

SAT Solvers in Computational Algorithm Design

- ◆ We used SAT and MAX-SAT solvers to design novel distributed algorithms
- ◆ **Idea:** Express the existence of an algorithm as a finite combinatorial problem
- ◆ **Results:** New optimal algorithms and impossibility results in three domains

Danny Dolev

The Hebrew University of Jerusalem

Juho Hirvonen

Janne H. Korhonen

Joel Rybicki

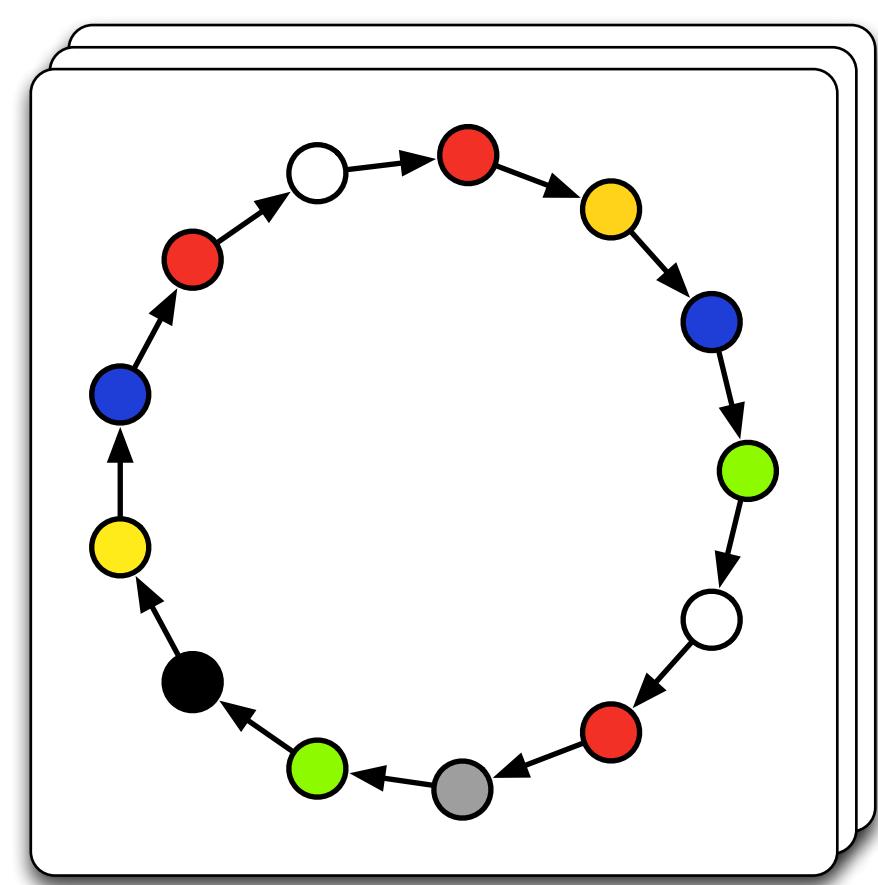
Jukka Suomela

HIIT and University of Helsinki

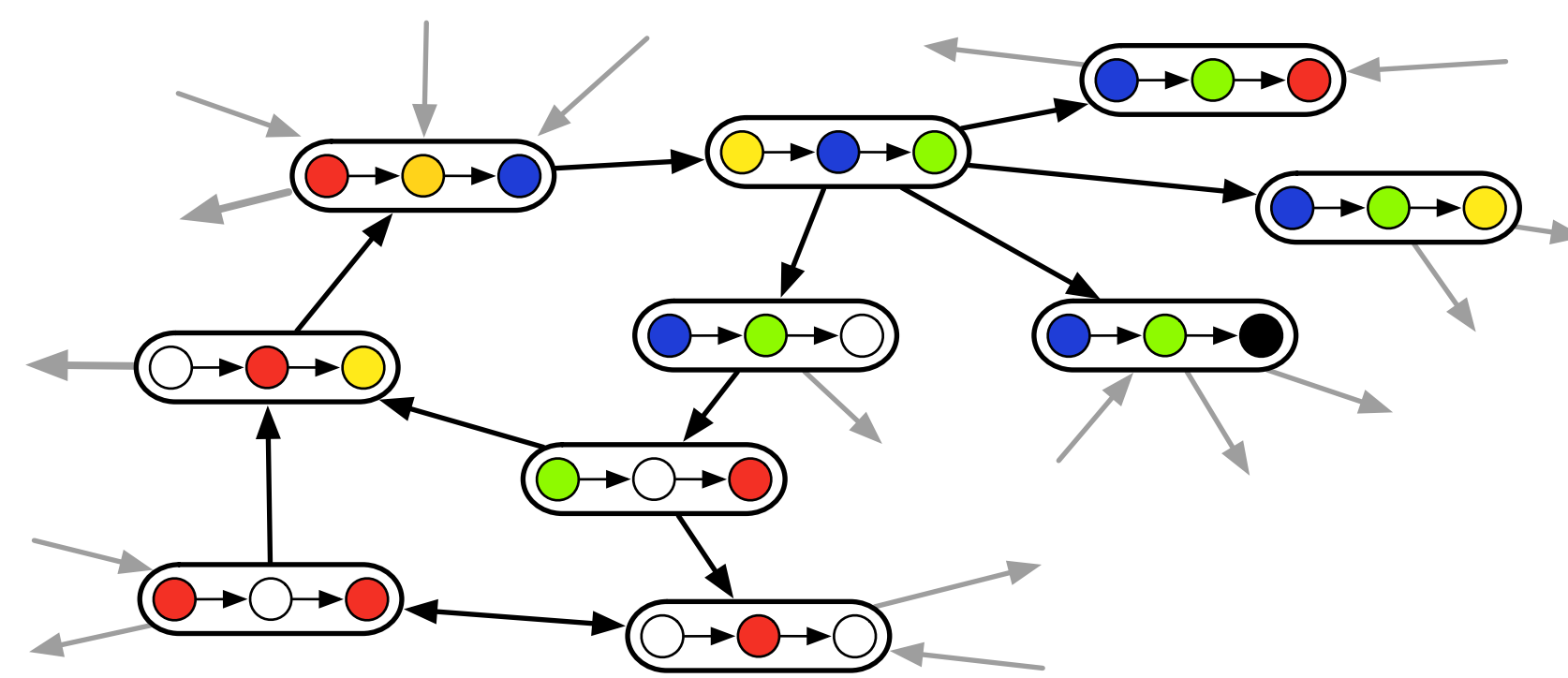
