

Population Protocols for Majority in Arbitrary Networks

George B. Mertzios, Sotiris E. Nikolettseas, Christoforos L. Raptopoulos,
and Paul G. Spirakis

School of Engineering and Computing Sciences, Durham University, UK
george.mertzios@durham.ac.uk

Computer Technology Institute and Press “Diophantus”, Greece
Computer Science Department, University of Patras, Greece
nikole@cti.gr

Computer Technology Institute and Press “Diophantus”, Greece
raptopox@ceid.upatras.gr

Computer Technology Institute and Press “Diophantus”, Greece
Department of Computer Science, University of Liverpool, UK
Computer Science Department, University of Patras, Greece
p.spirakis@liverpool.ac.uk

Abstract: We study the problem of determining the majority type in an arbitrary connected network, each vertex of which has initially two possible types. The vertices may have a few additional possible states and can interact in pairs only if they share an edge. Any (population) protocol is required to stabilize in the initial majority. We first present and analyze a protocol with 4 states per vertex that *always* computes the initial majority value, under any fair scheduler. This protocol is optimal, in the sense that there does not exist any population protocol that always computes majority with fewer than 4 states per vertex. However this does not rule out the existence of a protocol with 3 states per vertex that is correct with high probability (whp). To this end, we examine an elegant and very natural majority protocol with 3 states per vertex, introduced in [2] where its performance has been analyzed for the clique graph. In particular, we study the performance of this protocol in arbitrary networks, under the probabilistic scheduler. We prove that, when the two initial states are put uniformly at random on the vertices, the protocol of [2] converges to the initial majority with probability higher than the probability of converging to the initial minority. In contrast, we show that the resistance of the protocol to failure when the underlying graph is a clique causes the failure of the protocol in general graphs.

This abstract paper is based on our work [14], which appeared in the Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP) 2014.

1 Introduction and Motivation

One of the most natural computational problems in many physical systems is to compute the *majority*, i.e. to determine accurately which type of an element of the system appears more frequently. For instance, the majority problem is encountered in various settings such as in voting [9, 10], in epidemiology and interacting particles systems [13], in diagnosis of multiprocessor systems [19], in social networks [17, 15] etc. In distributed

computing, the majority problem is an important and natural special case of the central problem of reaching *consensus* within a system [12, 6], where a number of processes have to agree on any single data value (e.g. leader election [8]). In all these physical systems, some pairs of elements may interact with each other while other pairs may not be able to interact directly. This structure of the possible pairwise interactions between elements of the system can be modeled by a network (i.e. graph), where elements and possible interactions are represented by vertices and edges, respectively.

In order to solve the majority computation problem in a network, we first need to make some assumptions on the underlying model of computation. Much research has been done under the assumption that there exists a central authority, as well as unlimited available memory and full information about the whole network (see e.g. [20, 5]). However, in many real systems we do not have (or we do not wish to have) such a powerful computational model. The weaker the considered model of computation is (e.g. lack of central authority, partial or no information about the system, lack of memory etc.), the more challenging the majority computation becomes.

One of the ways to study distributed systems where agents may interact in pairs and each individual agent is extremely limited (in fact, being equipped only with a finite number of possible states) is by using *population protocols* [1, 3]. Then the complex behavior of the system emerges from the rules governing the possible pairwise interactions of the agents. Population protocols have been defined by analogy to population processes [11] in probability theory and have already been used in various fields, such as in statistical physics, genetics, epidemiology, chemistry and biology [4].

In particular, population protocols are *scalable*, i.e. they work independently of the size n of the underlying network (called the *interaction graph*) and the value of n is not even known to the protocol. Furthermore they are *anonymous*, i.e. there is only one transition function which is common to all entities/agents: the result of an interaction of an agent u at state q_u with an agent v at state q_v is the same regardless of the identity of u and v . The transition function of a population protocol only specifies the result of every possible interaction, without specifying *which* pairs of agents interact or *when* they are chosen to interact. Usually it is assumed that interactions between agents happen under some kind of a *fairness* condition. For a survey about population protocols we refer to [3].

