Selective Population Protocols

Adam Gańczorz[1*][0000-0001-9656-1643], Leszek Gąsieniec[2][0000-0003-1809-9814], Tomasz Jurdziński[1*][0000-0003-1908-9458], Jakub Kowalski[1†][0000-0003-1932-4278], and Grzegorz Stachowiak[1*][0000-0002-3128-4689]

[1] University of Wrocław, ul. Joliot-Curie 15 50-383 Wrocław, Poland
{adam.ganczorz,tju,jko,gst}@cs.uni.wroc.pl
[2] University of Liverpool, Liverpool, United Kingdom
l.a.gasieniec@liverpool.ac.uk

Abstract. The model of population protocols provides a universal platform to study distributed processes driven by pairwise interactions of anonymous agents. While population protocols present an elegant and robust model for randomized distributed computation, their efficiency wanes when tackling issues that require more focused communication or the execution of multiple processes. To address this issue, we propose a new, selective variant of population protocols by introducing a partition of the state space and the corresponding conditional selection of responders. We demonstrate on several examples that the new model offers a natural environment, complete with tools and a high-level description, to facilitate more efficient solutions.

In particular, we provide fixed-state stable and efficient solutions to two central problems: leader election and majority computation, both with confirmation. This constitutes a separation result, as achieving stable and efficient majority computation requires $\Omega(\log n)$ states in standard population protocols, even when the leader is already determined. Additionally, we explore the computation of the median using the comparison model, where the operational state space of agents is fixed, and the transition function determines the order between (arbitrarily large) hidden keys associated with interacting agents. Our findings reveal that the computation of the median of n numbers requires $\Omega(n)$ time. Moreover, we demonstrate that the problem can be solved in $O(n \log n)$ time, both in expectation and with high probability, in standard population protocols. In contrast, we establish that a feasible solution in selective population protocols can be achieved in $O(\log^4 n)$ time.

Keywords: Population Protocol · Stability · Conditional Interactions · Median Computation.

---

*Supported by the National Science Centre, Poland under project number 2020/39/B/ST6/03288.

†Supported in part by the National Science Centre, Poland under project number 2021/41/B/ST6/03691.

1 Introduction

The standard model of population protocols originates from the seminal paper [5], providing tools suitable for the formal analysis of *pairwise interactions* between simple, indistinguishable entities referred to as *agents*. These agents are equipped with limited storage, communication, and computation capabilities. When two agents engage in a direct interaction, their states change according to the predefined *transition function*, which is an integral part of the population protocol. The weakest possible assumptions in population protocols pertain to the fixed (constant size) operational *state space* of agents, and the size of the population n is neither known to the agents nor hard-coded in the transition function. It is assumed that a protocol starts in the predefined *initial configuration* of agents' states representing the input, and it *stabilizes* in one of the *final configurations* of states representing the solution to the considered problem. In the *probabilistic variant* of population protocols adopted here, in each step of a protocol, the *random scheduler* selects an ordered pair of agents: the *initiator* and the *responder*, which are drawn from the whole population uniformly at random. The lack of symmetry in this pair is a powerful source of random bits utilized by population protocols. In the probabilistic variant, in addition to efficient *state utilization*, one is also interested in the *time complexity*, where the *sequential time* refers to the number of interactions leading to the stabilization of a protocol in a final configuration. More recently, the focus has shifted to the *parallel time*, or simply the *time*, defined as the sequential time divided by the size nn of the whole population. The (parallel) time reflects on the parallelism of simultaneous independent interactions of agents utilized in *efficient population protocols* that stabilize in time $O(\text{poly} \log n)$. All protocols presented in this paper are *stable* (always correct) and guarantee stabilization time with high probability (whp) defined as $1 - n^{-\eta}$, for a constant $\eta > 0$.

There are already several efficient protocols known for solving central problems in distributed computing, including *leader election* [1, 13, 8], *majority computation* [2, 12], and the *plurality problem* [7]. While these protocols are efficient in terms of time, they rely on non-constant state space utilization and operate indefinitely. That is, they are not able to declare stabilization with probability 1. Moreover, the most efficient protocols are often non-trivial and hard to analyze. One can circumvent some of these deficiencies by relaxing probabilistic expectations, e.g., by dropping assumptions about the necessity of stabilization in protocols with predefined input [19], as well as in self-stabilizing protocols [9]. While such relaxation is beneficial, it does not solve some major deficiencies of the standard model, including depleting in time the number of meaningful interactions, limited computational power, and inefficient space-time trade-offs.

In order to circumvent some of these deficiencies, we propose a new *selective* variant of population protocols by imposing a simple *group (partition) structure* on the state space together with a conditional choice of the responder during random interacting pair selection. This model provides a natural extension of *passive mobile* sensor networks adopted in [5], where the focus is on single channel communication. In the new model the agents communicate over multiple communication channels, where each channel corresponds to one of a fixed number of groups (partitions) of the state space. Specifically, only agents currently listening on some specific communication channel C (their states belong to the corresponding group of states $\mathcal{G}_C$) are able to receive and respond to messages transmitted over this channel by agents with state indicating $\mathcal{G}_C$ as the *target group*. Alternative models with biased communication were previously used in the context of stochastic chemical reaction networks in [23] and data collection with non-uniform schedulers in [9]. The adopted selective model also refers to biased choices in nature studied earlier, in the context of small-world phenomena, where closer location in space results in a more likely interaction [18], or social preference, where agents with a greater array of similar attributes are more likely to know one another and, in turn, to interact [17]. A different motivation to study selective population protocols refers to more structural variant of population protocols known as *network constructors*, in which agents are allowed to be connected. As the expected parallel time to manipulate a specific edge is $\Theta(n)$, see, e.g., [20, 15], the design of truly efficient protocols in this model is not currently feasible. Utilizing the concept of selective population protocols, one can give a higher probabilistic bias to interactions along existing edges, enabling more efficient computation, comparable to graphical population protocols [3, 4].