Christoph Lenzen

MIT

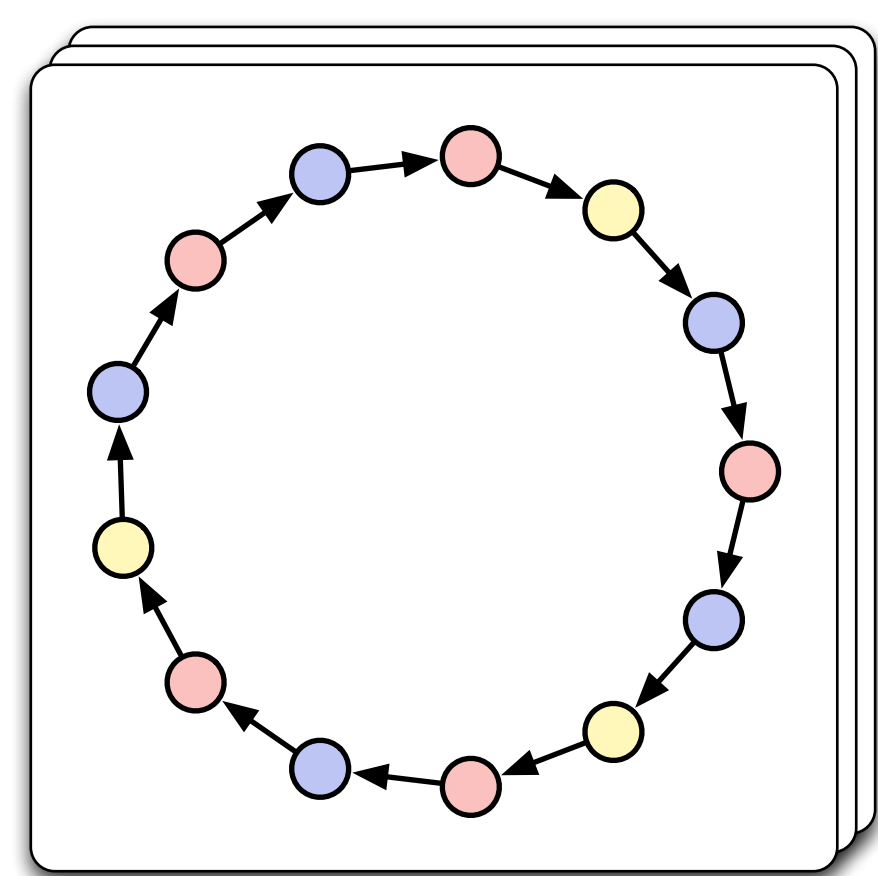
Deterministic graph coloring



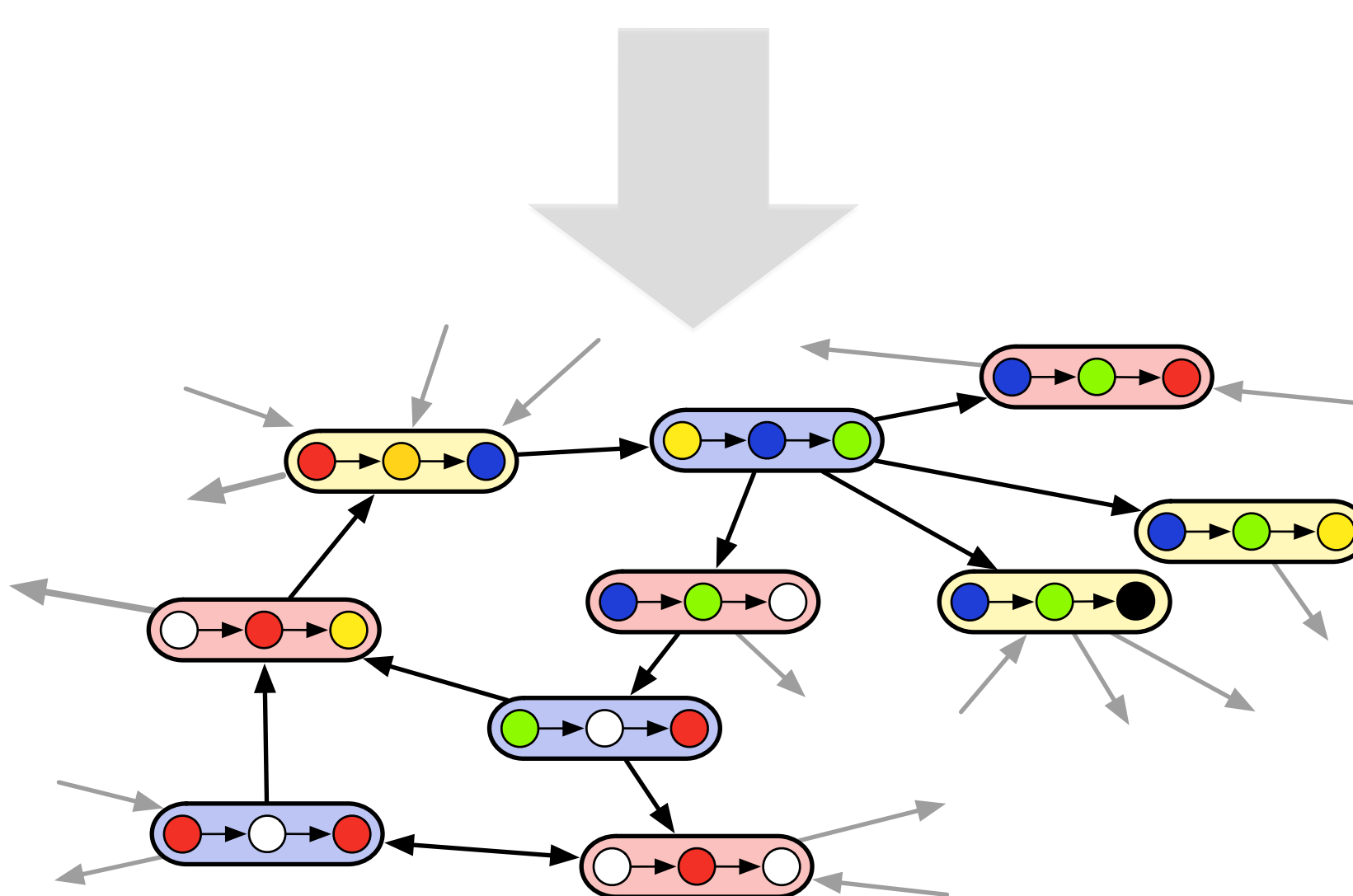
Input: n -coloring



neighborhood graph

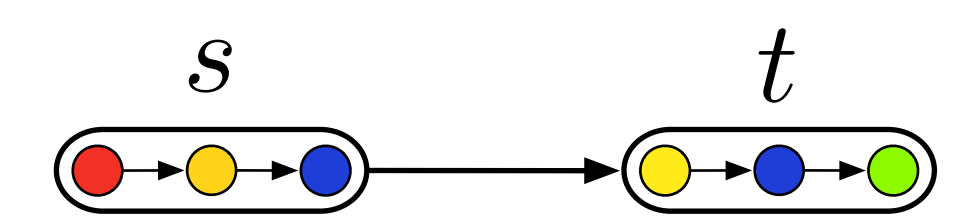


