2024.1.19 AFSA・数理CS合同セミナー@東京大学

Reconfiguration Problems, Hardness of Approximation, and Gap Amplification: What Are They?

Proc. 35th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA), 2024

Naoto Ohsaka

(CygerAgent, Inc.)

🔍 “Naoto Ohsaka” for paper link!!

Prologue: Sliding block puzzle

Klotski
Dad Puzzle
L'âne rouge
箱入り娘
华容道
…

- Complexity of reachability was open for 40 years...

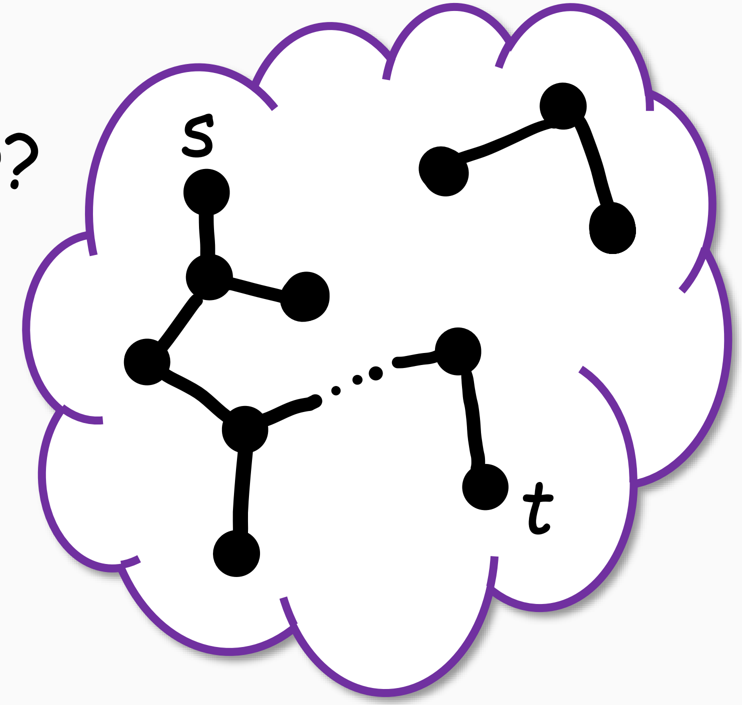
These puzzles are very much in want of a theory. Short of trial and error, no one knows how to determine if a given state is obtainable from another given state
[Martin Gardner. Scientific American 1964]

- **PSPACE**-complete [Flake-Baum. Theor. Comput. Sci. 2002]
even if only ◻◻ and ⊟ are available [Hearn-Demaine. Theor. Comput. Sci. 2005]

Reconfiguration Problems, Hardness of Approximation, and Gap Amplification: What Are They?

Intro of reconfiguration

Imagine **connecting** a pair of feasible solutions (of NP problem)
under a particular adjacency relation



Q. Is a pair of solutions reachable to each other?

Q. If so, what is the shortest transformation?

Q. If not, how can the feasibility be relaxed?

Q. Is the space of feasible solutions entirely connected?

Example 1-1

3-SAT Reconfiguration

[Gopalan-Kolaitis-Maneva-Papadimitriou. SIAM J. Comput. 2009]

- **Input:** 3-CNF formula φ & satisfying σ_s, σ_t
- **Output:** $\boldsymbol{\sigma} = \langle \sigma^{(0)}=\sigma_s, \ldots, \sigma^{(\ell)}=\sigma_t \rangle$ (reconf. sequence) s.t.
  $\sigma^{(i)}$ satisfies φ (feasibility)
  $\mathrm{Ham}(\sigma^{(i-1)}, \sigma^{(i)}) = 1$ (adjacency on hypercube)

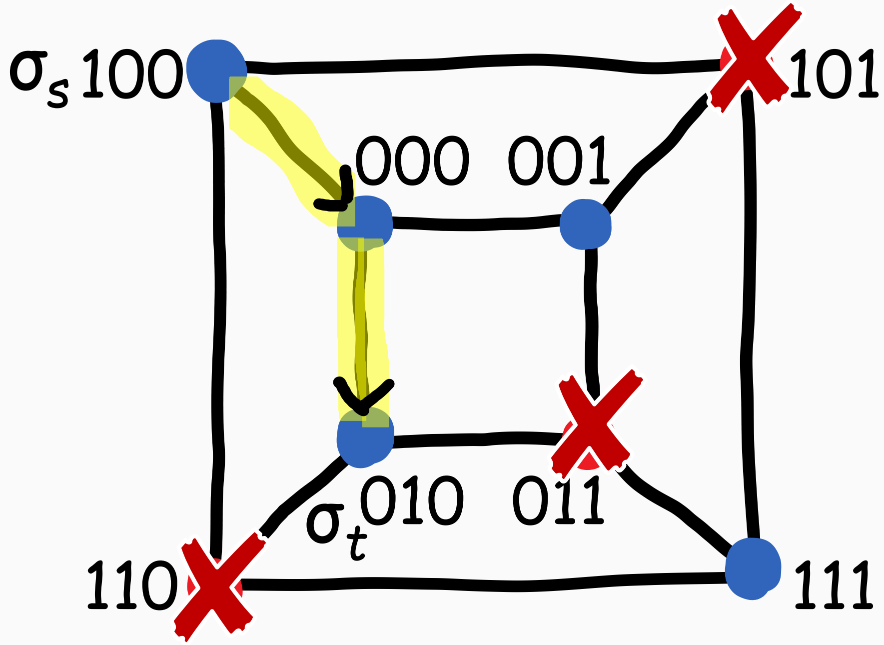
YES case

$\varphi = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z})$

$\sigma_s = (1,0,0)$

$\sigma_t = (0,1,0)$

⚠ Length of $\boldsymbol{\sigma}$ can be $2^{\Omega(\text{input size})}$



Example 1-2

3-SAT Reconfiguration

[Gopalan-Kolaitis-Maneva-Papadimitriou. SIAM J. Comput. 2009]

- **Input:** 3-CNF formula φ & satisfying σ_s, σ_t
- **Output:** $\boldsymbol{\sigma} = \langle \sigma^{(0)}=\sigma_s, \ldots, \sigma^{(\ell)}=\sigma_t \rangle$ (reconf. sequence) s.t.
  $\sigma^{(i)}$ satisfies φ (feasibility)
  $\mathrm{Ham}(\sigma^{(i-1)}, \sigma^{(i)}) = 1$ (adjacency on hypercube)

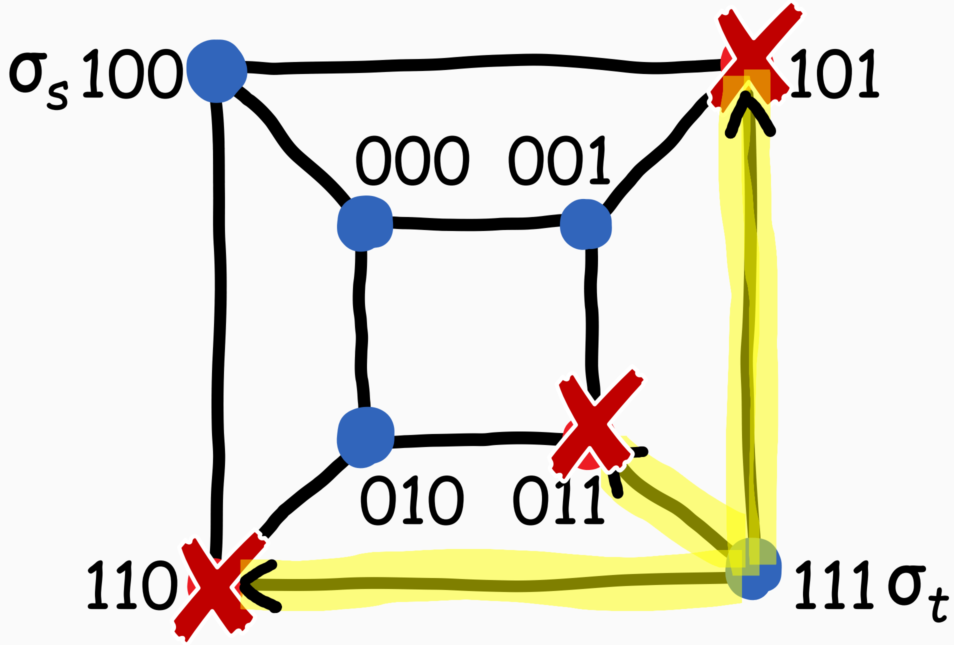
NO case

$\varphi = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z})$

$\sigma_s = (1,0,0)$

$\sigma_t = (1,1,1)$

⚠ Length of $\boldsymbol{\sigma}$ can be $2^{\Omega(\text{input size})}$



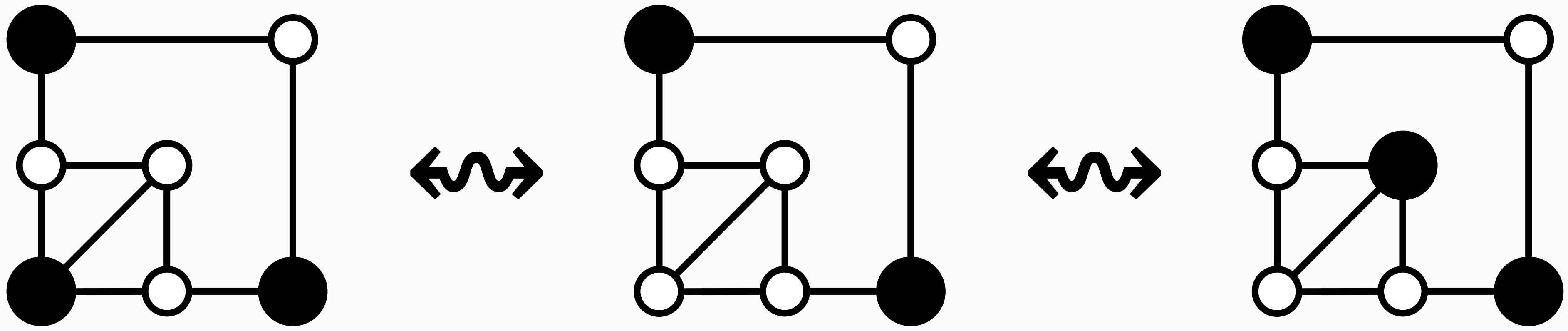
Example 2-1

Independent Set Reconfiguration

[Hearn-Demaine. Theor. Comput. Sci. 2005]

- **Input:** Graph G & independent sets I_s, I_t of size k
- **Output:** $\mathcal{J} = \langle I^{(0)}=I_s, \ldots, I^{(\ell)}=I_t \rangle$ (reconf. sequence) s.t.
  $I^{(i)}$ is independent & $|I^{(i)}| \geq k-1$ (feasibility)
  $|I^{(i-1)} \Delta I^{(i)}| = 1$ (adjacency called token-addition-removal)

YES case (k=3)



