SAT Solvers in Computational Algorithm Design

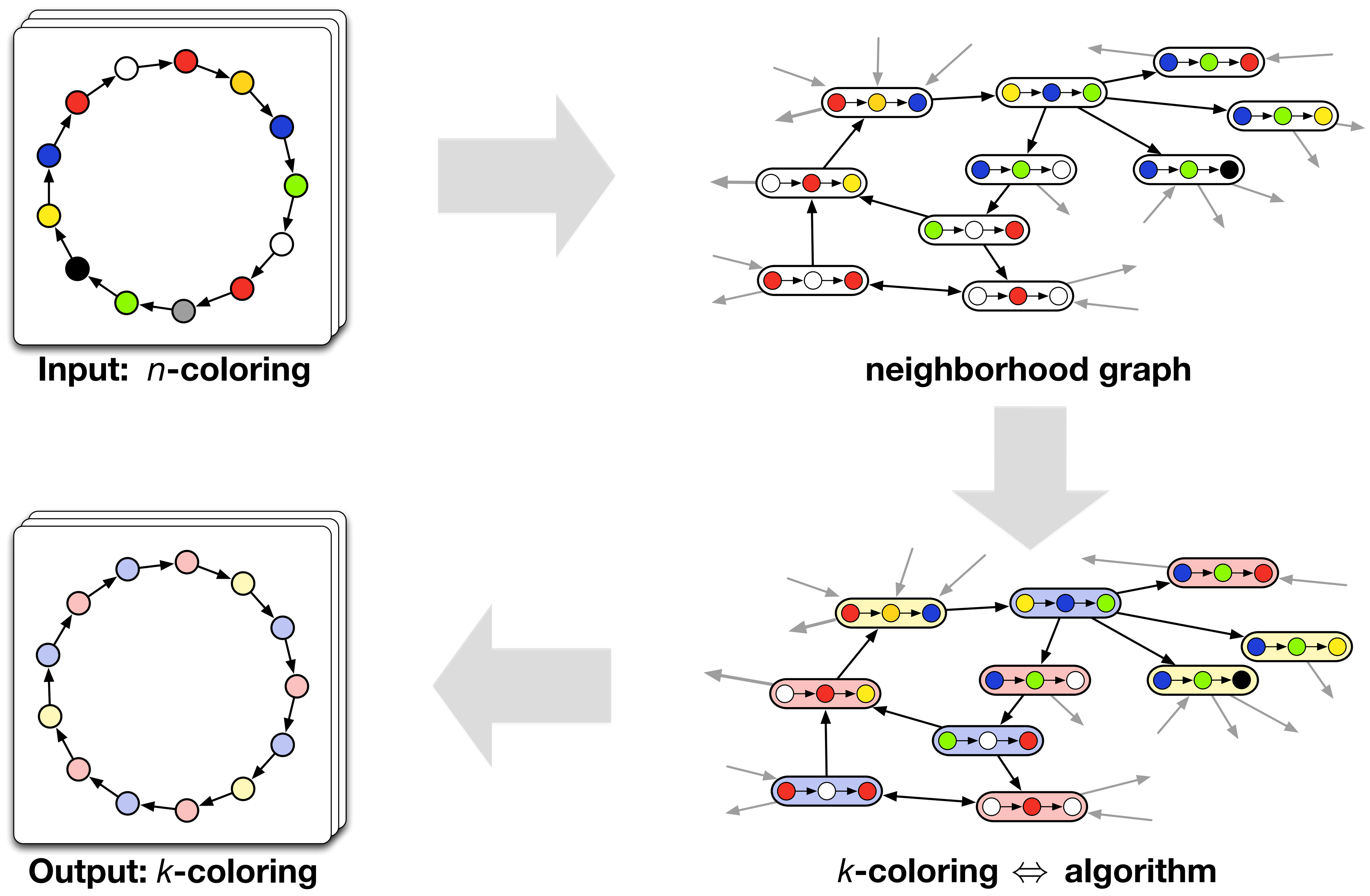
- We used SAT and MAX-SAT solvers to design novel distributed algorithms
- **Idea:** Express the existence of an algorithm as a finite combinatorial problem
- **Results:** New optimal algorithms and impossibility results in three domains

Danny Dolev
The Hebrew University of Jerusalem

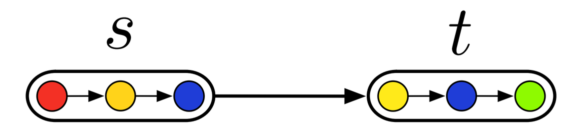
Juho Hirvonen
Janne H. Korhonen
Joel Rybicki
Jukka Suomela
HIIT and University of Helsinki