Output: k -coloring



k -coloring \Leftrightarrow algorithm

k -coloring as SAT instance



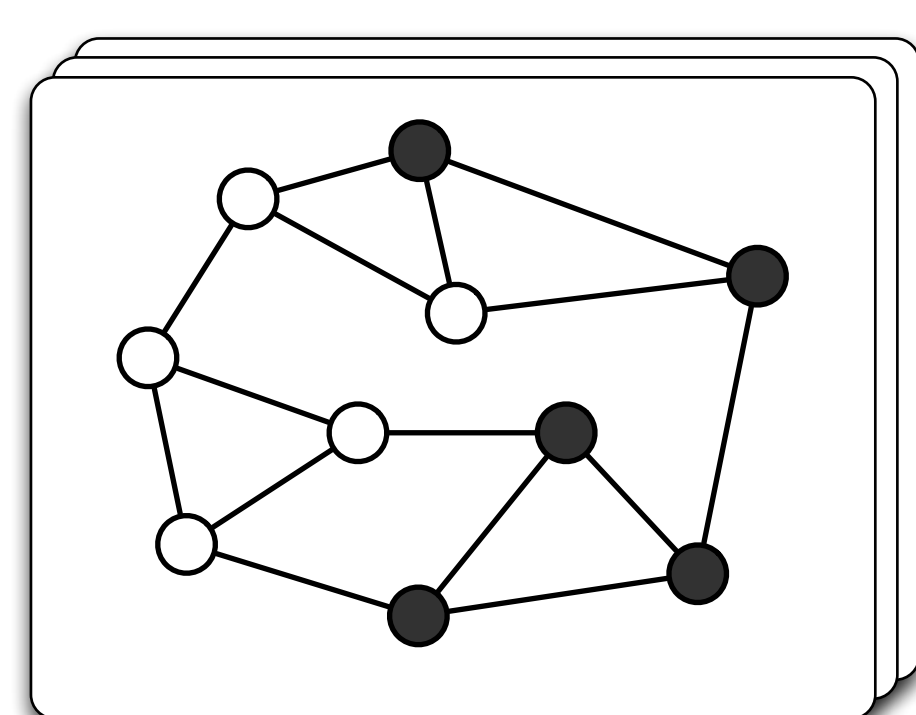
$$\begin{aligned} \bar{x}_{s,i} \wedge \bar{x}_{t,i} \quad i = 1, \dots, k \\ x_{s,1} \vee \dots \vee x_{s,k} \end{aligned}$$

