+(x,y)}\mathbf{x}_B - \mathbf{y}_A\mathbf{x}_B = -(\mathbf{y}_A - \mathbf{y}_{B+(x,y)})\mathbf{x}_B = -\mathbf{y}_{A-(B+(x,y))}\mathbf{x}_B = 0$
    - $\mathbf{w}\mathbf{w} = \cdots = \mathbf{x}_{A-B}\mathbf{y}_{B+(x,y)}$

Random facts about P_x, P_{xy}

$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\}$, $P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

For $B \subseteq P_x$ let $B + (x,y) = \{c \in A : (\exists b \in B)\ (b,c) \in P_{xy}\}$

- For $B \subseteq P_x$, we have $\mathbf{y}_{B+(x,y)} = \mathbf{x}_B + \mathbf{w}$,
  where $\mathbf{w}\mathbf{x}_B = 0$, and $\mathbf{w} = \mathbf{o}$ iff $B = B + (x,y) - (x,y)$.

A (correct) sequence of variables is called a pattern
$B + p, B - p$ defined in a natural way for a pattern p

Random facts about P_x, P_{xy}

$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\}$, $P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

For $B \subseteq P_x$ let $B + (x,y) = \{c \in A : (\exists b \in B)\ (b,c) \in P_{xy}\}$

- For $B \subseteq P_x$, we have $\mathbf{y}_{B+(x,y)} = \mathbf{x}_B + \mathbf{w}$,
  where $\mathbf{w}\mathbf{x}_B = 0$, and $\mathbf{w} = \mathbf{o}$ iff $B = B + (x,y) - (x,y)$.

A (correct) sequence of variables is called a pattern
$B + p, B - p$ defined in a natural way for a pattern p

For any $B \subseteq P_x$ and patterns p, q from x to x we have

- If $B + p = B$ then $B - p = B$

Random facts about P_x, P_{xy}

$P_{xy} = \{(a,b) \in A^2 : \mathbf{x}_a \mathbf{y}_b > 0\}$, $P_x = \{a \in A : \mathbf{x}_a \neq \mathbf{o}\}$

- P_{xy} is a subdirect subset of $P_x \times P_y$ (1-minimality)

For $B \subseteq P_x$ let $B + (x,y) = \{c \in A : (\exists b \in B)\ (b,c) \in P_{xy}\}$

- For $B \subseteq P_x$, we have $\mathbf{y}_{B+(x,y)} = \mathbf{x}_B + \mathbf{w}$,
  where $\mathbf{w}\mathbf{x}_B = 0$, and $\mathbf{w} = \mathbf{o}$ iff $B = B + (x,y) - (x,y)$.

A (correct) sequence of variables is called a pattern
$B + p, B - p$ defined in a natural way for a pattern p

For any $B \subseteq P_x$ and patterns p, q from x to x we have

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

Random facts about P_x, P_{xy} - summary

The new instance with constraints $P_{xy}(x, y)$ and subsets $P_x \subseteq A, x \in V$ satisfies
(for every $x, y \in V$, $B \subseteq P_x$ and patterns p, q from x to x)

- It is 1-minimal (P_{xy} is a subdirect subset of $P_x \times P_y$)
- If $B + p = B$ then $B - p = B$
- If $B + p + q = B$ then $B + p = B$

Weak Prague instance

Definition

An instance with constraints $P_{xy}(x, y)$ and subsets $P_x \subseteq A, x \in V$ is a weak Prague instance if
(for every $x, y \in V$, $B \subseteq P_x$ and patterns p, q from x to x)

- It is 1-minimal (P_{xy} is a subdirect subset of $P_x \times P_y$)
- If $B + p = B$ then $B - p = B$
- If $B + p + q = B$ then $B + p = B$

Weak Prague instance

Definition

An instance with constraints $P_{xy}(x, y)$ and subsets $P_x \subseteq A, x \in V$ is a weak Prague instance if
(for every $x, y \in V$, $B \subseteq P_x$ and patterns p, q from x to x)

