HAL
archives-ouvertes.fr

Computing with Large Populations Using Interactions

Olivier Bournez, Pierre Fraigniaud, Xavier Koegler

To cite this version:

Olivier Bournez, Pierre Fraigniaud, Xavier Koegler. Computing with Large Populations Using Interactions. Mathematical Foundations of Computer Science 2012, MFCS'2012, Aug 2012, Slovakia. pp.234-246. hal-00760669

HAL Id: hal-00760669

https://hal-polytechnique.archives-ouvertes.fr/hal-00760669

Submitted on 4 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing with Large Populations Using Interactions

Olivier Bournez[1⋆], Pierre Fraigniaud[2⋆⋆], and Xavier Koegler[2]

[1] LIX, Ecole Polytechnique and CNRS, France.
[2] LIAFA, CNRS and University Paris Diderot, France.

Abstract. We define a general model capturing the behavior of a population of anonymous agents that interact in pairs. This model captures some of the main features of opportunistic networks, in which nodes (such as the ones of a mobile ad hoc networks) meet sporadically. For its reminiscence to Population Protocol, we call our model *Large-Population Protocol*, or LPP. We are interested in the design of LPPs enforcing, for every $\nu \in [0,1]$, a proportion ν of the agents to be in a specific subset of marked states, when the size of the population grows to infinity; In which case, we say that the protocol *computes* ν. We prove that, for every $\nu \in [0,1]$, ν is computable by a LPP if and only if ν is algebraic. Our positive result is constructive. That is, we show how to construct, for every algebraic number $\nu \in [0,1]$, a protocol which computes ν.

1 Introduction

1.1 Motivation. So-called *opportunistic networks* (see, e.g., [32]) are characterized by connections between users that appear sporadically, which are as many opportunities for exchanging data or forwarding messages. As such, they form a subclass of the so-called delay-tolerant networks (DTNs). A typical and probably prominent example of opportunistic networks is sparse mobile ad hoc networks, as analyzed in, e.g., the Zebra project [29], as well as in several other projects aiming at understanding the potential of opportunistic networks [15, 27]. This paper is interested in the computing power of such networks.

More specifically, we are focussing on the *slicing* problem [28], which refers to the ability of creating a virtualized network running over multiple physical nodes, where the nodes are partitioned in multiple *slices*. Many metrics may be used to sort the nodes for assigning them to different slices. Typical metrics are available resources, such as memory, bandwidth, or computing power. However, as underlined in [28], slicing the network by focusing only on the size of the slices, also deserves to be investigated, for applications to systems involving devices with similar resource capabilities. Independently from the context, we stress that the slice sizes are usually expressed as a percentage of the network size. Slicing algorithms in the literature are usually designed for peer-to-peer systems in which powerful peers are capable of generating random numbers [28], or taking advantage of a storage capacity proportional to the population size [23] or to the

⋆ Supported by ANR project SHAMAN.
⋆⋆ The last two authors are supported by ANR projects DISPLEXITY and PROSE, and by INRIA project GANG.

size of the neighborhood [22]. In this paper, we show how to slice the nodes of opportunistic networks composed of tiny and simple devices, *deterministically*, using only the basic resources of *finite state* agents, by taking benefits of the randomized interactions between these agents.

To illustrate our objective, consider a population of nodes (e.g., sensors) moving in a somewhat restricted environment. Assume moreover, for the sake of simplifying the presentation and the analysis, that nodes meet uniformly at random over time. Let us say one wants to slice these nodes into two slices of equal size. The following trivial 2-state protocol achieves this. Nodes are either in state positive or negative, and whenever two nodes meet, one of them becomes positive while the other becomes negative. Nodes then tend over time to partition themselves into two slices of equal size, when the size of the population grows to infinity. As a more sophisticated example, assume that one wants to slice the nodes in order to construct a slice such that the probability that two nodes in this slice meet is $1/2$. That is, one wants to slice the nodes into two slices, with one slice amounting to a ratio $1/\sqrt{2}$ of the total number of nodes. This is ensured by the following 2-state protocol. Whenever two nodes meet, their actions depend on their current states $+$ or $-$: if the nodes are in different states then both become positive; otherwise, one of them becomes positive while the other becomes negative. This dynamic can be summarized by the four transition rules:

$$\begin{array}{lcl} +- & \rightarrow & ++ \\ -+ & \rightarrow & ++ \end{array} \qquad \begin{array}{lcl} ++ & \rightarrow & +- \\ -- & \rightarrow & +- \end{array} \tag{1}$$

This protocol has been extensively analyzed in [14], where it is proved that the proportion of positive nodes does converge to the desired ratio $1/\sqrt{2}$ over time, when the size of the population grows to infinity.

