Fast and Scalable Global Convergence in Single-Optimum Decentralized Coordination Problems

Luis R. Izquierdo, Segismundo S. Izquierdo, and Javier Rodríguez.

***Abstract*—Over the past few years, the scientific community has been studying the usefulness of evolutionary game theory to solve distributed control problems. In this paper we analyze a simple version of the *Best Experienced Payoff* (BEP) algorithm, a revision protocol recently proposed in the evolutionary game theory literature. This revision protocol is simple, completely decentralized and has minimum information requirements. Here we prove that adding some noise to this protocol can lead to efficient results in single-optimum coordination problems in little time, even in large populations of agents. We also test the algorithm under a wide range of different conditions using computer simulation. In particular, we consider different numbers of agents and of strategies, and we analyze the robustness of the algorithm to different updating schemes (e.g. synchronous vs asynchronous) and to different types of interaction networks (e.g. ring, preferential attachment, small world and complete). In all cases, using the noisy version of BEP, the agents quickly approach a small neighborhood of the optimal state from every initial condition, and spend most of the time in that neighborhood.**

***Index Terms*—Best Experienced Payoff, Decentralized Algorithms, Distributed Control, Evolutionary Dynamics, Evolutionary Game Theory, Large Population Double Limit, Small Noise Limit.**

I. Introduction

NOWADAYS there are many engineering systems that are difficult to control due to the large number of components that constitute them, the non-linear interdependencies that exist between them, and their distributed autonomy [1]. Examples include communication networks, transportation systems, wind farms, electrical networks, teams of autonomous vehicles, wireless sensor networks, and urban drainage systems.

The control of these large-scale distributed systems requires the design of individual decision rules (i.e. one for each of the system components) that guarantee the achievement of a common goal in a dynamic and highly uncertain environment. In this context, traditional control theory is usually not particularly useful, since in this type of distributed architecture there is no central entity with access and authority over all components of the system [2]. In fact, communication between components is often limited (e.g. due to economic and/or technological issues) and sometimes even simply unfeasible due to design requirements (e.g. due to privacy and/or stealth issues – as in military operations).

Over the past few years, the scientific community has realized the usefulness of evolutionary game theory to solve this kind of distributed control problems [3]. Evolutionary Game Theory (EGT) studies the interactions between autonomous agents who have only partial information about their environment and occasionally revise their strategies with the aim of improving their payoff. The two main components of EGT models are a population game and a revision protocol. The *population game* describes the payoffs that agents receive, given their individual strategy and the other agents' strategies. The *revision protocol* dictates when and how agents revise their strategies. A population game and a revision protocol together define an *evolutionary game dynamic*, which is a description of how the distribution of strategies in the population evolves over time [4].

Thus, the application of EGT to distributed control problems basically consists in finding a population game and a revision protocol such that the induced dynamics lead to the achievement of the overall objective pursued at the system level; all this considering the fact that individual agents may not have access to all the information needed to know the state of the system.

Following this approach, several promising results have been obtained in recent years. One of the most remarkable achievements has been the development of algorithms that allow to formalize numerous problems that often appear in engineering (e.g. dynamic resource allocation problems and

We are very grateful to four anonymous reviewers, whose comments greatly improved this manuscript. Financial support from the Spanish State Research Agency (PID2020-118906GB-I00/AEI/10.13039/501100011033), from the Regional Government of Castilla y León and the EU-FEDER program (CLU-2019-04), from the Spanish Ministry of Science, Innovation and Universities, and from the Fulbright Program (PRX19/00113, PRX21/00295), is gratefully acknowledged.

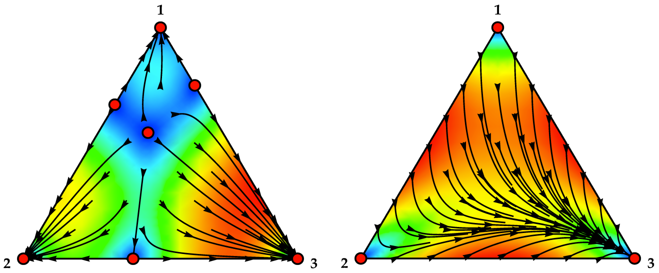
L.R. Izquierdo is with the Department of Management Engineering at Universidad de Burgos, Spain (e-mail: lrizquierdo@ubu.es).
S. S. Izquierdo is with the BioEcoUva Research Institute on Bioeconomy and with the Department of Industrial Organization at Universidad de Valladolid, Spain (e-mail: segis@eii.uva.es).
J. Rodríguez is with Telefonica I+D in Spain (e-mail: javiroma@gmail.com).

routing problems) as potential games [5]–[7],[1] since there are numerous revision protocols that ensure convergence to a Nash equilibrium in this type of games [4], [8]. However, even though this line of work has already produced several important results, it currently presents three notable limitations that are indicated below.

