Brief Announcement: Minimizing Energy Solves Relative Majority with a Cubic Number of States in Population Protocols

Tom-Lukas Breitkopf
TU Berlin
Berlin, Germany
t.breitkopf@tu-berlin.de

Julien Dallot
TU Berlin
Berlin, Germany
judafa@protonmail.com

Antoine El-Hayek
Institute of Science and Technology Austria
Klosterneuburg, Austria
antoine.el-hayek@ist.ac.at

Stefan Schmid
TU Berlin
Berlin, Germany
stefan.schmid@tu-berlin.de

Abstract

This paper revisits a fundamental distributed computing problem in the population protocol model. Provided n agents each starting with an input color in $[k]$, the relative majority problem asks to find the predominant color. In the population protocol model, at each time step, a scheduler selects two agents that first learn each other's states and then update their states based on what they learned.

We present the CIRCLES protocol that solves the relative majority problem with k^3 states. It is always-correct under weakly fair scheduling. Not only does it improve upon the best known upper bound of $O(k^7)$, but it also shows a strikingly simpler design inspired by energy minimization in chemical settings.

CCS Concepts

• **Theory of computation** → **Distributed algorithms**.

Keywords

Population protocols, k-majority problem

1 Introduction

Population protocols model a computation distributed across a population of n all-identical agents that interact in a chaotic, unpredictable manner. The model was introduced by Angluin et al. in 2006 [2] to model a network of small sensors; since then, it attracted a growing attention [1, 4, 7, 9] for its broad applications ranging from dynamics in social groups [5] to chemical reactions [8, 12] as well as its theoretical interest [3, 11].

Model. In this paper, we focus on the relative majority problem. In this problem, each agent is initially assigned one *input color* in $0, 1 \ldots k - 1$ and the goal is to find the color with the greatest support (we assume no ties). As part of the problem, a *scheduler* also specifies an infinite sequence of *pairwise interactions* between the agents. Then the protocol runs: pairs of agents interact one after the other according to the planned schedule; when two agents interact, they both 1) learn the *state* of the other agent and 2) update their current state as the protocol specifies. Two agents with the same state are perfectly identical: after it interacted, an agent's new state only depends on its previous own state and on the state of the other agent it just interacted with.

Definition 1.1 (Configuration). A configuration is a complete description of the population at a given time. As agents with the same state are identical, we define a configuration as the multiset that contains all the states of the population.

In the context of population protocols, we say that a protocol solves the problem if, for all possible input color assignments and all possible sequences of interactions, every agent eventually outputs the correct majority color, forever. However, an unconstrained scheduler makes the problem trivially impossible (by isolating some agents for instance), we therefore assume that the scheduler is *weakly fair*:

Definition 1.2 (Weakly Fair Scheduler). A weakly fair scheduler produces interaction schedules where each possible interaction pair happens infinitely often.

Contribution. We present CIRCLES, an always-correct protocol that solves the relative majority problem under a weakly fair scheduler. We designed CIRCLES with an emphasis on *state complexity*, which is the number of different states an agent can have. CIRCLES has a state complexity of k^3, which improves upon the best known upper bound of $O(k^7)$ [10] and narrows the gap with the best known lower bound of $\Omega(k^2)$ [12]. The emphasis on state complexity is motivated by applications where the memory space per agent is severely limited: tiny sensors in a network [2] or molecules in chemical applications [1, 7]. CIRCLES is always correct under a weakly fair scheduler and shows an elegant design inspired by energy minimization in chemical settings.