In some sense, the former protocol *computes* $1/2$, while the latter protocol computes $1/\sqrt{2}$. One may of course be interested in computing other values. To start with, beside $1/2$, is it possible to compute every rational in $[0, 1]$? And beside rationals, is the protocol for $1/\sqrt{2}$ extendable to ratios of the form $x^{-1/k}$ for every $x \geq 1$ and $k \geq 2$? More generally, what is the limit of such protocols in term of their computing power? For instance, is it possible to compute solutions of trigonometric equations? E.g., can we design a protocol insuring that, asymptotically, a ratio $\frac{\pi}{4}$ of the nodes are in some prescribed state?

1.2 Framework. In order to determine the computing power of a collection of nodes such as the ones involved in an opportunistic network, we abstract our model from the specific technological constraints to be faced when one is dealing with networks (security, forwarding mechanism, mobility, etc.). In fact, a network applying a protocol such as the one in Eq. 1 can be considered as a *population protocol*, in the spirit of the model introduced in [3]. Essentially, in population protocols, nodes (or agents) are anonymous, their interactions are supposed to be scheduled so as to satisfy some natural fairness property, and their individual actions are independent from the size of the population.

Classical population protocols are however designed to compute *predicates* over their input configuration whereas the protocol in Eq. 1 computes a (real) value (i.e., the proportion of agents in positive state), independently from the initial configuration. Another difference with classical population protocols is that the result of our computation is asymptotic, when the size of the population grows to infinity. These differences can be tackled by considering a setting that we call *Large-Population Protocols* (LPPs), which is essentially population protocols whose behaviors are analyzed when the population grows to infinity. In this context, for any given number $\nu \in [0,1]$, a LPP is said to *compute* ν if the proportion of agents in some specific states, say the *marked* states, converges over time to ν when the population grows to infinity.

We can now reformulate the question raised in the previous section slightly more formally by: what can be computed by a LPP? More precisely, we address the problem of determining which numbers can be computed by Large-Population Protocols. That is, we are aiming at identifying the set of real numbers $\nu \in [0,1]$ for which there exists a LPP computing ν.

1.3 Our results. We first define formally our model for Large-Population Protocols (LPP). The model is quite general in the sense that it encompasses all the models involving a "population" of agents, whenever they are dealing with anonymous agents that interact in pairs. We then prove that the execution of any LPP is well characterized by the behavior of a differential system. This characterization can be considered similar to what is usually done in mean field theory. However, we go beyond the simple application of mean field analysis by completely formalizing the connection between the execution of a LPP and the behavior of the corresponding differential system. Specifically, fix any protocol $\mathcal{P}$, and define $X = X(n,t)$ as the random variables equal to the proportion of agents in marked state at time t in a population of size n, during the execution of $\mathcal{P}$. We characterize the exact behavior of X when the size of population grows to infinity. Essentially, we prove that, whenever the initial state is close enough to a stable equilibrium, we have

$$X(n,t) \approx f(t) + \frac{1}{\sqrt{n}}\,\mathcal{N}(0,\chi) \tag{2}$$

where f is the solution of the differential system corresponding to $\mathcal{P}$, $\mathcal{N}(0,\chi)$ is a centered gaussian with covariance matrix χ depending on $\mathcal{P}$, and $\approx$ denotes a convergence in law after an appropriate rescaling.

Using the correspondence between LPPs and differential systems, we characterize the real numbers that are computable by LPPs as being all *algebraic* numbers in $[0,1]$. The fact that transcendental numbers cannot be computed by LPPs is a consequence of arguments from model theory (mainly Tarski's effective procedure for quantifier elimination over real closed fields). Our main result is a proof that all algebraic numbers can be computed by LPPs. Our proof is constructive, that is, for every algebraic number $\nu \in [0,1]$ described by some polynomial P, with rational coefficients, and satisfying $P(\nu) = 0$, we show how to construct a LPP computing ν in the sense of Eq. 2. That is, $X(n,t)$ satisfies the relation of Eq. 2, with $\lim_{t\to\infty} f(t) = \nu$.

The algorithmic construction proceeds in four stages. The first stage consists in constructing, for every *rational* ν, a LPP computing ν. The second stage of the construction consists in a form of derandomization for LPP. More precisely, we show that every protocol involving *probabilistic* transition rules where each probability is rational can be transformed into a protocol involving solely *deterministic* transition rules. Hence, the remaining two stages involve protocols using probabilistic transition rules. The third stage is heavily based on our characterization of LPP using differential systems. We construct a differential system corresponding to a LPP, and admitting ν as an equilibrium. This system is obtained by identifying a multivariate polynomial $\tilde{P}$ such that $\tilde{P}(x_1, \ldots, x_k) = P(x_1)$ whenever $x_i = x_{i-1}x_1$, where k is the degree of P. Interestingly, $\tilde{P}$ is *not* P in which x^i would be replaced by x_i. Instead, $\tilde{P}$ is specifically designed so that to yield a differential system which admits ν as an equilibrium, that can be in turn transformed into a protocol with rational probabilistic transitions. The fourth and last stage of the construction involves stability. We show how to modify the construction of the third stage to enforce that ν becomes a *stable* equilibrium. This is achieved by carefully modifying the polynomial $\tilde{P}$ so that the Routh-Hurwitz stability criterion can be applied, while preserving the ability to translate the differential system into a LPP.