Our contribution In this paper, we present initial studies on the (parallel) efficiency and stability of selective population protocols. We begin by discussing fundamental properties of this new promising variant, introducing the notion of *fragmented parallel time* as an equivalent measure to parallel time in the standard population protocol model. Additionally, we highlight that selective protocols offer a natural mechanism for deterministic emptiness (zero) testing. It is known, as indicated in [6], that such a test enables efficient simulation of $O(\log n)$-space Turing Machines with high probability. In contrast, we highlight that such simulations in selective protocols are not only efficient but also stable. Selective protocols can be utilized to design algorithms within this class that are both efficient and stable. Furthermore, we present fixed-state efficient and stable solutions to two central problems: leader election and majority computation (with confirmation, i.e., all agents stabilize while being aware of the conclusion of the process). This result is noteworthy as stable efficient majority computation requires $\Omega(\log n)$ states in standard population protocols [2], even when the leader is given. We also introduce the first non-trivial study on median computation in population protocols. We adopt a comparison model in which the operational state space of agents is constant, and the transition function determines the order between (arbitrarily large) hidden keys associated with the interacting agents. We demonstrate that computing the median of n numbers requires $\Omega(n)$ parallel time and the problem can be solved in $O(n \log n)$ parallel time in expectation and with high probability (whp) in standard population protocols. In contrast, we present an efficient median computation in selective population protocols, achieving $O(\log^4 n)$ parallel time. Furthermore, we delve into suitability of selective protocols for the high-level design of algorithms.

Please note that due to space restrictions the proofs of Lemmas 3, 4, 5, 6, 7, 8, 9, and Proposition 1 can be found in the full version of this paper [14].

2 Selective Population Protocols

As discussed in Section 1, in the standard population protocol model, the random scheduler draws consecutive pairs of interacting agents uniformly at random from the entire population. This is done irrespective of whether the states of interacting agents match some rule of the transition function or not. Consequently, many random pairwise interactions do not result in a transition and, in turn, do not bring the population closer to a final configuration.

In *selective population protocols* random interactions are scheduled differently. Specifically, the fixed-state space of agents $\mathcal{S}$ is partitioned into l groups of states $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_l$. In addition, any state s is mapped onto its *target group* $\mathcal{G}_{i(s)}$. We say that an interaction is *internal* if $s \in \mathcal{G}_{i(s)}$, and is *external* otherwise. This mapping is used during an attempt to form a random pair of interacting agents. The random scheduler first draws the initiator in state s uniformly from the whole population. This is followed by drawing the responder uniformly from all agents (different to the initiator) currently being in any state t belonging to the target group $\mathcal{G}_{i(s)}$. Such interaction is denoted by $s + \mathcal{G}_{i(s)}|t$, where $t = null$ when no responder is currently available in any state of $\mathcal{G}_{i(s)}$. The rules of the transition function refer to two types of outcomes of interaction attempts:

1. *Biased communication*
   Meaningful interaction (successful biased interaction attempt)
   $s + \mathcal{G}_{i(s)}|t \to s' + t'$.
   The purpose of meaningful interactions is to advance and in turn to maintain efficiency of the computing process.

2. *Interaction availability test*
   Emptiness (zero) test (unsuccessful external interaction attempt)
   $s + \mathcal{G}_{i(s)}|null \to s'$.
   Singleton test (unsuccessful internal interaction attempt)
   $s + \mathcal{G}_{i(s)}|null \to s'$.
   The two tests primarily confirm completion of computation processes.

One-way epidemic Consider a communication primitive known as *one-way epidemic* [6] in which state 1 of the source agent is propagated to all other agents initially being in state 0. The transition function has only one rule in the standard population protocol model $1 + 0 \to 1 + 1$. It is known that such epidemic process is stable and efficient, i.e., one-way epidemic stabilises with the correct answer in parallel time $O(\log n)$ whp, however during the final stages of the epidemic process the expected fraction of *meaningful interactions* decreases dramatically. Assume, that the state space $S = \{0, 1\}$ is partitioned into two singleton groups $\mathcal{G}_0 = \{0\}$ and $\mathcal{G}_1 = \{1\}$ in the new variant, and we have two transition rules instead:

(1) $1 + \mathcal{G}_0 \| 0 \longrightarrow 1 + 1$	(2) $1 + \mathcal{G}_0 \| null \longrightarrow Stop$

Now every interaction initiated by an agent in state 1 is either meaningful, when there are still uninformed agents, or changes the initiator's state to *Stop*, which indicates the end of the epidemic process, and in turn the next stage of computation not requiring group $\mathcal{G}_0$.