I_s I_t

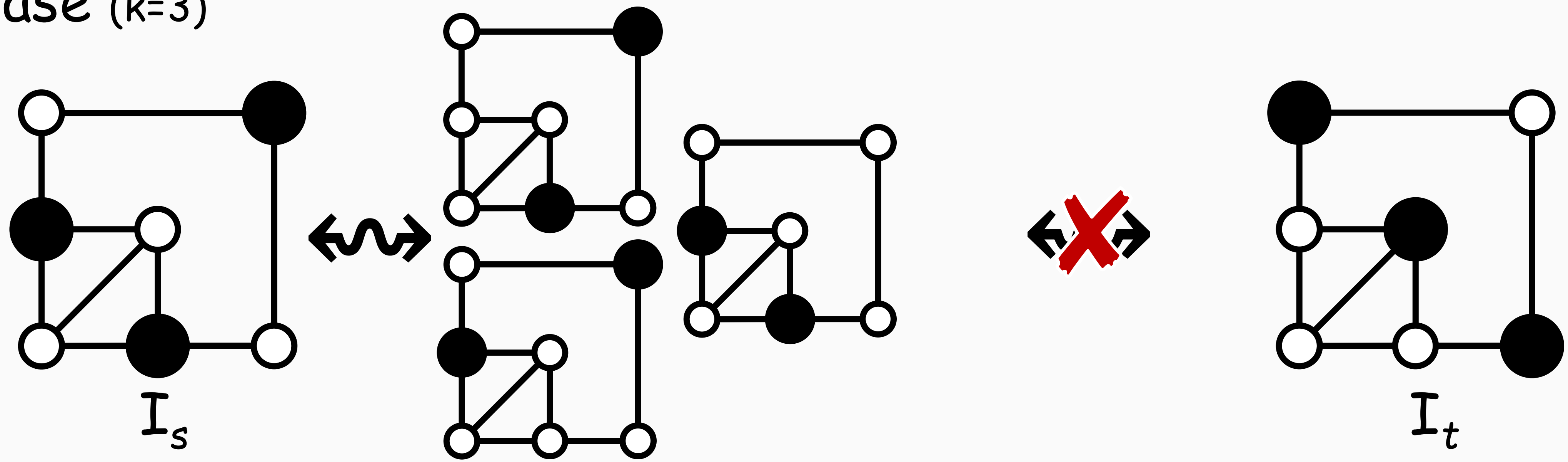
Example 2-2

Independent Set Reconfiguration

[Hearn-Demaine. Theor. Comput. Sci. 2005]

- **Input:** Graph G & independent sets I_s, I_t of size k
- **Output:** $\mathscr{I} = \langle I^{(0)}{=}I_s, \ldots, I^{(\ell)}{=}I_t \rangle$ (reconf. sequence) s.t.
  $I^{(i)}$ is independent & $|I^{(i)}| \geq k\text{-}1$ (feasibility)
  $|I^{(i-1)} \, \Delta \, I^{(i)}| = 1$ (adjacency called token-addition-removal)

NO case (k=3)



Recipe for defining reconfiguration problems

[Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

1. Source problem in NP

- Ask the existence of a feasible solution

  E.g., satisfying assignments; independent sets

2. Transformation rule

- Define a (symmetric) adjacency relation btw. a pair of solutions

  E.g., single assignment flip; addition or removal of a single vertex

Many reconfiguration problems derived from

Satisfiability, Coloring, Vertex Cover, Clique, Dominating Set, Feedback Vertex Set, Steiner Tree, Matching, Spanning Tree, Shortest Path, Set Cover, Subset Sum, …

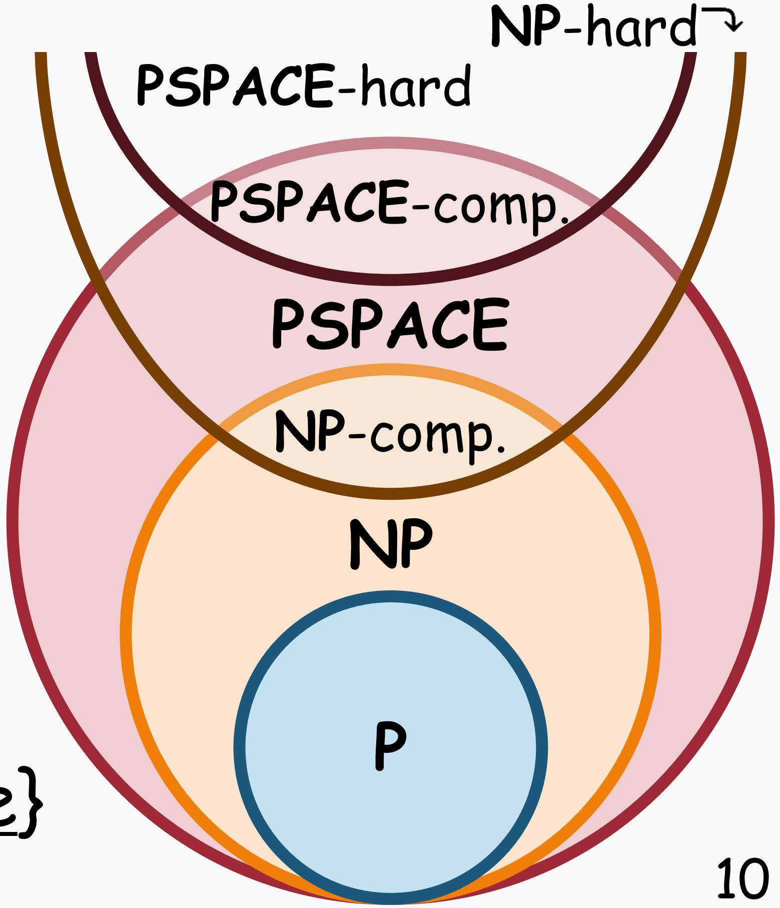
See [Nishimura. Algorithms 2018] [van den Heuvel. Surv. Comb. 2013] [Hoang. https://reconf.wikidot.com/]

What we want to do in CS Theory

Elucidate the computational complexity of reconfiguration problems

Q. How much resources are required (w.r.t. the input size) ?

time, space, randomness,
gates, nondeterminism, …



- P $\stackrel{\text{def}}{=}$ {probs. solvable in polynomial time}
- NP $\stackrel{\text{def}}{=}$ {probs. solvable in polynomial time given a polynomial-length witness}
- PSPACE $\stackrel{\text{def}}{=}$ {probs. solvable in polynomial space}

Complexity of reconfiguration problems

Source problem	Existence	Reconfiguration
Satisfiability	**NP**-complete	**PSPACE**-complete [Gopalan-Kolaitis-Maneva-Papadimitriou. SIAM J. Comput. 2009]
Independent Set	**NP**-complete	**PSPACE**-complete [Hearn-Demaine. Theor. Comput. Sci. 2005]
Matching	**P**	**P** [Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]
3-Coloring	**NP**-complete	**P** [Cereceda-van den Heuvel-Johnson. J. Graph Theory 2011]
Shortest Path	**P**	**PSPACE**-complete [Bonsma. Theor. Comput. Sci. 2013]
Independent Set on bipartite graphs	**P**	**NP**-complete [Lokshtanov-Mouawad. ACM Trans. Algorithms 2019; SODA 2018]

👆 **Nontrivial relation**

A personal motivation

"NATURAL" **PSPACE**-complete problems

- Connecting a pair of feasible solutions is a reasonable idea
- Simulating a (polynomial-space) nondeterministic Turing machine
  - ⚠ Quantified Boolean Formula is another **PSPACE**-complete problem

    $\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n\ \varphi(x_1, x_2, x_3, \ldots, x_n)$?
- Easily derived from **NP** problems

BLUE OCEAN…? (at least for hardness of approximation)

Reconfiguration Problems, Hardness of Approximation, and Gap Amplification: What Are They?

Gap Preserving Reductions Between Reconfiguration Problems

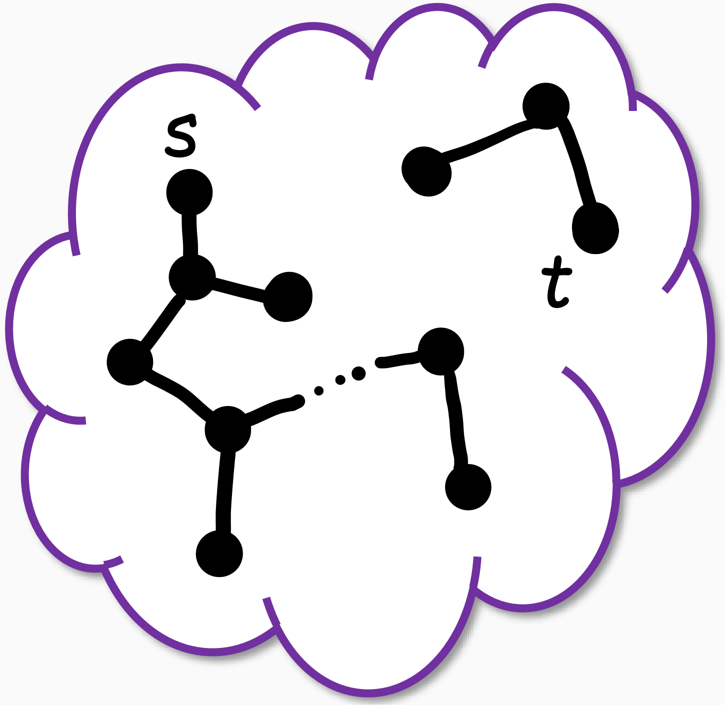
Naoto Ohsaka

40th Int. Symp. on Theoretical Aspects of Computer Science (STACS), 2023

Optimization versions of reconfiguration problems

Even if…

- 😖 NOT reconfigurable! and/or
- 😖 many problems are **PSPACE**-complete!



Still want an "approximate" reconf. sequence
(e.g.) made up of almost-satisfying assignments
or not-too-small independent sets

⬇

Let's **RELAX** feasibility!!

