Brief Announcement: Fast Graphical Population Protocols

Dan Alistarh
IST Austria, Klosterneuburg, Austria

Rati Gelashvili
Novi Research, Menlo Park, CA, USA

Joel Rybicki
IST Austria, Klosterneuburg, Austria

Abstract

Let G be a graph on n nodes. In the stochastic population protocol model, a collection of n indistinguishable, resource-limited nodes collectively solve tasks via pairwise interactions. In each interaction, two randomly chosen neighbors first read each other's states, and then update their local states. A rich line of research has established tight upper and lower bounds on the complexity of fundamental tasks, such as majority and leader election, in this model, when G is a *clique*. Specifically, in the clique, these tasks can be solved *fast*, i.e., in $n \operatorname{polylog} n$ pairwise interactions, with high probability, using at most $\operatorname{polylog} n$ states per node.

In this work, we consider the more general setting where G is an arbitrary graph, and present a technique for simulating protocols designed for fully-connected networks in any connected regular graph. Our main result is a simulation that is *efficient* on many interesting graph families: roughly, the simulation overhead is polylogarithmic in the number of nodes, and quadratic in the conductance of the graph. As an example, this implies that, in any regular graph with conductance φ, both leader election and exact majority can be solved in $\varphi^{-2} \cdot n \operatorname{polylog} n$ pairwise interactions, with high probability, using at most $\varphi^{-2} \cdot \operatorname{polylog} n$ states per node. This shows that there are fast and space-efficient population protocols for leader election and exact majority on graphs with good expansion properties.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases population protocols, leader election, majority

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.43

Related Version *Full Version*: https://arxiv.org/abs/2102.08808

Funding *Joel Rybicki*: This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 840605.

1 Introduction

In distributed computing, *population protocols* [4] have become a popular model for investigating the collective computational power of large collections of communication-bounded agents with limited computational capabilities. This model consists of n identical agents, seen as finite state machines, and computation proceeds via pairwise interactions of the agents, which trigger local state transitions. The sequence of interactions is provided by a scheduler, which picks pairs of agents to interact. Upon every interaction, the selected agents observe each other's states, and then update their local states. The goal is to have the system reach a configuration satisfying a given predicate, while minimising the number of interactions (time complexity) and the number of states per node (space complexity).

Early work on population protocols focused on the computational power of the model under various interaction graphs [4, 5]. More recently, the focus has shifted to complexity, often in the form of trade-offs between time and space complexity, e.g. [3, 17, 1, 9, 18, 16, 2].

This line of work almost exclusively focuses on the *uniform* stochastic scheduler, where each interaction pair is chosen uniformly at random *among all pairs* of agents in the population, and the time complexity of a protocol is measured by the number of interactions needed to solve a task. However, many natural systems exhibit spatial structure and this structure can significantly influence the system dynamics.

Indeed, there is a separation in terms of computational power for population protocols in the clique versus other interaction graphs: connected interaction graphs can simulate adversarial interactions on the clique graph by shuffling the states of the nodes [4] and population protocols on some interaction graphs can compute a strictly larger set of predicates than protocols on the clique; see e.g. [6] for a survey of computability results.

By comparison, surprisingly little is known about the *complexity* of basic tasks in general interaction graphs under the stochastic scheduler. So far, only a handful of protocols have been analysed on general graphs. Existing analyses tend to be complex, and specialised to specific algorithms on limited graph classes [15, 11, 8]. This is natural: given the intricate dependencies which arise due to the underlying graph structure, the design and analysis of protocols in the spatial setting is understood to be challenging.

We provide a general approach showing that standard problems in population protocols can be solved *efficiently* under *graphical* stochastic schedulers, by leveraging solutions designed for complete graphs.

