Borel Local Lemma for graphs of slow growth

Anton Bernshteyn
joint with **Jing Yu**, Fudan University

University of California, Los Angeles

North American Annual Meeting of the ASL

The Lovász Local Lemma

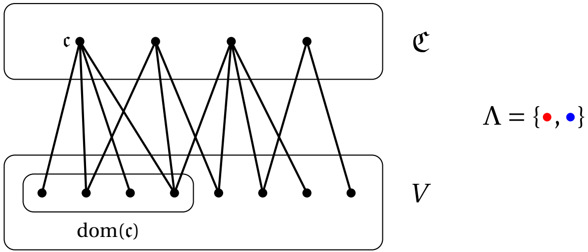
The **Lovász Local Lemma** (the **LLL**) is a powerful probabilistic tool.

- Introduced by Erdős and Lovász in '75.
- Useful for proving existence results.
- Used throughout combinatorics.
- Recently found a number of applications in other areas (topological dynamics, ergodic theory, descriptive set theory).

This talk is about using the LLL for **Borel constructions**.

Constraint satisfaction problem (**CSP**) is $\boxed{\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})}$:

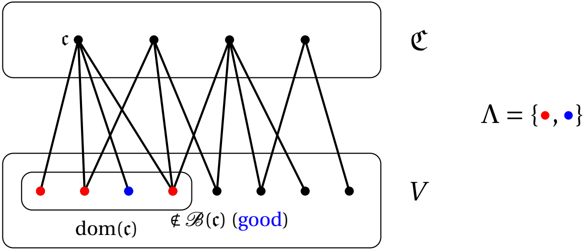
- V, Λ, and $\mathfrak{C}$ are sets of variables, labels, and constraints,
- for each constraint $\mathfrak{c} \in \mathfrak{C}$:
  - $\mathrm{dom}(\mathfrak{c}) \subseteq V$ is a finite set, the domain of $\mathfrak{c}$,
  - $\mathscr{B}(\mathfrak{c}) \subseteq \Lambda^{\mathrm{dom}(\mathfrak{c})}$ is the set of bad labelings of $\mathrm{dom}(\mathfrak{c})$.



Framework: Constraint Satisfaction Problems

Constraint satisfaction problem (**CSP**) is $\boxed{\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})}$:

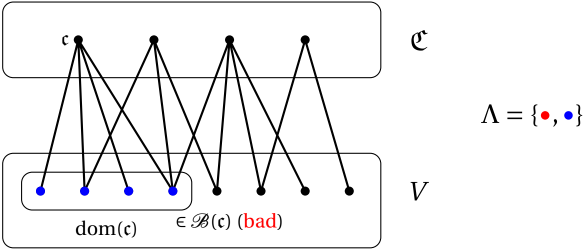
- V, Λ, and $\mathfrak{C}$ are sets of variables, labels, and constraints,
- for each constraint $\mathfrak{c} \in \mathfrak{C}$:
  - $\mathrm{dom}(\mathfrak{c}) \subseteq V$ is a finite set, the domain of $\mathfrak{c}$,
  - $\mathscr{B}(\mathfrak{c}) \subseteq \Lambda^{\mathrm{dom}(\mathfrak{c})}$ is the set of bad labelings of $\mathrm{dom}(\mathfrak{c})$.



Framework: Constraint Satisfaction Problems

Constraint satisfaction problem (**CSP**) is $\boxed{\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})}$:

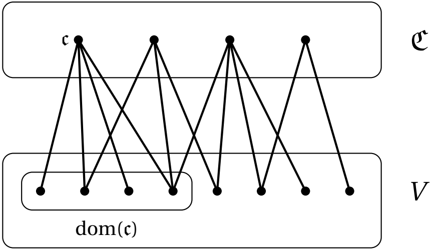
- V, Λ, and $\mathfrak{C}$ are sets of variables, labels, and constraints,
- for each constraint $\mathfrak{c} \in \mathfrak{C}$:
  - $\mathrm{dom}(\mathfrak{c}) \subseteq V$ is a finite set, the domain of $\mathfrak{c}$,
  - $\mathscr{B}(\mathfrak{c}) \subseteq \Lambda^{\mathrm{dom}(\mathfrak{c})}$ is the set of bad labelings of $\mathrm{dom}(\mathfrak{c})$.



Framework: Constraint Satisfaction Problems

Constraint satisfaction problem (**CSP**) is $\boxed{\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})}$:

- V, Λ, and $\mathfrak{C}$ are sets of variables, labels, and constraints,
- for each constraint $\mathfrak{c} \in \mathfrak{C}$:
  - $\mathrm{dom}(\mathfrak{c}) \subseteq V$ is a finite set, the domain of $\mathfrak{c}$,
  - $\mathscr{B}(\mathfrak{c}) \subseteq \Lambda^{\mathrm{dom}(\mathfrak{c})}$ is the set of bad labelings of $\mathrm{dom}(\mathfrak{c})$.