Example 1+

Maxmin 3-SAT Reconfiguration

[Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

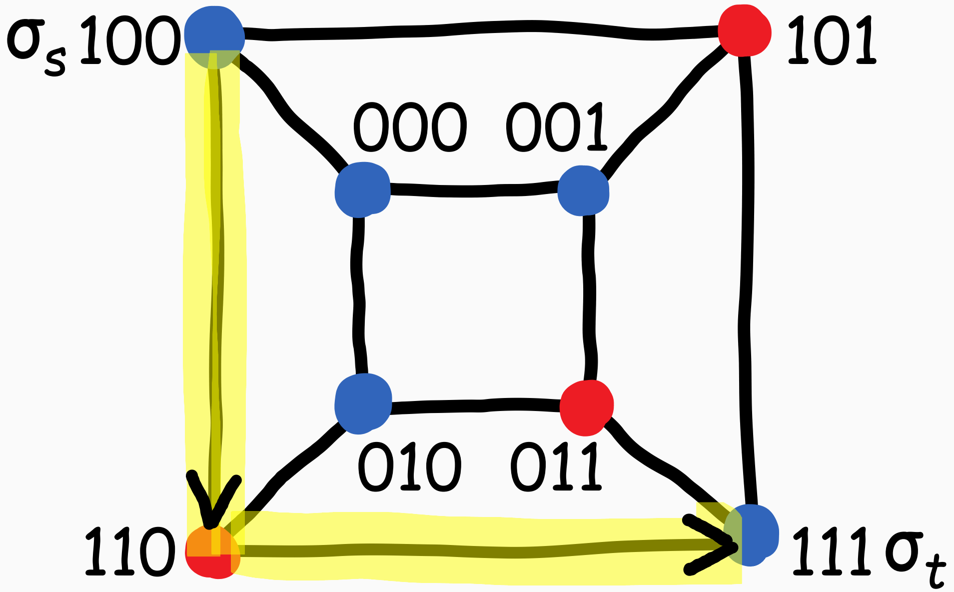
- **Input:** 3-CNF formula φ & satisfying σ_s, σ_t
- **Output:** $\boldsymbol{\sigma} = \langle \sigma^{(0)}=\sigma_s, \ldots, \sigma^{(\ell)}=\sigma_t \rangle$ (reconf. sequence) s.t.
  ~~$\sigma^{(i)}$ satisfies φ~~ (feasibility)
  $\mathrm{Ham}(\sigma^{(i-1)}, \sigma^{(i)}) = 1$ (adjacency on hypercube)
- **Goal:** $\max_{\boldsymbol{\sigma}} \mathrm{val}_\varphi(\boldsymbol{\sigma}) \stackrel{\text{def}}{=} \min_i$ (frac. of satisfied clauses by $\sigma^{(i)}$)

$\varphi = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z})$

- $\sigma_s = (1,0,0)$
- $\sigma_t = (1,1,1)$

→ $\mathrm{val}_\varphi(\boldsymbol{\sigma}) = \min\{1, \frac{2}{3}, 1\} = \frac{2}{3}$

⚠ Length of $\boldsymbol{\sigma}$ can be $2^{\Omega(\text{input size})}$

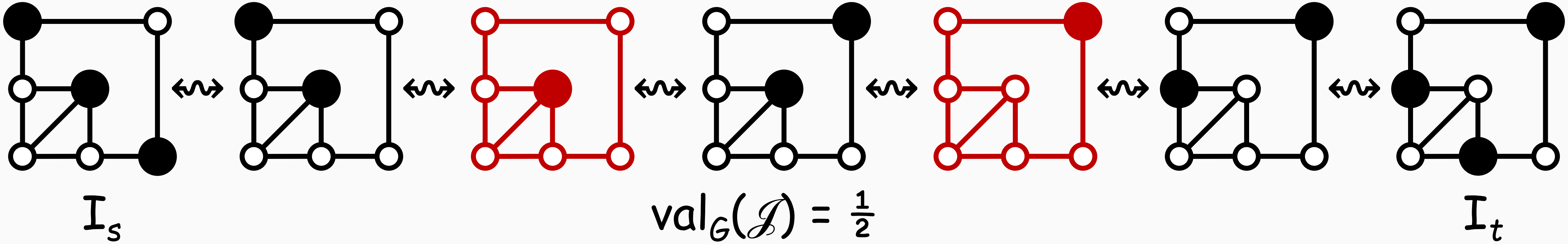


Example 2+

Maxmin Independent Set Reconfiguration

[Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

- **Input:** Graph G & independent sets I_s, I_t of size k
- **Output:** $\mathscr{I} = \langle I^{(0)}=I_s, \ldots, I^{(\ell)}=I_t \rangle$ (reconf. sequence) s.t.
  $I^{(i)}$ is independent $\cancel{\& \; |I^{(i)}| \geq k-1}$ (feasibility)
  $|I^{(i-1)} \,\Delta\, I^{(i)}| = 1$ (adjacency called token-addition-removal)
- **Goal:** $\max_{\mathscr{I}} \mathrm{val}_G(\mathscr{I}) \stackrel{\text{def}}{=} \min_i \dfrac{|I^{(i)}|}{k-1}$



Questions of interest about approximate reconfiguration

Algorithmic side

- How well can we approximate reconfiguration problems?

  Set Cover Reconf.
  [Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

  Subset Sum Reconf. [Ito-Demaine. J. Comb. Optim. 2014]

  Submodular Reconf. [O.-Matsuoka. WSDM 2022]

Hardness side

- How hard is it to approximate reconfiguration problems?

  👆**My interest** [STACS 2023 & SODA 2024]

Known results on hardness of approximation

NP-hardness of approx. for Maxmin SAT & Ind. Set Reconf.

[Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

- Not optimal $\because$ SAT Reconf. & Ind. Set Reconf. are **PSPACE**-comp.
- Rely on **NP**-hardness of approximating Max SAT & Max Ind. Set

5. Open problems

There are many open problems raised by this work, and we mention some of these below:

- Can the MATCHING RECONFIGURATION problem for edge-weighted graphs be solved also in polynomial time? We conjecture that the answer is positive.
- Is the TRAVELING SALESMAN RECONFIGURATION problem (where two tours are adjacent if they differ in two edges) PSPACE-complete?
- Are there better approximation algorithms for the MINMAX POWER SUPPLY RECONFIGURATION problem? Lower bounds?
- Are the problems in Section 4 PSPACE-hard to approximate (not just NP-hard)?

Known results on hardness of approximation

NP-hardness of approx. for Maxmin SAT & Ind. Set Reconf.

[Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

- Not optimal $\because$ SAT Reconf. & Ind. Set Reconf. are **PSPACE**-comp.
- Rely on **NP**-hardness of approximating Max SAT & Max Ind. Set

Significance of showing **PSPACE**-hardness

- no polynomial-time algorithm (**P** $\neq$ **PSPACE**)
- no polynomial-length sequence (**NP** $\neq$ **PSPACE**)

(probabilistically checkable proof)

Reconfiguration analogue of the PCP theorem

[Arora-Lund-Motwani-Sudan-Szegedy. J. ACM 1998] [Arora-Safra. J. ACM 1998]

Our working hypothesis [O. STACS 2023]

Reconfiguration Inapproximability Hypothesis (**RIH**)

Binary CSP G & satisfying ψ_s, ψ_t, **PSPACE**-hard to distinguish btw.

- (Completeness) $\exists \psi \; \mathrm{val}_G(\psi) = 1$ (some sequence violates no constraint)
- (Soundness) $\forall \psi \; \mathrm{val}_G(\psi) < 1\text{-}\varepsilon$ (any sequence violates >ε-frac. of constraints)

Binary CSP (Constraint Satisfaction Problem) in short:
Given a constraint system over variable pairs
Color each variable to satisfy as many constraints as possible
E.g., 3-Coloring & 2-SAT

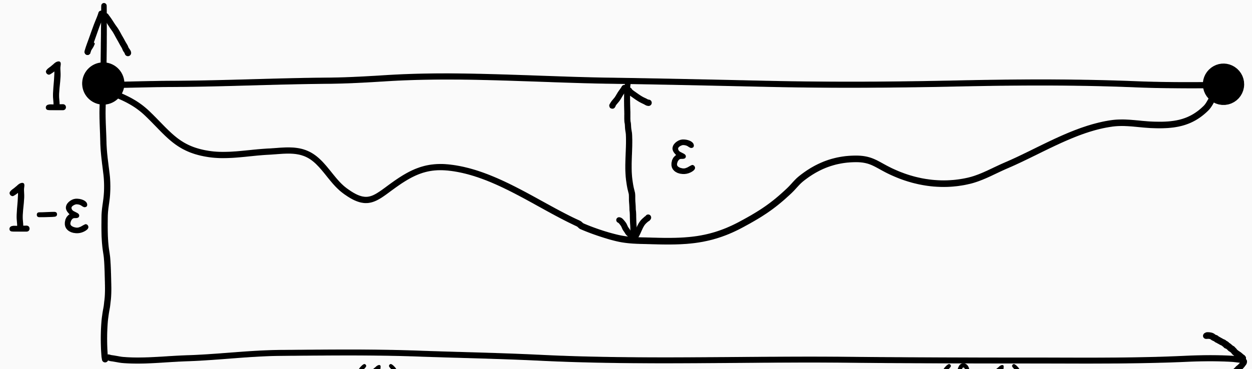
Q. Which reconfiguration problems are
PSPACE-hard to approximate under (seemingly) plausible **RIH**?

Our working hypothesis [O. STACS 2023]

Reconfiguration Inapproximability Hypothesis (**RIH**)

Binary CSP G & satisfying ψ_s, ψ_t, **PSPACE**-hard to distinguish btw.

- (Completeness) $\exists\psi\ \mathrm{val}_G(\psi) = 1$ (some sequence violates no constraint)
- (Soundness) $\forall\psi\ \mathrm{val}_G(\psi) < 1\text{-}\varepsilon$ (any sequence violates >ε-frac. of constraints)



$\langle \psi_s,\ \psi^{(1)},\ \ldots,\ \psi^{(\ell-1)},\ \psi_t \rangle$

Q. Which reconfiguration problems are **PSPACE**-hard to approximate under (seemingly) plausible **RIH**?

Our (previous) results [O. STACS 2023]

- 😋Under **RIH**, many problems are **PSPACE**-hard to approximate

How? <u>Gap-preserving reductions</u>!!

$\exists\psi \;\; \mathrm{val}_G(\psi) = 1$

(Perfect) Completeness⤴

Soundness⤵

$\forall\psi \;\; \mathrm{val}_G(\psi) < 1\text{-}\varepsilon$

polynomial-time transformation

$\exists\sigma \;\; \mathrm{val}_\varphi(\sigma) = 1$

$\forall\sigma \;\; \mathrm{val}_\varphi(\sigma) < 1\text{-}\delta$

Gap[1 vs. 1-ε] Binary CSP Reconf.

PROMISE: $\varepsilon \in (0,1)$ is const.

Gap[1 vs. 1-δ] 3-SAT Reconf.

😋$\delta \in (0,1)$ depends only on ε

Related work

Probabilistically checkable debate systems
[Condon-Feigenbaum-Lund-Shor. Chic. J. Theor. Comput. Sci. 1995]

- PCP-like charact. of **PSPACE**
- $\Longrightarrow$ Quantified Boolean Formula is **PSPACE**-hard to approx.

Other optimization variants of reconfiguration (orthogonal to this study)

- Shortest sequence
  [Bonamy-Heinrich-Ito-Kobayashi-Mizuta-Mühlenthaler-Suzuki-Wasa. STACS 2020]
  [Ito-Kakimura-Kamiyama-Kobayashi-Okamoto. SIAM J. Discret. Math. 2022]
  [Kamiński-Medvedev-Milanič. Theor. Comput. Sci. 2011]
  [Miltzow-Narins-Okamoto-Rote-Thomas-Uno. ESA 2016]
- Incremental optimization
  [Blanché-Mizuta-Ouvrard-Suzuki. IWOCA 2020]
  [Ito-Mizuta-Nishimura-Suzuki. J. Comb. Optim. 2022]
  [Yanagisawa-Suzuki-Tamura-Zhou. COCOON 2021]

Reconfiguration Problems, Hardness of Approximation, and **Gap Amplification**: What Are They?

Gap Amplification for Reconfiguration Problems*

Naoto Ohsaka†

Proc. 35th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA), 2024

Limitation of [O. STACS 2023]

😖 Inapprox. factors are not explicitly shown

Recall from [O. STACS 2023]

- **RIH** claims "$\exists \varepsilon > 0$, Gap[1 vs. 1-ε] Binary CSP Reconf. is **PSPACE**-h."
- Can reduce to $\boxed{\text{Gap[1 vs. 1-}\delta\text{]}}$ ∗∗ Reconf.