2.1 Beyond Presburger Arithmetic

We find in [6] that the emptiness test, also known as *zero test*, is a powerful tool enabling efficient simulation of $O(\log n)$-space Turing Machine. The two-stage randomised simulation from [6] is based on simulation of *Register Machines* known to be equivalent with $O(\log n)$-space Turing Machines [21]. This approach hinges on the presence of a unique leader, crucial for achieving efficient and stable computations, which efficient computation requires at least $\Omega(\log \log n)$ states [1]. An alternative randomized two-stage simulation of Turing Machines, detailed in [23] within the context of a related *stochastic chemical reaction network* model, utilises the concept of *clockwise Turing Machines* [22]. Both simulations rely on zero tests, the correctness of which can be assured only with high probability in the adopted models, rendering them unsuitable for deployment in stable protocols. In contrast, selective protocols equipped with deterministic emptiness test provide a suitable platform for the design of efficient and stable *fixed-state* solutions in $O(\log n)$-space complexity class.

As the primary focus of this paper centers on the (parallel) *efficiency* of selective population protocols, and the computational power of such protocols is inherited from standard population protocols, we direct the reader to [6] for full simulation details. Instead, our current study delves into the parallelism of selective protocols, presenting several separation results. This includes *majority computation*, where any efficient stable algorithm in standard protocols requires $\Omega(\log n)$ states while a fixed-state space allows for an $O(\log n)$-time stable solution, as demonstrated in Section 2.4.

Several efficient algorithms presented in this paper follow a more direct approach, relying on a single stabilization process. Examples include the efficient and stable leader election and majority computation discussed in Sections 2.3 and 2.4, respectively. However, in more complex solutions, the need arises for a leader to act as the *"program counter,"* overseeing the proper execution of potentially numerous individual stabilization processes encoded in the transition function in the correct order. This encompasses the preparation of input for each individual process, ensuring its proper termination, and further interpreting the output. It's worth noting that, due to state partitioning in selective protocols, each individual stabilization process can be executed on a distinct partition of states. This allows several processes to run efficiently at the same time, as demonstrated in the efficient ranking protocol presented in Section 4, where multiple leaders are employed. The leader is also responsible for translating the output from one process to the input of its successor. This is achieved by rewriting states (from one partition to another) via one-way epidemic. Ultimately, the termination of any process, including rewriting, is recognized through either an emptiness or singleton test.

2.2 Parallelism of Selective Protocols

Recall that in population protocols, the (parallel) time of a sequence I of interactions is defined as $|I|/n$. This definition is motivated by the observation that in a sequence of xn interactions, each agent has, on average, x interactions. However, it is noteworthy that only for $x = \Omega(\log n)$ does each agent engage in $\Theta(x)$ interactions whp. An interesting finding presented in [11] demonstrates that in this latter case, the sequence I can be simulated in time $\Theta(x)$ on a parallel computer whp. In the new variant, where the choice of the responder is likely biased, we must adopt a more nuanced definition of parallelism.

Fragmented Parallel Time In the novel selective variant of population protocols, the initiator is uniformly chosen at random. Consequently, in a sufficiently long sequence of interactions, any agent serves as the initiator with the same frequency, aligning with the pattern observed in the standard model. This stands in contrast to the selection of responders, where certain agents are more likely to be chosen than others. For example, consider an epidemic process with the state space $S = \{0, 1, 1^*\}$ partitioned into groups: $\mathcal{G}_0 = \{0\}$ with uninformed agents, $\mathcal{G}_1 = \{1\}$ with active informers, and $\mathcal{G}_{1^*} = \{1^*\}$ with informed and already rested agents, governed by two transition rules:

(1) $0 + \mathcal{G}_1|1 \longrightarrow 1^* + 1$ (2) $1 + \mathcal{G}_0|null \longrightarrow 1^*$

If in the initial configuration there is exactly one informed agent in state 1, all other agents in state 0 contact this agent to get informed and rest, see rule (1). In the last meaningful interaction rule (2) rests the unique informer. While the number of interactions before stabilisation with all agents resting in state 1^* is $O(n \log n)$ whp, the parallelism of this epidemic process is very poor as only one agent informs others as the responder.

This potential imbalance in the workload of individual agents can be captured by tracking the frequency at which agents act as responders. To handle this imbalance, we propose a more subtle definition of (parallel) time. This new definition is whp asymptotically equivalent to the definition and the properties of time used in the standard model, see Lemma 1.

Definition 1 (Fragmented parallel time). *Consider ways to divide the sequence of interactions I into subsequent disjoint* chunks*, where each chunk is a sequence of consecutive interactions in which any agent has at most* $10 \ln n$ *interactions as the responder. If the minimum number of chunks for such divisions is k, then we say that the* fragmented parallel time*, or in short the* fragmented time*, is $T_F = k \ln n$.*

Recall that η is the quality parameter in the definition of high probability.

Lemma 1. *Consider a sequence of interactions I in the standard population protocol model executed in time $T = |I|/n$. If the fragmented time $T_F = k \ln n$, for $k > 11\eta/35$ and large enough n, then $T/10 \le T_F \le 2T$ whp.*

Proof. The total number of interactions during fragmented time T_F does not exceed $10kn \ln n$, ensuring $\frac{T}{10} \le T_F$. It remains to show that $T_F \le 2T$ whp.