Notations We define a few notations used in the protocol's definition and proofs.

multisets. For sets S and T we write S^T to denote the set of functions $f : T \to S$. If T is finite we call the elements of $\mathbb{N}^T$ *multisets* over T. In this paper, the subset $\subseteq$, the union $\cup$ and the set substraction $\setminus$ operations are systematically generalized to multisets.

remainder. For $x \in \mathbb{Z}$ and $p \in \mathbb{N}^*$, we define $x \bmod p$ as the remainder of the euclidean division of x by p. Note that this is a number in $\mathbb{N}$, not $\mathbb{Z}/n\mathbb{Z}$.

ranges. Let $x, y \in \mathbb{N}$ such that $x \leq y$. $[x, y]$ denotes the set $\{x, x+1, \ldots, y-1, y\}$, and (x, y) denotes the set $\{x+1, x+2, \ldots, y-2, y-1\}$.

modulo ranges. Let $x, y \in \mathbb{N}$. We define modulo ranges: $[x, y]_p$ denotes the set $\{x \bmod p, (x+1) \bmod p, \ldots, (x+(y-x) \bmod p - 1) \bmod p, (x+(y-x) \bmod p) \bmod p\}$, and $(x, y)_p$ denotes the set $\{(x+1) \bmod p, (x+2) \bmod p, \ldots, (x+(y-x) \bmod p - 2) \bmod p, (x+(y-x) \bmod p - 1) \bmod p\}$. For instance, $[2, 7]_{10} = \{2, 3, 4, 5, 6, 7\}$ and $(8, 3)_{10} = \{9, 0, 1, 2\}$.

bra-ket. We abuse the bra-ket notation, frequently used in quantum mechanics, to note ordered pairs. For $i, j \in \mathbb{N}$, we write $\langle i|j \rangle$ simply to distinguish the different roles of i and j. For an agent storing a bra-ket $\langle i|j \rangle$, we refer to i as its bra and j as its ket.

2 The Circles protocol

We present thereafter the Circles protocol that runs in every agents: its set of states, input function (to convert the input color into one of the protocol's states), output function (to ask an agent what the majority color is) and the transition function (to specify how two agents update their states when they interact).

- **States:** The set of states Q contains every triples $(i, j, o) \in [0, k-1]^3$. In the remainder of this paper we will use the bra-ket notation $\langle i|j \rangle$ to refer to the first two numbers of the triple, *bra* refers to i and *ket* refers to j, while *out* refers to o.
- **Input:** each agent is initialized with $\langle i|i \rangle$ and $out = i$, where i is the input color of the agent.
- **Output:** return *out*
- **Transition function:** We define *weights* for each bra-ket $\langle i|j \rangle$ as follows:

$$w(\langle i|j \rangle) = \begin{cases} k & \text{if } i = j \\ (j - i) \bmod k & \text{otherwise} \end{cases}$$

Two agents a and b that interact perform two successive operations:

(1) a and b exchange their kets in case this strictly decreases the minimum weight of their two bra-kets.
(2) If either a or b is of the form $\langle i|i \rangle$ for some $i \in [k]$, set $out_a = out_b = i$.

3 Proof of correctness

We prove the protocol's correctness in Theorems 3.4 and 3.7, Theorem 3.7 directly derives from Lemma 3.6. We beforehand introduce Greedy Independent sets, a construction on the input colors used to prove the protocol's correctness, as well as two preliminary lemmas 3.2 and 3.3.

Definition 3.1 (Greedy Independent Sets). Consider the multiset of input colors to our protocol. We partition this multiset into sets $G_1, G_2, \ldots, G_q$ as follows: store in G_1 as many inputs as possible as

long as G_1 does not contain two equal colors; then apply the same on the remaining inputs to obtain G_2, G_3, and so on.

Lemma 3.2. *(Majority Color).* Assume that there exists a unique color μ in relative majority, then it holds that $G_q = \{\mu\}$ and there is no $j \neq \mu$ and $p \in [1, q]$ such that $G_p = \{j\}$.