Key results

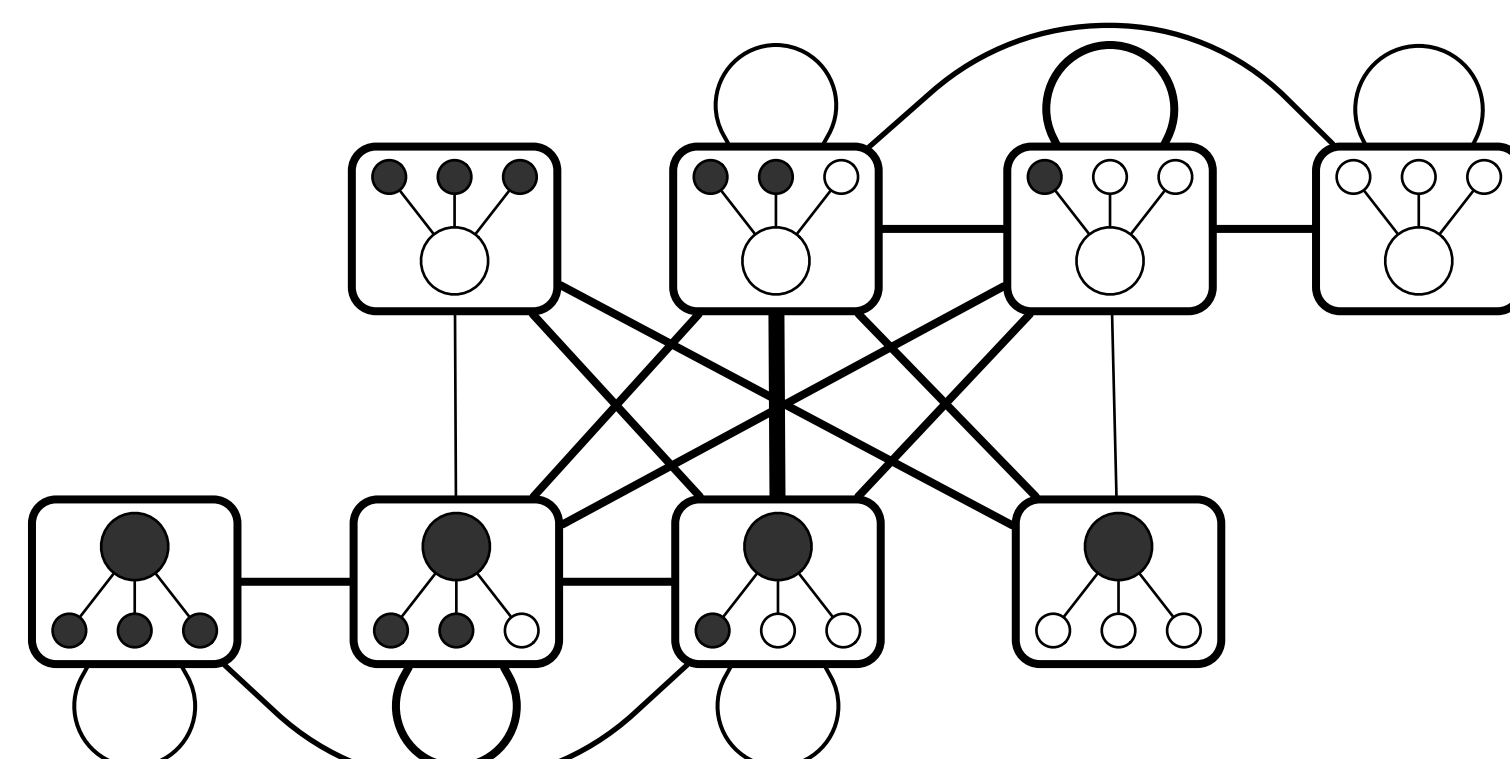
Positive: $\frac{1}{2}(\log^* n + 7) \rightarrow \frac{1}{2}(\log^* n + 3)$