The first limitation is that the Nash equilibria achieved may be highly inefficient from the point of view of the overall objective to be achieved at the system level [9]. The second limitation is that most of the results are valid only for potential games, but there are numerous problems in engineering that cannot be formalized as a potential game [10]. An example with these characteristics is the distributed control of wind farms with the objective of maximizing total energy production. Currently we lack precise engineering knowledge of the aerodynamic interactions that occur between the turbines, so it is not possible to know the effect of modifying the variables of a single turbine on the total energy production [11]. And the third limitation is that the most promising algorithms proposed up to date (e.g. [10]) scale very badly with the number of components, requiring extremely long times to reach their asymptotic behavior even for moderate numbers of components.

Here we study the *Best Experienced Payoff* (BEP) protocol [12], [13], which presents three characteristics that address each of the limitations outlined above: it converges to very efficient states (not necessarily Nash equilibria) in several games that have inefficient Nash equilibria, it has minimum information requirements (agents only need access to the action they played and the payoff they received),[2] and its speed of convergence is very high. Its main drawback is the limited scope of games for which convergence to the optimum is guaranteed (as opposed to e.g. the general algorithm proposed by Marden et al. [10]).[3] Specifically, this paper shows that the BEP protocol with some noise can be used as a scalable decentralized algorithm to quickly reach the optimal outcome in a large class of coordination problems.

The paper is structured as follows. Section II presents a motivating example. In Section III, we specify the problem we are dealing with and define *single-optimum coordination* (SOC) games. In Section IV we present the general BEP algorithm and propose a noisy version called nBEPA1. Section V is devoted to the formal analysis of the nBEPA1 dynamics in SOC games. It includes several propositions that together characterize the transient and asymptotic dynamics of the nBEPA1 protocol in SOC games. In Section VI we present various simulation experiments aimed at exploring the dependence of the nBEPA1 dynamics on the number of agents, the number of strategies, the level of noise, and the way agents are scheduled to revise their strategies. We also study the nBEPA1 dynamics on networks, where agents can interact only with a small subset of the population. Lastly, we summarize our main conclusions in Section VII. All figures and simulation experiments in this paper can be replicated using open-source software that can be downloaded using the links provided in Appendix I.



(a) Replicator Dynamics (b) BEP mean dynamics

Fig. 1: Phase portraits of the coordination game with payoff matrix (1), with $n = 3$ strategies. The simplex on the left (a) shows the replicator dynamics, and the simplex on the right (b) shows the BEP mean dynamics (solving ties at random).

II. A MOTIVATING EXAMPLE

As an illustration, consider the following problem: A set of drones sent to a hostile environment must choose a channel to communicate from among n possible ones. Two drones can communicate if and only if they both use the same channel, and there is no risk of any channel becoming saturated by the drones. However, the efficiency of each channel is different, it is not possible to know which channel is optimal a priori, and it is possible that the optimal channel is different at different times (i.e. stochastic game). In this setting, the best outcome would be one where, at any moment, every drone is using the optimal channel. This situation could be modeled as a symmetric two-player (single-optimum coordination) game with the following payoff matrix:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & & n-1 & 0 \\ 0 & 0 & \dots & 0 & n \end{pmatrix} \tag{1}$$

This game has n pure Nash equilibria (and several mixed ones), but only one is optimal (i.e. both players choose channel n). Fig. 1 shows the phase map of this game with $n = 3$ strategies, placed in a population context, a) under the replicator dynamics (Figure 1a) and b) the mean dynamics when agents use the BEP revision protocol (Figure 1b).

In this problem, most of the dynamics studied in the EGT literature (e.g. *Replicator*, *Smith*, *Brown-von Neumann-Nash*, *best-response*, and *imitate-the-better-realization* [18], [19]) [4] and most algorithms used for coordination problems in the

[1] Potential games are games that admit a potential function, which is a scalar function that contains all the payoff information that is relevant to study the game. In particular, in potential games, the change in any player's payoff from a unilateral deviation equals the change in potential. This means that profitable strategy revisions increase the value of the potential function [4, chapter 3].

[2] This means that the algorithm is payoff-based [10], completely uncoupled [14], [15], radically uncoupled [16] and local [17].

[3] Notably, the notion of "convergence" in our paper is different from [10]. We focus on the *large population double limit* – where convergence is guaranteed in finite time – while results in [10] refer to the *small noise limit*, where convergence time is not bounded. This is explained in detail in Section V.