$\Lambda = \{\bullet, \bullet\}$

A solution to Π: a labeling $f \colon V \to \Lambda$ s.t. $f|_{\mathrm{dom}(\mathfrak{c})} \notin \mathscr{B}(\mathfrak{c})$ $\forall \mathfrak{c} \in \mathfrak{C}$.

Example: sinkless orientation

Let G be a locally finite graph. An orientation of G is sinkless if it has no sinks, i.e., vertices with outdegree 0.

Example: sinkless orientation

Let G be a locally finite graph. An orientation of G is sinkless if it has no sinks, i.e., vertices with outdegree 0.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+, -\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

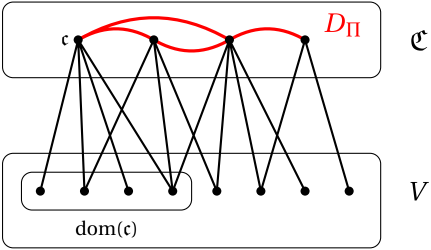
- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

More definitions (sorry!)

Let $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$ be a CSP.

Dependency graph D_Π:

- $V(D_\Pi) := \mathfrak{C}$,
- $E(D_\Pi) := \big\{\{\mathfrak{c}, \mathfrak{c}'\} : \mathfrak{c} \neq \mathfrak{c}' \text{ and } \mathrm{dom}(\mathfrak{c}) \cap \mathrm{dom}(\mathfrak{c}') \neq \varnothing\big\}$

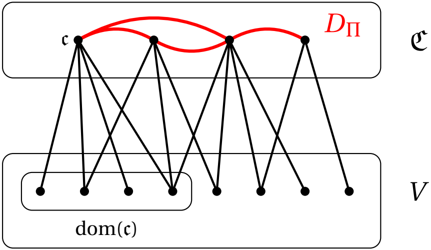


More definitions (sorry!)

Let $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$ be a CSP.

Dependency graph D_Π:

- $V(D_\Pi) := \mathfrak{C}$,
- $E(D_\Pi) := \big\{\{\mathfrak{c}, \mathfrak{c}'\} : \mathfrak{c} \neq \mathfrak{c}' \text{ and } \mathrm{dom}(\mathfrak{c}) \cap \mathrm{dom}(\mathfrak{c}') \neq \varnothing\big\}$



We also need some extra structure:

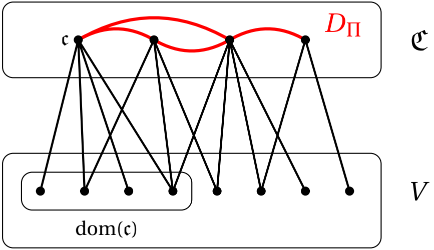
$(\Lambda, \mathbb{P})$ is a probability space s.t. each $\mathscr{B}(\mathfrak{c})$ is $\mathbb{P}^{\mathrm{dom}(\mathfrak{c})}$-measurable.

More definitions (sorry!)

Let $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$ be a CSP.

Dependency graph D_Π:

- $V(D_\Pi) := \mathfrak{C}$,
- $E(D_\Pi) := \big\{\{\mathfrak{c}, \mathfrak{c}'\} : \mathfrak{c} \neq \mathfrak{c}' \text{ and } \mathrm{dom}(\mathfrak{c}) \cap \mathrm{dom}(\mathfrak{c}') \neq \varnothing\big\}$



We also need some extra structure:

$(\Lambda, \mathbb{P})$ is a probability space s.t. each $\mathscr{B}(\mathfrak{c})$ is $\mathbb{P}$-measurable.

Finally, the Local Lemma

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

Then Π has a solution.

Finally, the Local Lemma

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0, 1)$, $d \in \mathbb{N}$ such that:

- $\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p$ for all $\mathfrak{c} \in \mathfrak{C}$,

Then Π has a solution.

Finally, the Local Lemma

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

- $\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p$ for all $\mathfrak{c} \in \mathfrak{C}$,
- the maximum degree of D_Π is $\leqslant d$, and

Then Π has a solution.

Finally, the Local Lemma

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

- $\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p$ for all $\mathfrak{c} \in \mathfrak{C}$,
- the maximum degree of D_Π is $\leqslant d$, and
- $\boxed{e \cdot p \cdot (d+1) < 1}$. $e = 2.71828\ldots$

Then Π has a solution.

Example: sinkless orientation

Say G is a Δ-regular graph. Want to find a sinkless orientation of G.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+, -\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

Example: sinkless orientation

Say G is a Δ-regular graph. Want to find a sinkless orientation of G.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+, -\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

Take $\mathbb{P}$ to be the uniform measure on $\{+, -\}$.

Example: sinkless orientation

Say G is a Δ-regular graph. Want to find a sinkless orientation of G.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+, -\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

Take $\mathbb{P}$ to be the uniform measure on $\{+, -\}$.