Negative: $\frac{1}{2}(\log^* n - 3) \rightarrow \frac{1}{2}(\log^* n + 1)$

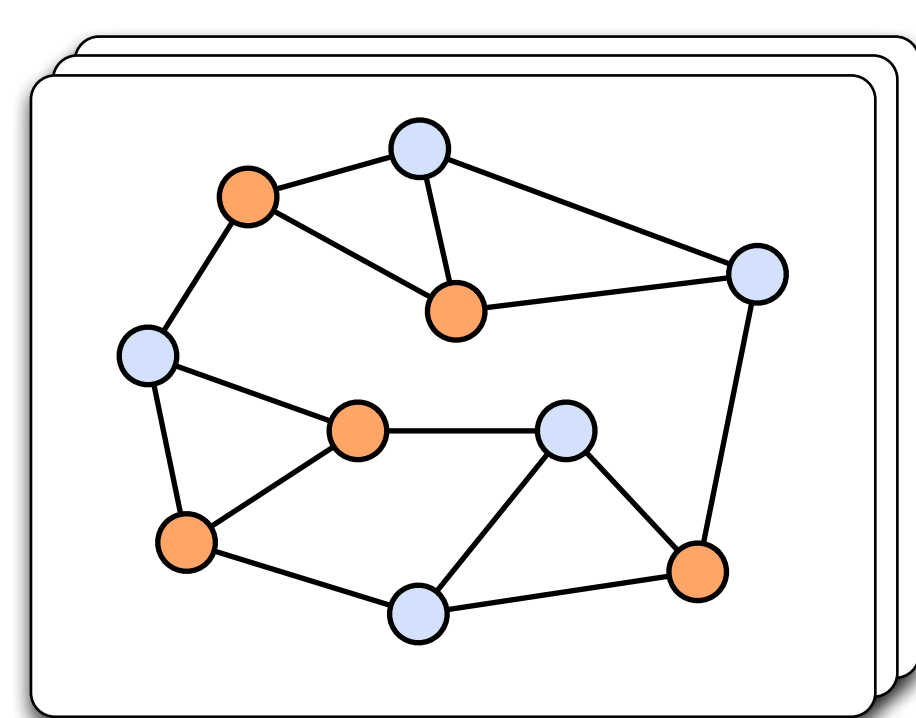
Randomized max cut



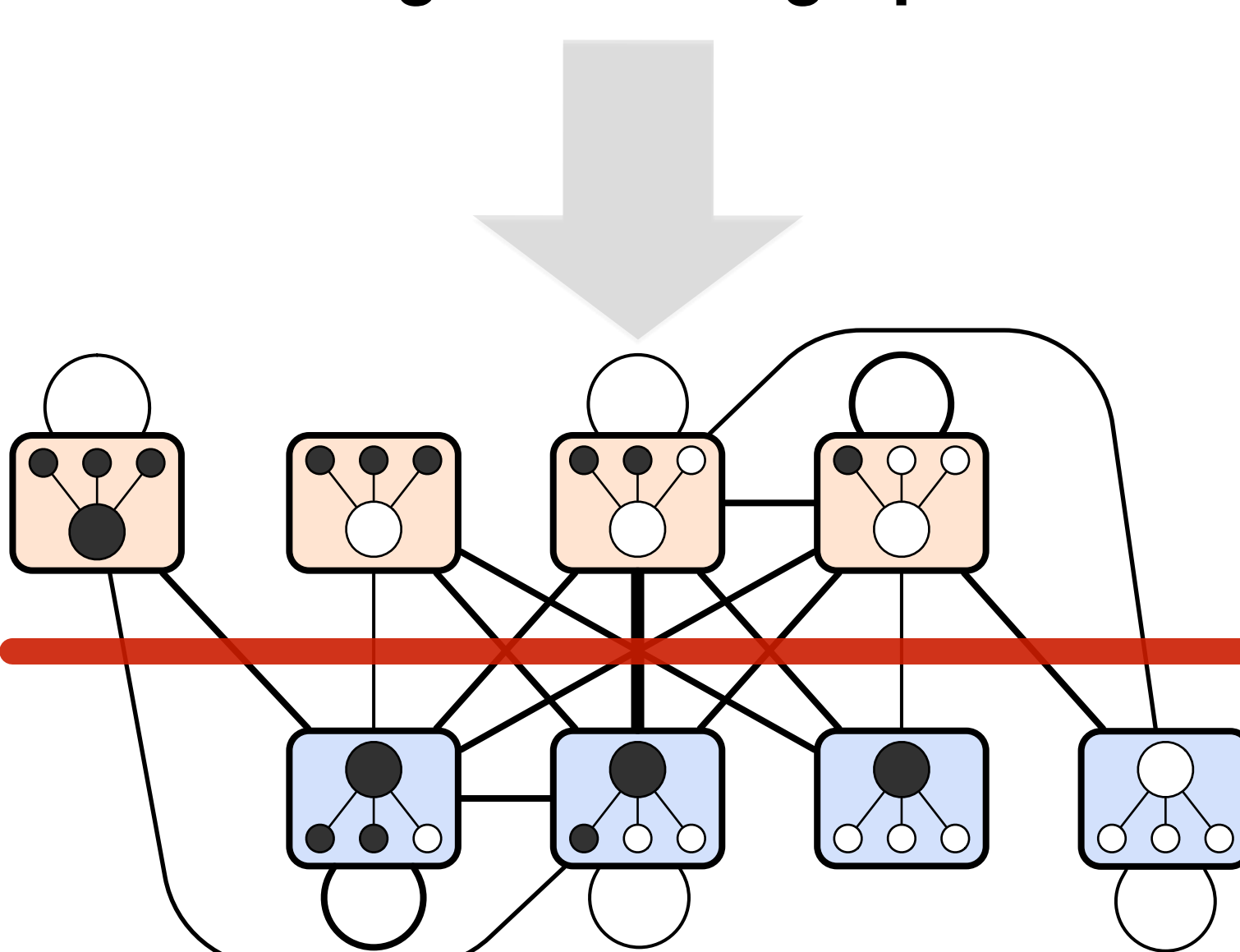
Input: random cut



neighborhood graph

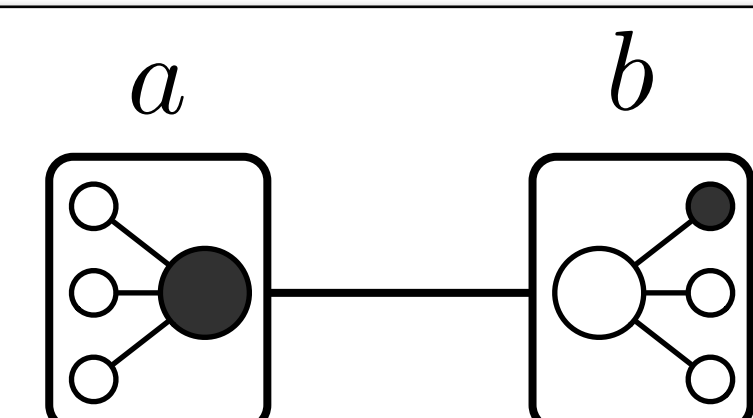