Let us first estimate the probability that a given chunk corresponds to time smaller than $\ln n$. This probability is not greater than the probability that in time $\ln n$ (starting at the beginning of the chunk) some agent has the responder type interactions greater than $10 \ln n$. By Chernoff bound[§], the probability that in time $\ln n$ a given agent experiences $X > 10 \ln n = 10\mathbb{E}X$ interactions as the responder can be estimated by

$$\begin{aligned}\Pr(X > 10 \ln n) &= \Pr\left(X > (1+9)\mathbb{E}X\right) \\ &< \exp\left(-\frac{9^2}{2+9}\mathbb{E}X\right) = \exp\left(-\frac{81}{11}\ln n\right) = n^{-81/11}.\end{aligned}$$

By the union bound the probability that in time $\ln n$ some agent interacts as the responder more than $10 \ln n$ times is smaller than $n^{-70/11}$. Thus, for n large enough and $11\eta/35 < k$, the probability that at least half of k chunks correspond to time smaller than $\ln n$ does not exceed

$$\binom{k}{k/2}\left(n^{-70/11}\right)^{k/2} < 2^k n^{-35k/11} < (2^k n^{-k/5})n^{-\eta} < n^{-\eta}.$$

In turn, whp we obtain time at least $\frac{k}{2}\ln n$, and $T_F \le 2T$. □

We introduce a lemma for analyzing fragmented time in the new model, crucial for the examination of leader election and majority computation protocols.

Lemma 2. *Consider an interval of interactions I, s.t., $|I| > \frac{110}{35}\eta n \ln n$. If every agent acts as the responder in an external interaction in I at most once, then the fragmented parallel time of I is $\Theta(|I|/n)$ whp.*

Proof. The total number of fragmented time chunks is at least $k \ge \frac{|I|}{10n \log n}$, thus $T_F \ge \frac{|I|}{10n}$. We demonstrate that $T_F \le \frac{2|I|}{n}$ with high probability.

Consider a fixed agent in a given interaction. We first observe that the probability of an event A, that an interaction is internal and this agent acts as the responder is at most $1/n$. For an agent belonging to a group of size 1, when this agent can be counted as both the initiator and the responder, the probability of event A does not exceed $1/n$. For an agent belonging to a group of size $g > 1$ this probability is at most $\frac{g-1}{n} \cdot \frac{1}{g-1} = \frac{1}{n}$.

[§]We utilise the Chernoff bound variant: $\Pr\left(X > (1+\delta)\mathbb{E}X\right) < \exp\left(-\delta^2\mathbb{E}X/(1+\delta)\right)$ for $\delta > 0$.

Let us divide all interactions into maximal subseries such that for any fixed agent there are at most $10 \ln n - 1$ events A involving this agent. Note that these subseries are simultaneously chunks of interactions in which any agent acts as the responder at most $10 \ln n$ times since any agent is a responder in I in an external interaction at most once. Let us first estimate the probability that a given subseries has less than $n \ln n$ interactions. This probability is not greater than the probability that during $n \ln n$ interactions (counting from the beginning of the subseries) event A happens for some agent at least $10 \ln n$ times. Using calculations from Lemma 1, one can estimate that this probability for a specific agent is smaller than $n^{-81/11}$.

By the union bound the probability that in a given subseries event A occurs for some agent at least $10 \ln n$ times does not exceed $n^{-70/11}$. Analogously to the proof of Lemma 1, for sufficiently large n and $k \geq \frac{|I|}{10n \ln n} > \frac{11\eta}{35}$, the probability that at least half of the k subseries correspond to times smaller than $\ln n$ is $n^{-\eta}$.

As this occurs with negligible probability, we get time at least $\frac{2|I|}{n}$ whp. □

Recall that if a group is a singleton, an attempt to execute pairwise interaction within this group fails. This is observed by the initiator via singleton test. Note also that such failed interactions do not affect parallelism as each failed interaction is attributed to the initiator. The next lemma enables the analysis of more complex protocols using the leader.

Lemma 3. *Consider an interval of interactions I, s.t., $|I| > \frac{60}{13}\eta n \ln n$. Assume also that every agent acts as the responder in an external interaction, which is not initiated by the leader in $|I|$ at most once, then the fragmented time of I is $\Theta(|I|/n)$ whp.*

2.3 Leader Election

In *leader election* (with confirmation) at least one agent from the initial configuration is a candidate to become the *unique leader*, and all other agents start as followers. The main goal in leader election is to distinguish and report selection of the unique leader, and to declare all other agents as followers. The state space of the leader election protocol presented below is $S = \{L, L^*, F, F^*\}$, where all initial leader and follower candidates are in states L and F, respectively. The remaining states include L^* referring to the confirmed unique leader, and F^* utilised by confirmed followers. The state space is partitioned into two groups $\mathcal{G}_0 = \{L, L^*, F^*\}$ and $\mathcal{G}_1 = \{F\}$.

LE-protocol: As at least one agent starts in state L, these agents target group $\mathcal{G}_0$ using a double rule (1) and (2) and when state L^* is eventually reached, with exactly one agent being in this state, the epidemic process defined by the transition rules (3)-(4) informs all followers about successful leader election.

(1) $L + \mathcal{G}_0 \vert L \longrightarrow L + F$	(3) $L^* + \mathcal{G}_1 \vert F \longrightarrow L^* + F^*$
(2) $L + \mathcal{G}_0 \vert null \longrightarrow L^*$	(4) $F^* + \mathcal{G}_1 \vert F \longrightarrow F^* + F^*$

Lemma 4. *The fragmented time of fixed-state LE-protocol is $O(\log n)$ whp.*

2.4 Majority Computation

The state space of the majority protocol is $S = \{G, G^*, R, R^*, N\}$, and in the initial configuration each agent is either in state G or R. The main goal is to decide which subpopulation of agents in state G or R is greater than the other. If the subpopulation in state G is greater, all agents are expected to stabilise in state G^*. Otherwise, they must stabilise in state R^*. The majority protocol **M-protocol** described below uses also neutral state N. The state space is partitioned into three groups $\mathcal{G}_R = \{R, R^*\}$, $\mathcal{G}_N = \{N\}$, and $\mathcal{G}_G = \{G, G^*\}$.