Multi-Agent Systems literature (e.g. HCR [17], EM [17], WSLpS [20] and *Majority Action* [21]) converge to one or another of the n pure Nash equilibria, depending on initial conditions, with all Nash equilibria having a sizable basin of attraction (as in Figure 1a).[4] Consequently, using these dynamics, there is a high probability of ending up in an inefficient state, especially if the optimal channel changes in time.

However, if agents use the BEP protocol – breaking ties at random –, the agents will be able to coordinate on the optimal communication channel starting from nearly all initial conditions. In fact, coordination of the entire population on the optimal channel is an almost globally asymptotically stable state of the BEP mean dynamics [13, prop. 5.11], which is reached from every state except for the $(n-1)$ inefficient pure states where the whole population is coordinated on a suboptimal channel. This suggests that, if there is some small variability or *noise* in the decision protocol (or we add it by design), agents will be able to quickly coordinate on the optimal channel from any initial condition, even if the optimal channel changes at some point.

III. The population game

We assume that there is a population of N agents that may engage in a 2-player symmetric game $G^{PC} = \{S, A\}$, where $S = \{1, ..., n\}$ is the set of pure strategies and A is the payoff matrix, with elements a_{ij} satisfying the following payoff conditions:

Payoff Conditions 3.1: There is a strategy $s \in S$ such that the following two conditions hold:

- $a_{ss} > \max_{i \neq s} a_{ij} \quad \forall j \in S.$
- $a_{sj} \geq \max_{i \neq s} \min(a_{ij}, a_{is}) \quad \forall j \neq s.$ □

We use superscript ^{PC} in G^{PC} to emphasize that G^{PC} is a game that satisfies Payoff Conditions 3.1. The first condition states that there is an optimal strategy s in the sense that payoff a_{ss} is greater than any payoff that can be obtained with any of the other strategies. In particular, the first condition implies that, in a population context, the optimal monomorphic state is the one where every agent is choosing strategy s. The second condition is satisfied if strategy s is weakly dominant in an auxiliary game in which, for each of the other strategies $i \neq s$, payoff a_{ij} is replaced with $\min(a_{ij}, a_{is})$.

A particularly relevant family of games that satisfy both conditions is the set of *Single-Optimum Coordination* (SOC) games. We define SOC games as 2-player symmetric games where there is a unique maximum payoff a_{ss} that is obtained if both players choose the same strategy s, and players using different strategies obtain the same payoff $b < a_{ss}$.

$$\begin{pmatrix} a_{11} & b & b & \dots & b \\ b & a_{22} & b & \dots & b \\ b & b & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b \\ b & b & \dots & b & a_{nn} \end{pmatrix}$$

The game with payoff matrix (1) is a SOC game. In the Economics literature, SOC games have been used extensively to study the evolution of social norms (see e.g. [23]–[26]) and the diffusion of technological innovations (see e.g. [22], [27], [28]). In such settings, agents must choose whether to keep using a status-quo norm or technology, or adopt a superior innovation. SOC games also appear naturally in engineering contexts where a set of devices are required to coordinate on an optimal action, such as swarm robotics and wireless sensor networks (see e.g. [20]).

IV. The Best Experienced Payoff protocol

The *Best Experienced Payoff* (BEP) protocol is based on the so-called "procedurally rational agents", initially proposed by Osborne and Rubinstein [29] and analyzed for the first time in an evolutionary context by Sethi [30].

In accordance with the EGT literature, here we assume that there is a population of N agents who can revise their strategy occasionally and independently, one agent at a time. We define one unit of time as the lapse of time over which each agent expects to receive one revision opportunity. Thus, the expected number of revisions per time unit in the population is N.

Under the general BEP revision protocol, the revising agent tests a subset of its available strategies by trying out each of them a predetermined number of times [12], [13]. In this paper we use the simplest version of the BEP protocol, i.e., the version where revising agents consider all their strategies and they try each of them only once. Crucially, every time the revising agent tries one strategy, she draws a new random agent to play with, following a uniform distribution; thus, each strategy trial is conducted with a potentially different co-player, and every agent is equally likely to be selected. Once each of the candidate strategies has been tried against one opponent, the revising agent chooses the strategy that provided the greatest payoff in the test, resolving the possible ties at random. The version of BEP that we use in this paper – where revising agents test all their strategies, each against one new random agent, and break ties uniformly at random – is henceforth called BEPA1.[5,6]

In some games G^{PC} that we study in this paper, the BEPA1 algorithm has several absorbing states that are inefficient. An example would be the state where the whole population is using strategy 1 (or strategy 2) in the SOC game with payoff matrix (1) and $n = 3$. This is undesirable, especially in stochastic games, because the population will be unable to adapt to changing conditions, such as a shift in the optimal channel in SOC game (1).