In this direction, a very natural and simple population protocol for the majority problem on the clique (i.e. the complete graph), where initially every vertex has one of two possible types (states), has been introduced and analyzed in [2]. In particular, the protocol of [2] assigns only 3 possible states to every agent (i.e. there is a 3×3 transition table capturing all possible interactions) and the interactions between agents are dictated by a *probabilistic scheduler* (i.e. all pairs have the same probability to interact at any step). Every vertex has an identity v , but it is unaware of the identity of any other vertex, as well as of its own identity. Although the underlying interaction graph in [2] is assumed to be a clique, the authors distinguish in their protocol the agents u and v participating in an interaction into an “initiator” and a “responder” of the interaction (when agents u and v interact, each of them becomes initiator or responder with equal probability). Their main result is that, if initially the difference between the initial majority from the initial minority in the complete graph with n vertices is $\omega(\sqrt{n} \log n)$, their protocol converges to the correct initial majority value in $O(n \log n)$ time with high probability.

Most works on population and majority dynamics so far considered only two entity types (e.g. the voter model [9], the Moran process [16]). The analysis of population dynamics with more than two types is challenging. As an example we refer to the model of [2], in which, although agents can have initially one of only two types (red and green), the protocol itself allows every agent to be in one among three different states (red, green and blank) at every subsequent time point. Even though this model is quite simple, it is very hard to be analyzed. Computing the majority with as few states as possible in the more general case, where the interaction graph has an arbitrary structure (as opposed to the complete graph that has been mainly considered so far) remained an open problem.

2 Our Contribution

In [14] we study the majority problem in an *arbitrary* underlying interaction graph G , where initially every vertex has two possible states (red and green). We consider the weakest and simplest possible model of computation. In particular, we assume the existence of no central authority and we allow every vertex of G to have only a (small) constant number of available types (or states). Although every vertex of G has a unique identity, no vertex is aware of its own identity or the identity of any other vertex. Furthermore, although only two adjacent vertices can interact, vertices of G do not even know to which other vertices they are adjacent.

2.1 Stable computation of majority

We initially focus on the problem of *always* computing the correct majority value in an *arbitrary* (directed or undirected) interaction graph G , regardless of how large the initial difference between the majority and the minority is. In particular, assuming that the interacting pairs of vertices are chosen by an arbitrary fair scheduler, we derive matching lower and upper bounds on the number of available states, for which there exists a population protocol that always computes the correct majority value. For the lower bound, we prove the following:

Theorem 1 *Any population protocol that stably computes the majority function uses at least 4 states.*

The proof is based on a contradiction argument; assuming there is a population protocol computing majority using 3 states, we construct two distinct instances C_1 and C_2 that (a) have different initial majorities and (b) there is a fair scheduler that brings both C_1 and C_2 to the same intermediate configuration (after which the protocol will have no way of determining whether it started from C_1 or C_2).

On the other hand, for the matching upper bound we provide a population protocol with 4 states per vertex, namely the *Ambassador protocol*, which always computes the correct majority value, even if initially the difference between majority and minority is 1. A different 4 state protocol for majority was independently presented in [7]. Using standard results on random walks on graphs and coupon collector arguments, we provide polynomial upper bounds on the expected time needed by our Ambassador protocol to converge, and we show that in certain cases the running time is $O(n \log n)$, i.e. the same as for the fast protocol of [2].

2.2 The protocol of [2] on arbitrary graphs

In [14], we also provide a detailed analysis of the 3-state protocol of [2] on an arbitrary interaction graph G . Our first result in this direction is the following:

Theorem 2 *For any strongly connected directed graph G , if the initial assignment of individuals to the vertices of G is random, then the majority protocol described in [2] correctly identifies the initial majority with probability at least $\frac{1}{2}$.*

The proof of this relies on a well known result in extremal combinatorics (in particular, on Hall’s marriage Theorem).