M-protocol: The protocol has the following transition rules:

(1) $R + \mathcal{G}_G \mid G \longrightarrow N + N$	(4) $G + \mathcal{G}_\mathcal{R} \mid null \longrightarrow G^*$
(2) $R + \mathcal{G}_G \mid null \longrightarrow R^*$	(5) $R^* + \mathcal{G}_\mathcal{N} \mid N \longrightarrow R^* + R^*$
(3) $G + \mathcal{G}_\mathcal{R} \mid R \longrightarrow N + N$	(6) $G^* + \mathcal{G}_\mathcal{N} \mid N \longrightarrow G^* + G^*$

Transition rules (1) and (3) instruct agents in states R and G to become neutral for as long as pairs $R + G$ and $G + R$ can be formed. As soon as one of these states is no longer present in the population either rule (2) or (4) is used to change state R to R^* or G to G^*, respectively. In addition, either rule (5) or (6) is used to change neutral state to R^* or G^*, respectively. Alternatively, if all states G and R disappear after application of rules (1) and (3), the population stabilises in the neutral state N.

Lemma 5. *The fragmented time of fixed-state M-protocol is $O(\log n)$ whp.*

3 Computing the Median

In this section, we consider computing the median of n distinct keys, each of which is held by one of the n agents. For agents a, b belonging to the set S of agents, the relation $b < a$ denotes key$(a) <$ key(b). We adopt here a *comparison model* in which the transition function depends not only on the states of the agents, but also on the order of their keys. The keys are hidden and there is no other way to access them. The number of states remains fixed. A similar limited use of large keys can be found in *community protocols* in [16] to handle Byzantine failures.

For any agent $c \in S$, let $\mathbb{A}_c$ and $\mathbb{B}_c$ be the set of all agents above and below c respectively. The agent m is the unique *median* if $|\mathbb{B}_m| - |\mathbb{A}_m| = 0$, for odd n, or one of the two medians if $||\mathbb{B}_m| - |\mathbb{A}_m|| = 1$, for even n. In this version we assume that all keys are different and n is odd. The arbitrary case requires minor amendments, as the answer may refer to two agents. Before we consider selective protocols, we first consider median computation in comparison model with a standard random scheduler.

3.1 Median Computation in Standard Model

Theorem 1. *Finding the median in the comparison model requires $\Omega(n)$ time in expectation.*

Proof. Assume that n is odd and agents a_1 and a_2 share between themselves the median and the key immediately succeeding it (in the total order of keys). One can observe that before the first interaction between a_1 and a_2, all consecutive configurations of states of all agents are independent from whether a_1 or a_2 is associated with the median. Thus before the first interaction between a_1 and a_2 no algorithm can declare either of them as the median. And since the expected number of interactions preceding the first interaction between a_1 and a_2 is $\Omega(n^2)$, the thesis of the theorem follows. □

Now we formulate an almost optimal median population protocol in the adopted model. All agents start this protocol in neutral state N. In due course, agents change their states, s.t., eventually all agents associated with keys smaller than the median end up in state B, those with keys greater than the median in state A, and the median conclude in state N. The median protocol uses the following symmetric transition function:

(1) $N + N \xrightarrow{<} B + A$ ▷ initialisation	(3) $A + N \xrightarrow{<} N + A$ ▷ fix order
(2) $A + B \xrightarrow{<} B + A$ ▷ fix order	(4) $N + B \xrightarrow{<} B + N$ ▷ fix order

Note that there is always the same number of agents in states B and A and one agent will remain in state N as n is an odd number.

Theorem 2. *The fixed-state median protocol operates in $O(n \log n)$ time both in expectation and whp.*

Proof. We say that a pair of agents a and b is *disordered* if an interaction between a and b is meaningful. We define the *disorder* $d(C)$ of a configuration C as the total number of disordered pairs of agents in this configuration. Since all agents start in state N, any initial interaction is meaningful, via application of rule (1). And in turn the disorder of the initial configuration is $d(C_0) = \binom{n}{2}$. In the final configuration C_∞, when the agent with the median key is in state N, and all agents with smaller and larger (than the median) keys are in states B and A respectively, the disorder $d(C_\infty) = 0$, as none of the rules can be applied.

Proposition 1. *Any meaningful interaction reduces the disorder of a configuration.*

The probability of making a meaningful interaction in a configuration C, s.t., $d(C) = i$ is $p_i = i/\binom{n}{2}$. Let the random variable T_i be the number of interactions needed to observe a meaningful interaction for a configuration C when $d(C) = i$. We have $\mathbb{E}[T_i] = \binom{n}{2}/i$. The expected number of interactions $\mathbb{E}[T]$ to transition from C_0 to C_∞ is

$$\mathbb{E}[T] \le \mathbb{E}[T_1] + \mathbb{E}[T_2] + \cdots + \mathbb{E}\left[T_{\binom{n}{2}}\right] = \binom{n}{2} H_{\binom{n}{2}} = O(n^2 \log n).$$

One can also prove that $T = O(n^2 \log n)$ whp applying Janson's bound. However, we show here an alternative proof by a potential function argument.