1.4 Related work. The model that we consider in this paper captures the behavior of any *large* population of *indistinguishable* agents interacting *in pairs* in a *Markovian* manner. This framework includes many models from nature, physics, and biology (see, e.g., [31]). Several papers have already demonstrated the benefit of using an algorithmic approach for understanding such models (see, e.g., the recent papers [12, 18, 20]). Conversely, models from nature, physics, and biology can be viewed as alternative paradigms of computation (see, e.g., [1, 11]).

Classical models for capturing the dynamics of populations include *Lotka-Volterra* dynamics for *predator-prey* models, *replicator* dynamics, and, more generally, all kinds of models from evolutionary game theory. In particular, it is known that a subclass of protocols designed in the context of evolutionary game theory correspond to Lotka Volterra dynamics [17], which are in turn known to be equivalent to replicator dynamics [26]. The connections between the dynamics of games and population protocols has been studied in [13, 17].

Population protocols have been introduced in [3]. The model was designed to decide logic predicates, and predicates computable by classical population protocols have been characterized [3, 4] as being precisely the semi-linear predicates, that is, those predicates on counts of input agents definable in first-order Presburger arithmetic. Variants of the original model considered so far include restrictions on communications [2, 5], random interactions [3, 8, 7], and mediated interactions [30]. Various kinds of fault tolerance have also been considered for population protocols [6, 21]. We refer to [9] for a comprehensive introduction to population protocols, and to [16, 19] for the description of formal methods for verifying such protocols.

A few papers addressed the asymptotic behavior of population protocols, when the population size grows to infinity. In [24], a framework for translating

certain subclasses of differential equation systems into practical protocols for distributed systems, assuming a large population, is described. In [17], the authors study the dynamics and the stability of (probabilistic) population protocols via ordinary differential equations. [14] proves that there exists a close relationship between, on the one hand, classical finite population protocols, and, on the other hand, models obtained by ordinary differential equations. The protocol computing $1/\sqrt{2}$ described in Eq. 1 has been thoroughly studied in [14] where convergence is proved using weak-convergence methods for stochastic processes. In [10], the authors address the issue of convergence speed. It is proved that it is possible to compute $1/\sqrt{2}$ with arbitrary precision $\epsilon > 0$ in a time polynomial in $1/\epsilon$, using a number of agents polynomial in $1/\epsilon$.

2 Large-Population Protocols

In this section, we define Large-Population Protocols (LPP), and state formally what is meant by computing with LPPs. The general idea of the model has been introduced in [10] and [14]. The following subsection recalls the main features of the model. Our first contribution is a formal specification of the asymptotic behavior of a LPP.

2.1 The model. We consider a population of n anonymous agents, each of which can be in finitely many possible states, from a finite set Q. This population evolves with time. We assume a synchronous discrete-time system, and, at each round, two agents a and b are selected among the n agents. The selection is performed uniformly at random, independently from the past. Note that the original population protocol model [3] just assumes a specific fairness hypothesis for the interactions between the agents, which are under the control of an adversary with restricted power. Nevertheless, when the size of the population goes to infinity, uniform sampling of agents appears to be a natural way to extend the fairness hypothesis used in classical population protocols. Moreover, uniform sampling is consistent with the interpretation of agents as autonomous entities moving at random. (See also [9] for a discussion on the random adversary in finite state systems.)

The two agents a, b that are selected, can interact and change their states according to a set Δ of transition rules of the form $q_i\ q_j \rightarrow q_k\ q_l$ where $(q_k, q_l) = \Delta(q_i, q_j)$.

Note that the transition is not necessarily symmetric, i.e., the selected pair (a, b) may cause a transition different from the one caused by the pair (b, a). In other words, we do not necessarily assume $\Delta(q_i, q_j) = \Delta(q_j, q_i)$. Let us identify a specific subset Q^+ of states of Q, say $\{q_1, q_2, \cdots, q_m\}$, to be the *marked state*, and denote $Q = \{q_1, q_2, \cdots, q_{|Q|}\}$. The pair (Q, Δ) entirely defines a protocol $\mathcal{P}$. Such a protocol is called *large-population protocol* because, informally, we will say that $\mathcal{P}$ computes some given number ν if $\mathcal{P}$ enforces the proportion of agents in states $q \in Q^+$ to converge to ν along with time, when the size n of the population grows to infinity.