- It is 1-minimal (P_{xy} is a subdirect subset of $P_x \times P_y$)
- If $B + p = B$ then $B - p = B$
- If $B + p + q = B$ then $B + p = B$

- Slightly weaker notion than Prague strategy
- Every Prague strategy has a solution (if P_{xy}'s are invariant under $\mathrm{Pol}\,\Gamma$...) BK

Weak Prague instance

Definition

An instance with constraints $P_{xy}(x, y)$ and subsets $P_x \subseteq A, x \in V$ is a weak Prague instance if
(for every $x, y \in V$, $B \subseteq P_x$ and patterns p, q from x to x)

- It is 1-minimal (P_{xy} is a subdirect subset of $P_x \times P_y$)
- If $B + p = B$ then $B - p = B$
- If $B + p + q = B$ then $B + p = B$

- Slightly weaker notion than Prague strategy
- Every Prague strategy has a solution (if P_{xy}'s are invariant under $\operatorname{Pol}\Gamma$...) BK
- Every weak Prague strategy has a solution K

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W.........W......OOO......RRR.........K......K.......

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W..........W......OOO......RRR.........K......K.......
- ....W..........W....O......O.....R.....R......K...K.......

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W..........W......OOO......RRR.........K......K.......
- ....W..........W....O......O.....R.....R......K...K.......
- ....W...W...W....O......O......RRR........KK.............

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W..........W......OOO......RRR.........K......K.......
- ....W..........W....O......O.....R.....R......K...K.......
- ....W...W...W....O......O......RRR........KK.............
- ....W...W...W....O......O.....R.....R......K...K.........

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W..........W......OOO......RRR.........K......K.......
- ....W..........W....O......O.....R.....R......K...K.......
- ....W...W...W....O......O......RRR........KK.............
- ....W...W...W....O......O.....R.....R......K...K..........
- .......W...W.........OOO.......R.......R....K......K......

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W..........W......OOO......RRR.........K......K.......
- ....W..........W....O......O.....R......R......K...K.......
- ....W...W...W....O......O......RRR........KK.............
- ....W...W...W....O......O.....R......R......K...K..........
- .......W...W.........OOO.......R........R...K......K......
- ....

General case

- $\mathrm{SDPOpt}(\Gamma) = 1 - \varepsilon$, ε small
- We define $P_{xy} = \{(a, b) : \mathbf{x}_a \mathbf{x}_b > \delta\}$
- If δ is big enough then for almost all x, y we have $P_{xy} \subseteq R_{xy}$
- If δ is small enough then the calculations will almost work...
- ....
- ....
- ....W.........W......OOO......RRR.........K......K.......
- ....W.........W....O......O.....R.....R......K...K.......
- ....W...W...W....O......O......RRR........KK.............
- ....W...W...W....O......O.....R.....R......K...K..........
- .......W...W.........OOO.......R.......R....K......K......
- ....
- QED

Final remarks

- Is the quantitative dependence optimal?

Final remarks

- Is the quantitative dependence optimal?
- How to improve derandomization to match the randomized version?

Final remarks

- Is the quantitative dependence optimal?
- How to improve derandomization to match the randomized version?
- What can we say about the quantitative dependence on ε in general?

  Wild guess: NU $\Rightarrow$ polynomial loss

Final remarks

- Is the quantitative dependence optimal?
- How to improve derandomization to match the randomized version?
- What can we say about the quantitative dependence on ε in general?

  Wild guess: NU $\Rightarrow$ polynomial loss
- SDP, LP outputs $\leftrightarrow$ consistency notions (within CSP).
  What is the precise connection?
  Is there any connection beyond CSPs?

Final remarks

- Is the quantitative dependence optimal?
- How to improve derandomization to match the randomized version?
- What can we say about the quantitative dependence on ε in general?

  Wild guess: NU $\Rightarrow$ polynomial loss
- SDP, LP outputs $\leftrightarrow$ consistency notions (within CSP).
  What is the precise connection?
  Is there any connection beyond CSPs?
- Thank you!