Proof. Each time a set G_p is constructed according to Definition 3.1, any color whose count is not zero yet is added into G_p and its count is decremented by one. The color μ appears in the population strictly more than any other, it therefore holds that $\forall p \in [1, q]$, $\mu \in G_p$. Let now $i \in [0, k-1]$ be a color contained in a set G_p for some $p \in [1, q]$. It holds that $\forall l \leq p, i \in G_l$ because color i is available to populated G_p so i was also available when G_l was filled earlier. It therefore cannot be that a color $j \neq \mu$ is contained in G_q as j would be contained in all sets $G_1 \ldots G_q$ and thus j would also be in relative majority, a contradiction. □

Lemma 3.3. *(Global Braket Invariant).* In every configuration and for all $i \in [0, k-1]$, the number of bras $\langle i|$ and the number of kets $|i\rangle$ are equal.

Proof. Every agent is initialized with bra-ket $\langle i|i \rangle$ for some $i \in [0, k-1]$ and the claim initially holds. Agents subsequently only ever update their bra-ket, by exchanging kets among each other. The overall number of bras and kets in the population therefore does not change during a computation. □

Theorem 3.4. *(Stabilization).* The agents exchange their kets a finite number of times.

Proof. We call ω the smallest ordinal number greater than all the integers. We prove the claim by exhibiting a non-negative quantity that strictly decreases at each ket exchange. Given a configuration C, let $w_1(C), w_2(C) \ldots w_n(C)$ be the bra-ket's weights of each agent sorted in increasing order. Define

$$g(C) = \omega^{n-1} \cdot w_1(C) + \omega^{n-2} \cdot w_2(C) + \cdots + \omega \cdot w_{n-1} + 1 \cdot w_n$$

Assume that two agents exchange their kets. Let p be the lowest index such that $w_p(C)$ changes before and after the ket exchange. By design of the protocol, $w_p(C)$ strictly decreases. This implies that g strictly decreases when two agents exchange their kets. As an ordinal number cannot decrease infinitely many times, the number of ket exchanges is therefore finite. □

We define the following special sets of bra-kets only to formulate Lemma 3.6.

Definition 3.5 (Circle Bra-ket Sets). For a given greedy independent set G_p with $p \in [1, q]$ (Definition 3.1), let $g_0, g_1, \ldots, g_m$ be the elements of G_p sorted in increasing order and define

$$f(G_p) = \{\langle g_0|g_1 \rangle, \langle g_1|g_2 \rangle, \ldots, \langle g_m|g_0 \rangle\}$$

Lemma 3.6. *After Stabilization (Theorem 3.4), let C be the multiset of bra-kets of the agents. We have that:*

$$C = \bigcup_{p=1 \ldots q} f(G_p)$$

PROOF. We prove the predicate $H(r)$ by induction on $r \in [0, q]$:

$$\bigcup_{p=1...r} f(G_p) \subseteq C \qquad (H(r))$$

The base case for $r = 0$ is trivial. Let $r \in [0, q-1]$, we assume that $H(r)$ holds and show that $H(r + 1)$ also holds. We define the subconfiguration $C[r + 1] = C \setminus \cup_{p=1...r} f(G_p)$.

Case 1: $\cup_{p=r+1}^{q} G_p$ contains only elements from one color. Let i be that color.

Then for any other color $j \neq i$, there are at most r many bras $\langle j|$ and as many kets $|j\rangle$, which are all included in $\cup_{p=1...r} f(G_p)$. Thus all agents in $C[r + 1]$ are of the form $\langle i|i\rangle$. Since $\{\langle i|i\rangle\} = f(G_{r+1})$, we have $\cup_{p=1...r+1} f(G_p) \subseteq C$.

Case 2: There are at least two different colors in G_{r+1}.