Nonetheless, this drawback can be easily overcome by adding some probability of experimentation (or noise) to the algorithm so that, with a small probability ϵ, revising agents adopt any of the n possible strategies with equal probability – and with probability $(1-\epsilon)$ they select the new strategy following the BEPA1 protocol. This noisy generalization of

[4] Noisy dynamics with sufficiently large levels of noise will have only one global attractor, but this global attractor may well be far from the optimal outcome (e.g. see [4, Example 6.2.2, pp. 191-3] and [22] for logit dynamics).

[5] Recent papers that analyze the BEP dynamics in other contexts include [31] in repeated games, [32] in the Prisoner's Dilemma with several trials, and [33] in the Centipede game testing two strategies only.

[6] In general, the results in this paper do not apply to versions of BEP where not all the strategies are tested [13].

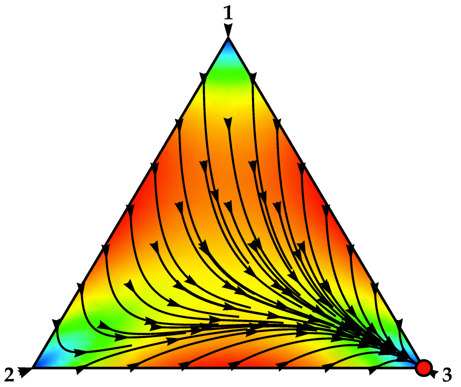


Fig. 2: Phase portrait showing the mean dynamics of the nBEPA1$^{\epsilon=10^{-4}}$ in the single-optimum coordination game with payoff matrix (1), with $n = 3$ strategies.

BEPA1 is called nBEPA1, and it is the main object of study in this paper.

Fig. 2 shows the mean dynamic of the nBEPA1$^{\epsilon=10^{-4}}$ protocol in the SOC game with payoff matrix (1), with $n = 3$ strategies. The introduction of noise implies that the algorithm has no absorbing states anymore and its mean dynamic has a unique globally asymptotically stable state where most of the population – i.e. more than 99.9% of the agents – are coordinated on the optimal strategy. This means that if the optimal channel changes, the population will quickly coordinate on the new optimal strategy.

V. Analytical results

A. Notation

A population state x is characterized by the fraction x_i of agents using strategy $i \in S$. Thus, the set of population states is the simplex in $\mathbf{R}^n$, i.e. $\Delta S = \{x \in \mathbf{R}^n_+ : \sum_{i\in S} x_i = 1\}$. We use e_i to denote the monomorphic (or pure) state where all agents use strategy i, i.e., the state x such that $x_i = 1$ and $x_j = 0$ for all $j \neq i$. Recall that in every game G^{PC} (i.e. 2-player symmetric game satisfying Payoff Conditions 3.1) played in a population context, there is a unique optimal pure state e_s where every agent is choosing strategy s.

B. The nBEPA1 dynamics as a Markov chain

Defining the population state x by the fraction of agents that are using each strategy, the dynamics induced by the nBEPA1 protocol can be usefully seen as a Markov chain $\{X^{N,\epsilon}_t\}$ whose finite set of states is the grid $\Delta^N S = \{x \in \Delta S : Nx \in \mathbf{Z}^n\}$.

In the following sections we study the transient and the asymptotic behavior of the Markov chain $\{X^{N,\epsilon}_t\}$ induced by the nBEPA1 protocol. Following [34], we assume that at discrete times $t = 0, \frac{1}{N}, \frac{2}{N}, \ldots$ exactly one individual chosen at random is given the opportunity to change strategy, but this assumption may be relaxed.[7] Thus, each individual agent is expected to revise its strategy exactly once over one unit of time, which we call a tick.

C. Transient behavior: the mean dynamic

In this section we derive and analyze the mean dynamic of the nBEPA1 stochastic process, which is a set of differential equations that approximate the transient dynamics of the Markov chain remarkably well, especially for finite time horizons and large populations [4, chapter 10].

To derive the mean dynamic, we need to introduce some notation first. Assume for now that there is no noise. Let a battery of tests conducted by a revising agent be the process of testing each of her available strategies and assigning to each strategy the corresponding experienced payoff. (Recall that each strategy is tested with a new randomly drawn coplayer.)