Since G is Δ-regular, we can take

$$d = \Delta \qquad \text{and} \qquad p = 2^{-\Delta}.$$

Example: sinkless orientation

Say G is a Δ-regular graph. Want to find a sinkless orientation of G.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+, -\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

Take $\mathbb{P}$ to be the uniform measure on $\{+, -\}$.

Since G is Δ-regular, we can take

$$d = \Delta \qquad \text{and} \qquad p = 2^{-\Delta}.$$

Since $\boxed{ep(d+1)} = e2^{-\Delta}(\Delta + 1) < 1$ for all $\Delta \geqslant 4$, we conclude:

Example: sinkless orientation

Say G is a Δ-regular graph. Want to find a sinkless orientation of G.

A sinkless orientation of G is a solution to a CSP with

- $V = E(G)$, $\mathfrak{C} = V(G)$, $\Lambda = \{+,-\}$ (coding the direction of an edge),

where for each $x \in V(G)$:

- $\mathrm{dom}(x) = \{\text{edges incident to } x\}$,
- $\mathscr{B}(x) = \{\text{the unique labeling of } \mathrm{dom}(x) \text{ making } x \text{ a sink}\}$.

Take $\mathbb{P}$ to be the uniform measure on $\{+,-\}$.

Since G is Δ-regular, we can take

$$d = \Delta \qquad \text{and} \qquad p = 2^{-\Delta}.$$

Since $\boxed{ep(d+1)} = e2^{-\Delta}(\Delta+1) < 1$ for all $\Delta \geqslant 4$, we conclude:

If $\Delta \geqslant 4$, G has a sinkless orientation.

Example: sinkless orientation

Remarks:

- Very flexible: the exact nature of the problem is irrelevant.
- There is a “slack” in the $ep(d+1) < 1$ bound.

Example: sinkless orientation

Remarks:

- Very flexible: the exact nature of the problem is irrelevant.
- There is a "slack" in the $ep(d+1) < 1$ bound.

Downside: Nonconstructive!

Theorem (Thornton '20)

For any $\Delta \in \mathbb{N}$, there exists a Δ-regular Borel graph G with **no** Borel sinkless orientation. ☹

Powerful assumption: Subexponential growth

From now on, **all CSPs are Borel** (V, Λ, and $\mathfrak{C}$ are standard Borel spaces and dom and $\mathscr{B}$ are Borel in the natural sense).

When does a Borel CSP have a Borel solution?

Powerful assumption: Subexponential growth

From now on, **all CSPs are Borel** (V, Λ, and $\mathfrak{C}$ are standard Borel spaces and dom and $\mathscr{B}$ are Borel in the natural sense).

When does a Borel CSP have a Borel solution?

Let G be a locally finite graph.

Growth function: $\gamma_G(R) := \sup_{x \in V(G)} |B_G(x, R)|$

Exponential growth rate: $\mathrm{egr}(G) := \lim_{R \to \infty} \sqrt[R]{\gamma_G(R)}$

The graph G is of subexponential growth if $\mathrm{egr}(G) = 1$.

Powerful assumption: Subexponential growth

Theorem (ERDŐS–LOVÁSZ '75): the **Lovász Local Lemma**

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Powerful assumption: Subexponential growth

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

(some technical assumption)

If D_Π is $\boxed{\text{of subexponential growth}}$, then Π has a **Borel** solution.

Powerful assumption: Subexponential growth

Theorem (Erdős–Lovász '75): the **Lovász Local Lemma**

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is } \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

(some technical assumption)

If D_Π is of subexponential growth, then Π has a **Borel** solution.

For example, this implies the following for $\Delta \geqslant 4$:

Theorem (Thornton '20)

If $\Delta \geqslant 3$, every Borel Δ-regular graph G of subexponential growth has a Borel sinkless orientation.

Powerful assumption: Subexponential growth

Theorem (ERDŐS–LOVÁSZ '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (CSÓKA–GRABOWSKI–MÁTHÉ–PIKHURKO–TYROS)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

(some technical assumption)

If D_Π is $\boxed{\text{of subexponential growth}}$, then Π has a **Borel** solution.

Lots of applications!

Theorem (AB–YU '25): Uses the CGMPT Borel LLL

Borel graphs of polynomial growth are hyperfinite.

Powerful assumption: Subexponential growth

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

$(*)$ Assume that $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})|$ and $|\Lambda|$ are finite.

If D_Π is $\boxed{\text{of subexponential growth}}$, then Π has a Borel solution.

Powerful assumption: Subexponential growth

Theorem (Erdős–Lovász '75): the **Lovász Local Lemma**

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

$(*)$ Assume that $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})|$ and $|\Lambda|$ are finite.

If D_Π is $\boxed{\text{of subexponential growth}}$, then Π has a Borel solution.

Condition $(*)$ typically holds in applications.

