Fast Space Optimal Leader Election in Population Protocols

Leszek Gąsieniec, Grzegorz Stachowiak

Paul Huber

Introduction

- Population protocols refers to agents

- The emphasis is on the space complexity in fast leader election

- Main result of the paper:
  - New fast and space optimal leader election protocol

  - Parallel time: $O(log^2\ n)$ and $O(log\ log\ n)$ states for each agent

- Population protocols adopted to the model of Angulin et al.
- A population protocol terminates with success if the whole population eventually stabilises.
- In leader election is a single agent expected to conclude in a leader state.
- Leader election cannot be solved in sublinear time.

- Alistarh et al. show a lower bound of $\Omega(\mathit{log}\ n)$ states for any protocol which stabilises in $O(n^c)$ time, for any constant $c <= 1$.
- The new algorithm, the new results and lower bound from Angulin et al. provide a complete suit of protocols for the time and space optimal leader election and majority computation.
- The algorithm utilises a fast and small space reduction of potential leaders in the population.
- The main result is based on rapid computation of junta of leaders followed by fast election of a single leader.

Leader election

- Originally studied in networks with nodes having distinct labels.

- Focuses on the ring topology in synchronous as well as in asynchronous models.

- Also, in networks populated by mobile agents the leader election was studied first in networks with labeled nodes.

Preliminaries

- Population protocols defined on the complete graph of interactions.
- Random scheduler picks uniformly at random pairs of agents, which are anonymous.
- All agents start in the same initial state.
- Interactions refer to ordered pairs of agents (responder, initiator).
- Agents change their states (a, b) into (a', b')
  - $(a, b) \rightarrow (a', b')$

- Two complexity measures:
  1. space complexity
  2. time complexity

- The emphasis in this paper is on parallel time of the solution.

- This work aims at protocols formed of $O(n * poly\ log\ n)$ interactions.

One-way epidemics

- Refers to the population protocol with the state space $\{0, 1\}$.

- Transition rule $x, y \rightarrow \max\{x, y), y$.

- In order to conclude one-way epidemic (infect all agents) one needs $\theta(n\ log\ n)$ pairwise interactions with high probability.

Phase clock

- Subprotocol used to count off intervals of $\theta(log\ n)$ parallel time.
- Angluin et al. studied phase clocks under the assumption of having already determined a unique leader.
- Phase clocks also work when the unique leader is replaced by a junta of leaders.
- Once the phase clock is in motion, reduce to a single leader with the help of coin flipping combined with propagation via one-way epidemic
- Election of a single leader in expected $O(n\ log^2\ n)$ interactions

- Processed in two loops, one nested inside the other.
    - The internal loop operates in parallel time θ(*log n*) interactions, to distribute 1's via one-way epidemic
    - The external loop is used to count θ (*log n*) executions of the internal loop.
- The states of agents controlling the phase clock protocol are structured in pairs (x, b).
    - The entry b has value **leader** for leaders in the junta and **follower** for all other agents.
    - The entry x represents a phase from the set Zm = $\{0, 1, 2, \ldots, m-1\}$

Forming a junta

- The purpose of this protocol is to rapidly elect from n agents a junta of $O((n\ log\ n)^{1/2})$ leaders assuming each agent utilises $O(log\ log\ n)$ states.

- The states of agents are represented as pairs (l, a) where a $\in \{0, 1\}$

- The protocol stabilises when all agents conclude with a = 0.

- On the conclusion of this protocol there are $O((n\ log\ n)^{1/2})$ agents, with the highest computed value l.

- All agents start in the same state (l, a) = (0, 1).

- When an agent in state (0, 1) interacts with any agent in state (0, 1), the final state of the initiator is (1, 1) and (0, 0) of the responder.
  - $(0, 1), (0, 1) \rightarrow (0, 0), (1, 1)$

Leader election

- The protocols are combined to obtain a new fast population protocol for leader election.

- The protocol assumes a non-empty subset of agents which are candidates for leaders and this subset is gradually reduced to a singleton
  - The protocol consists of (log n) repetitions of the external loop.
  - Each formed of (n log n) interactions controlled by the ordinary mode of the phase clock.
  - During each repetition every candidate picks independently at random either bit 0 or 1 by tossing a fair coin.

- The protocol selects a unique leader during $\theta(log\ n)$ repetitions

- To realise this protocol a counter of $\theta(log\ n)$ repetitions must be implemented and a multibroadcast of $1s$ which requires $\theta(n\ log\ n)$ interactions.

- The leader election protocol starts with a single execution of protocol to form a junta which is followed by the leader reduction mechanism allowing to reduce the original junta team to a single leader.

- All agent enter the leader election protocol in the same state.

- The current state of an agent is represented by a vector (l, a, b, x, y).

- All agents start the leader election protocol with (l, a, b, x, y) = (0, 1, leader, 0, 0).

- All agents run *Forming junta* protocol in state (*l, a*), for as long as $b = leader$.
- As soon as b gets value follower, according to *Forming junta* protocol the state of the relevant agent becomes (0,0).
- This happens only when l is not at the highest level in the population.
- each agent should conclude the protocol in the first (n log n) interactions.

Phase clocks on different levels

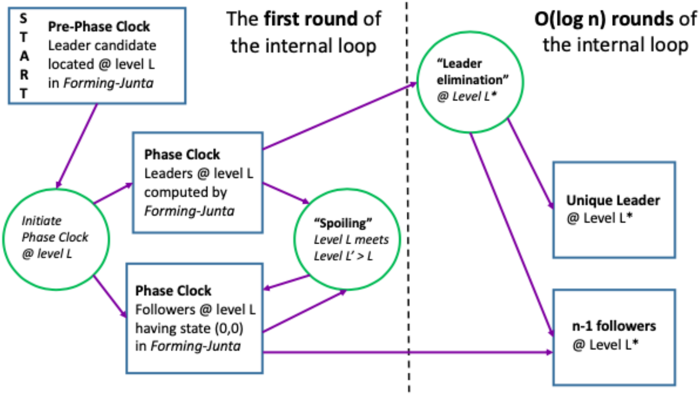
- Once value a becomes 0, the agent starts its phase clocks on level l as the leader.

- When an agent with a = 0 at level l interacts with an agent with the phase clocks on a higher level $l' > l$, its state is rewritten (l, a, b, x, y) $\leftarrow$ (l', 0, follower, 0, 0).

- The agent aligns its phase clocks in phase 0 on level l' and ends up in state (0, 0)

Conclusion

- The new fast and space efficient leader election in population protocols stabilises in (parallel) time $O(log^2\ n)$ when each agent is equipped with $O(log\ log\ n)$ states.



Thank you for your attention!