Brief Announcement: New Clocks, Fast Line Formation and Self-Replication Population Protocols

Leszek Gąsieniec
University of Liverpool, United Kingdom

Paul Spirakis
University of Liverpool, United Kingdom

Grzegorz Stachowiak
Uniwersytet Wrocławski, Poland

Abstract

In this paper we consider a known variant of the standard population protocol model in which agents can be connected by edges, referred to as the *network constructor* model. During an interaction between two agents the relevant connecting edge can be formed, maintained or eliminated by the transition function. The state space of agents is fixed (constant size) and the size n of the population is not known, i.e., not hard-coded in the transition function.

Since pairs of agents are chosen uniformly at random the status of each edge is updated every $\Theta(n^2)$ interactions in expectation which coincides with $\Theta(n)$ parallel time. This phenomenon provides a natural lower bound on the time complexity for any non-trivial network construction designed for this variant. This is in contrast with the standard population protocol model in which efficient protocols operate in $O(\text{poly} \log n)$ parallel time.

The main focus in this paper is on efficient manipulation of linear structures including formation, self-replication and distribution (including pipelining) of *complex information* in the adopted model.

- We propose and analyse a novel edge based phase clock counting parallel time $\Theta(n \log n)$ in the network constructor model, showing also that its leader based counterpart provides the same time guaranties in the standard population protocol model. Note that all currently known phase clocks can count parallel time not exceeding $O(\text{poly} \log n)$.
- The new clock enables a nearly optimal $O(n \log n)$ parallel time spanning line construction (a key component of universal network construction), which improves dramatically on the best currently known $O(n^2)$ parallel time protocol, solving the main open problem in the considered model [9].
- We propose a new *probabilistic bubble-sort* algorithm in which random comparisons and transfers are allowed only between the adjacent positions in the sequence. Utilising a novel potential function reasoning we show that rather surprisingly this probabilistic sorting (via conditional pipelining) procedure requires $O(n^2)$ comparisons in expectation and whp, and is on par with its deterministic counterpart.
- We propose the first population protocol allowing self-replication of a *strand* of an arbitrary length k (carrying a k-bit message of size independent of the state space) in parallel time $O(n(k + \log n))$. The pipelining mechanism and the time complexity analysis of the strand self-replication protocol mimic those used in the probabilistic bubble-sort. The new protocol permits also simultaneous self-replication, where l copies of the strand can be created in time $O(n(k + \log n) \log l)$. Finally, we discuss application of the strand self-replication protocol to pattern matching.

Our protocols are always correct and provide time guaranties with high probability defined as $1 - n^{-\eta}$, for a constant $\eta > 0$.

2012 ACM Subject Classification Theory of computation Distributed algorithms; Theory of computation → Distributed algorithms

Keywords and phrases Population protocols, network constructors, probabilistic bubble-sort, self-replication

36th International Symposium on Distributed Computing (DISC 2022).
Editor: Christian Scheideler; Article No. 15; pp. 15:1–15:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.DISC.2022.15

Category Brief Announcement

Related Version *Full Version*: https://arxiv.org/abs/2111.10822

1 Introduction

The model of *population protocols* originates from the seminal paper of Angluin et al. [1]. This model provides tools for the formal analysis of *pairwise interactions* between simple indistinguishable entities referred to as *agents*. The agents are equipped with limited storage, communication and computation capabilities. When two agents engage in a direct interaction their states are amended according to the predefined *transition function*. The weakest possible assumptions in population protocols, also adopted here, limit the state space of agents to a fixed (constant) size disallowing utilisation of the size of the population n in the transition function. In the *probabilistic variant* of population protocols adopted in this paper, in each step the *random scheduler* selects from the whole population an ordered pair of agents formed of the *initiator* and the *responder*, uniformly at random. The lack of symmetry in this pair is a powerful source of random bits often used by population protocols. In this variant, in addition to *state utilisation* one is also interested in the *time complexity* of the proposed solutions. In more recent work on population protocols the focus is on *parallel time* defined as the total number of pairwise interactions (sequential time) leading to the solution divided by the size n of the whole population. For example, a core dissemination tool in population protocols known as *one-way epidemic* [2] distributes simple (e.g., 0/1) messages to all agents in the population utilising $\Theta(n \log n)$ interactions or equivalently $\Theta(\log n)$ parallel time. The parallel time is meant to reflect on massive parallelism of simultaneous interactions. While this is a simplification [4], it provides a good estimation on locally observed time expressed in the number of interactions each agent was involved in throughout the computation process.