Let Φ_i be the probability with which strategy i is selected in a battery of tests. Strategy i is selected if it is the only strategy that obtains the greatest experienced payoff in the battery of tests or, if there are more strategies with the same greatest experienced payoff, if it is selected (uniformly at random) among this set of best-performing strategies. If, when tested, strategy i meets an agent using strategy j, strategy i obtains payoff a_{ij}. The conditional probability that payoff π_k obtained by strategy $k \neq i$ is lower than payoff a_{ij} (obtained by strategy i) is $P(\pi_k < a_{ij}) = \sum_{m:a_{km}<a_{ij}} x_m$. And the conditional probability that strategy $k \neq i$ obtains the same payoff as strategy i is $P(\pi_k = a_{ij}) = \sum_{m:a_{km}=a_{ij}} x_m$.

For each payoff a_{ij} that strategy i may obtain, let $\Theta_i(a_{ij})$ be the set of strategies other than i that may obtain the same payoff a_{ij}, i.e.:

$\Theta_i(a_{ij}) = \{k \in (S \setminus i) : a_{km} = a_{ij} \text{ for some } m \in S\}$.

Let $\mathcal{P}(\Theta_i(a_{ij}))$ be the power set of $\Theta_i(a_{ij})$, i.e., the set of subsets of strategies other than i that may obtain the same payoff a_{ij} as strategy i (when it meets a j-strategist), including the empty set. The probability Φ_i is then:

$$\Phi_i = \sum_j x_j \left[\sum_{\theta \in \mathcal{P}(\Theta_i(a_{ij}))} \frac{1}{\#\theta + 1} \left(\prod_{k\in\theta} P(\pi_k = a_{ij}) \right) \left(\prod_{k\notin\theta, k\neq i} P(\pi_k < a_{ij}) \right) \right] \tag{2}$$

where $\#\theta$ is the cardinality of subset θ. The term x_j in (2) is the probability that strategy i obtains payoff a_{ij}. The term in square brackets is the conditional probability that each of the other strategies obtains either a lower payoff, or the same payoff as strategy i and strategy i is the one selected from the set of best-performing strategies.

Introducing now noise in the process, the nBEPA1 mean dynamic equations can be expressed as:

$$\dot{x}_i = (1-\epsilon)\Phi_i + \frac{\epsilon}{n} - x_i \tag{3}$$

The outflow (negative) term $-x_i$ in (3) corresponds to agents who are currently using strategy i (whose proportion is x_i) and revise their strategy, potentially adopting another strategy. The inflow (positive) terms correspond to the revising agents who adopt strategy i. Specifically, the inflow is the probability that noise plays no role in the revision $(1-\epsilon)$ multiplied by the probability with which the revising agent

[7] Alternatively, we can assume that each agent has a rate 1 "Poisson clock" that sets her revision times, with all clocks being statistically independent.

adopts strategy i in the absence of noise Φ_i, plus the probability with which strategy i is selected under uniform random noise $(\frac{\epsilon}{n})$.

As an example, the inflow of the nBEPA1 stochastic process without noise in SOC games with payoff matrix (1) reads:

$$\Phi_i = x_i \prod_{j=i+1}^{n} (1-x_j) + \frac{1}{n}\prod_{j=1}^{n}(1-x_j)$$

The first inflow term $x_i \prod_{j=i+1}^{n}(1-x_j)$ corresponds to the probability that a revising agent, when testing strategy i, meets another agent using strategy i too (obtaining payoff i), and when testing any strategy $j > i$, it meets an agent using any strategy other than j (obtaining a payoff of 0). The second inflow term $\frac{1}{n}\prod_{j=1}^{n}(1-x_j)$ corresponds to the probability that a revising agent, when testing each strategy j, meets an agent using any strategy other than j, so all strategies obtain the same payoff (0) in the test, and are chosen with probability $\frac{1}{n}$.

The following proposition states that, in any SOC game, nearly all trajectories of the BEPA1 mean dynamics converge to the optimal state e_s, i.e., the state where $x_s = 1$.

Proposition 5.1: The optimal state e_s is an almost global attractor of the BEPA1 mean dynamics ((3) with $\epsilon = 0$) in any SOC game. This state e_s attracts all trajectories except possibly those starting at the other monomorphic states in which all players use the same strategy.