⚠ δ (as well as ε) can be arbitrarily small, because...

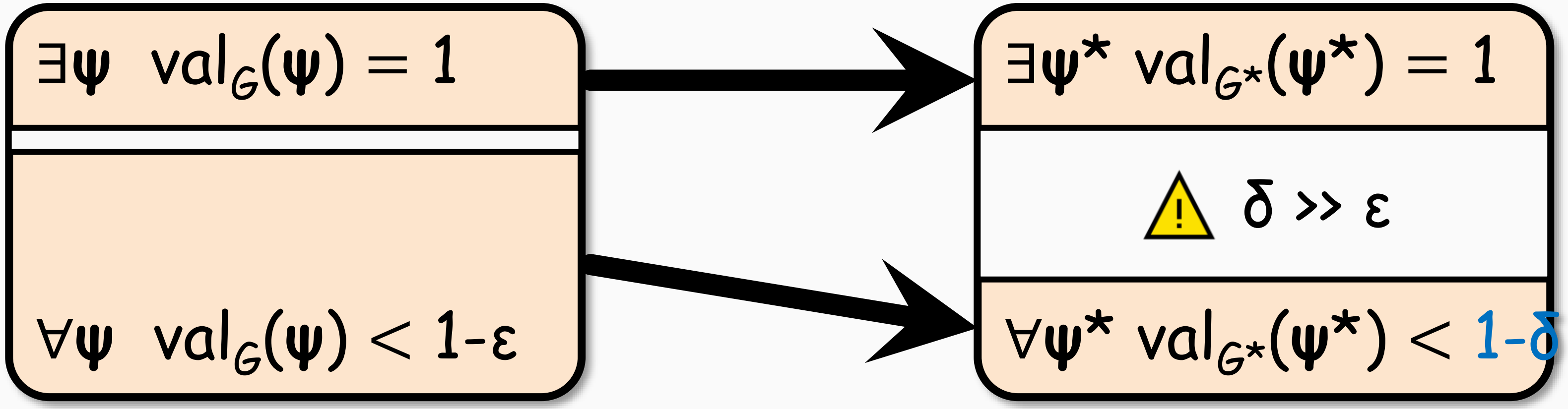
- δ depends on ε (e.g., $\delta = \varepsilon^2$)
- **RIH** doesn't specify any value of ε (e.g., $\varepsilon = 1/2^{10000}$)

→ May not rule out 0.999⋯999-approx. for ∗∗ Reconf.

🎯 Gap[1 vs. 0.999] ∗∗ Reconf. is **PSPACE**-hard only assuming **RIH**

🎯Our target: **Gap amplification**

- (Polynomial-time) reduction that makes a tiny gap into a larger gap



In **NP** world...

The parallel repetition theorem [Raz. SIAM J. Comput. 1998]

➜😋Gap[1 vs. 0.000⋯001] Binary CSP is **NP**-hard (i.e. gap ≈ 1)

In ***reconfiguration*** world...

😖Naïve parallel repetition fails to amplify gap ε of Gap[1 vs. 1-ε] Binary CSP Reconf. [O. arXiv 2023]

🎯Our target: **Gap amplification**

- (Polynomial-time) reduction that makes a ti... p into a larger gap

$\exists w$ $\mathrm{val}_G(w)$ ... * val... = 1

Can we derive explicit factors of
PSPACE-hardness of approx.
only assuming **RIH**?

...s **NP**-ha... (i.e. gap ≈ 1)

In **reconfigur...ion** world...

😖Naïve parallel repetition fails to amplify gap ε of
Gap[1 vs. 1-ε] Binary CSP Reconf. [O. arXiv 2023]

Our results [O. SODA 2024]

😋Can derive explicit inapproximability factors only assuming **RIH**!!

	Maxmin Binary CSP Reconfiguration	Minmax Set Cover Reconfiguration
PSPACE-hardness under **RIH**	**0.9942** (this paper)	**1.0029** (this paper)
NP-hardness rely on parallel repetition theorem [Raz. SIAM J. Comput. 1998]	>0.75 (this paper) 0.993 [Ito et al. Theor. Comput. Sci. 2011] [O. STACS 2023]	1.0029 (this paper)
approximability	≈0.25 [O. arXiv 2023]	2 [Ito et al. Theor. Comput. Sci. 2011]

Main result [O. SODA 2024]

Gap amplification for Binary CSP Reconf.

- We prove gap amplification à la Dinur [Dinur. J. ACM 2007]

(Informal) For any small const. $\varepsilon \in (0,1)$,

gap	alphabet size	degree	spectral expansion
1 vs. $1-\varepsilon$	W	d	λ
1 vs. $1-0.0058$	$W^* = W^{d^{O(\varepsilon^{-1})}}$	$d^* = \left(\frac{d}{\varepsilon}\right)^{O(\varepsilon^{-1})}$	$\lambda^* = O\left(\frac{\lambda}{d}\right) d^*$

- 😋 Can make λ^*/d^* arbitrarily small by decreasing λ/d
- 😖 Alphabet size W^* gets gigantic depending on ε^{-1}

Application [O. SODA 2024]

Inapprox. of Minmax Set Cover Reconf.

- **PSPACE**-hard to approx. within 1.0029 under **RIH**
- 2-approximation is known [Ito-Demaine-Harvey-Papadimitriou-Sideri-Uehara-Uno. Theor. Comput. Sci. 2011]

(Informal) Gap-preserving reduction from Gap[1, ε] Binary CSP Reconf. (with small λ/d) to Gap[1, $\approx 2-\sqrt{\varepsilon}$] Set Cover Reconf.

- Based on [Lund-Yannakakis. J. ACM 1994] but expander mixing lemma [Alon-Chung. Discret. Math. 1988] is needed

BREAK: Why is it accepted to SODA?
(from my personal point of view)

- (Of course) I was lucky... 190/652≈29%
- Open up the hardness-of-approximation for reconf. problems
- **RIH** seems to be considered somewhat important (within review)
- Nontrivial extension of Dinur's gap ampl. [Dinur. J. ACM 2007] to reconf.
  From arbitrarily small gap to universal const.

- Demonstrate usefulness of alphabet squaring trick [O. STACS 2023]
  (explained later)

In the remainder of this talk…

Proof sketch of gap amplification

1. Preprocessing step

- Degree reduction [O. STACS 2023]
- Expanderization (skipped)

2. Powering step

- Simple appl. of [Dinur. J. ACM 2007] [Radhakrishnan. ICALP 2006] to Binary CSP Reconf. looses perfect completeness
- TRICK: **Alphabet squaring** [O. STACS 2023] & **modified verifier**

Recap: Max Binary CSP

- **Input:** Binary CSP $G = (V,E,\Sigma,\Pi=(\pi_e)_{e\in E})$, where $\pi_e \subseteq \Sigma^2$
- **Output:** $\psi: V \to \Sigma$

  ψ satisfies (v,w) if $(\psi(v), \psi(w)) \in \pi_{(v,w)}$
- **Goal:** $\max_\psi \mathrm{val}_G(\psi) \stackrel{\text{def}}{=}$ (frac. of edges satisfied by ψ)

Example

- 3-Coloring: $\Sigma = \{R,G,B\}$, $\pi_e = \{(R,G), (G,R), (G,B), (B,G), (B,R), (R,B)\}$
- 2-SAT: $\Sigma = \{0,1\}$, $\pi_C = \{$asgmt. satisfying 2-literal clause $C\}$

Recap: Dinur's powering, in a nutshell

[Dinur. J. ACM 2007]

🎯**Two goals:**

(Completeness) $\exists\psi \ \mathrm{val}_G(\psi) = 1 \quad \Longrightarrow \quad \exists\psi^* \ \mathrm{val}_{G^*}(\psi^*) = 1$

(Soundness) $\forall\psi \ \mathrm{val}_G(\psi) < 1\text{-}\varepsilon \quad \Longrightarrow \quad \forall\psi^* \ \mathrm{val}_{G^*}(\psi^*) < 1 - \Omega(T\cdot\varepsilon)$

const. parameter

How? Virtually examine T edges simultaneously:

- **1**. Each vertex has "opinions" about the color of all vertices

  for simplicity

- **2**. Sample a length-T random walk **W** with endpoints x & y
- **3**. Constraint & agreement test over opinions of x & y along with **W**

Recap: Dinur's powering [Dinur. J. ACM 2007]

Graph construction

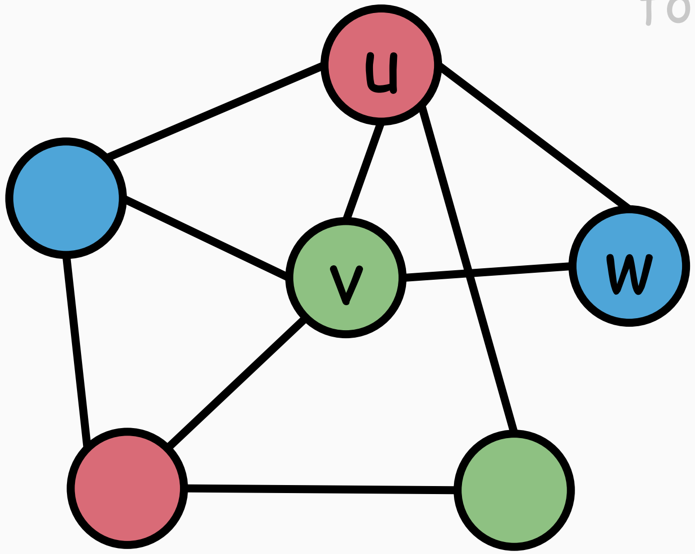
Say 3-Coloring $\Sigma=\{R,G,B\}$

Original $G = (V,E,\Sigma,\Pi=(\pi_e)_{e\in E})$ ➜ New $G^* = (V,E^*,\Sigma^*,\Pi^*)$

⚠ G must be EXPANDER

Asgmt. ψ: $V\to\Sigma$ ➜ Asgmt. ψ^*: $V\to\Sigma^V$

for simplicity



- $\psi^*(x)[v] \overset{\text{def}}{=}$ "opinion" of x about the color of v
- edge of G^* = a length-T random walk over G

const. parameter

Recap: Dinur's powering [Dinur. J. ACM 2007]