In particular, we show that for two subsequent configurations C and C' we have $\mathbb{E}[d(C')] \leq \left(1 - \frac{2}{n^2}\right) d(C)$. This inequality becomes trivial when $d(C) = 0$. Otherwise, when $d(C) \neq 0$,

$$\mathbb{E}[d(C')] \leq \left(1 - \frac{2d(C)}{n^2}\right) d(C) + \frac{2d(C)}{n^2} \left(d(C) - 1\right) = \left(1 - \frac{2}{n^2}\right) d(C).$$

After t interactions beyond configuration C_0 we get $\mathbb{E}[d(C_t)] \leq \left(1 - \frac{2}{n^2}\right)^t d(C_0) \leq \exp\left(-\frac{2t}{n^2}\right) \binom{n}{2}$. We obtain $\mathbb{E}[d(C_t)] < n^{-\eta}$ for $t \geq \binom{n}{2} \left(\ln \frac{n^2}{2} + \ln \eta\right)$. Finally, when $\mathbb{E}[d(C_t)] < n^{-\eta}$, by Markov's inequality, we get $\Pr(d(C_t) \geq 1) < n^{-\eta}$. This is equivalent to $\Pr(C_t \neq C_\infty) < n^{-\eta}$. □

3.2 Fast Median Computation

We present and analyse here efficient median computation in selective population protocols. The proposed solution is done by breaking up the full protocol into smaller blocks implemented as independent stabilisation processes with clearly defined inputs and outputs, as well as efficient and stable solutions. Each of these independent processes (see Fast-median algorithm below) including leader (pivot) election, partitioning agents wrt the key of the pivot, and majority computation, is executed on distinct partitions of states. Recall from Section 2.1 that the leader elected in the beginning of the computation process executes the code of the solution embedded in the transition function, and manages all input/output operations.

Algorithm 1 Fast-median.

Input: S – all agents set, $C = S$ – median candidate set

Select randomly leader (as pivot) $p \in S$ ▷ leader election

repeat

  Partition S to $\mathbb{B}_p = \{x \in S : x < p\}, \mathbb{A}_p = \{x \in S : x > p\}, \{p\}$;

  switch ▷ majority computation

    case $(|\mathbb{B}_p| > |\mathbb{A}_p|) \longrightarrow C = C \cap \mathbb{B}_p$

    case $(|\mathbb{B}_p| < |\mathbb{A}_p|) \longrightarrow C = C \cap \mathbb{A}_p$

    case $(|\mathbb{B}_p| = |\mathbb{A}_p|) \longrightarrow C = \{p\}$

  $p \leftarrow$ randomly chosen (by current p) agent in C ▷ leader hand over

until $(|\mathbb{B}_p| = |\mathbb{A}_p|)$

return p ▷ result announcement

Recall that leader election and majority computation, were discussed in Sections 2.3 and 2.4, respectively. Thus the focus in this section is on efficient partitioning of all agents in S to $\mathbb{B}_p = \{x \in S : x < p\}, \mathbb{A}_p = \{x \in S : x > p\}$, and $\{p\}$. Note that such partitioning is not trivial as due to the restrictions in the model the pivot p cannot distribute the value of its key to all agents in the population. Instead, the agents gradually learn their relationship with respect to the pivot by comparing their keys with other agents.

Theorem 3. *Fixed-state* Fast-median *protocol stabilises in fragmented parallel time* $O(\log^4 n)$ *whp.*

Proof. The proof is located at the end of Section 3.2. ☐

Partitioning via Coloring We commence by providing an overview of the argument. The partitioning of all agents occurs in multiple phases, represented by consecutive stabilization processes. The objective in each phase is to correctly partition a constant fraction of agents that have not been partitioned yet. Each phase has a time complexity of $O(\log^2 n)$. Since partitioning requires $O(\log n)$ phases, the overall time complexity becomes $O(\log^3 n)$.

In the median protocol, we execute $O(\log n)$ partitioning steps, which constitute the majority of the computation time. Consequently, the total time complexity for computing the median is $O(\log^4 n)$. To analyze a single phase of partitioning, we categorize the set of uncolored agents above the pivot into $2 \log n$ buckets, and similarly for those below the pivot. We then demonstrate that within $O(\log^2 n)$ time, the algorithm successfully colors $\log n$ agents from the first bucket. In each successive time period indexed by $i = 2, 3, \ldots$, with a duration of $O(\log n)$, the algorithm colors $2^{i-1} \log n$ agents in the i-th bucket. Consequently, after $O(\log^2 n)$ time from the initiation of a phase, a constant fraction of uncolored agents acquires colors.

The input for the partitioning process consists of the leader agent in the pivot state P and all other agents in the state N_{in}. We utilize two groups of states: $\mathcal{G} = \{P, B_0, \ldots, B_{21}, A_0, \ldots, A_{21}, N\}$ and $\mathcal{G}_{in} = \{N_{in}\}$. We interpret states A_t, B_t, and N as above, below, and neutral colors, respectively. An agent adopts state A_t (B_t) as soon as it learns that its key is above (below) the key of the pivot.

During each phase, we color approximately a fraction of 1/22 of yet uncolored agents. Upon the conclusion of the phase, these agents are moved to a different group, i.e., they do not participate in the partitioning of uncolored agents in the remaining phases.