To get some intuition of how to formally define computation, assume first that n is fixed, and assume $m = 1$. The evolution (with time) of the population can be modeled by a discrete-time homogeneous Markov chain whose states are all the possible configurations of the system. For the sake of simplifying the discussion, assume first that the Markov chain is irreductible. (Whether it is the case or not depends on the protocol $\mathcal{P}$.) Let $Y_i(t)$ be the random variables equal to the numbers of agents in state q_i at time t, and let $Y(t) = (Y_1(t), \ldots, Y_{|Q|}(t))$. Let us now consider the Markov chain defined by $\overline{Y}(t) = \frac{1}{n}(Y_1(t), \cdots, Y_{|Q|}(t))$. A consequence of the Ergodic Theorem (this is where we use the irreducibility assumption) is that the chain $\overline{Y}(t)$ admits a unique stationary distribution, say $\mu = (\mu_1, \mu_2, \ldots, \mu_{|Q|})$. Hence, for any initial state of the population, the distribution of $\overline{Y}(t)$ converges to μ when t goes to infinity. In particular, the distribution of $\overline{Y}_1(t)$, the proportion of agents in the marked state q_1 at time t, converges to the distribution μ_1 when t goes to infinity. As a consequence, the expected value of $\overline{Y}_1(t)$ converges to the expected value $\mathbf{E}\mu_1$ of μ_1. Intuitively, we are interested in the limit of $\mathbf{E}\mu_1$ when n grows to infinity. The difficulty comes from the fact that a protocol is dealing with $\overline{Y}_1(t)$, which depends on both t and n. The study of this double limit must be treated with care in the general case, which is the purpose of the remainder of this section.

Notice that the limit of $\mathbf{E}\mu_1$ can be a non-rational real number, whereas, when n is fixed, the expected value of μ_1 is necessarily a rational number since, for every i, we have $\overline{Y}_i(t) \in \left\{\frac{1}{n}, \ldots, \frac{n}{n}\right\}$. So $\overline{Y}_i(t)$ is a Markov chain over this latter set. As a consequence, the distribution μ_i is a distribution over this latter set. Since the stationary distribution μ is the solution of a set of linear equations with rational coefficients, μ is necessarily weighting the elements of $\left\{\frac{1}{n}, \ldots, \frac{n}{n}\right\}$ with rational quantities. In particular, the expected value of μ_1 satisfies $\mathbf{E}\mu_1 = \sum_{i=1}^{n} \mu_1(\frac{i}{n}) \cdot \frac{i}{n}$, and thus is a rational number.

To handle the growth of the population, one must perform a time rescaling. Let us redefine the notations so that to capture explicitely the size n of the population. Let $Y_i^{(n)}(t)$ be the random variable equal to the numbers of agents in state q_i at time t in a population of size n, and let $\overline{Y}_i^{(n)}(t) = \frac{1}{n} \cdot Y_i^{(n)}(t)$. Let $\overline{Y}^{(n)}(t) = (\overline{Y}_1^{(n)}(t), \ldots, \overline{Y}_{|Q|}^{(n)}(t))$. Then let

$$X^{(n)}(t) = \overline{Y}^{(n)}(\lfloor nt \rfloor) + (nt - \lfloor nt \rfloor) \cdot (\overline{Y}^{(n)}(\lfloor nt + 1 \rfloor) - \overline{Y}^{(n)}(\lfloor nt \rfloor)).$$

By definition, $X^{(n)}(t)$ is a continuous-time Markov chain obtained by linear interpolation of $\overline{Y}^{(n)}$ with a time-acceleration of factor n. After rescaling, the number of interactions occurring in one time-unit is proportional to the number of agents in the population. To capture the asymptotic behavior of $X^{(n)}(t)$, we use a *balance* equation. Let $(e_q)_{q \in Q}$ be the canonical base of $\mathbb{R}^{|Q|}$, and let $b : \mathbb{R}^{|Q|} \to \mathbb{R}^{|Q|}$ be the function defined by:

$$b(x) = \sum_{(q_1,q_2) \in Q^2} \Big(x_{q_1} x_{q_2} \Big(- e_{q_1} - e_{q_2} + \sum_{(q_3,q_4) \in Q^2} \delta_{q_1,q_2,q_3,q_4} (e_{q_3} + e_{q_4}) \Big) \Big) \quad (3)$$

where $\delta_{q_1,q_2,q_3,q_4} = 1$ if $\Delta(q_1, q_2) = (q_3, q_4)$, and 0 otherwise. The function b acts as a balance equation. That is, assuming that the proportion of agents in

state q is $x_q \in \mathbb{R}$ for every $q \in Q$, then one expects each rule $q_i\ q_j \to q_k\ q_l$ to happen with probability $x_{q_i} x_{q_j}$. Accounting for this balance for all rules, and considering that all produced states must be consumed by some rule, yield that if the proportion of states converges to some equilibrium x, then this equilibrium must satisfy $b(x) = 0$. The following has been proved in [14].