Powerful assumption: Subexponential growth

Theorem (Erdős–Lovász '75): the Lovász Local Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ such that:

$$\mathbb{P}[\mathscr{B}(\mathfrak{c})] \leqslant p \ \forall \mathfrak{c} \in \mathfrak{C}, \quad \text{max. deg. of } D_\Pi \text{ is} \leqslant d, \quad \text{and } e \cdot p \cdot (d+1) < 1.$$

Then Π has a solution.

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

(∗) Assume that $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})|$ and $|\Lambda|$ are finite.

If D_Π is of subexponential growth, then Π has a Borel solution.

Condition (∗) typically holds in applications.

But do we *really* need it?..

A refinement

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that Λ is finite and $\mathbb{P}$ is the uniform measure on Λ.

Assume further that there exist $\Delta \in \mathbb{N}$ and $\varepsilon > 0$ s.t.:

- $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})| \leqslant \Delta$ and $\sup_{x \in V} |\mathrm{dom}^{-1}(x)| \leqslant \Delta$,
- $\mathsf{egr}(D_\Pi) < (1+\varepsilon)^{2/3}$.

If $\boxed{e \cdot p \cdot (d+1) \cdot |\Lambda|^{\varepsilon\Delta} < 1}$, then Π has a Borel solution.

A refinement

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that Λ is finite and $\mathbb{P}$ is the uniform measure on Λ.

Assume further that there exist $\Delta \in \mathbb{N}$ and $\varepsilon > 0$ s.t.:

- $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})| \leqslant \Delta$ and $\sup_{x \in V} |\mathrm{dom}^{-1}(x)| \leqslant \Delta$,
- $\mathsf{egr}(D_\Pi) < (1+\varepsilon)^{2/3}$.

If $\boxed{e \cdot p \cdot (d+1) \cdot |\Lambda|^{\varepsilon\Delta} < 1}$, then Π has a Borel solution.

This can be applied when D_Π is of slow exponential growth…

A refinement

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that Λ is finite and $\mathbb{P}$ is the uniform measure on Λ.

Assume further that there exist $\Delta \in \mathbb{N}$ and $\varepsilon > 0$ s.t.:

- $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})| \leqslant \Delta$ and $\sup_{x \in V} |\mathrm{dom}^{-1}(x)| \leqslant \Delta$,
- $\mathsf{egr}(D_\Pi) < (1+\varepsilon)^{2/3}$.

If $\boxed{e \cdot p \cdot (d+1) \cdot |\Lambda|^{\varepsilon\Delta} < 1}$, then Π has a Borel solution.

This can be applied when D_Π is of slow exponential growth...

...but the base of the exponent depends on $|\Lambda|$ (and Δ)

A refinement

Theorem (Csóka–Grabowski–Máthé–Pikhurko–Tyros)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that Λ is finite and $\mathbb{P}$ is the uniform measure on Λ.

Assume further that there exist $\Delta \in \mathbb{N}$ and $\varepsilon > 0$ s.t.:

- $\sup_{\mathfrak{c} \in \mathfrak{C}} |\mathrm{dom}(\mathfrak{c})| \leqslant \Delta$ and $\sup_{x \in V} |\mathrm{dom}^{-1}(x)| \leqslant \Delta$,
- $\mathsf{egr}(D_\Pi) < (1+\varepsilon)^{2/3}$.

If $\boxed{e \cdot p \cdot (d+1) \cdot |\Lambda|^{\varepsilon\Delta} < 1}$, then Π has a Borel solution.

This can be applied when D_Π is of slow exponential growth...

...but the base of the exponent depends on $|\Lambda|$ (and Δ)

In particular, this can only be used when $\mathsf{egr}(D_\Pi) < 2^{2/3} \approx 1.587\ldots$

Our result

We prove a Borel version of the LLL under less restrictive growth assumptions and with no restrictions on Λ:

Theorem (AB–Yu)

Suppose there exist $p \in (0, 1)$, $d \in \mathbb{N}$ as in the LLL.

Assume further that $\operatorname{egr}(D_\Pi) < s$ for some $s > 1$.

If $\boxed{p \cdot (e \cdot (d+1))^s < 1}$, then Π has a Borel solution.

The growth rate bound only depends on p and d.

Our result

We prove a Borel version of the LLL under less restrictive growth assumptions and with no restrictions on Λ:

Theorem (AB–Yu)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume further that $\mathrm{egr}(D_\Pi) < s$ for some $s > 1$.

If $\boxed{p \cdot (e \cdot (d+1))^s < 1}$, then Π has a Borel solution.

The growth rate bound only depends on p and d.

Corollary (AB–Yu)

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL (so $e \cdot p \cdot (d+1) < 1$).

If D_Π is of subexponential growth, then Π has a Borel solution.

Example: Borel sinkless orientations

Theorem (Thornton '20)

If $\Delta \geqslant 3$, every Borel Δ-regular graph G of subexponential growth has a Borel sinkless orientation.