In order to limit the activity of colored agents and, in turn, the duration of each phase, we introduce the concept of *tickets*. While the pivot has an unlimited number of tickets, any newly colored agent receives a fixed pool of 21 tickets. For as long as any colored agent has tickets, it targets agents in group $\mathcal{G}_{in}$ trying to color them. During such an interaction, a colored agent loses one ticket and moves one agent from $\mathcal{G}_{in}$ to $\mathcal{G}$. Once a colored agent loses all its tickets, it starts targeting group $\mathcal{G}$. The (partial) coloring phase concludes when the group $\mathcal{G}_{in}$ becomes empty. The set of relevant rules is given below.

(1) $P + \mathcal{G}_{in} \vert N_{in} \xrightarrow{\leq} P + A_{21}$	(3) $A_{t>0} + \mathcal{G}_{in} \vert N_{in} \xrightarrow{\leq} A_{t-1} + A_{21}$
(1) $P + \mathcal{G}_{in} \vert N_{in} \xrightarrow{>} P + B_{21}$	(3) $A_0 + \mathcal{G} \vert N \xrightarrow{\leq} A_0 + A_{21}$
(2) $B_{t>0} + \mathcal{G}_{in} \vert N_{in} \xrightarrow{\leq} B_{t-1} + N$	(4) $N + \mathcal{G} \vert P \xrightarrow{\leq} B_{21} + P$
(2) $B_{t>0} + \mathcal{G}_{in} \vert N_{in} \xrightarrow{>} B_{t-1} + B_{21}$	(4) $N + \mathcal{G} \vert P \xrightarrow{>} A_{21} + P$
(2) $B_0 + \mathcal{G} \vert N \xrightarrow{>} B_0 + B_{21}$	(4) $N + \mathcal{G} \vert B_t \xrightarrow{\leq} B_{21} + B_t$
(3) $A_{t>0} + \mathcal{G}_{in} \vert N_{in} \xrightarrow{>} A_{t-1} + N$	(4) $N + \mathcal{G} \vert A_t \xrightarrow{>} A_{21} + A_t$

We formulate here a tail bound that works for hypergeometric sequences for the case of small fraction p of black balls. We need this more sensitive bound in some of our proofs.

Lemma 6. *Assume we have an urn with n balls where pn of them are black. Let X_i be a binary random variable equal to one iff in the ith draw without replacement the drawn ball was black, and let $X = \sum_{i=1}^{\kappa} X_i$. Then, for $\kappa \leq n$ and $0 < \delta < 1$, we get* $\Pr\left(X < (1-\delta)p\kappa - p\right) < \kappa \exp\left(\frac{-\delta^2 p\kappa}{2}\right)$.

Denote the number of agents participating (not previously colored) in the phase by m. For any interaction t, let $k(t)$ be a number of agents in group $\mathcal{G}$ in t and $Inf_x(t)$ be a number of informed agents which are in states A or B in bucket x. The sequence of useful technical lemmas leading to the thesis of Theorem 4 follows.

Lemma 7. *If during interaction t the number of colored agents is $r \geq \log^2 n$, then $k(t+500n) \geq \min\{20r, m\}$ whp.*

Lemma 8. *The fragmented time of one phase of partitioning by coloring is $O(\log^2 n)$ whp.*

Lemma 9. *There is a constant $c > 0$, such that if $|[t_0, t_1)| = cn\log^2 n$ and $|[t_{i-1}, t_i)| = cn\log n$ for all $i > 1$, then whp*

- $k(t_i) \geq \min\left\{20 \cdot 2^{i-1}\log^2 n, m\right\}$,
- $Inf_i(t_i) > \min\left\{2^{i-1}\log n, \frac{m}{10\log n}\right\}$.

The phase ends when group $\mathcal{G}_{in}$ becomes empty. Each agent is relocated from $\mathcal{G}_{in}$ to $\mathcal{G}$ only if it either gets properly colored or it was given a ticket. Since each colored agent has only 21 tickets to utilise, we can formulate the following fact.

Fact 1 *After single coloring phase a fraction of $\frac{1}{22}$ uncolored agents gets colored.*

Thus, after $O(\log n)$ iterations of the coloring phase all agents are properly colored. This leads to the following theorem.

Theorem 4. *Partitioning by coloring stabilises in $O(\log^3 n)$ fragmented time.*

We conclude with the proof of Theorem 3.

Proof (of Theorem 3). The structure of the solution replicates the logic of a standard median computation protocol. Thus, the correctness of the solution follows from the correctness of the individual routines including leader election, majority computation and partitioning.

Concerning the time complexity, leader election and majority computation are implemented in fragmented parallel time $O(\log n)$ whp, see Lemmas 4 and 5. By Theorem 4, each partitioning stage takes $O(\log^3 n)$ time whp, and with probability $\frac{1}{2}$ at most $\frac{3}{4}$ candidates remain in C. Thus, with high probability after at most $O(\log n)$ iterations of this routine set C is reduced to a singleton containing the median.