Lemma 1 (Theorem 4 of [14]). *For every initial condition $Y^{(n)}(0)$ with $Y^{(n)}(0) \to x$ when $n \to \infty$, the sequence of random processes $X^{(n)}$ converges in law to the solution of the stochastic differential equation (with degenerated brownian motion): $dX(t) = b(X(t))dt$, with $X(0) = x$.*

2.2 Computing with LPPs. In view of Lemma 1, we get that the behavior of a protocol can be well approached by an ordinary differential equation, when the size of the population becomes large. In particular, if x^* is some stable equilibrium of the differential equation, then one expects $\overline{Y}^{(n)}(t)$ to converge to x^* whenever it starts close enough to x^*. Unfortunately, the notion of convergence involved in Lemma 1 (i.e., convergence in law) is too weak to derive this conclusion directly. On the other hand, it is actually possible to go further in the analysis of population protocol, in order to provide a deeper understanding of the convergence. By doing so, we are able to provide an asymptotic development for $\overline{Y}^{(n)}(t)$, as stated in the following result.

Theorem 1. *Assume that the ordinary differential in Lemma 1 has a stable equilibrium $b(x^*) = 0$. Then there exists a neighborhood of x^* such that, whenever $\overline{Y}^{(n)}(0)$ belongs to this neighborhood, we have $\overline{Y}^{(n)}(t) \approx x^* + \frac{1}{\sqrt{n}}\,\mathcal{N}(0,\chi)$ when $t \to \infty$, where $\mathcal{N}(0,\chi)$ is the centered gaussian distribution with covariance matrix χ, for some χ, and $\approx$ denotes convergence in law of the rescaling of $\overline{Y}^{(n)}(t)$ when $t \to \infty$.*

By an adaptation of the arguments in [10], one can also show that if the ordinary differential equation in Lemma 1 has a stable equilibrium $b(x^*) = 0$, then, for every $\epsilon > 0$, and for every $0 < p < 1$, there is a neighborhood U of x^* and some integers n and t, both polynomial in $1/\epsilon$, which guarantee that, with probability at least p, we have $\|\overline{Y}^{(n)}(t) - x^*\| \leq \epsilon$ whenever the initial configuration belongs to U.

We have now all ingredients to formally define computing with LPPs.

Let $\mathcal{P} = (Q, \Delta)$ be a LPP. A vector of real numbers $x^* = (x_1, \ldots, x_{|Q|}) \in [0,1]^{|Q|}$ is said to be an *equilibrium* of a $\mathcal{P}$ if and only if $b(x^*) = 0$, that is to say the constant solution $f(t) = (x_1, \ldots, x_{|Q|})$ is a fix-point solution of the differential equation in Lemma 1. An equilibrium x^* of $\mathcal{P}$ is said to be *stable* if it is the (exponentially) stable equilibrium of the associated ordinary differential equation. In other words, there is a neighborhood U of the equilibrium x^* such that any trajectory starting from U converges exponentially fast to the equilibrium. This is equivalent to saying that the Eigenvalues of the Jacobian matrix of b in x^* has negative real parts [25].

Definition 1. *A real number ν is said to be computable by LPP if there exists a vector $x^* = (x_1, x_2, ..., x_k) \in [0,1]^k$ such that $\sum_{i=1}^{k} x_i = 1$, and a LPP $\mathcal{P}$, admitting finitely many equilibria, such that $(x_1, x_2, ..., x_k)$ is a stable equilibrium of $\mathcal{P}$ and $\sum_{q_i \in Q^+} x_i = \nu$ where Q^+ is the set of marked states for $\mathcal{P}$.*

Notice that the above definition requires the system to have finitely many equilibria. This assumption is mainly to avoid pathological cases, in particular the case of idle systems $q\ q' \to q\ q'$ for all q, q'. Indeed, in idle systems, all initial states are equilibria, and such a system would compute any real of $[0,1]$, depending on the initial configuration.

3 The computational power of LPPs

In this section, we establish our main result:

Theorem 2. *Every $\nu \in [0,1]$ is computable by a LPP if and only if it is algebraic.*

We first prove that there is an intrinsic limitation to the power of LPPs, namely not a single transcendental number can be computed by LPPs. Indeed, a direct consequence of arguments from model theory (mainly Tarski's effective procedure for quantifier elimination over real closed fields) allows us to prove the following lemma:

Lemma 2. *For every $\nu \in [0,1]$, if ν is computable by a LPP then ν is algebraic.*

The remaining part of the section is entirely dedicated to proving that every algebraic number is indeed computable by a LPP. The proof is constructive, meaning that we describe how to construct a LPP computing ν, for any given algebraic number $\nu \in [0,1]$. The construction of the protocol is made in four stages, corresponding to the following four subsections. The first stage consists in the design of LPPs computing rational numbers. The second stage consists in using the computation of rational numbers as a subroutine for the emulation of probabilistic transition rules. This stage will allow us to consider LPPs with transition rules of the form

$$q_i\ q_j \to \alpha_{i,j,k,l}\ q_k\ q_l$$

to be understood as: the interaction between two agents in respective states q_i and q_j results in the two agents moving to respective states q_k and q_l with probability $\alpha_{i,j,k,l}$. Then, the third stage of our proof is the construction of a (probabilistic) protocol $\mathcal{P}$ admitting ν as an equilibrium. We assume we are given a degree-δ polynomial $P \in \mathbb{Q}[X]$ with root ν. The protocol $\mathcal{P}$ is based on one specific choice for another degree-δ polynomial $P' \in \mathbb{Q}[X]$, and, essentially, satisfies that $(x_1, \ldots, x_\delta) \in [0,1]^\delta$ is an equilibrium of $\mathcal{P}$ if and only if (1) $P'(x_1) = 0$, (2) $x_i = x_1^i$ for every $1 \leq i < \delta$, and (3) $x_\delta = 1 - \sum_{i=1}^{\delta} x_i$. Finally, the fourth stage of the construction consists in proving that we can actually enforce this protocol $\mathcal{P}$ to be stable near the equilibrium.