Christoph Lenzen
MIT

Deterministic graph coloring



k-coloring as SAT instance



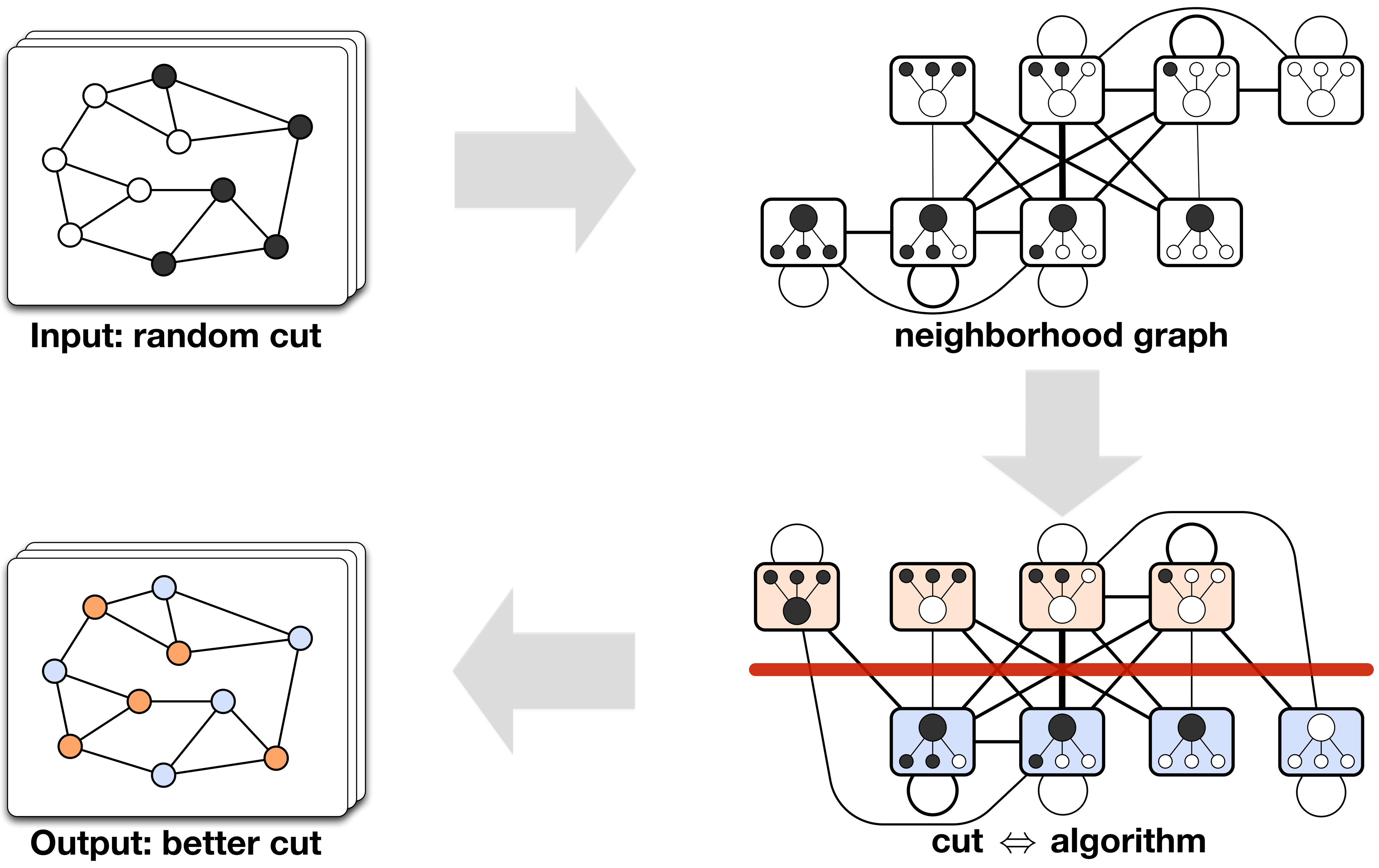
$$\overline{x}_{s,i} \wedge \overline{x}_{t,i} \quad i = 1, \dots, k$$
$$x_{s,1} \vee \cdots \vee x_{s,k}$$

Key results

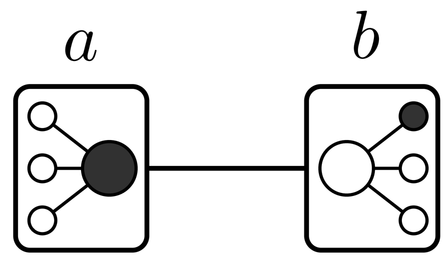
Positive: $\frac{1}{2}(\log^* n + 7) \to \frac{1}{2}(\log^* n + 3)$