Example: Borel sinkless orientations

Theorem (Thornton '20)

If $\Delta \geqslant 3$, every Borel Δ-regular graph G of subexponential growth has a Borel sinkless orientation.

Our result further implies that:

$$\mathrm{egr}(G) \lesssim \frac{\Delta}{\log_2 \Delta} \quad \Longrightarrow \quad \text{a Borel sinkless orientation.}$$

Proof overview

(1) **Moser–Tardos Algorithm**: a **randomized** procedure to find a solution under the LLL assumptions

(2) **Problem:** The MTA requires too much randomness!

- CGMPT: Randomness conservation: use the same random input multiple times

(3) **Our solution:** Probability boosting

Moser–Tardos Algorithm

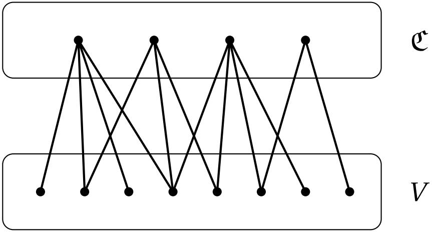
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

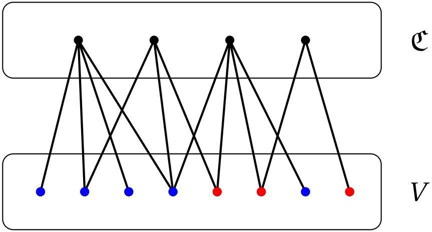
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

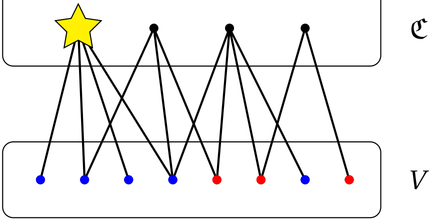
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

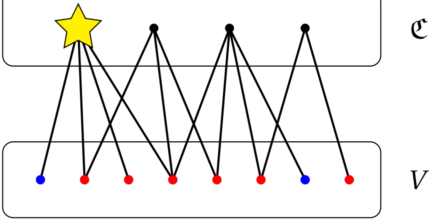
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\textcolor{red}{\bullet}, \textcolor{blue}{\bullet}\}$

bad = constant

Moser–Tardos Algorithm

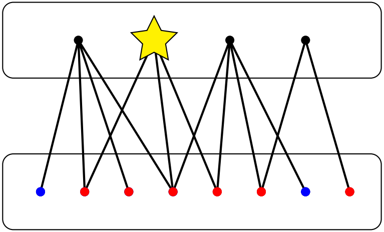
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\color{red}{\bullet}, \color{blue}{\bullet}\}$

bad = constant

Moser–Tardos Algorithm

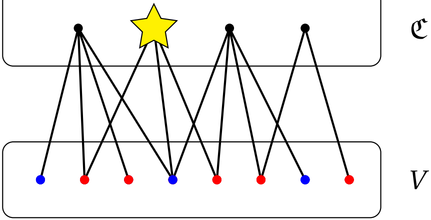
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\color{red}{\bullet}, \color{blue}{\bullet}\}$

bad = constant

Moser–Tardos Algorithm

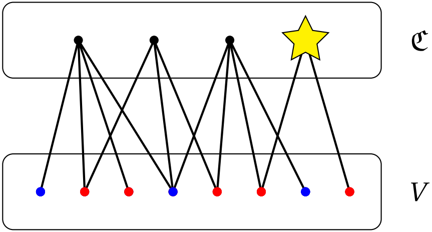
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

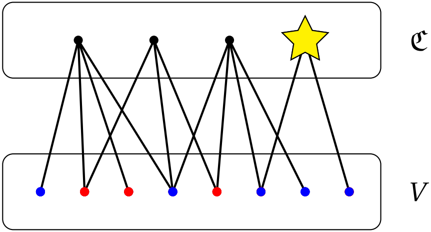
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

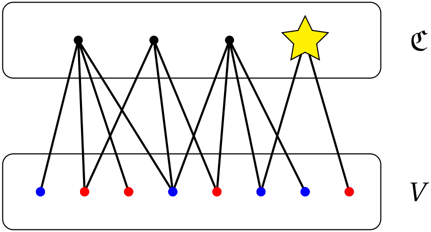
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm

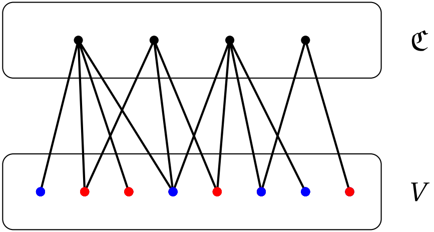
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\color{red}{\bullet}, \color{blue}{\bullet}\}$

bad = constant

Moser–Tardos Algorithm

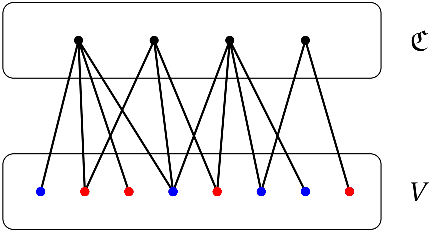
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

MOSER–TARDOS '10: If $ep(d+1) < 1$, then every variable $x \in V$ gets resampled finitely many times with probability 1.