Finally, Fast-median protocol stabilises in $O(\log^4 n)$ parallel time. □

4 Further Discussion

Suitable Programming Environment It is a natural solution to articulate solutions in any computational model using pseudocode. Such representation enhances the readability and understanding of the proposed solution within the context of the main features of the underlying computational model. Subsequently, this aids in conducting rigorous mathematical analysis. Various pseudocodes have been explored in the past to address challenges in population protocols, encompassing simple protocols [19], separation bounds [10], and leader-based computation [6]. The latter work forms the basis for our approach, which here focuses on the development of efficient parallel protocols. Our objective is to champion selective population protocols, enabling the development of simpler and more structured efficient solutions presented at higher programming level. The primary reasons for advocating this approach stem from the partitioning of the state space. Each partition represents the local variables of an independent process, supported by conditional interactions that also facilitate independent interaction availability (zero) tests. This, in turn, eliminates the necessity for a global clocking mechanism through the application of event-based distributed computation. For further detail consult the full version of this paper [14].

Final Comment We would like to postulate that the efficiency of selective protocols stand out when tackling problems that demand more extensive memory utilization and yield intricate outputs. A good example is the *ranking problem*, requiring assignment of unique labels from the range 1 to n to all agents, examined recently in the context of leader election in self-stabilizing protocols [9], and related sorting problem in the constructors model [15]. For these two problems, currently, no efficient solutions based on a polynomial number of states are known in standard population protocols. We would like to assert the following.

Conjecture 1. Any efficient solution to the sorting problem necessitates exponential state space in standard population protocols.

On the contrary, the evidence presented in the full version [14] demonstrates that selective protocols can efficiently solve sorting by ranking using a much smaller number of states. Specifically, we present transition rules of an efficient quick-sort-like, selective sorting by ranking. This algorithm has polynomial in n state space, utilises $O(n)$ partitions and stabilises in time $O(\log^2 n)$.

References

1. Alistarh, D., Aspnes, J., Eisenstat, D., Gelashvili, R., Rivest, R.: Time-space trade-offs in population protocols. In: Proc. SODA. pp. 2560–2579 (2017)
2. Alistarh, D., Aspnes, J., Gelashvili, R.: Space-optimal majority in population protocols. In: Proc. SODA. pp. 2221–2239 (2018)

3. Alistarh, D., Gelashvili, R., Rybicki, J.: Fast graphical population protocols. In: Proc. OPODIS. vol. 217, pp. 14:1–14:18 (2021)
4. Alistarh, D., Rybicki, J., Voitovych, S.: Near-optimal leader election in population protocols on graphs. In: PODC. pp. 246–256 (2022)
5. Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M., Peralta, R.: Computation in networks of passively mobile finite-state sensors. In: Proc. PODC. pp. 290–299 (2004)
6. Angluin, D., Aspnes, J., Eisenstat, D.: Fast computation by population protocols with a leader. Distributed Comput. **21**(3), 183–199 (2008)
7. Bankhamer, G., Berenbrink, P., Biermeier, F., Elsässer, R., Hosseinpour, H., Kaaser, D., Kling, P.: Population protocols for exact plurality consensus: How a small chance of failure helps to eliminate insignificant opinions. In: PODC. pp. 224–234 (2022)
8. Berenbrink, P., Giakkoupis, G., Kling, P.: Optimal time and space leader election in population protocols. In: STOC. pp. 119–129 (2020)
9. Burman, J., Chen, H., Chen, H., Doty, D., Nowak, T., Severson, E., Xu, C.: Time-optimal self-stabilizing leader election in population protocols. In: Proc. PODC. pp. 33–44 (2021)
10. Czerner, P.: Brief announcement: Population protocols decide double-exponential thresholds. In: Proc. PODC. pp. 28–31 (2023)
11. Czumaj, A., Lingas, A.: On parallel time in population protocols. Inf. Process. Lett. **179**, 106314 (2023)
12. Doty, D., Eftekhari, M., Gąsieniec, L., Severson, E., Uznanski, P., Stachowiak, G.: A time and space optimal stable population protocol solving exact majority. In: FOCS. pp. 1044–1055 (2021)
13. Gąsieniec, L., Stachowiak, G.: Enhanced phase clocks, population protocols, and fast space optimal leader election. J. ACM **68**(1), 2:1–2:21 (2021)
14. Gańczorz, A., Gąsieniec, L., Jurdziński, T., Kowalski, J., Stachowiak, G.: Selective Population Protocols. ArXiv **abs/1906.04238** (2024)
15. Gąsieniec, L., Spirakis, P., Stachowiak, G.: New clocks, optimal line formation and self-replication population protocols. In: Proc. STACS. pp. 33:1–33:22 (2023)
16. Guerraoui, R., Ruppert, E.: Names trump malice: Tiny mobile agents can tolerate byzantine failures. In: Proc. ICALP'09. pp. 484–495 (2009)
17. Kennedy, J.: Thinking is social: Experiments with the adaptive culture model. Journal of Conflict Resolution **42**(1), 56–76 (1998)
18. Kleinberg, J.: The small-world phenomenon: an algorithmic perspective. In: Proc. STOC. pp. 163–170 (2000)
19. Kosowski, A., Uznański, P.: Brief announcement: Population protocols are fast. In: Proc. PODC. pp. 475–477 (2018)
20. Michail, O., Spirakis, P.: Network constructors: A model for programmable matter. In: Proc. SOFSEM. pp. 15–34 (2017)
21. Minsky, M.: Computation: Finite and Infinite Machines. Prentice-Hall (1967)
22. Neary, T., Woods, D.: Four small universal turing machines. Fundam. Informaticae **91**(1), 123–144 (2009)
23. Soloveichik, D., Cook, M., Winfree, E., Bruck, J.: Computation with finite stochastic chemical reaction networks. Nat. Comput. **7**(4), 615–633 (2008)