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

Randomized max cut



Weighted MAX-SAT instance

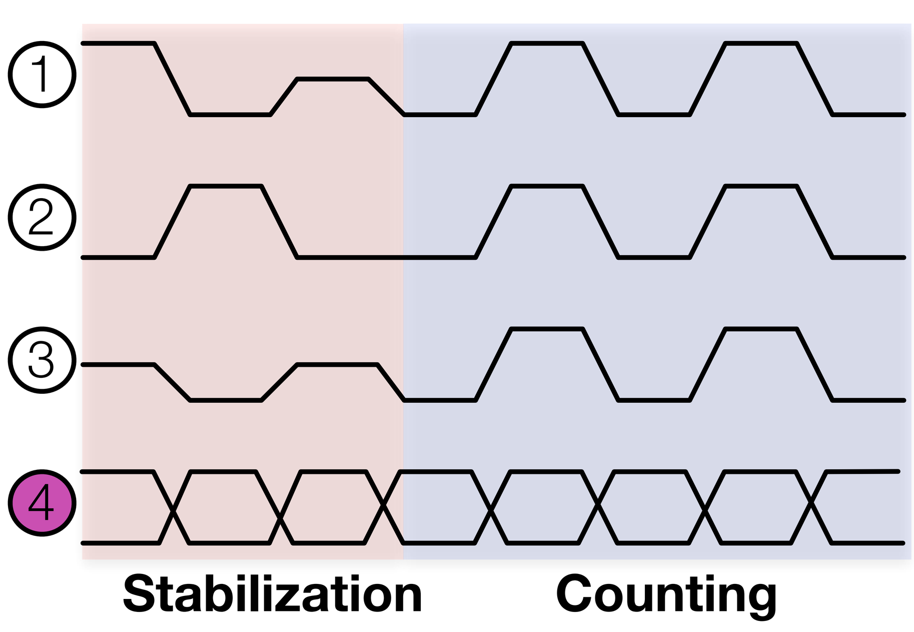


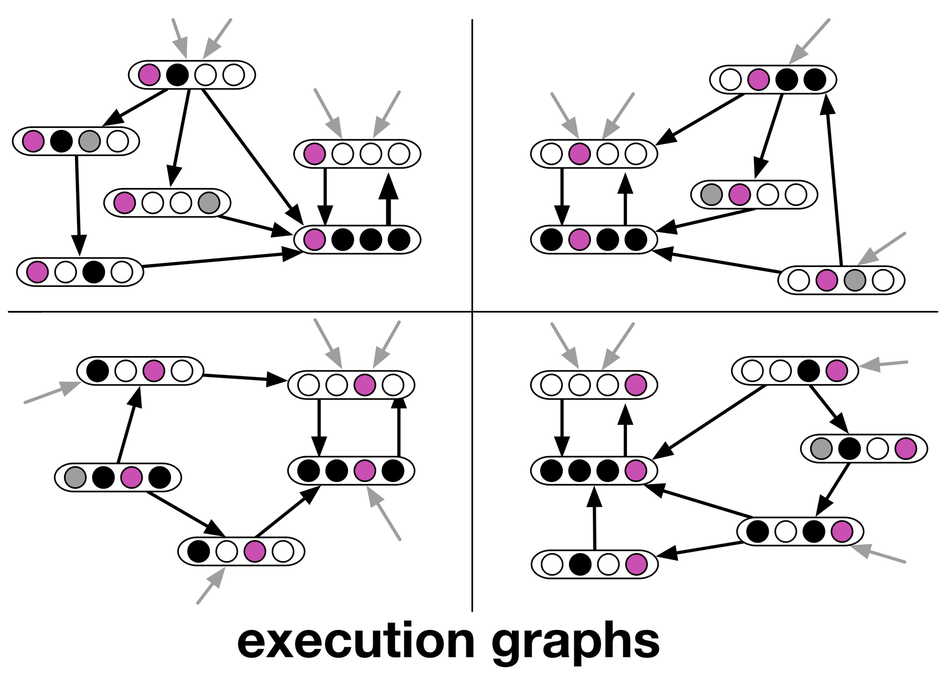
$$(x_a \vee x_b) \colon w(a,b)$$
$$(\overline{x}_a \vee \overline{x}_b) \colon w(a,b)$$

Key results

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

Self-stabilizing counting with Byzantine failures





Key results

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