Moser–Tardos Algorithm

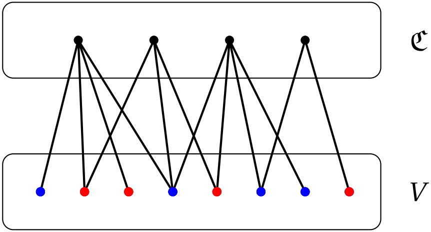
We are given a CSP $\Pi = (V, \Lambda, \mathfrak{C}, \mathrm{dom}, \mathscr{B})$, want a solution.

Moser–Tardos Algorithm

Sample the initial value $f(x) \in \Lambda$ for each $x \in V$ independently.

While at least one constraint is violated **do:**

- Pick a violated constraint $\mathfrak{c} \in \mathfrak{C}$ (adversarially)
- **Resample** f on $\mathrm{dom}(\mathfrak{c})$



$\Lambda = \{\bullet, \bullet\}$

bad = constant

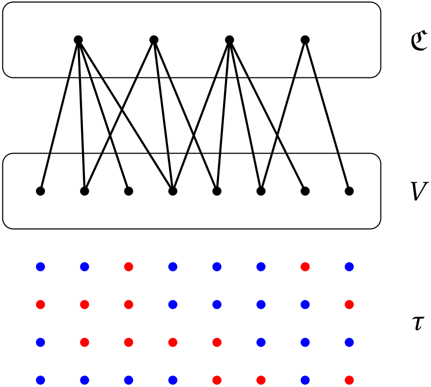
Moser–Tardos '10: If $ep(d+1) < 1$, then every variable $x \in V$ gets resampled finitely many times with probability 1.

Moser–Tardos Algorithm: Borel version

Let $\tau\colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\color{red}{\bullet}, \color{blue}{\bullet}\}$

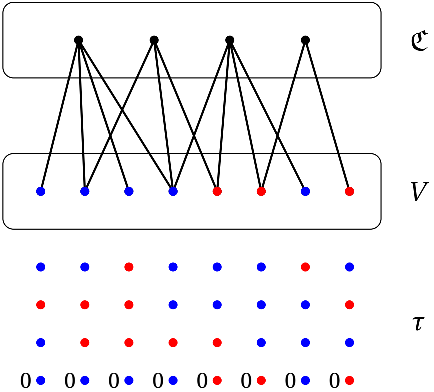
bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau\colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



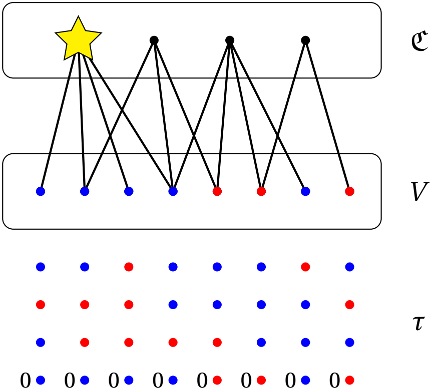
$\Lambda = \{\bullet, \bullet\}$

bad = constant

Let $\tau\colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



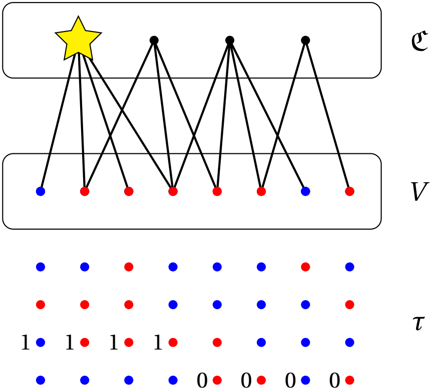
$\Lambda = \{\bullet, \bullet\}$

bad = constant

Let $\tau\colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



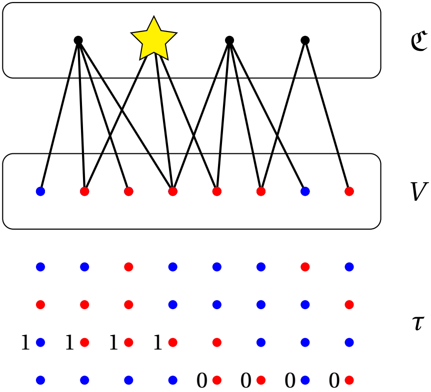
$\Lambda = \{\bullet, \bullet\}$

bad = constant

Let $\tau\colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\bullet, \bullet\}$

