

Self-Stabilizing Algorithms in the Uniform Port Model

Yuval Emek

Technion — Israel Institute of Technology

Workshop on Foundations of Distributed and Parallel Graph Algorithms
Venice, Italy
May 18–21, 2026

Joint work with Liam Brinker and Oren Louidor (in preparation)

- Uniform computational model:
algorithm designer oblivious to instance size
 - E.g., Turing machines vs. Boolean circuits
- Advantage: easier to manufacture/install *computational devices*



- Amplified in **DistComp**:
execution includes multitude of computational devices

1 Uniform Distributed Graph Algorithms

2 The Uniform Port Model

3 Our Results

4 Conclusion

Interacting automata

- Computational devices = *automata* that **interact** with other automata
- Interactions determined by graph $G = (V, E)$
 - G serves as **problem instance**
- Syntax: $\text{Alg} = \langle Q, \delta, \omega \rangle$
 - Q = state set
 - δ = (randomized) *state transition function*
next state $\leftarrow \delta(\text{current state, current states of interacting automata})$
 - ω = function from states to *output labels* (or \perp)
 - **Output**: apply ω to states of all automata
- (Synchronous) *execution* is sequence C^0, C^1, \dots of *configurations*
 C^t : automata \rightarrow states
- Non-uniform: $\forall n \exists \text{Alg} \forall G \in \mathcal{G}_n$ Uniform: $\exists \text{Alg} \forall n \forall G \in \mathcal{G}_n$

Theorem ([Angluin 1980])

Randomized anonymous uniform algorithms cannot elect a leader (if termination detection is required).

- **Initial configuration** C^0 ?

- *Graceful initialization*: $q_{\text{init}} \in Q$ determined by Alg's designer
- *Self-stabilization*: adversarial
 - Recovery from **transient faults**¹
 - Graceful initialization = *fault free*
 - Robust solution concept (subroutine composition)

- *Correct* Alg stabilizes to **legal output** by time t w.p. $\rightarrow 1$ as $t \rightarrow \infty$
 - For all graphs
 - Self-stab: for all initial configurations
- *Runtime* = earliest t such that output stabilizes by time t
 - Bound w.h.p.

¹Lamport: "A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."

Truly uniform distributed algorithms

- For many years: large (often infinite) Q
- Turning point: DistComp in **micro-scale** biological networks [Afek et al. 2011]
- Aim for local automata with **constant** $|Q|$ and $|\delta|$
 - Independent of any graph parameter
 $n = |V|$, $m = |E|$, $D = \text{diam}(G)$, $\Delta = \max_{v \in V} \deg(v)$
 - *Truly uniform*
- **Caveat:** automaton may interact with many other automata
 - too many for δ to encode
- **Solution:** argument of $\delta =$ **set** of states of interacting automata [Hella et al. 2015]
 - set, rather than vector or multiset
 - Related to *beeping* communication scheme [Cornejo, Kuhn 2010]

Node-centric truly uniform model

- *Stone age model*: truly uniform automaton in each node [E, Wattenhofer 2013]
 - Interactions = graph edges
- Same constant $|\delta|$ automaton in all nodes (of all degrees) in all graphs



- **Fault free SA algs:**

- MIS in $O(\log^2 n)$ time [E, Wattenhofer 2013]
- k -leader selection in $D \log^{O(1)} n$ time [Afek, E, Kolikant 2018a]
- Simulation of linear space Turing machines on bounded degree graphs [Afek, E, Kolikant 2018b]
- Leader election (without termination detection) in $O(D^2 \log n)$ time [Vacus, Ziccardi 2025]

Self-stabilizing stone age algorithms — CONS



- Efficient self-stab SA algs for non-trivial problems in general graphs?
 - MIS on bounded diameter graphs in $O(\log^2 n)$ time [E, Keren 2021]
 - Leader election on bounded diameter graphs in $O(\log n)$ time [E, Keren 2021]
 - MIS on Erdős-Rényi random graphs in $\log^{O(1)} n$ time [Giakkoupis, Ziccardi 2023]
- Low expressivity (in graphs of unbounded degrees)
 - Not much beyond MIS, leader election, k -leader selection

1 Uniform Distributed Graph Algorithms

2 The Uniform Port Model

3 Our Results

4 Conclusion

Half-edge-centric truly uniform model

- *Half-edges* (a.k.a. *ports*) of $G = (V, E)$:
 (v, e) such that $v \in V$ and $v \in e \in E$
- Assigning output labels to ports is much more expressive
 - edge subsets, edge orientations, paths, trees
 - Common practice nowadays in [LCL literature](#)
- *Uniform port model*: truly uniform automaton in each port
 - Same constant $|\delta|$ automaton in [all ports](#) in [all graphs](#)



Formal definition

For each port $p = (v, \{u, v\})$, state transition function

$$\delta : Q \times Q \times 2^Q \rightarrow Q$$

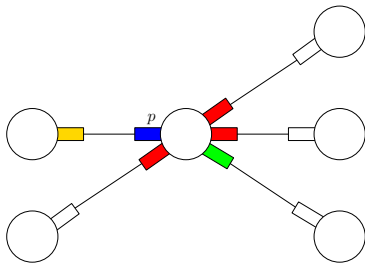
(randomly) maps

$$C^t(p), \quad C^t(\bar{p}), \quad \text{and} \quad \{C^t(p') \mid p' \in \text{sibs}(p)\}$$

to

$$C^{t+1}(p) = \delta(C^t(p), C^t(\bar{p}), \{C^t(p') \mid p' \in \text{sibs}(p)\})$$

- $\bar{p} = (u, \{u, v\})$
- $\text{sibs}(p) = \{(v, \{u', v\}) \mid u' \in N(v) - \{u\}\}$



$$C^{t+1}(p) = \delta(\text{blue}, \text{gold}, \{\text{red}, \text{green}\})$$

1 Uniform Distributed Graph Algorithms

2 The Uniform Port Model

3 **Our Results**

4 Conclusion

Self-stabilizing uniform port algorithms

- MM in $O(\log^5 n)$ time
- Sinkless orientation in $O(\log^2 n)$ time
- MIS in $O(\log^2 n)$ time
- Maximal node c -coloring in $O(\log^2 n)$ time
- Maximal edge c -coloring in $O(\log^5 n)$ time

The MIS algorithm

- Generate stream of i.i.d. coins $r(v) \sim \mathcal{U}(\{0, 1\})$ at each node $v \in E$
- Orient each edge $e = \{v_1, v_2\} \in E$:
 - If $r(v_1) \neq r(v_2)$, then orient e toward v_i s.t. $r(v_i) = 1$
 - Otherwise, keep current orientation
- If v is sink, then decide IN
 - $r(v) \leftarrow 1$ subsequently
- If v has IN-neighbor, then decide OUT
 - $r(v) \leftarrow 0$ subsequently

1 Uniform Distributed Graph Algorithms

2 The Uniform Port Model

3 Our Results

4 Conclusion

- Efficient self-stab SA algs on general graphs
 - MIS, leader election, k -leader selection
 - The *2-state MIS alg* is **not** efficient on general graphs
[Giakkoupis, Ziccardi 2023], [E, Louidor]
- Efficient self-stab UP algorithms for additional problems
 - Including *global problems*
- Self-stab *simulation* of SA under UP
- Runtime *lower bounds* beyond $\Omega(\log n)$
 - Exploit constant $|\delta|$
 - Exploit self-stab

GRAZIE MILLE