What can be computed in a simple chaotic way?[⋆]
(Invited talk)

Emanuele Natale[⋆⋆]

Max-Planck Institut für Informatik
enatale@mpi-inf.mpg.de

Abstract. We provide a brief overview of a set of analytical results regarding some very simple randomized protocols, called *dynamics*, for solving the plurality consensus problem, together with some examples on how to build efficient and robust distributed algorithms for other fundamental tasks using these protocols as a core subroutine. More specifically, we look at the 3-Majority and the Undecided-State dynamics, and at how to use them to solve fundamental problems such as information propagation and clock synchronization.

1 Introduction

Since the advent of the computer, scientists found themselves with a new telescope which provided them with the capacity to observe the computational universe at a new scale. Through simulations, they were able to look far beyond their mathematical understanding of the relationship between local interactions among the tiny parts of a system and its global behavior and, in the last decades, they were astounded by the unexpected appearance of global complexity from local simplicity. However, such enthusiasm brought from the shocking new ability to explore the universe of computational rules has been counterbalanced from the inability to develop a new mathematics which could account for our new observations.

Within the world of theoretical computer science, a particularly appealing tool to look at the interplay of local/individual and global/collective behavior is the theory of distributed computing. From programmable matter to chemical reaction networks, from sensor networks to the behaviour of insect colonies, there is a huge part of the distributed computing discipline driven by the aspiration to develop a theory analogous to that built by statistical mechanics for interacting particle systems, when we replace "particle" with "agent". With this respect, distributed computing researchers have recently managed to start analyzing analytically some particularly simple protocols, collectively called *dynamics* [7]. Informally, a dynamics is a distributed computing protocol in which

[⋆] The content of this extended abstract has been adapted from the first chapter of the author's PhD thesis.

[⋆⋆] The author would like to thank all the collaborators involved in the joint works mentioned in this paper.

each node of the graph updates its state according to a time- and topology-invariant function which only depends on the node's current state and on that of its neighbors.

In the next section, we briefly discuss two of the most notable examples of dynamics and some of the associated results, whose development the author had the pleasure to take part of. In the subsequent section, we finally discuss some way to use such results to improve the state of the art of some fundamental distributed computing problems.

1.1 Dynamics for majority consensus

One of the simplest, non-trivial[1] dynamics one can think of is the 3-Majority dynamics [3], in which at each round each node looks at the states of three random neighbors and then switch to the majority state among these three ones (breaking ties uniformly at random).

Rather surprisingly, we found that the convergence time of the 3-Majority dynamics in solving the consensus problem, is essentially linear in the number of initial different opinions in the system. We further proved that the situation does not change if instead of the 3-Majority we consider any protocol within a wide class of dynamics (h-input dynamics), and that the 3-Majority dynamics was already optimal, in a precise sense, w.r.t. all those dynamics which basically consist in exchanging opinions making use of at most 3 inputs.

These results were very bad news for the potential use of majority-like dynamics as a building block for more complex protocols and as an efficient dynamics per-se, and motivated the further investigation of faster dynamics for achieving plurality consensus. After exploring the vast space of possible candidates for quite a while, stumbled upon the Undecided-State dynamics.

The Undecided-State dynamics was already famous in computer science as an elegant solution to more restricted majority consensus problems (i.e. asynchronous communication models with binary initial states). After some attempts at proving upper bounds on its convergence time w.r.t. the standard hypotheses that are assumed in majority consensus problems, we noticed that under its deceptively simple structure the Undecided-State dynamics shows an evolution with an unexpected anatomy [2]. Namely, its behaviour and convergence time, instead of depending on few crucial parameters, are a function sensible to the *whole* initial configuration. We named this function the *monochromatic distance*. By inspecting the monochromatic distance we see that the Undecided-State dynamics has the advantage of having a convergence time which is at least as good as that of the 3-Majority dynamics (for a number of opinions in the system which can be as large as $\sqrt{n}/\log n$), and exponentially faster for a wide range of configurations. Thus, it is a simple but way more effective dynamics in many applicative contexts, such as congested communication models [1].

[1] A *trivial* dynamics is, for example, the Polling Process, in which at each round each node copies the state of a random neighbor. While the analysis of the latter process already poses many mathematical challenges, from a computational point of view it doesn't exhibit surprising features.

1.2 Applications of majority dynamics

In addition to the purely theoretical interest and potential applications in technological contexts (e.g. sensor and ad-hoc networks), the investigation of dynamics is also partially motivated by biological questions. In the biological world, information propagation and majority consensus are common phenomena in a wide range of systems. Examples of such processes include a single ant that has found food and recruits others, few cells that trigger large population responses, a school of fish that reaches consensus around a group of leaders, or a small number of observant individuals that alert their herd. Such information propagation is achieved despite what appear to be highly unpredictable, uncoordinated, noisy and limited communication settings.