Output: better cut



cut \Leftrightarrow algorithm

Weighted MAX-SAT instance

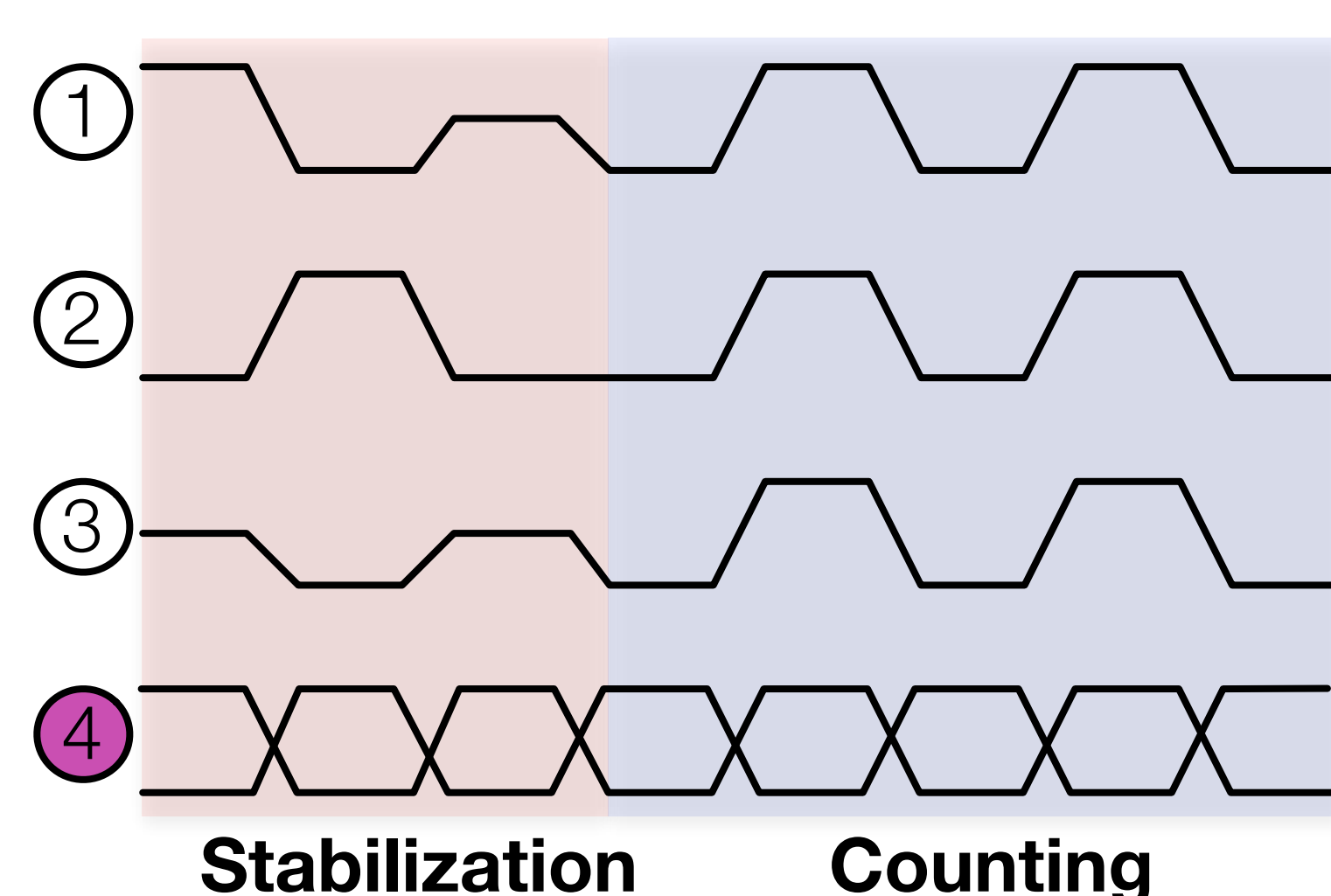


$$\begin{aligned} (x_a \vee x_b) : w(a, b) \\ (\bar{x}_a \vee \bar{x}_b) : w(a, b) \end{aligned}$$