Graph construction

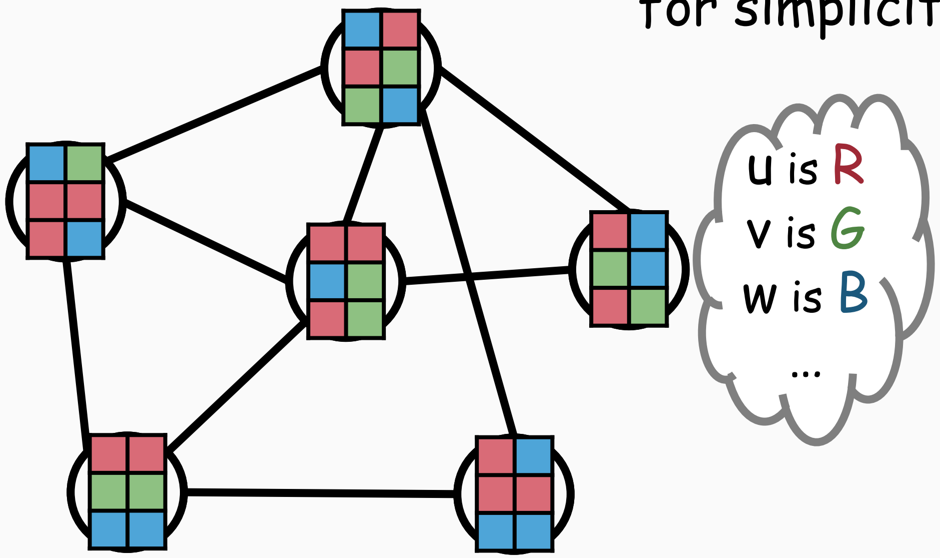
Say 3-Coloring $\Sigma=\{R,G,B\}$

Original $G = (V,E,\Sigma,\Pi=(\pi_e)_{e\in E})$ ➜ New $G^* = (V,E^*,\Sigma^*,\Pi^*)$

⚠ G must be EXPANDER

Asgmt. ψ: $V\to\Sigma$ ➜ Asgmt. ψ^*: $V\to\Sigma^V$

for simplicity



- $\psi^*(x)[v] \stackrel{\text{def}}{=}$ "opinion" of x about the color of v
- edge of G^* = a length-T random walk over G

const. parameter

Recap: Dinur's powering [Dinur. J. ACM 2007]

Graph construction

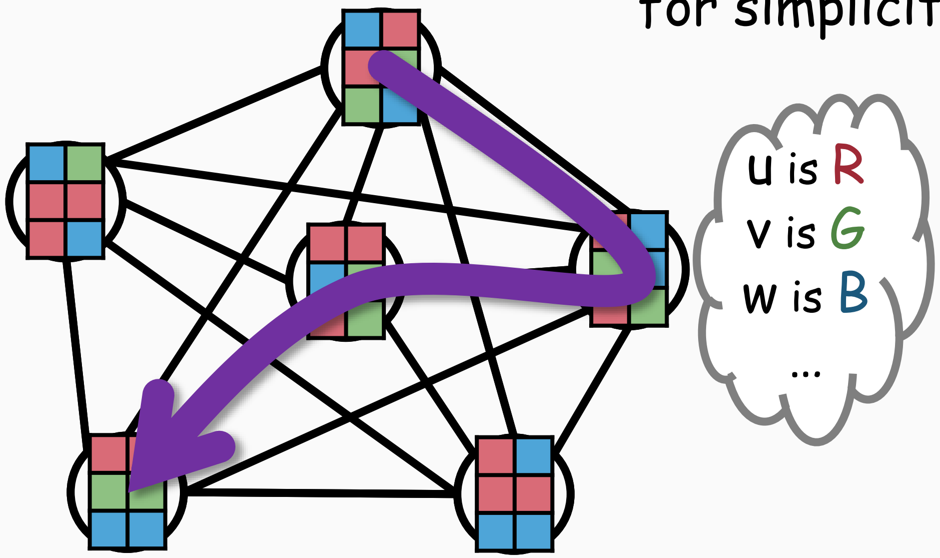
Say 3-Coloring $\Sigma=\{R,G,B\}$

Original $G = (V,E,\Sigma,\Pi=(\pi_e)_{e\in E})$ ➜ New $G^* = (V,E^*,\Sigma^*,\Pi^*)$

⚠ G must be EXPANDER

Asgmt. ψ: $V\to\Sigma$ ➜ Asgmt. ψ^*: $V\to\Sigma^V$

for simplicity



- $\psi^*(x)[v] \stackrel{\text{def}}{=}$ "opinion" of x about the color of v
- edge of G^* = a length-T random walk over G

const. parameter

Recap: Dinur's powering [Dinur. J. ACM 2007]

Verifier's test on G^* (1) [Radhakrishnan. ICALP 2006]

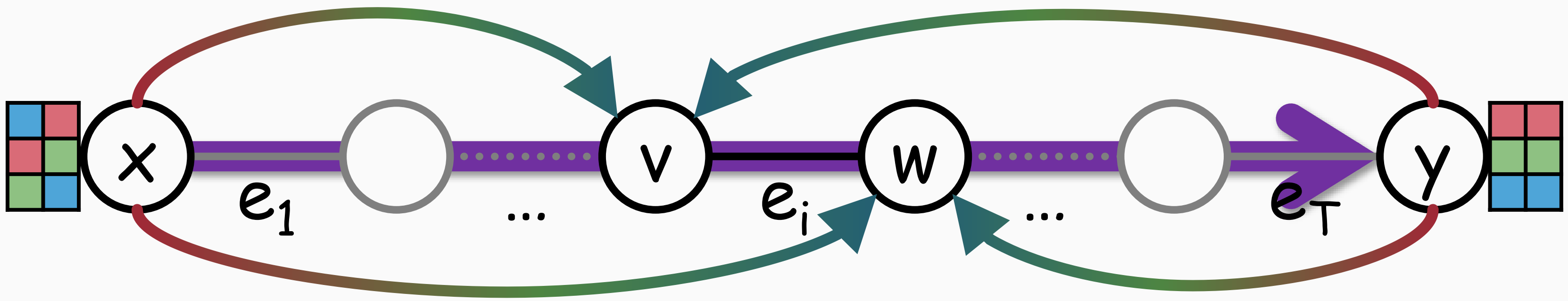
Pick a random walk $\mathbf{W} = \langle e_1, \ldots, e_T \rangle$ from x to y

$\psi^*(x)$ & $\psi^*(y)$ pass the test at $e_i = (v,w)$ if

x & y agree on color of (v,w)

opinions about (v,w) satisfy $\pi_{(v,w)}$

ψ^* satisfies $\mathbf{W}$ $\overset{\text{def}}{\iff}$ $\psi^*(x)$ & $\psi^*(y)$ pass test at every edge in $\mathbf{W}$



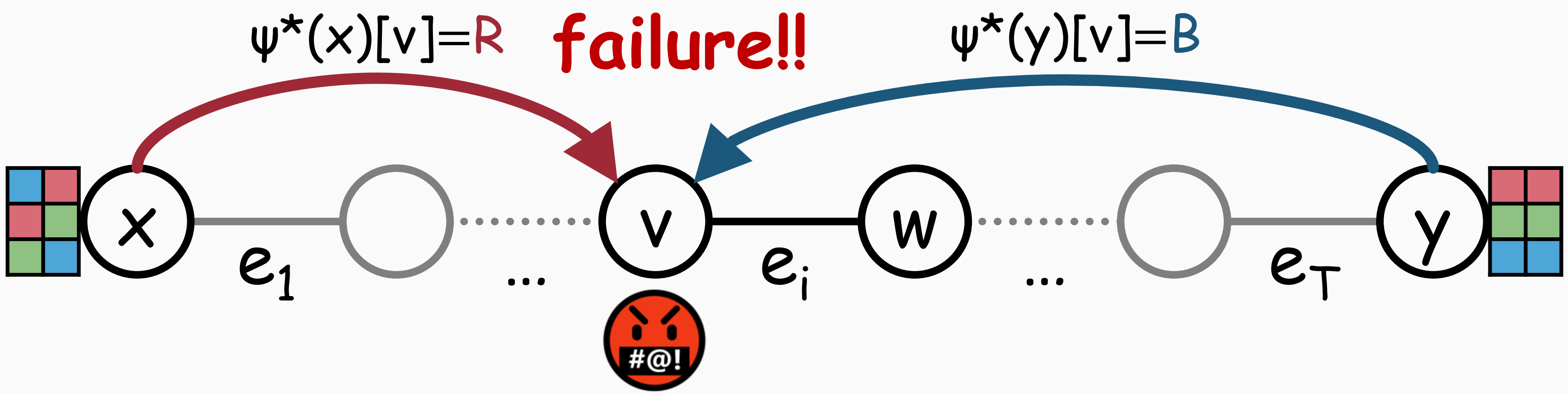
Recap: Dinur's powering [Dinur. J. ACM 2007]

Verifier's test on G* (2) [Radhakrishnan. ICALP 2006]

Pick a random walk **W** $= \langle e_1, \ldots, e_T \rangle$ from x to y

ψ*(x) & ψ*(y) <u>pass the test</u> at $e_i = (v,w)$ if

- ψ*(x)[v] = ψ*(y)[v]
- ψ*(x)[w] = ψ*(y)[w]
- (ψ*(x)[v], ψ*(x)[w]) satisfies e_i



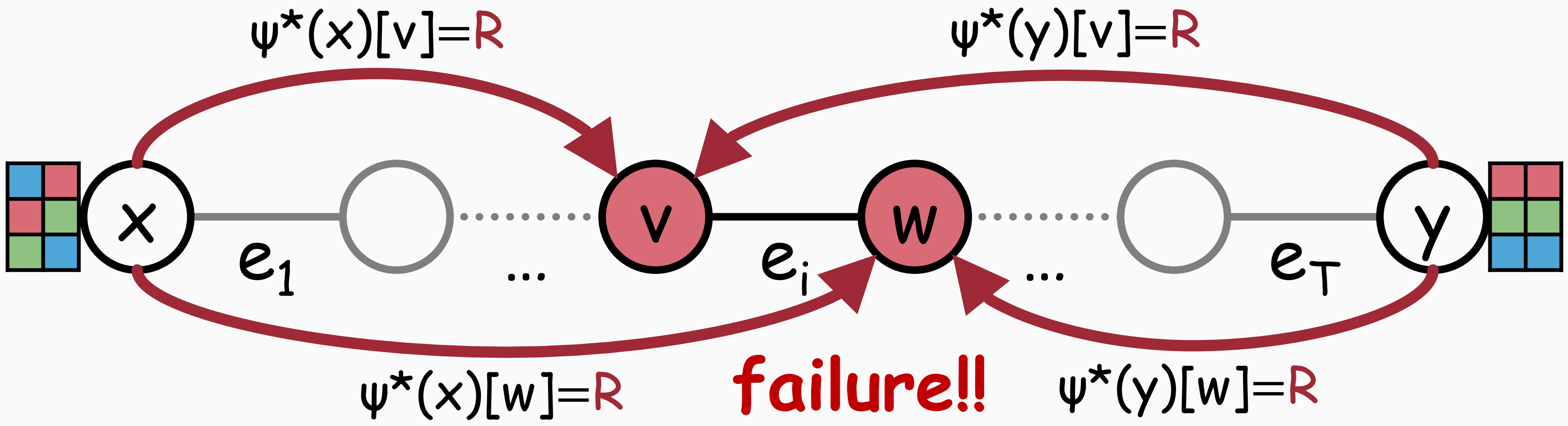
Recap: Dinur's powering [Dinur. J. ACM 2007]

Verifier's test on G* (3) [Radhakrishnan. ICALP 2006]