We note $g_0, g_1, \ldots, g_m$ the elements of G_{r+1} sorted in increasing order. Let $l \in [0, m]$. To lighten notations, we will mean $l+s \mod m+1$ each time we write $l + s$ in the remainder of this proof. We prove that, if there is no agent with bra-ket $\langle g_l|g_{l+1}\rangle$ in $C[r + 1]$, then there exist two agents whose interaction creates that bra-ket, a contradiction with the stability hypothesis. First notice that $\langle g_l|$ and $|g_{l+1}\rangle$ are in $C[r + 1]$. Indeed, note that there are at least $r + 1$ many $\langle g_l|$ and $r + 1$ many $\langle g_{l+1}|$ in C, as there are at least $r + 1$ many agents with color g_l and g_{l+1} initially. By Theorem 3.3, this means we have at least $r + 1$ many $|g_{l+1}\rangle$ in C. Because each f_p for $p \leq r$ contains exactly one $\langle g_l|$ and one $|g_{l+1}\rangle$, we have that both $\langle g_l|$ and $|g_{l+1}\rangle$ are in $C[r + 1]$.

Assuming by contradiction that there is no agent with bra-ket $\langle g_l|g_{l+1}\rangle$ in $C[r + 1]$, then there exists an agent with bra-ket $\langle g_l|j\rangle$ and an agent with bra-ket $\langle i|g_{l+1}\rangle$ in $C[r + 1]$ for some i and j. We show that those two agents exchange their kets if they interact.

CLAIM 1. $i, j \notin (g_l, g_{l+1})_m$

PROOF. By contradiction, assume that i is in $(g_l, g_{l+1})_m$. A $\langle i|$ in $C[r + 1]$ indicates that i had initially at least $r + 1$ agents supporting it, as by contradiction if it wasn't the case, all the $\langle i|$ would have been in $\cup_{p=1}^{r} f(G_p)$. By construction of G_{r+1}, we must have that i is in G_{r+1}, and thus, g_l and g_{l+1} are not consecutive in the ordered list of G_{r+1}, a contradiction. The case $j \in (g_l, g_{l+1})_m$ is symmetric. □

If $i \neq g_{l+1}$, it holds by Claim 1 that

$$w(\langle g_l|g_{l+1}\rangle) = (g_{l+1} - g_l) \mod k < (g_{l+1} - i) \mod k = w(\langle i|g_{l+1}\rangle)$$

Otherwise, if $i = g_{l+1}$:

$$w(\langle g_l|g_{l+1}\rangle) = (g_{l+1} - g_l) \mod k < k = w(\langle i|g_{l+1}\rangle)$$

Similarly, if $j \neq g_l$, it holds by Claim 1 that

$$w(\langle g_l|g_{l+1}\rangle) = (g_{l+1} - g_l) \mod k < (j - g_l) \mod k = w(\langle g_l|j\rangle)$$

Otherwise, if $j = g_l$:

$$w(\langle g_l|g_{l+1}\rangle) = (g_{l+1} - g_l) \mod k < k = w(\langle g_l|j\rangle)$$

This proves that exchanging kets between $\langle g_l|j\rangle$ and $\langle i|g_{l+1}\rangle$ reduces the minimum weight. An interaction between the two agents eventually happens as the scheduler is weakly fair, this interaction would therefore trigger a ket exchange, which is in contradiction with the stability hypothesis: we deduce that $\langle g_l|g_{l+1}\rangle \in C[r + 1]$, which implies $f(G_{r+1}) \subseteq C[r + 1]$ and therefore $H(r + 1)$ holds.

We proved by induction that $H(q)$ holds:

$$\bigcup_{p=1...q} f(G_p) \subseteq C$$

As $|\cup_{p=1...q} f(G_p)| = |C|$ we can rewrite $H(q)$ as an equality and the claim holds. □

THEOREM 3.7. *(Correctness). Assume that there exists a unique color μ in relative majority. In the CIRCLES protocol, all agents eventually output μ under a weakly fair scheduler.*

PROOF. By Lemma 3.6 and Lemma 3.2, after Stabilization (Theorem 3.4), since we assumed that there is only one majority color, there exists at least one agent in bra-ket $\langle \mu|\mu\rangle$ and none in bra-ket $\langle j|j\rangle$ for $j \neq \mu$. The agent(s) with bra-ket $\langle \mu|\mu\rangle$ will transmit their output color to the rest of the population and the claim follows. □

4 Extensions

We plan to expand the functionalities of CIRCLES to handle ties and/or to operate in an unordered setting. Those extensions will be published as a more complete version of the present work.