Unless stated otherwise we assume that any protocol starts in the predefined *initial configuration* with all agents being in the same *initial state*. A population protocol *terminates with success* if the whole population stabilises eventually, i.e., it arrives at and stays indefinitely in the *final configuration* of states representing the desired property of the solution.

1.1 Our results and their significance

We study here several central problems in distributed computing by focusing on the adopted variant of population protocols. These include the concept of *phase clocks*, a distributed synchronisation tool with good space and accuracy guarantees. The first study of leader based $O(1)$ space phase clocks can be found in the seminal paper by Angluin *et al.* in [2]. Further extensions including the junta based clock and nested clocks counting any $\Theta(\text{poly} \log n)$ parallel time were analysed in [6]. In a very recent work [5] Doty *et al.* study constant resolution phase clocks utilising $O(\log n)$ states as the main engine in the optimal *majority* computation protocols. In this work we propose and analyse a new phase clock based on a matching allowing to count $\Theta(n \log n)$ parallel time. This is the first clock able to confirm the conclusion of the slow leader election protocol based on direct duels between the (remaining) leader candidates. We also propose an edge-less variant of this clock based on the computed leader. This clock powers a nearly optimal $O(n \log n)$ parallel time spanning line construction (a key component of universal network construction), improving dramatically on the best currently known $O(n^2)$ parallel time protocol, solving the main open problem from [9].

We also consider a probabilistic variant of the classical bubble-sort algorithm, in which any two consecutive positions in the sequence are chosen for comparison uniformly at random. We show that rather surprisingly this variant is on par with its deterministic counterpart as it requires $\Theta(n^2)$ random comparisons whp. While this new result is of an independent algorithmic interest, together with the edge-less clock they conceptually power the strand (line-segment carrying information) self-replication protocol studied at the end of this paper.

In a wider context, *self-replication* is a property of a dynamical system which allows reproduction. Such systems are of increasing interest in biology, e.g., in the context of how life could have begun on Earth [8], but also in computational chemistry [10], robotics [7] and other fields. In our case a larger chunk of information (well beyond the limited state capacity) is stored collectively in a strand (line-segment) of agents. Such strands may represent strings in pattern matching or a large value, or a code in more complex distributed process. In such cases the replication mechanism facilitates an improved accessibility to this information. We propose the first strand self-replication protocol allowing to reproduce a strand of size k in parallel time $O(n(k + \log n))$. This protocol permits simultaneous replication, where l copies of a strand can be generated in parallel time $O(n(k + \log n) \log l)$. The parallelism of this protocol is utilised in efficient pattern matching.

The full version of this paper including definitions, algorithms and formal proofs is available on arXiv.org [3].

2 Open Problems

We conjecture that our line formation protocol is optimal, i.e., no population protocol can construct a line containing all agents in parallel time $o(n \log n)$ whp. Going beyond the proposed strand self-replication protocol one could investigate whether other network structures can self-replicate and at what cost.

References

1. D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. In *Proc. PODC 2004*, pages 290–299, 2004.
2. D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. *Distributed Comput.*, 21(3):183–199, 2008.
3. L. Gąsieniec, P.G. Spirakis, and G. Stachowiak. New clocks, optimal line formation and efficient replication population protocols (making population protocols alive). *CoRR*, abs/2111.10822, 2021. URL: `https://arxiv.org/abs/2111.10822`, `arXiv:2111.10822`.
4. A. Czumaj and A. Lingas. On truly parallel time in population protocols. *CoRR*, abs/2108.11613, 2021.
5. D. Doty, M. Eftekhari, L. Gąsieniec, E.E. Severson, G. Stachowiak, and P. Uznański. A time and space optimal stable population protocol solving exact majority. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 1044–1055, 2021.
6. L. Gąsieniec and G. Stachowiak. Enhanced phase clocks, population protocols, and fast space optimal leader election. *J. ACM*, 68(1):2:1–2:21, 2021.
7. R.A. Freitas Jr and R.C. Merkle. *Kinematic Self-Replicating Machines*. Landes Bioscience, Georgetown, TX, 2004.
8. T.A. Lincoln and G.F. Joyce. Self-sustained replication of an rna enzyme. In *Science, Vol 323, Issue 5918*, pages 1229–1232. American Association for the Advancement of Science, 2009.
9. O. Michail and P. Spirakis. Simple and efficient local codes for distributed stable network construction. *Distributed Computing*, 29(3):207–237, 2016.
10. E. Moulin and N. Giuseppone. Dynamic combinatorial self-replicating systems. In *Constitutional Dynamic Chemistry*, pages 87–105. Springer, 2011.