Pick a random walk **W** $= \langle e_1, \ldots, e_T \rangle$ from x to y

ψ*(x) & ψ*(y) <u>pass the test</u> at $e_i = (v,w)$ if

- ψ*(x)[v] = ψ*(y)[v]
- ψ*(x)[w] = ψ*(y)[w]
- (ψ*(x)[v], ψ*(x)[w]) satisfies e_i



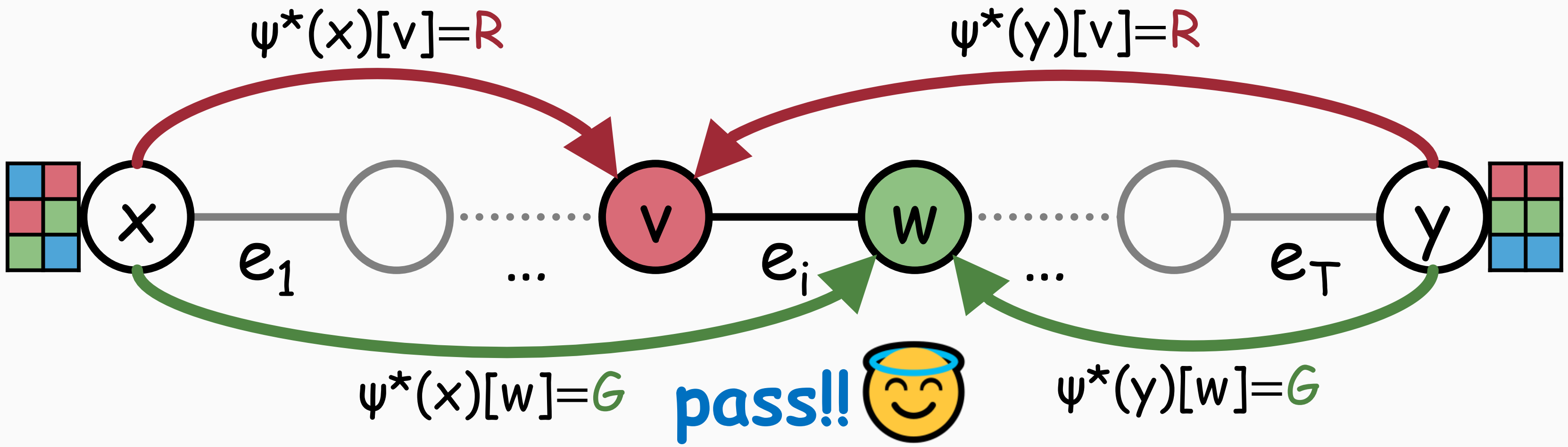
Recap: Dinur's powering [Dinur. J. ACM 2007]

Verifier's test on G* (4) [Radhakrishnan. ICALP 2006]

Pick a random walk **W** $= \langle e_1, \ldots, e_T \rangle$ from x to y

ψ*(x) & ψ*(y) <u>pass the test</u> at $e_i = (v,w)$ if

- ψ*(x)[v] = ψ*(y)[v]
- ψ*(x)[w] = ψ*(y)[w]
- (ψ*(x)[v], ψ*(x)[w]) satisfies e_i

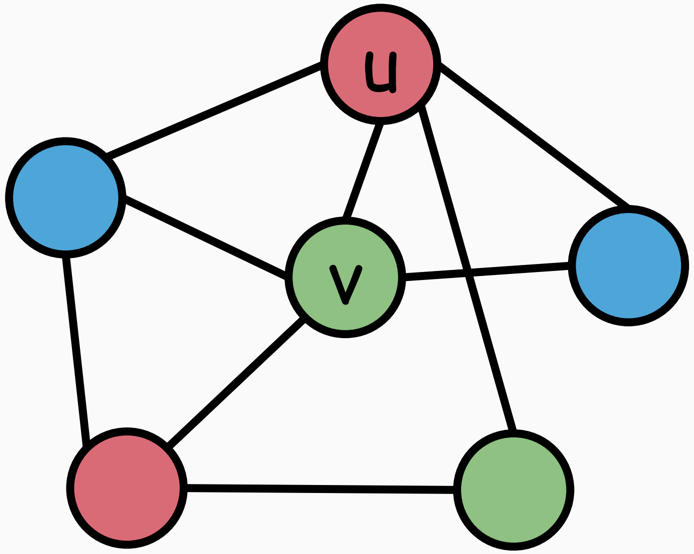


Recap: Dinur's powering [Dinur. J. ACM 2007]

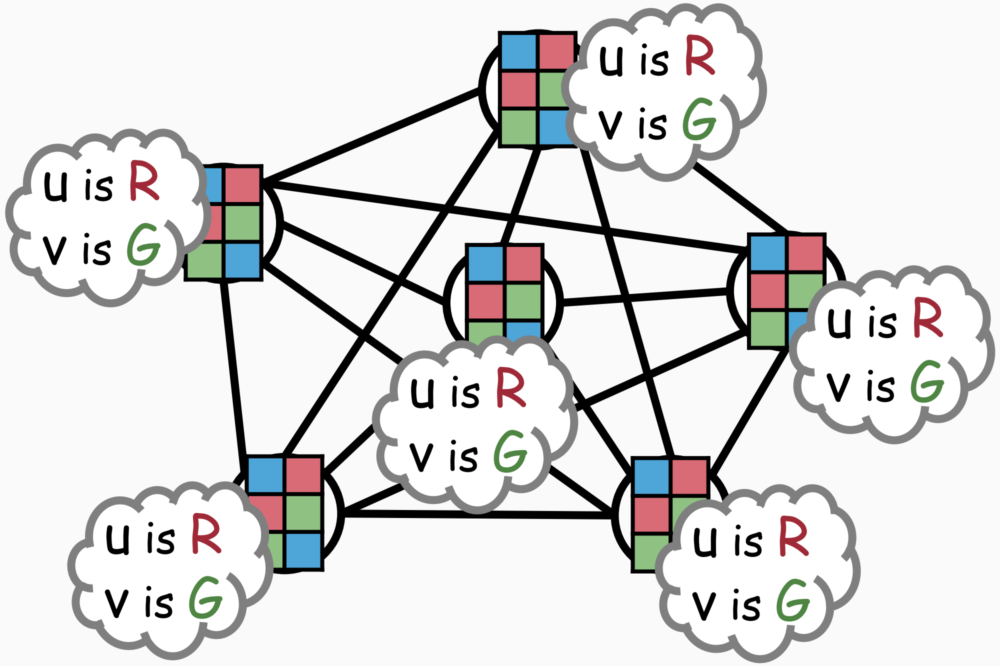
Completeness side

🎯**Goal:** $\exists\psi\ \mathrm{val}_G(\psi)=1 \implies \exists\psi^*\ \mathrm{val}_{G^*}(\psi^*)=1$

Optimal $\psi: V\to\Sigma$ ➜ let $\psi^*(x)[v]=\psi^*(y)[v]=\cdots=\psi(v)$



➜



Recap: Dinur's powering [Dinur. J. ACM 2007]

Soundness side [Radhakrishnan. ICALP 2006]

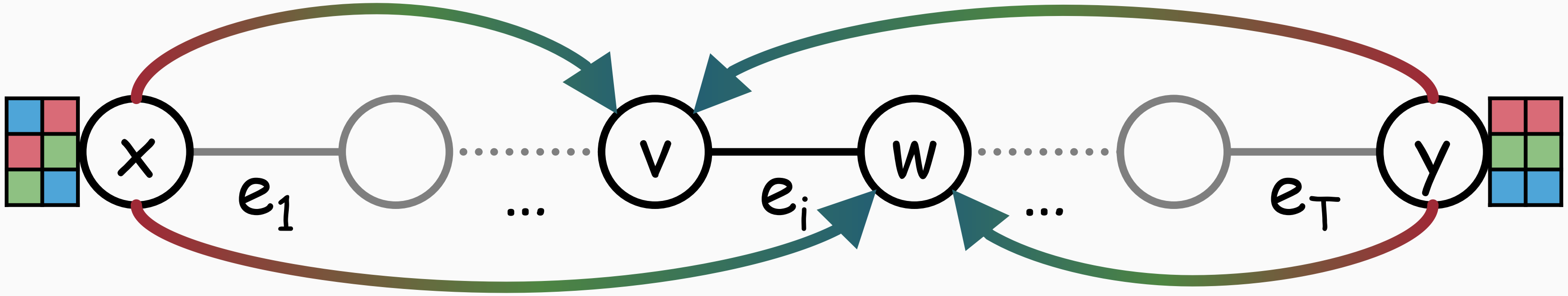
🎯**Goal:** $\forall\psi\ \mathrm{val}_G(\psi) < 1\text{-}\varepsilon \quad \Longrightarrow \quad \forall\psi^*\ \mathrm{val}_{G^*}(\psi^*) < 1\text{-}\Omega(T\cdot\varepsilon)$

Some $\psi: V\to\Sigma$ ⇠ Optimal $\psi^*: V\to\Sigma^V$

plurality vote

- If verifier checks one of ε-frac. unsat. edges e_i w.r.t. ψ, ψ^* doesn't pass test at e_i w.p. $\Omega(1)$
- Edges in RWs **W** are pairwise independent & uniform (almost)
  this is where expansion is applied

➜😋verifier rejects w.p. $\approx\Omega(1)\cdot\varepsilon\cdot\mathbb{E}[\text{length of } \mathbf{W}] = \Omega(T\cdot\varepsilon)$



Maxmin Binary CSP Reconfiguration

[Ito et al. Theor. Comput. Sci. 2011] [O. STACS 2023]

- **Input**: Binary CSP $G = (V,E,\Sigma,\Pi=(\pi_e)_{e\in E})$ & satisfying $\psi_s, \psi_t: V\to\Sigma$
- **Output**: $\psi = \langle \psi^{(0)}=\psi_s, \ldots, \psi^{(\ell)}=\psi_t \rangle$ (reconf. sequence) s.t.
  ~~ψ satisfies all edges of G~~ (feasibility)
  $\mathrm{Ham}(\psi^{(i-1)}, \psi^{(i)}) = 1$ (adjacency on hypercube)
- **Goal**: $\max_\psi \mathrm{val}_G(\psi) \stackrel{\text{def}}{=} \min_i$ (frac. of edges satisfied by $\psi^{(i)}$)
  $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) \stackrel{\text{def}}{=}$ max. value of $\uparrow$

RIH $\Longrightarrow \exists\varepsilon > 0$, Gap[1 vs. 1-ε] Binary CSP Reconf. is **PSPACE**-hard:

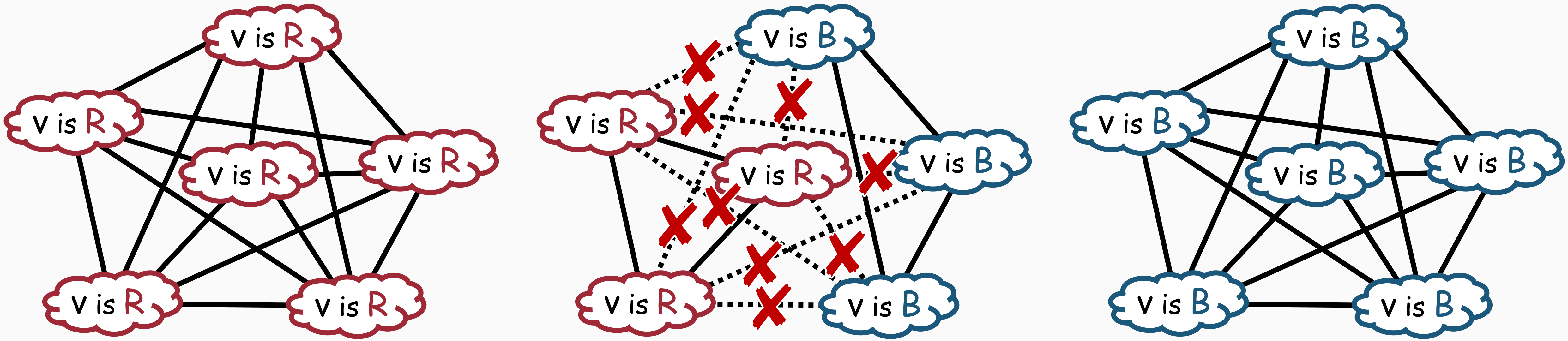
- $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) = 1$ ($\exists\psi$ every $\psi^{(i)}$ satisfies all edges), or
- $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) < 1-\varepsilon$ ($\forall\psi$ some $\psi^{(i)}$ violates ε-frac. of edges)

Difficulty of powering Binary CSP Reconf.