Handling ties. We can extend CIRCLES to handle ties in multiple ways. For instance, all agents can indicate a tie with a special state (tie report), agree on one unique winning color (tie break), or output their own color if their input color wins while the losers output any winning color (tie share). It is possible to implement all those ways to handle ties by adding simple extra-layer protocols on top of CIRCLES while keeping the state complexity at $O(k^3)$.

Unordered setting. The CIRCLES protocol, which we introduce in this work, relies on numerical representations of the colors in order to compute some kind of distance between them. It can be adapted to the unordered setting (in which agents are only able to compare colors for equality and memorize them) using $O(k^4)$ states. For that we propose a new protocol to generate an ordering between colors using $O(k^2)$ states. Adapting a protocol proposed in [6] we perform leader-election between all agents of the same color (using the asymmetry of interactions) and have the leaders increment a numeric label every time they meet another leader with the same label. The non-leaders simply copy the label of their leader. Similar to [12] we then combine the ordering protocol with CIRCLES by re-initializing agents of some color whenever their numeric label (representing that color) changes. For that we need to put agents into special states in which they wait to undo changes they previously made to the population until they are "consistent" again and ready to be re-initialized. In order to use as few states as possible we do not explicitly store the output of the ordering protocol, but write it directly to the bra of an agent.

Acknowledgments

References

[1] Dan Alistarh and Rati Gelashvili. 2018. Recent Algorithmic Advances in Population Protocols. *ACM SIGACT News* 49, 3 (2018), 63–73.

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

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

[4] Gregor Bankhamer, Petra Berenbrink, Felix Biermeier, Robert Elsässer, Hamed Hosseinpour, Dominik Kaaser, and Peter Kling. 2022. Population Protocols for Exact Plurality Consensus: How a small chance of failure helps to eliminate insignificant opinions. In *41st Annual ACM Symposium on Principles of Distributed Computing (PODC 2022)*. Salerno (Italy), 224–234.

[5] Luca Becchetti, Andrea E. F. Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. 2015. Plurality consensus in the gossip model. In *26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*. San Diego (USA), 371–390.

[6] Shukai Cai, Taisuke Izumi, and Koichi Wada. 2012. How to Prove Impossibility Under Global Fairness: On Space Complexity of Self-Stabilizing Leader Election on a Population Protocol Model. *Theory of computing systems* 50, 3 (2012), 433–445.

[7] Philipp Czerner, Javier Esparza, and Jérôme Leroux. 2023. Lower bounds on the state complexity of population protocols. *Distributed computing* 36, 3 (2023), 209–218.

[8] David Doty. 2014. Timing in chemical reaction networks. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014)*. Portland (USA), 772–784.

[9] Robert Elsässer and Tomasz Radzik. 2018. Recent Results in Population Protocols for Exact Majority and Leader Election. In *The Distributed Computing Column*, Stefan Schmid (Ed.).

[10] Leszek Gąsieniec, David Hamilton, Russell Martin, Paul G. Spirakis, and Grzegorz Stachowiak. 2017. Deterministic Population Protocols for Exact Majority and Plurality. In *20th International Conference on Principles of Distributed Systems (OPODIS 2016)*. Madrid (Spain), 14:1–14:14.

[11] Richard J. Lipton. 1976. The Reachability Problem Requires Exponential Space. (1976).

[12] Emanuele Natale and Iliad Ramezani. 2019. On the Necessary Memory to Compute the Plurality in Multi-Agent Systems. In *11th International Conference on Algorithms and Complexity (CIAC 2019)*. Rome (Italy), 323–338. arXiv:1901.06549 http://arxiv.org/abs/1901.06549