3.1 Computing Rationals.

Lemma 3. *Let $\nu \in [0,1]$ be a rational number. There exists a LPP computing ν.*

Proof. We first show that, for every integer $k \in \mathbb{N}$, there exists a protocol that, given any initial configuration, converges to the unique equilibrium $(\frac{1}{k}, \ldots, \frac{1}{k})$. For this purpose, consider the protocol $\mathcal{M}$ over states $Q_k = \{1, \ldots, k\}$ given by the following transition rules: $i\ j \to (i+1)\ (j+1)$ where, for $q \in Q_k$, $q+1$ stands for $(q \bmod k) + 1$. The dynamic system describing this protocol is

$$\frac{dx_i}{dt} = 2(x_{i-1} - x_i).$$

If $f : \mathbb{R} \to [0,1]^k$ is a solution of this differential system, then, considering $g(t) = \|f(t) - (\frac{1}{k}, \ldots, \frac{1}{k})\|^2$, where $\|x\|$ is the euclidian norm of vector x, we get

$$\frac{dg(t)}{dt} = 4 \sum_{i=1}^{k} x_i(x_{i-1} - x_i).$$

A simple induction on $k \in \mathbb{N}$ enables to show that $\frac{dg(t)}{dt} \leq 0$ for every vector $x \in [0,1]^k$, and $\frac{dg(t)}{dt} = 0$ if and only if $x_1 = x_2 = \ldots = x_k$, thereby proving that f converges to $(\frac{1}{k}, \cdots, \frac{1}{k})$ when $t \to \infty$. This, in particular, guarantees that $(\frac{1}{k}, \ldots, \frac{1}{k})$ is the only stable equilibrium of $\mathcal{M}$.

Now, let $\nu = p/q \in \mathbb{Q}$. Computing ν is achieved by using $\mathcal{M}$ as above, with $k = q$, and setting marked states as the first p states of Q. □

3.2 Derandomization. We now prove that considering LPPs with probabilistic transition rules where the probabilities are rational does not change the computing power of LPPs. Note that this result has its own independent interest. It essentially says that the random choice of the agents involved in the transition provides enough randomness, and that there are no further benefits from using randomization in the transition rules. Nevertheless, using probabilistic LPPs, or PLPPs for short, considerably simplifies the construction of LPPs computing algebraic numbers.

We focus on PLPPs, where the transition rules are defined by

$$q_i\ q_j \to \alpha_{i,j,k,l}\ q_k\ q_l$$

where all $\alpha_{i,j,k,l}$ are rational numbers[1]. Recall that such probabilistic transition rules mean that an interacting pair of agents in respective states q_i and q_j will move to respective states q_k and q_l, with probability $\alpha_{i,j,k,l}$. Of course, for such rules to be well-defined, we assume that for every pair $(q_i, q_j) \in Q^2$, we have

- for every $(q_k, q_l) \in Q^2$, $\alpha_{i,j,k,l} \geq 0$, and
- $\sum_{(q_k,q_l)\in Q^2} \alpha_{i,j,k,l} = 1$.

[1] In fact, our derandomization technique could be extended to the case in which the $\alpha_{i,j,k,l}$ are computable by a LPP. This would however overload the presentation, and the stronger assumption that $\alpha_{i,j,k,l} \in \mathbb{Q}$ is anyway sufficient for our purpose.

Notice that all previous definitions and statements can be easily extended to PLPPs. In particular, Lemma 1 and Theorem 1 still hold, by replacing function b in Eq. 3 by

$$b(x) = \sum_{(q_1,q_2)\in Q^2} x_{q_1}x_{q_2}\Big(- e_{q_1} - e_{q_2} + \sum_{(q_3,q_4)\in Q^2} \alpha_{q_1,q_2,q_3,q_4}(e_{q_3} + e_{q_4})\Big).$$

Lemma 4. *Let $\nu \in [0,1]$, and assume that there exists a probabilistic LPP computing ν, with rational probabilities. Then there exists a (deterministic) LPP computing ν.*

3.3 Constructing Equilibria. In view of the previous two subsections, one can freely use probabilistic LPPs, whenever the probabilities are rational, in order to compute any algebraic number $\nu \in [0,1]$. In this section, we will not yet produce a probabilistic LPPs computing an algebraic number ν, as we will ignore stability which is only discussed in the next section, and solely focus on constructing a protocol with ν as an equilibrium.