😭Loosing perfect completeness

🎯**Goal:** $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) = 1 \not\Rightarrow \mathrm{OPT}_{G^*}(\psi^*_s \leftrightsquigarrow \psi^*_t) = 1$

All vertices should have the SAME opinion about the color of v



$\forall x \;\; \psi^*_s(x)[v] \stackrel{\mathrm{def}}{=} R$ $\qquad \exists x,y \;\; \psi^{*(i)}(x)[v] \neq \psi^{*(i)}(y)[v]$ $\qquad \forall x \;\; \psi^*_t(x)[v] \stackrel{\mathrm{def}}{=} B$

😖Verifier rejects↩

Our solution

Alphabet squaring trick [O. STACS 2023]

🎯Think as if opinion could take a pair of colors!

- Original $\Sigma = \{R, G, B\}$
- New $\Sigma_{sq} = \{R, G, B, \mathbf{RG}, \mathbf{GB}, \mathbf{BR}\}$
- α & β are **consistent** $\Leftrightarrow$ $\alpha \subseteq \beta$ or $\alpha \supseteq \beta$

	R	RG	G	GB	B	BR
R	●	●				●
RG	●	●	●			
G		●	●	●		
GB			●	●	●	
B				●	●	●
BR	●				●	●

⚠️Asgmt. on G^* is now $\psi^*: V \to (\Sigma_{sq})^V$, not ~~$\psi^*: V \to \Sigma^V$~~

Our solution

Modifying verifier's test (1)

🎯Think as if opinion could take a pair of colors!

- Original $\Sigma \quad = \{R, G, B\}$
- New $\quad\quad \Sigma_{sq} = \{R, G, B, \mathbf{RG}, \mathbf{GB}, \mathbf{BR}\}$
- α & β are **consistent** $\Leftrightarrow$ $\alpha \subseteq \beta$ or $\alpha \supseteq \beta$

	R	RG	G	GB	B	BR
R	●	●				●
RG	●	●	●			
G		●	●	●		
GB			●	●	●	
B				●	●	●
BR	●				●	●

Pick RW $\mathbf{W} = \langle e_1, \ldots, e_T \rangle$ from x to y as before

$\psi^*(x)$ & $\psi^*(y)$ <u>pass modified test</u> at $e_i = (v,w)$ if

opinions of x & y are **consistent** at (v,w)

opinions about (v,w) satisfy $\pi_{(v,w)}$

Our solution

Modifying verifier's test (2)

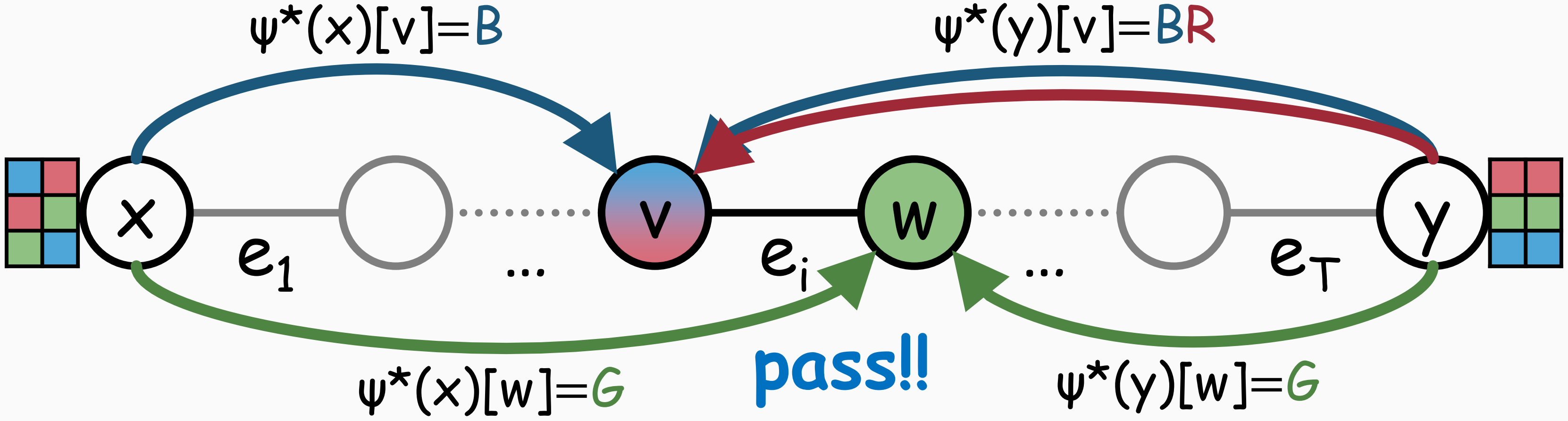
	R	RG	G	GB	B	BR
R	●	●				●
RG	●	●	●			
G		●	●	●		
GB			●	●	●	
B				●	●	●
BR	●				●	●

🎯Think as if opinion could take a pair of colors!

- Original $\Sigma = \{R, G, B\}$
- New $\Sigma_{sq} = \{R, G, B, \mathbf{RG}, \mathbf{GB}, \mathbf{BR}\}$
- α & β are **consistent** $\Leftrightarrow$ $\alpha \subseteq \beta$ or $\alpha \supseteq \beta$

Pick RW $\mathbf{W} = \langle e_1, \dots, e_T \rangle$ from x to y as before

$\psi^*(x)$ & $\psi^*(y)$ <u>pass modified test</u> at $e_i = (v,w)$ if



Our solution

Modifying verifier's test (3)

	R	RG	G	GB	B	BR
R	●	●				●
RG	●	●	●			
G		●	●	●		
GB			●	●	●	
B				●	●	●
BR	●				●	●

🎯Think as if opinion could take a pair of colors!

- Original $\Sigma = \{R, G, B\}$
- New $\Sigma_{sq} = \{R, G, B, \mathbf{RG}, \mathbf{GB}, \mathbf{BR}\}$
- α & β are **consistent** $\Leftrightarrow$ $\alpha \subseteq \beta$ or $\alpha \supseteq \beta$

Pick RW **W** $= \langle e_1, \ldots, e_T \rangle$ from x to y as before

$\psi^*(x)$ & $\psi^*(y)$ <u>pass modified test</u> at $e_i = (v,w)$ if

(C1) $\psi^*(x)[v]$ & $\psi^*(y)[v]$ are **consistent**

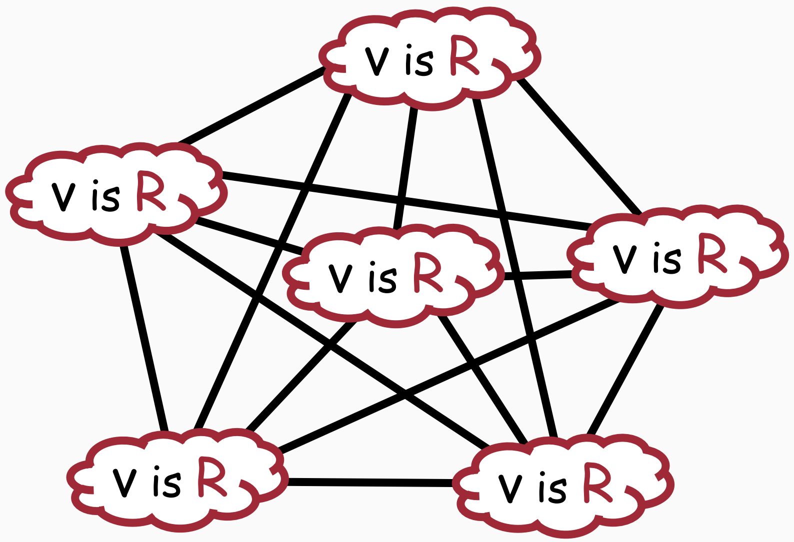
(C2) $\psi^*(x)[w]$ & $\psi^*(y)[w]$ are **consistent**

(C3) $(\psi^*(x)[v] \cup \psi^*(y)[v]) \times (\psi^*(x)[w] \cup \psi^*(y)[w]) \subseteq \pi_{(v,w)}$

⚠️This verifier is "much weaker" than before

😋 Alphabet squaring preserves perfect completeness

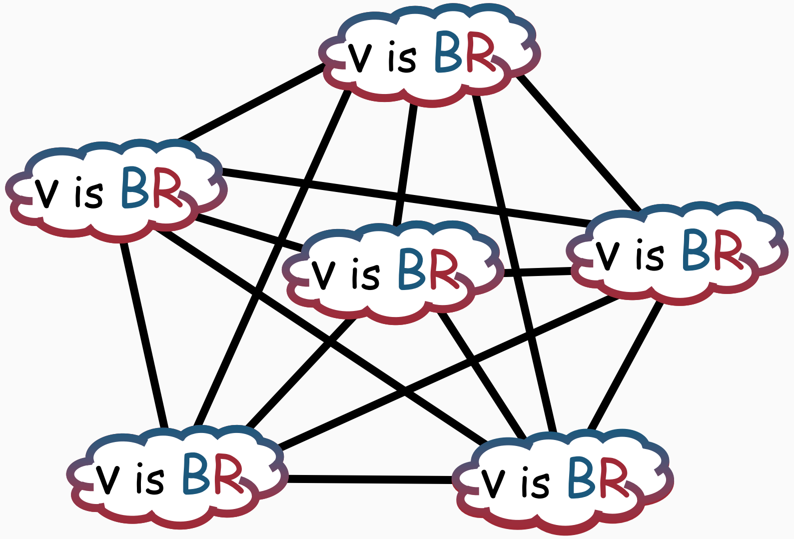
🎯 **Goal:** $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) = 1 \quad \Longrightarrow \quad \mathrm{OPT}_{G^*}(\psi^*{}_s \leftrightsquigarrow \psi^*{}_t) = 1$