On the other hand, we proved the following:

Theorem 3 *There exists an infinite family $\{G_n\}_{n \in \mathbb{N}}$ of interaction graphs where the protocol fails with high probability, even when the initial difference between majority / minority is $n - \Theta(\log n)$.*

Theorem 4 *There exists an infinite family $\{G'_n\}_{n \in \mathbb{N}}$ of interaction graphs where the protocol terminates in exponential expected time.*

In particular, Theorem 4 rules out the possibility to use a Markov chain Monte-Carlo approach to approximate the probability that the protocol of [2] converges to the correct majority value.

The proofs of Theorems 3 and 4 employ an intermediate result concerning the robustness of the protocol of [2] on the clique. In particular, *for any $\varepsilon > 0$, if the minority has size at most $(\frac{1}{7} - \varepsilon)n$ in the complete graph with n vertices, then the protocol of [2] converges to the initial minority with exponentially small probability.* This result shows that, although the performance of the protocol of [2] can drop significantly when the interaction graph G is not the complete graph, it is quite robust when G is the complete graph. After submission of our original work in [14], we became aware of the paper [18], the results of which can also be used to prove a tighter version of our result concerning the robustness of the protocol of [2] on the clique. However, the techniques used there cannot be applied to graph structures other than the clique. On the other hand, our proof technique can also be applied to graph structures that are different than (but close to being) cliques by using a non-trivial coupling argument which can be of independent interest.

References

- [1] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18:235–253, 2006.
- [2] D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.
- [3] J. Aspnes and E. Ruppert. An introduction to population protocols. In B. Garbinato, H. Miranda, and L. Rodrigues, editors, *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer-Verlag, 2009.
- [4] M. Cook, D. Soloveichik, E. Winfree, and J. Bruck. Programmability of chemical reaction networks. In A. Condon, D. Harel, J. N. Kok, A. Salomaa, and E. Winfree, editors, *Algorithmic Bioprocesses*, Natural Computing Series, pages 543–584. Springer Berlin Heidelberg, 2009.
- [5] G. De Marco and A. Pelc. Randomized algorithms for determining the majority on graphs. *Combinatorics, Probability and Computing*, 15(6):823–834, 2006.
- [6] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

- [7] M. Draief and M. Vojnovic. Convergence speed of binary interval consensus. *SIAM Journal on control and Optimization*, 50(3):1087–1109, 2012.
- [8] M. Fischer and H. Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *Proceedings of the 10th International Conference on Principles of Distributed Systems (OPODIS)*, 2006.
- [9] R. A. Holley and T. M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3:643–663, 1975.
- [10] M. Kearns and J. Tan. Biased voting and the democratic primary problem. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, pages 639–652, 2008.
- [11] T. G. Kurtz. *Approximation of Population Processes*. Society for Industrial and Applied Mathematics, 1987.
- [12] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
- [13] T. M. Liggett. *Interacting Particle Systems*. Springer-Verlag, 2004.
- [14] G. B. Merzios, S. E. Nikolettseas, C. L. Raptopoulos, and P. G. Spirakis. Determining majority in networks with local interactions and very small local memory. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*, volume 1, pages 871–882, 2014.
- [15] A. Mizrachi. Majority vote and monopolies in social networks. Master’s thesis, Department of Communication Systems Engineering, Faculty of Engineering, Ben-Gurion University of the Negev, 2013.
- [16] P. A. P. Moran. Random processes in genetics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 54(1):60–71, 1958.
- [17] E. Mossel, J. Neeman, and O. Tamuz. Majority dynamics and aggregation of information in social networks. *Autonomous Agents and Multi-Agent Systems*, 28(3):408–429, 2014.
- [18] E. Perron, D. Vasudevan, and M. Vojnovic. Using three states for binary consensus on complete graphs. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pages 2527–2535, 2009.
- [19] F. P. Preparata, G. Metze, and R. T. Chien. On the connection assignment problem of diagnosable systems. *IEEE Transactions on Electronic Computers*, 16:848–854, 1967.
- [20] M. Saks and M. Werman. On computing majority by comparisons. *Combinatorica*, 11(4):383–387, 1991.