Proof: In any SOC game, we have:

$$\dot{x}_s \geq x_s + \frac{1}{n}\prod_{i=1}^{n}(1-x_i) - x_s = \frac{1}{n}\prod_{i=1}^{n}(1-x_i)$$

The inflow term x_s corresponds to the probability that a revising agent, when testing strategy s, meets another agent using strategy s too, obtaining the highest possible payoff. The inflow term $\frac{1}{n}\prod_{i=1}^{n}(1-x_i)$ corresponds to the probability that a revising agent, when testing each strategy i, meets an agent using any other strategy $j \neq i$, so all strategies obtain the same payoff (b) and are chosen with probability $\frac{1}{n}$. For SOC games with $a_{ii} < b$ for some i, the inflow would include more terms. The growth rate for x_s is then strictly positive except possibly at states e_i where $x_i = 1$, proving the result. ■

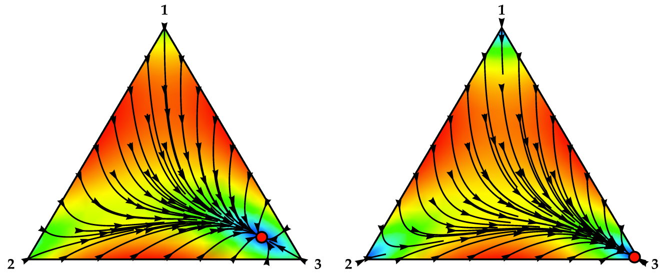
Proposition 5.2 below shows that, if the noise level is low, all trajectories of the nBEPA1 mean dynamic (3) in any game G^{PC} converge to a small neighborhood around the optimal pure state e_s, i.e., there is a small neighborhood around e_s that is globally asymptotically stable. Fig. 3 illustrates this result in the SOC game with payoff matrix (1), with $n = 3$ strategies.

Proposition 5.2: For any positive $\delta < 1$, there is a threshold noise level $\epsilon_\delta > 0$ such that, for all positive noise levels $\epsilon < \epsilon_\delta$, all trajectories of the dynamics (3) in any game G^{PC} converge to the set $O_\delta(e_s) \equiv \{x \in \Delta S : x_s \geq 1 - \delta\}$.

Proof: We will show that the function $L_\delta \colon \Delta S \to [0, 1-\delta]$ defined by $L_\delta(x) = \max(0, 1-\delta-x_s)$ is a strict Lyapunov function for the set $O_\delta(e_s)$, proving the result. It is easy to check that $L_\delta^{-1}(0) = O_\delta(e_s)$, and, for $x_s < 1-\delta$, $\dot{L}_\delta(x) = -\dot{x}_s$. If $x_s < 1-\delta$, then there is some strategy $j \neq s$ with $x_j > \frac{\delta}{n}$. This implies that $\Phi_s \geq x_s + \frac{1}{n}\frac{\delta}{n}(\frac{\delta}{n}x_s)^{n-1}$, where the term x_s is the probability that a revising agent, when testing strategy s, meets an s-strategist (so it obtains payoff a_{ss} in that test, and strategy s is selected); the last term is a lower bound on the inflow from revising agents who, when testing strategy s meet a j-strategist (obtaining the payoff a_{sj} in the test of strategy s, which happens with a probability of at least $\frac{\delta}{n}$) and when testing each of the other $n-1$ strategies (generically, strategy $i \neq s$) do not obtain a greater payoff because they meet either an s-strategist or a j-strategist (whichever provides the lower payoff $\min(a_{ij}, a_{is})$), considering that $\forall i \neq s, a_{sj} \geq \min(a_{ij}, a_{is})$ and $\min(\frac{\delta}{n}, x_s) \geq \frac{\delta}{n}x_s$. For dynamics (3) we then have

$$\dot{x}_s \geq (1-\epsilon)\frac{1}{n}\frac{\delta}{n}\left(\frac{\delta}{n}x_s\right)^{n-1} + \epsilon\left(\frac{1}{n} - x_s\right) \equiv \lambda. \tag{4}$$

If $x_s < \frac{1}{n}$, then the last term in the lower bound λ in (4) is positive and, given that the first term is non-negative, we have $\dot{x}_s > 0$ and, if $x_s < 1-\delta$, then $\dot{L}_\delta(x) < 0$. If $\frac{1}{n} \leq x_s < 1-\delta$, then $\lim_{\epsilon\to 0}\lambda = \frac{1}{n}\frac{\delta}{n}(\frac{\delta}{n}x_s)^{n-1} > 0$, which means that there is an $\epsilon_\delta > 0$ such that for $\epsilon < \epsilon_\delta$, we have $\lambda > 0$, $\dot{x}_s > 0$ and $\dot{L}_\delta(x) < 0$, completing the proof. ■



(a) $\epsilon = 10^{-1}$ (b) $\epsilon = 10^{-2}$

Fig. 3: Phase portraits of the SOC game with payoff matrix (1), with $n = 3$ strategies. The simplex on the left (a) shows the mean dynamics of the nBEPA1$^{\epsilon=10^{-1}}$ protocol and the simplex on the right (b) shows the mean dynamics of the nBEPA1$^{\epsilon=10^{-2}}$ protocol.