Can transform all R opinions into all B opinions via BR's

😋 Alphabet squaring preserves perfect completeness

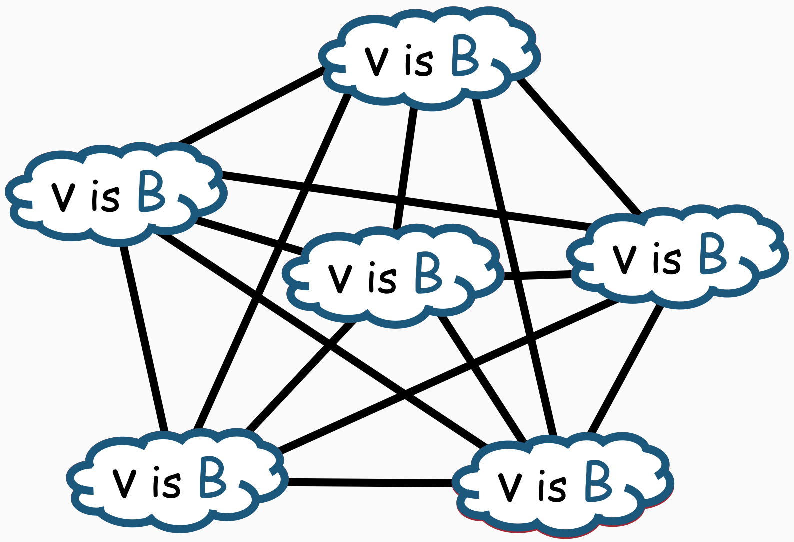
🎯 **Goal:** $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) = 1 \quad \Longrightarrow \quad \mathrm{OPT}_{G^*}(\psi^*{}_s \leftrightsquigarrow \psi^*{}_t) = 1$



Can transform all R opinions into all B opinions via BR's

Alphabet squaring preserves perfect completeness

Goal: $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) = 1 \quad \Longrightarrow \quad \mathrm{OPT}_{G^*}({\psi^*}_s \leftrightsquigarrow {\psi^*}_t) = 1$



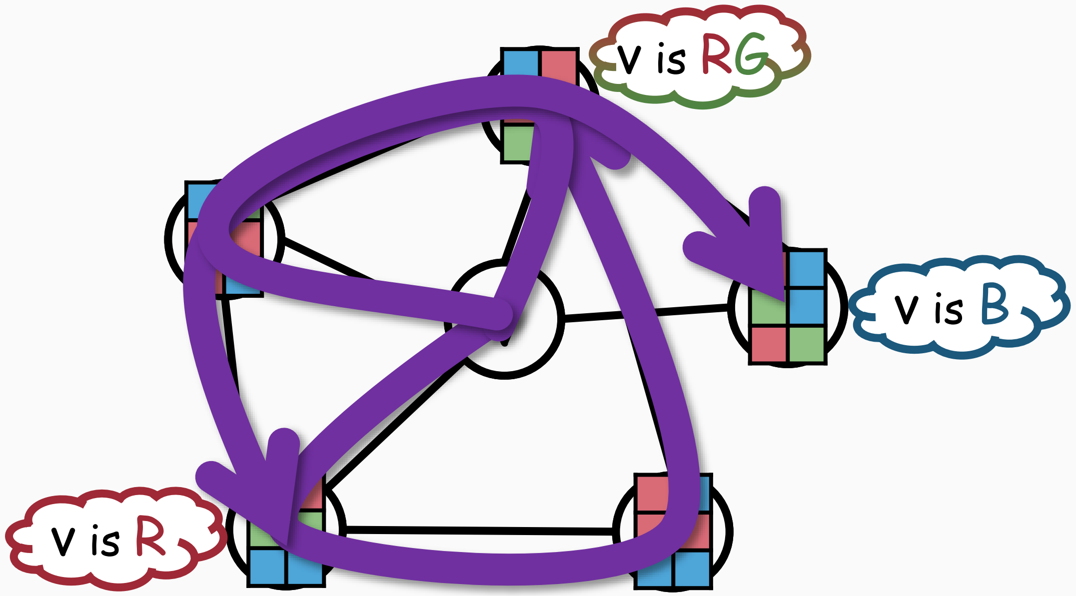
Can transform all R opinions into all B opinions via BR's

Why the modified verifier works

Soundness: Overview

🎯**Goal:** $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) < 1\text{-}\varepsilon \implies \mathrm{OPT}_{G^*}(\psi^*_s \leftrightsquigarrow \psi^*_t) < 1\text{-}\Omega(T\cdot\varepsilon)$

$\psi = \langle \psi^{(0)}, \ldots, \psi^{(\ell)} \rangle \xleftarrow[\text{plurality vote}]{} \text{Optimal } \psi^* = \langle \psi^{*(0)}, \ldots, \psi^{*(\ell)} \rangle$



R	G	B
2	1	1

➔ $\psi(v) \stackrel{\text{def}}{=} R$

Why the modified verifier works

Soundness: Overview

🎯**Goal:** $\mathrm{OPT}_G(\psi_s \leftrightsquigarrow \psi_t) < 1\text{-}\varepsilon \implies \mathrm{OPT}_{G^*}(\psi^*_s \leftrightsquigarrow \psi^*_t) < 1\text{-}\Omega(T\cdot\varepsilon)$

$\psi = \langle\psi^{(0)}, \ldots, \psi^{(\ell)}\rangle \xleftarrow{\text{plurality vote}}$ Optimal $\psi^* = \langle\psi^{*(0)}, \ldots, \psi^{*(\ell)}\rangle$

- Can show "$\exists i\ \mathrm{val}_G(\psi^{(i)}) < 1\text{-}\varepsilon\text{+}o(1)$" (slightly nontrivial)
- Suppose $\psi^{(i)}$ violates (v,w) of G

$$\boxed{\Pr[\psi^{*(i)} \text{ fails modified test at } (v,w) \mid \mathbf{W} \text{ touches } (v,w)] = \Omega(1)}$$

⚠DIFFERENT from [Radhakrishnan. ICALP 2006] $\because\ \psi^{*(i)}: V \to (\Sigma_{sq})^V$ but $\psi^{(i)}: V \to \Sigma$

{R, G, B, RG, GB, BR} {R, G, B}

Why the modified verifier works

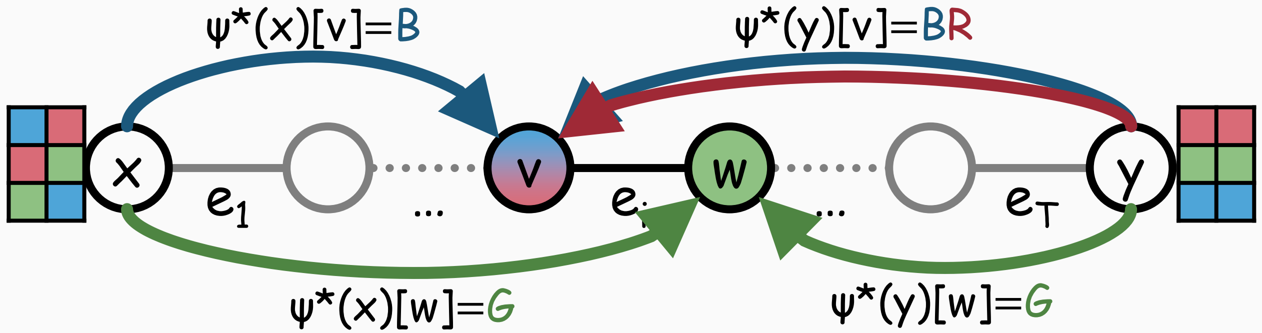
Soundness: Bounding failure probability

🎯Bound ⭐ $\overset{\text{def}}{=}$ **Pr**[ψ* fails modified test at (v,w) | **W** touches (v,w)]

assuming "plurality vote ψ violates $\pi_{(v,w)}$"

$$\left.\begin{array}{l} p_v \overset{\text{def}}{=} \mathbf{Pr}_x[\psi^*(x)[v] \ni \psi(v)] \\ p_w \overset{\text{def}}{=} \mathbf{Pr}_y[\psi^*(y)[w] \ni \psi(w)] \end{array}\right\}$$

Prob. random opinion over RW from v or w is consistent with plurality vote



p_v & p_w are UNKNOWN, but…

(1) $\mathbf{Pr}_{x,y}[\psi^*(x)[v]$ & $\psi^*(y)[v]$ are consist.$] \leq \boxed{2\cdot p_v}$

(2) $\mathbf{Pr}_{x,y}[\psi^*(x)[w]$ & $\psi^*(y)[w]$ are consist.$] \leq \boxed{2\cdot p_w}$

2-factor loss from [Radhakrishnan. ICALP 2006]

(3) $\mathbf{Pr}_{x,y}[\psi^*(x)[v] \ni \psi(v)$ & $\psi^*(y)[w] \ni \psi(w)] \geq p_v \cdot p_w$

➜😋 ⭐ $\geq \max\{1\text{-}2\cdot p_v,\ 1\text{-}2\cdot p_w,\ p_v\cdot p_w\} \geq (\sqrt{2}-1)^2$

Why the modified verifier works

Where 2-factor loss comes from

- $\boldsymbol{\lambda}$ & $\boldsymbol{\mu}$: distribution over Σ_{sq}
- $a_{PLR} \stackrel{\text{def}}{=} \operatorname{argmax}_{a \in \Sigma} \Pr_{X \sim \boldsymbol{\lambda}}[a$ & X are consistent$]$ (depending only on $\boldsymbol{\lambda}$)

  this is exactly plurality vote
- $p \stackrel{\text{def}}{=} \Pr_{X \sim \boldsymbol{\lambda}}[a_{PLR}$ & X are **consistent**$]$
- $q \stackrel{\text{def}}{=} \Pr_{X \sim \boldsymbol{\lambda},\, Y \sim \boldsymbol{\mu}}[X$ & Y are **consistent**$]$

🤪 $q \leq 2p$

E.g.

- $a_{PLR} = R$
- $p = 0.51,\ q = 1$

	R	B	RB
$\boldsymbol{\lambda}$	0.51	0.49	0
$\boldsymbol{\mu}$	0	0	1

Reconfiguration Problems, Hardness of Approximation, and Gap Amplification: What Are They?

Conclusions: We have seen...

Reconfiguration
- Brand-new, puzzle-like **PSPACE**-complete problems

PSPACE-hardness of approximation
- May require a theory beyond the PCP theorem for **NP**

Gap amplification
- We *partially* made it (à la Dinur)!!

MANY OPEN QUESTIONS
- **Algorithmic results?** Proof of **RIH**? Optimal inapprox.?

Breaking news: A few weeks ago...

Proof of **RIH**

- Independently announced by
  [Karthik C. S.-Manurangsi. 2023. https://arxiv.org/abs/2312.17140]
  [Hirahara-O. 2024. https://arxiv.org/abs/2401.00474]
- Both applying PCP of proximity
  [Ben-Sasson, Goldreich, Harsha, Sudan, Vadhan. SIAM J. Comput. 2006]
  [Dinur-Reingold. SIAM J. Comput. 2006]

Tight **NP**-hardness [Karthik C. S.-Manurangsi. 2023]

- Binary CSP Reconf. is **NP**-hard to approx. within $\frac{1}{2}+\varepsilon$
- Set Cover Reconf. is **NP**-hard to approx. within $2-\varepsilon$

To be continued...