Lemma 5. *For every algebraic number $\nu \in [0,1]$, there exist $\delta \in \mathbb{N}$, $\lambda \in \mathbb{Q}$, and a protocol $\mathcal{P}$ such that $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ is an equilibrium of $\mathcal{P}$.*

Proof. Let $\nu \in (0,1]$ be an algebraic number, and let $P(X) = \sum_{i=0}^{\delta} a_i X^i, P \in \mathbb{Q}[X]$, be a polynomial such that $P(\nu) = 0$, and $P(0) > 0$. We claim that there exist a rational number $\epsilon \neq 0$, and a protocol $\mathcal{P}_\epsilon$ with equilibrium $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots$ $\ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1-\sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ described by the following differential equations:

$$\begin{cases} dx_1 = \epsilon(a_0 + a_1x_1 + \sum_{i=2}^{\delta-1} \frac{a_{i+1}}{\lambda^{i-1}} x_{i-1}x_1) \\ dx_i = \lambda x_1 x_{i-1} - x_i \ \text{ for every } i = 2, \ldots, \delta-1 \\ dx_\delta = -\sum_{i=1}^{\delta-1} dx_i. \end{cases} \tag{4}$$

where λ is a rational number such that $\lambda > 0$, and $\sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i \leq 1$. To establish that claim, we explicitly construct a protocol $\mathcal{P}_\epsilon$ over set of states $Q = \{1, \ldots, \delta\}$ with 1 serving as our marked state. Fix $\lambda \in \mathbb{Q}$, $\lambda > 0$ small enough, so that $\sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i \leq 1$. Then let

$$M = \max\Big(\{|\frac{a_{i+1}}{\lambda^{i-1}} + 2a_0 + a_1|, i \in \{2, ..., \delta-1\}\} \cup \{|a_2 + a_0 + a_1|, |2a_0 + a_1|, a_0\}\Big)$$

and fix $\epsilon \in \mathbb{Q}$, $0 < \epsilon < \frac{1}{M}(1 - \frac{\lambda}{2})$. We define the family $(\alpha_{i,j,k,l})_{1\leq i,j,k,l\leq\delta}$, that yields the transition rules for the protocol $\mathcal{P}_\epsilon$ as follows:

$$\begin{cases} i = 1, j = 1 & \Longrightarrow \quad \alpha_{1,1,1,1} = \epsilon\, \frac{a_2+a_1+a_0}{2} + \frac{1}{2} \text{ and } \alpha_{1,1,2,2} = \frac{\lambda}{2} \\ i = 1, 1 < j < \delta-1 & \Longrightarrow \quad \alpha_{1,j,1,1} = \epsilon\, \frac{\frac{a_{j+1}}{\lambda^{j-1}}+2a_0+a_1}{4} + \frac{1}{2} \text{ and } \alpha_{1,j,j+1,j+1} = \frac{\lambda}{4} \\ i = 1, j = \delta-1 & \Longrightarrow \quad \alpha_{1,j,1,1} = \epsilon\, \frac{\frac{2a_\delta}{k^{\delta-2}}+2a_0+a_1}{4} + \frac{1}{2} \\ i = 1, j = \delta & \Longrightarrow \quad \alpha_{1,j,1,1} = \epsilon\, \frac{2a_0+a_1}{4} + \frac{1}{2} \\ 1 < i < \delta-1, j = 1 & \Longrightarrow \quad \alpha_{i,1,1,1} = \epsilon\, \frac{\frac{a_{i+1}}{\lambda^{i-1}}+2a_0+a_1}{4} + \frac{1}{2} \text{ and } \alpha_{i,1,i+1,i+1} = \frac{\lambda}{4} \\ i = \delta-1, j = 1 & \Longrightarrow \quad \alpha_{i,1,1,1} = \epsilon\, \frac{\frac{2a_\delta}{k^{\delta-2}}+2a_0+a_1}{4} + \frac{1}{2} \\ i = \delta, j = 1 & \Longrightarrow \quad \alpha_{i,1,1,1} = \epsilon\, \frac{2a_0+a_1}{4} + \frac{1}{2} \\ i > 1, j > 1 & \Longrightarrow \quad \alpha_{i,j,1,1} = \epsilon\, \frac{a_0}{2} \end{cases}$$

And, for all $(k,l) \neq (\delta,\delta)$ not considered above, we set $\alpha_{i,j,k,l} = 0$. Finally, if $(k,l) = (\delta,\delta)$, then $\alpha_{i,j,\delta,\delta} = 1 - \sum_{(k,l)\neq(\delta,\delta)} \alpha_{i,j,k,l}$.