Going back to the actual stochastic process $\{X_t^{N,\epsilon}\}$ induced by the nBEPA1 protocol – and considering the relationship between the stochastic process and its mean dynamic in large populations [34] –, we have shown that, for large enough population sizes and moderate noise, in any game G^{PC}, the stochastic process $\{X_t^{N,\epsilon}\}$ – starting from any initial state – will approach a small neighborhood of the optimal pure state e_s with probability close to one, and we can make this neighborhood as small as we like by making the noise ($\epsilon > 0$) sufficiently small.

This is an important difference with the (many) dynamics where several pure states in SOC games can have a sizable basin of attraction (see Section II). In such dynamics, even if some (sufficiently low) noise is added, there will still be different attractors with sizable basins of attraction; i.e., in SOC games, those dynamics will not converge to the vicinity of the optimal state from every initial condition. If noise is increased in those dynamics until there is one single global attractor, this global attractor may well be far away from the optimal state. Exactly how far the global attractor will be from the optimal state will generally depend on the specific dynamic employed, on the magnitude of the payoffs and, naturally, on the level of noise (e.g. see [4, Example 6.2.2, pp. 191-4] for logit dynamics). This can be easily checked using EvoDyn-3s software [35].

D. Asymptotic dynamics: the limiting distribution

Having seen that the stochastic process $\{X_t^{N,\epsilon}\}$ induced by the nBEPA1 protocol in any game G^{PC} – with N sufficiently large and ϵ adequately small – will approach the surroundings of the optimal pure state e_s from any initial condition, we turn our attention to the question of whether the process will stay in that area for long. To answer this question, we must study the limiting distribution $\mu^{N,\epsilon}$ of $\{X_t^{N,\epsilon}\}$, when it exists.

In general, the stochastic process induced by nBEPA1 in the absence of noise (i.e. BEPA1) may have several absorbing states, so the asymptotic dynamics of this process may well depend on initial conditions. An example of this situation is the SOC game with payoff matrix (1): if the process starts at any of the n pure states, it will stay there forever.

In contrast with the protocol without noise, the nBEPA1 protocol with $\epsilon > 0$ has full support, so the Markov chain it induces is irreducible and aperiodic [4, section 11.1.1]. This means that there is a unique stationary distribution $\mu^{N,\epsilon}$, which describes the infinite horizon behavior of $\{X_t^{N,\epsilon}\}$ regardless of initial conditions (i.e. $\mu^{N,\epsilon}$ is the limiting distribution) and it also represents the long-run fraction of time that the process spends in each state (i.e. $\mu^{N,\epsilon}$ is the occupancy distribution, or the limiting empirical distribution). This distribution $\mu^{N,\epsilon}$ has full support, i.e. fixing N and ϵ, every state is visited infinitely often, but the probability mass is often concentrated on a small set of states. Here we show that, for sufficiently low levels of noise ϵ, as the population size N grows, the limiting distribution $\mu^{N,\epsilon}$ in any game G^{PC} concentrates all its probability mass on a small neighborhood around the optimal pure state e_s.

Proposition 5.3: In any game G^{PC} played in a population context, for any positive $\delta < 1$ there is a threshold noise level $\epsilon_\delta > 0$ such that for all positive noise levels $\epsilon < \epsilon_\delta$, the probability of the set $O_\delta(e_s) \equiv \{x \in \Delta S : x_s \geq 1-\delta\}$ under the stationary distribution $\mu^{N,\epsilon}$ of the nBEPA1 process $\{X_t^N\}$ tends to 1 as the population size N grows, i.e., $\lim_{N\to\infty}\mu^{N,\epsilon}(O_\delta(e_s)) = 1$.