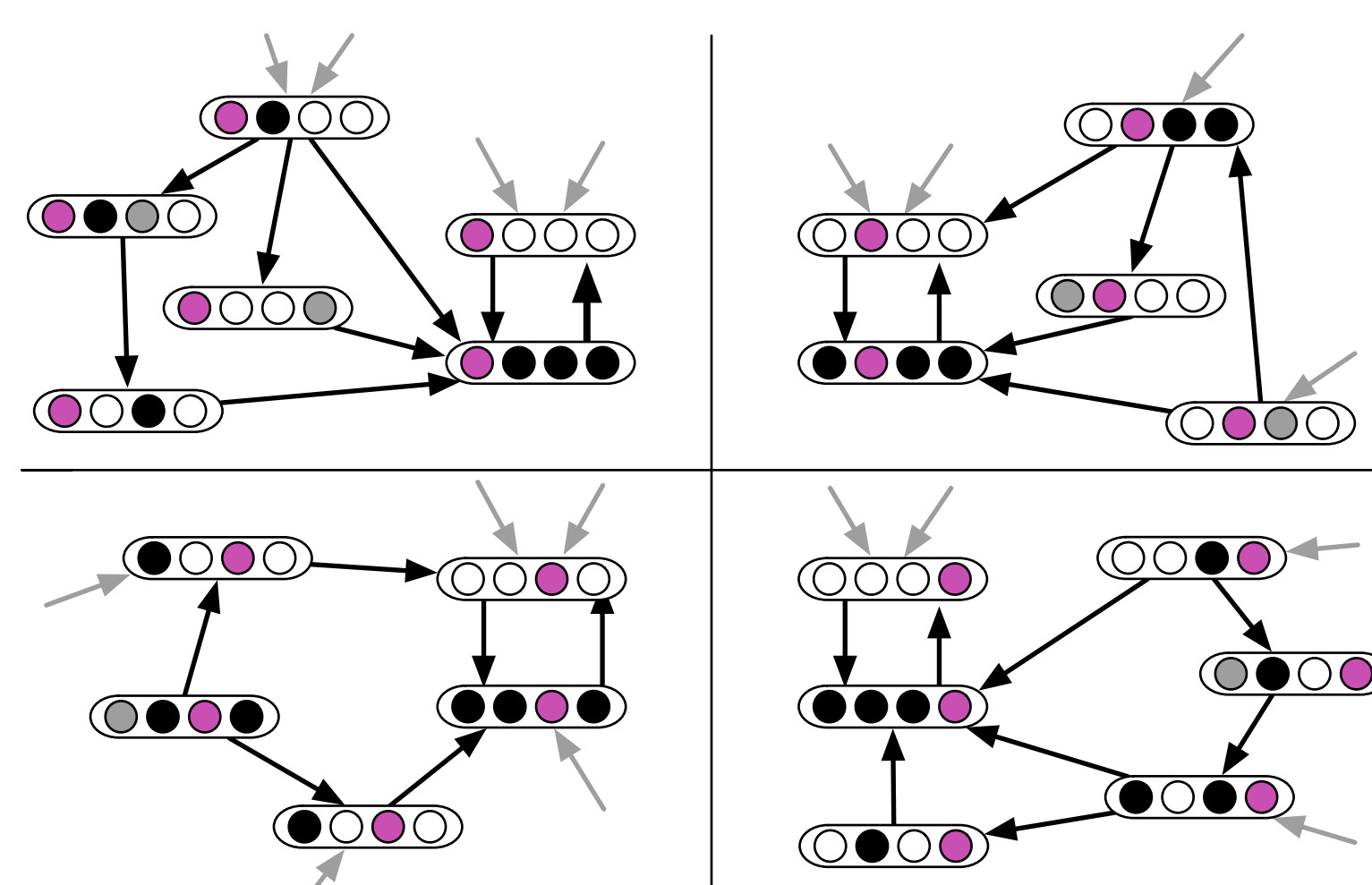
Key results

An optimal cut algorithm: $\frac{1}{2} + \Theta\left(\frac{1}{\sqrt{d}}\right)$

Self-stabilizing counting with Byzantine failures



Stabilization Counting



execution graphs

Key results

	Nodes	Auxiliary states
Positive	≥ 4	1
	≥ 6	0
Negative	≤ 5	0