First, we give a general framework for simulating a large class of *synchronous* protocols designed for *fully-connected networks*, in the graphical stochastic population protocol model. Thus, the user can design efficient (and simple to analyse) synchronous algorithms on a clique model, and transport the analysis automatically to the population protocol model on a large class of interaction graphs. For instance, on any d-regular graph with edge expansion $\beta > 0$, the resulting overhead in parallel time and state complexity is in the order of $(d/\beta)^2 \cdot \operatorname{polylog} n$. As concrete applications, we show that for any d-regular graph with edge expansion $\beta > 0$, there exist protocols for leader election and exact majority that stabilise both in expectation and with high probability in $(d/\beta)^2 \cdot \operatorname{polylog} n$ parallel time, using $(d/\beta)^2 \cdot \operatorname{polylog} n$ states.

Second, to complement the results following from the simulation, we also show that, on any graph G with diameter $\operatorname{diam}(G)$ and m edges, leader election can be solved both in expectation and with high probability in $O(\operatorname{diam}(G) \cdot mn^2 \log n)$ parallel time, using a constant-state protocol. This result provides the first running time analysis of the protocol of [7].

Our reduction framework combines several techniques from different areas, and can be distilled down to the following ingredients.

We start by defining a simple *synchronous, fully-connected* model of communication for the n nodes, called the *k-token shuffling model.* This is the model in which the algorithm should be designed and analysed, and is similar, and in some ways simpler, relative to the standard population model. Specifically, nodes proceed in *synchronous* rounds, in which every node v first generates k tokens based on its current state. Tokens are then shuffled uniformly at random among the nodes. At the end of a round, every node v updates its local state based on its current state, and the tokens it received in the round. This simple model is quite powerful, as it can simulate both *pairwise* and *one-way* interactions between all sets of agents, for well-chosen settings of the parameter k.

Our key technical result is that any algorithm specified in this round-synchronous k-token shuffling model can be *efficiently* simulated in the graphical population model. Although intuitive, formally proving this result, and in particular obtaining bounds on the efficiency of the simulation, is non-trivial. First, to show that simulating *a single round* of the k-token shuffling model can be done efficiently, we introduce new type of *card shuffling process* [12, 10, 19], which we call the k-stack interchange process, and analyse its mixing time by linking it to random walks on the symmetric group.

Table 1 Protocols for exact majority (EM) and leader election (LE) for different graph classes. The state complexity is the number of states used by the protocol. The parallel time column gives the expected parallel time (expected number of interaction steps divided by n) to stabilise. (*) In [15], the running time of the protocol is bounded by the initial discrepancy in the inputs and the spectral properties of the contact rate matrix; bounds in terms of n are only given for select graph classes (paths, cycles, stars, random graphs and cliques). No sublinear in n bounds on parallel time are given in [15]. Protocols marked with ($\star$) stabilise also in non-regular graphs in poly(n) time.

Graph class	Task	States	Parallel time	Note
cliques	EM	4	$O(n \log n)$	[15]
	EM	$O(\log n)$	$\Theta(\log n)$	[13]
	LE	2	$\Theta(n)$	[14]
	LE	$\Theta(\log \log n)$	$\Theta(\log n)$	[9]
connected	EM	4	$\mathrm{poly}(n)$	[15, 8], (*)
	LE	6	$O(\mathrm{diam}(G) \cdot mn^2 \log n)$	**new** analysis of [7]
d-regular	EM	$(d/\beta)^2 \cdot \mathrm{polylog}\, n$	$(d/\beta)^2 \cdot \mathrm{polylog}\, n$	**new**, ($\star$)
	LE	$(d/\beta)^2 \cdot \mathrm{polylog}\, n$	$(d/\beta)^2 \cdot \mathrm{polylog}\, n$	**new**, ($\star$)

Second, to allow correct and efficient asynchronous simulation of the synchronous token shuffling model, we introduce two new gadgets: (1) a *graphical* version of *decentralised phase clocks* [1, 17], combined with (2) an *asynchronous* token shuffling protocol, which simulates the k-token interchange process in a graphical population protocol. The latter ingredient is our main technical result, as it requires both efficiently combining the above components, and carefully bounding the probability bias induced by simulating a synchronous model under asynchronous pairwise-random interactions.

Finally, we instantiate this framework to solve exact majority and leader election in the graphical setting. We provide simple token-shuffling protocols for these problems, as well as backup protocols to ensure their correctness in all executions.