In [5], the authors investigate "natural" protocols for solving the information propagation and majority consensus problems in a noisy version of the uniform $\mathcal{PUSH}$ model, where each message can be corrupted before being received. We generalize the previous problem to the case with multiple opinions [6]. Such generalization required us to solve both conceptual and technical issues. On the conceptual side, while in the binary-message case the noise merely consists in the fact that with some probability one of the two values can be "flipped", in the multivalued case it is not clear what is the *right* way of modeling the fact that messages can be misunderstood. Here, the "right" modeling is the formalization that allows to separate in the most natural way the settings in which the problem can be solved from those in which it is not solvable. On the technical side, the problem shares the following usual difficulty of generalizing a finite-volume process from dimension one to more than one. In the binary case, the fact that there are only two possible values provides the possibility to "take the complement" of quantities regarding one value, to get those regarding the other one. This possibility, which is often a key ingredient of the analysis, vanishes when we introduce further degrees of freedom in the process by allowing more than two possible values. While the aforementioned generalization in the end is still far from being a dynamics per-se, the subroutines at the core of the algorithm rely on a generalization of the 3-Majority dynamics.

The above noise variant of the rumor spreading problem deals with reaching consensus despite the presence of *noise* in the system. The consensus problem, however, has a wider general significance within the natural world. In a biological setting, maintaining consensus is an evolutionary convenient strategy to cope with the limitations of single individuals in acquiring information from the environment, and to maximize the probability of survival in general. As an example, let us imagine a group of birds on a wire[2]. At some point some bird starts to fly. The other birds have the legitimate doubt that the moving one is leaving her spot on the wire because she has caught sight of a predator. Therefore, other birds start flying as well. Perhaps, shortly after leaving her point on the wire the first bird lands again on it, since her original intention was only to move to a better place. The other alarmed birds then realize that it was a false alarm, and

[2] The author thanks Amos Korman for the nice "stubborn bird" example.

they also start landing again on the wire. On the other hand, the first bird may also stubbornly continue her escape from an imminent threat; although initially many others may not follow her, after few moments more and more other birds start leaving the wire as they see other fellows doing it, so that and eventually realizes that it might be wiser to take off.

From the previous anecdotal example, we can abstract the following distributed computing problem. We have a system of agents in the $\mathcal{PULL}$ model, and one of them, the *source*, has some important piece of information that the system could use, which we call *input bit*. At the outset, each agent as a guess on the value of the input bit. However, there is no assumption on the initial states of the agents: some of them, for example, may hold a wrong assumption on the value of the input bit. Therefore, we would like to devise a strategy, as simple as possible, such that the system can rapidly reach consensus on the true value of the input bit, starting from an arbitrary initial configuration of the agents' states.

Equivalently, given the previous abstract formulation, we are essentially asked to solve the *self-stabilizing* consensus problem in the setting in which there is one agent (the source) which *does not change her mind* (she *knows* the true input bit). By leveraging on the power of simple dynamics, we show a sound connection between the self-stabilizing information propagation problem and the problem of synchronizing clocks, in a self-stabilizing manner, in the uniform $\mathcal{PULL}$ model [4]. We thus end up devising a solution for the self-stabilizing clock synchronization problem. Our solution uses, as a subroutine, any dynamics for majority consensus such as the 3-Majority dynamics and, in the uniform $\mathcal{PULL}$ model using messages of 3 bits only, the presented solution allows the agents to synchronize a clock modulo T in time essentially logarithmic in T and the size of the system. This allows to remove the assumption of an initial common time notion from an entire class of protocols, and provides a general solution for the self-stabilizing *majority* information propagation problem, which is a generalization of the aforementioned information propagation problem which includes the majority consensus problem as a special case.

References

1. L Becchetti, A. Clementi, E. Natale, F. Pasquale, and G. Posta. Self-stabilizing repeated balls-into-bins. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'15)*, pages 332–339. ACM, 2015.
2. L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri. Plurality consensus in the gossip model. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15)*, pages 371–390. SIAM, 2015.
3. L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan. Simple dynamics for plurality consensus. In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'14)*, pages 247–256. ACM, 2014.
4. L. Boczkowski, A. Korman, and E. Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. In *Proceed-*

ings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17), pages 2540–2559. SIAM, 2017.

5. O. Feinerman, B. Haeupler, and A. Korman. Breathe before speaking: Efficient information dissemination despite noisy, limited and anonymous communication. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '14)*, pages 114–123. ACM, 2014.
6. P. Fraigniaud and E. Natale. Noisy rumor spreading and plurality consensus. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC'16)*, pages 127–136. ACM, 2016.
7. Emanuele Natale. *On the computational power of simple dynamics.* PhD thesis, 2017.