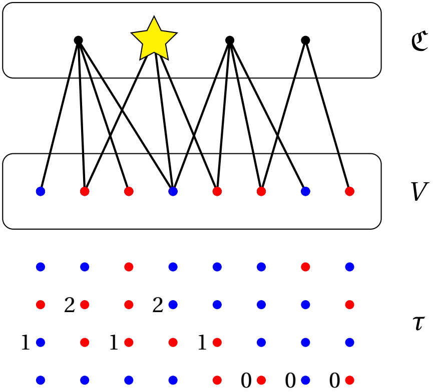
bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x, 0), \tau(x, 1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\bullet, \bullet\}$

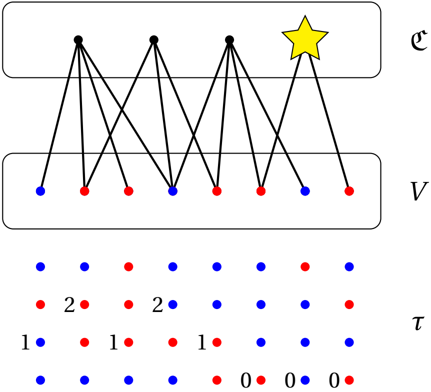
bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x, 0), \tau(x, 1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\bullet, \bullet\}$

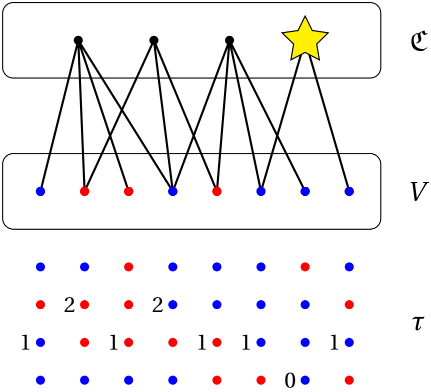
bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau : V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x, 0), \tau(x, 1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



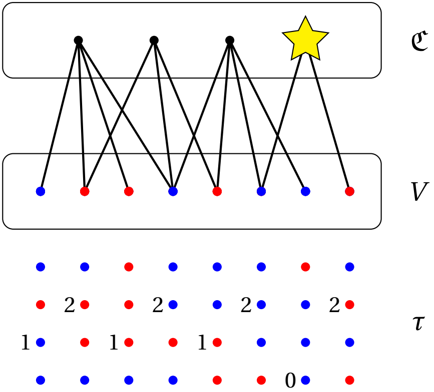
$\Lambda = \{\color{red}{\bullet}, \color{blue}{\bullet}\}$

bad = constant

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\bullet, \bullet\}$

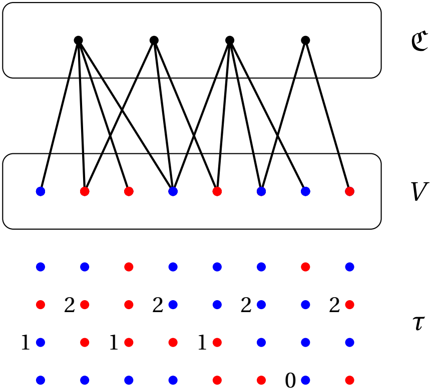
bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau : V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.



$\Lambda = \{\bullet, \bullet\}$

bad = constant

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.

If $\ell(x)$ eventually stabilizes, then we can define

$$f(x) := \tau\big(x, \text{eventual value of } \ell(x)\big).$$

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.

If $\ell(x)$ eventually stabilizes, then we can define

$$f(x) := \tau\big(x, \text{eventual value of } \ell(x)\big).$$

If τ is Borel and f is defined everywhere, then f is a Borel solution!

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.

If $\ell(x)$ eventually stabilizes, then we can define

$$f(x) := \tau\big(x, \text{eventual value of } \ell(x)\big).$$

If τ is Borel and f is defined everywhere, then f is a Borel solution!

MT '10: If $\tau \sim \mathbb{P}^V$, then $\forall x,\ f(x)$ exists with prob. 1.

Moser–Tardos Algorithm: Borel version

Let $\tau \colon V \to \Lambda^{\mathbb{N}}$ be a function, which we call a table.

Instead of random sampling, we look up labels for each $x \in V$ in the corresponding string $\tau(x) = (\tau(x,0), \tau(x,1), \ldots)$.

A counter $\ell(x)$ keeps track of the position of x's current label.

If $\ell(x)$ eventually stabilizes, then we can define

$$f(x) := \tau\big(x, \text{eventual value of } \ell(x)\big).$$

If τ is Borel and f is defined everywhere, then f is a Borel solution!

MT '10: If $\tau \sim \mathbb{P}^V$, then $\forall x$, $f(x)$ exists with prob. 1. Not Borel! ☹

Finding a table that works

Goal: Find a Borel table τ such that $\mathsf{MTA}(\tau)$ converges everywhere.