By definition of M and ϵ, it follows that, for any pair (i,j), if $(k,l) \neq (\delta,\delta)$, then $0 \leq \alpha_{i,j,k,l} \leq 1$. Moreover, we have $0 \leq \sum_{(k,l)\neq(\delta,\delta)} \alpha_{i,j,k,l} \leq 1$. Thus, for every (i,j), $0 \leq \alpha_{i,j,\delta,\delta} \leq 1$. Therefore, the family $(\alpha_{i,j,k,l})$ properly defines a protocol $\mathcal{P}_\epsilon$. We now show that this protocol satisfies our needs. By construction, the dynamic of $\mathcal{P}_\epsilon$ is captured by the following system :

$$\forall k \in \{1,\ldots,\delta\},\ dx_k = \sum_{l=1}^{\delta} \sum_{i,j} (\alpha_{i,j,k,l} + \alpha_{i,j,l,k}) x_i x_j - x_k$$

which precisely yields Eq. 4. □

3.4. Enforcing Stability. Perhaps surprisingly, stability does not come for free, and the construction of the previous section is not sufficient to conclude. One needs to enforce stability. For that purpose, the protocol of the previous section is modified in order to satisfy the stability criteria from the theory of dynamic systems. The following result completes the proof of Theorem 2.

Lemma 6. *For every algebraic number $\nu \in [0,1]$, there exist $\delta \in \mathbb{N}$, $\lambda \in \mathbb{Q}$, and a protocol $\mathcal{P}$ such that $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ is a* stable *equilibrium of $\mathcal{P}$.*

References

1. L.M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021, 1994.
2. D. Angluin, J. Aspnes, M. Chan, M. J. Fischer, H. Jiang, and R. Peralta. Stably computable properties of network graphs. In *DCOSS*, volume 3560 of *LNCS*, pages 63–74. Springer-Verlag, 2005.
3. D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. In *PODC*, pages 290–299, 2004.
4. D. Angluin, J. Aspnes, and D. Eisenstat. Stably computable predicates are semilinear. In *PODC*, pages 292–299, 2006.
5. D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.
6. D. Angluin, J. Aspnes, M. J. Fischer, and H. Jiang. Self-stabilizing population protocols. In *OPODIS*, LNCS, pages 79–90. Springer, 2005.
7. Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, 2008.
8. Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.
9. J. Aspnes and E. Ruppert. An introduction to population protocols. *Bulletin of the EATCS*, 93:106–125, 2007.
10. G. Aupy and O. Bournez. On the number of binary-minded individuals required to compute $\sqrt{1/2}$. *Theoretical Computer Science*, 412(22):2219–2456, 2010.

11. G. Berry. The chemical abstract machine. *TCS*, 96(1):217–248, 1992.
12. V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *44th IEEE Conf. on Decision and Control*, pages 2996–3000, 2005.
13. O. Bournez, J. Chalopin, J. Cohen, X. Koegler, and M. Rabie. Computing with pavlovian populations. In *OPODIS*, pages 409–420, 2011.
14. O. Bournez, P. Chassaing, J. Cohen, L Gerin, and X. Koegler. On the convergence of population protocols when population goes to infinity. *Applied Math. and Computation*, 2009.
15. A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.
16. I. Chatzigiannakis, O. Michail, and P. G. Spirakis. Algorithmic verification of population protocols. In *SSS*, pages 221–235, 2010.
17. I. Chatzigiannakis and P. G. Spirakis. The dynamics of probabilistic population protocols. In *DISC*, volume 5218 of *LNCS*, pages 498–499, 2008.
18. B. Chazelle. Natural algorithms. In *SODA*, pages 422–431, 2009.
19. J. Clément, C. Delporte-Gallet, H. Fauconnier, and M. Sighireanu. Guidelines for the verification of population protocols. In *ICDCS*, pages 215–224, 2011.
20. F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.
21. C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, and E. Ruppert. When birds die: Making population protocols fault-tolerant. In *DCOSS*, volume 4026 of *LNCS*, pages 51–66. Springer, 2006.
22. A. Fernández, V. Gramoli, E. Jiménez, A.-M. Kermarrec, and M. Raynal. Distributed slicing in dynamic systems. In *ICDCS*, 2007.
23. V. Gramoli, Y. Vigfusson, K. Birman, A.-M. Kermarrec, and R. van Renesse. Slicing distributed systems. *IEEE Trans. Computers*, 58(11):1444–1455, 2009.
24. I. Gupta, M. Nagda, and C.F. Devaraj. The design of novel distributed protocols from differential equations. *Distributed Computing*, 20(2):95–114, 2007.
25. M. W. Hirsch, S. Smale, and R. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press, 2003.
26. J. Hofbauer and K. Sigmund. Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 4:479–519, 2003.
27. P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *WDTN*, pages 244–251, 2005.
28. M. Jelasity and A.-M. Kermarrec. Ordered slicing of very large-scale overlay networks. In *P2P*, pages 117–124, 2006.
29. P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Shiuan Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS*, pages 96–107, 2002.
30. O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Mediated population protocols. *Theor. Comput. Sci.*, 412(22):2434–2450, 2011.
31. J. D. Murray. *Mathematical Biology. I: An Introduction*. Springer, 3rd ed., 2002.
32. Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. In *WTDN*, pages 229–236, 2005.