Our results imply new and improved upper bounds on the time and state complexity of majority and leader election for a wide range of graph families. In some cases, they improve upon the best known upper bounds for these problems. Please see Table 1 for a systematic comparison. While our protocols guarantee *fast* stabilisation in regular graphs with high expansion, they will stabilise in polynomial expected time in *any connected graph*.

Our results suggest the existence of a similar complexity gap in the graphical setting. Specifically, on d-regular graphs with good expansion, such that $d/\beta \in \mathrm{polylog}\, n$, we provide polylogarithmic-time protocols for both leader election and exact majority. This opens a significant complexity gap relative to known constant-state protocols on graphs. For instance, the 4-state exact majority protocol for general graphs [15] requires $\Omega(n)$ parallel time even in regular graphs with high expansion, if node degrees are $\Theta(n)$. Yet, our protocols guarantee stabilisation in only $\mathrm{polylog}\, n$ parallel time in both low and high degree graphs, as long as d/β is at most $\mathrm{polylog}\, n$.

References

1 Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*. SIAM, 2018. doi:10.1137/1.9781611975031.144.

2 Dan Alistarh and Rati Gelashvili. Recent algorithmic advances in population protocols. *SIGACT News*, 49(3):63–73, 2018. doi:10.1145/3289137.3289150.

3 Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and exact majority in population protocols. In *Proc. 34th ACM Symposium on Principles of Distributed Computing (PODC 2015)*, pages 47–56, 2015.

4 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.

5 Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

6 James Aspnes and Eric Ruppert. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer, 2009.

7 Joffroy Beauquier, Peva Blanchard, and Janna Burman. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *International Conference on Principles of Distributed Systems*, pages 38–52. Springer, 2013. URL: `https://hal.archives-ouvertes.fr/hal-00867287v2`.

8 Petra Berenbrink, Tom Friedetzky, Peter Kling, Frederik Mallmann-Trenn, and Chris Wastell. Plurality consensus in arbitrary graphs: Lessons learned from load balancing. In *Proc. 24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57, pages 10:1–10:18, 2016. `doi:10.4230/LIPIcs.ESA.2016.10`.

9 Petra Berenbrink, George Giakkoupis, and Peter Kling. Optimal time and space leader election in population protocols. In *Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020)*, pages 119–129, 2020.

10 Pietro Caputo, Thomas M. Liggett, and Thomas Richthammer. Proof of Aldous' spectral gap conjecture. *Journal of the American Mathematical Society*, 23(3):831–851, 2010.

11 Colin Cooper, Tomasz Radzik, Nicolás Rivera, and Takeharu Shiraga. Fast plurality consensus in regular expanders. In *Proc. 31st International Symposium on Distributed Computing (DISC 2017)*, pages 13:1–13:16, 2017. `doi:10.4230/LIPIcs.DISC.2017.13`.

12 Persi Diaconis and Laurent Saloff-Coste. Comparison techniques for random walk on finite groups. *The Annals of Probability*, 21(4):2131–2156, 1993.

13 David Doty, Mahsa Eftekhari, and Eric Severson. A stable majority population protocol using logarithmic time and states, 2020. `arXiv:2012.15800`.

14 David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *Distributed Computing*, 31(4):257–271, 2018.

15 Moez Draief and Milan Vojnović. Convergence speed of binary interval consensus. *SIAM Journal on Control and Optimization*, 50(3):1087–1109, 2012.

16 Robert Elsässer and Tomasz Radzik. Recent results in population protocols for exact majority and leader election. *Bulletin of the EATCS*, 126, 2018. URL: `http://bulletin.eatcs.org/index.php/beatcs/article/view/549/546`.

17 Leszek Gąsiniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, 2018. `doi:10.1137/1.9781611975031.169`.

18 Leszek Gąsiniec, Grzegorz Stachowiak, and Przemyslaw Uznański. Time and space optimal exact majority population protocols, 2020. `arXiv:2011.07392`.

19 Johan Jonasson. Mixing times for the interchange process. *Latin American Journal of Probability and Mathematical Statistics*, 9(2):667–683, 2012.