Finding a table that works

Goal: Find a Borel table τ such that $\mathsf{MTA}(\tau)$ converges everywhere.

CGMPT: Partition V into "sparse" Borel sets $V_1, \ldots, V_k$ and, for each $1 \leqslant i \leqslant k$, use the same random values for all $x \in V_i$.

Finding a table that works

Goal: Find a Borel table τ such that $\mathsf{MTA}(\tau)$ converges everywhere.

CGMPT: Partition V into “sparse” Borel sets $V_1, \ldots, V_k$ and, for each $1 \leqslant i \leqslant k$, use the same random values for all $x \in V_i$.

Our approach: Reduce the problem to a different CSP, with much, much better parameters!

Finding a table that works

Key Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that there exists $s > 1$ s.t.:

- $\mathsf{egr}(D_\Pi) < s$,
- $p \cdot (e \cdot (d+1))^s < 1$.

Finding a table that works

Key Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that there exists $s > 1$ s.t.:

- $\mathsf{egr}(D_\Pi) < s$,
- $p \cdot (e \cdot (d+1))^s < 1$.

Then, for any parameter $N \in \mathbb{N}$, there exists a Borel CSP

$$\Pi_N = \left(V, \Lambda^{\mathbb{N}}, \mathfrak{C}, \mathrm{dom}', \mathscr{B}_N\right)$$

such that if $\tau \colon V \to \Lambda^{\mathbb{N}}$ is a solution to Π_N, then in the Moser–Tardos Algorithm with table τ, each variable is resampled $\leqslant N$ times ☺

Key Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that there exists $s > 1$ s.t.:

- $\mathsf{egr}(D_\Pi) < s$,
- $p \cdot (e \cdot (d+1))^s < 1$.

Then, for any parameter $N \in \mathbb{N}$, there exists a Borel CSP

$$\Pi_N = \left(V, \Lambda^{\mathbb{N}}, \mathfrak{C}, \mathrm{dom}', \mathscr{B}_N\right)$$

such that if $\tau \colon V \to \Lambda^{\mathbb{N}}$ is a solution to Π_N, then in the Moser–Tardos Algorithm with table τ, each variable is resampled $\leqslant N$ times ☺, the dependency graph of Π_N has maximum degree independent of N

Finding a table that works

Key Lemma

Suppose there exist $p \in (0,1)$, $d \in \mathbb{N}$ as in the LLL.

Assume that there exists $s > 1$ s.t.:

- $\mathsf{egr}(D_\Pi) < s$,
- $p \cdot (e \cdot (d+1))^s < 1$.

Then, for any parameter $N \in \mathbb{N}$, there exists a Borel CSP

$$\Pi_N = \left(V, \Lambda^{\mathbb{N}}, \mathfrak{C}, \mathrm{dom}', \mathscr{B}_N\right)$$

such that if $\tau : V \to \Lambda^{\mathbb{N}}$ is a solution to Π_N, then in the Moser–Tardos Algorithm with table τ, each variable is resampled $\leqslant N$ times ☺, the dependency graph of Π_N has maximum degree independent of N, and $\sup_{\mathfrak{c} \in \mathfrak{C}} \mathbb{P}[\mathscr{B}_N(\mathfrak{c})]$ goes to 0 as $N \to \infty$.

Last step

Lemma

Suppose a Borel CSP Π satisfies, for some $p \in (0,1)$ and $d \in \mathbb{N}$:

- $\mathbb{P}[B(\mathfrak{c})] \leqslant p$ for all $\mathfrak{c} \in \mathfrak{C}$,
- the maximum degree of D_Π is $\leqslant d$, and
- $\boxed{p \cdot (d+1)^{d+1} < 1}$.

Then Π has a Borel solution.

A greedy construction using method of conditional probabilities from computer science + "large section" uniformization.

Last step

Lemma

Suppose a Borel CSP Π satisfies, for some $p \in (0,1)$ and $d \in \mathbb{N}$:

- $\mathbb{P}[B(\mathfrak{c})] \leqslant p$ for all $\mathfrak{c} \in \mathfrak{C}$,
- the maximum degree of D_Π is $\leqslant d$, and
- $\boxed{p \cdot (d+1)^{d+1} < 1}$.

Then Π has a Borel solution.

A greedy construction using method of conditional probabilities from computer science + "large section" uniformization.

PROOF OF THE MAIN THEOREM:

- Construct Π_N using the **Key Lemma** for very large N.
- Find a Borel solution $\tau \colon V \to \Lambda^{\mathbb{N}}$ to Π_N using the lemma above.
- Run $\mathsf{MTA}(\tau)$ to build a Borel solution $f \colon V \to \Lambda$. ∎

Open problems

Any interesting applications with atomless prob. spaces?

Can “subexponential growth” be replaced by “amenable”?

What about continuous solutions?

Thank you!