Proof: The proof is based on [34, Prop. 4], which shows that, for large populations, a Markov chain satisfying some conditions almost surely spends almost all time, in the long run, at the Birkhoff center of the flow. Benaïm and Weibull [36, Remark 2] show that this result holds under more general assumptions than those considered in [34], including our framework. The nBEPA1 protocol defines a Markov chain $\{X_t^{N,\epsilon}\}$ whose finite set of states is the grid $\Delta^N S = \{x \in \Delta S : Nx \in \mathbf{Z}^n\}$. For $\epsilon > 0$, this Markov chain is irreducible and aperiodic, and presents a unique stationary distribution $\mu^{N,\epsilon}$. Let $F^N(x)$ be the expected increment in the population state between two consecutive revisions, times the population size N, when the process is at state x. Considering the transition probabilities associated to the revision process (see Section V-C), we have $F_i^N(x) = (1-\epsilon)\Phi_i^N + \frac{\epsilon}{n} - x_i$, where, in order to calculate Φ_i^N, the formula for Φ_i in (2) needs to be adjusted. Specifically, for a revising agent using strategy i in a population of size N, the probability of meeting a co-player using the same strategy i is $\frac{Nx_i-1}{N-1}$ (instead of x_i), and the probability of meeting a co-player using strategy $j \neq i$ is $\frac{Nx_j}{N-1}$ (instead of x_j). It is then easy to check that the sequence of functions $\{F^N\}$ converges uniformly to the vector field F presented in (3), i.e., $\lim_{N\to\infty}\max_{x\in X^N}|F^N(x) - F(x)| = 0$ (see [18] for a similar case). On the other hand, from Proposition 5.2 we know that for any positive $\delta < 1$ there is a threshold noise level $\epsilon_0 > 0$ such that for all positive noise levels $\epsilon < \epsilon_0$ the function $L_{\delta/2}$ is a strict Lyapunov function under (3) for the set $O_{\delta/2}(e_s) \equiv \{x \in \Delta S : x_s \geq 1-\frac{\delta}{2}\}$. This implies that for $\epsilon < \epsilon_0$ the (relatively) open set $O'_\delta(e_s) \equiv \{x \in \Delta S :\ x_s > 1-\delta\}$ contains the closure of the set of recurrent points of the noisy mean dynamic (3), so it contains the Birkhoff center [34] of (3). The results in [36, Remark 2] and [34, Prop. 4] then imply that $\lim_{N\to\infty}\mu^{N,\epsilon}(O_\delta(e_s)) = 1$. ■

Proposition 5.3 implies that, in any game G^{PC} played in a population context, the optimal pure state e_s is *uniquely stochastically stable in the large population double limit* [4, p. 458] under the nBEPA1 protocol, i.e., for any (relatively) open set O containing e_s:

$$\lim_{\epsilon\to 0}\lim_{N\to\infty}\mu^{N,\epsilon}(O) = 1$$

E. The small noise limit

Interestingly, note that Proposition 5.3 does not imply anything about the *small noise limit* $\lim_{\epsilon\to 0}\mu^{N,\epsilon}$ or about the *small noise double limit* $\lim_{N\to\infty}\lim_{\epsilon\to 0}\mu^{N,\epsilon}$, which do not generally agree with the large population double limit in games G^{PC}. To study the small noise limit, it is useful to start characterizing the set of absorbing states in the process without noise, i.e., the BEPA1 stochastic process.

Proposition 5.4: In any 2-player symmetric game played in a population context with more than 2 agents ($N > 2$), a state of the Markov chain $\{X_t^{N,0}\}$ induced by the BEPA1 protocol (i.e. the nBEPA1$^{\epsilon=0}$ protocol) is absorbing if and only if all agents are playing the same pure strategy $i \in S$ and strategy profile (i, i) is a strict Nash equilibrium.

Proof: Let BR(i) be the set of strategies that provide the maximum possible payoff when meeting an i-strategist, i.e., the set of pure best replies to i. At any given population state, consider an agent using strategy i and let S_{-i} be the set of strategies played by the other players in the population.

Under the BEPA1 protocol, a revising agent using strategy i may change its strategy unless $\min_{j\in S_{-i}} a_{ij} > \max_{k\neq i, j\in S_{-i}} a_{kj}$. This condition implies that, for every $j \in S_{-i}$, BR$(j) = \{i\}$, which is consequently a necessary condition for every strategy i in the support of an absorbing state: any strategy i in the support has to be the unique best reply to each of the other strategies in the support, and also to itself if there is more than one player using strategy i.

An absorbing state in a population with more than two agents must be monomorphic, i.e., all agents must play the same strategy i, because:

- An absorbing state cannot include more than two different strategies in its support. Suppose there are three different strategies, namely $i \neq j \neq k$, being played at an absorbing state. This implies $\mathrm{BR}(k) = \{i\}$ and $\mathrm{BR}(k) = \{j\}$, which is a contradiction.
- An absorbing state in a population with more than two agents cannot have exactly two different strategies in its support. Suppose that there is an absorbing state at which two or more players use strategy i and one or more players use strategy $j \neq i$. This implies that $\mathrm{BR}(i) = \{i\}$ and $\mathrm{BR}(i) = \{j\}$, which is a contradiction.

Consequently, at an absorbing state in a population with more than two agents, all agents must play the same strategy i such that $\mathrm{BR}(i) = \{i\}$, i.e., such that the strategy profile (i, i) is a strict Nash equilibrium of the two-player stage game. It is easy to check that states satisfying this necessary condition are indeed absorbing, so the condition is both necessary and sufficient. ■

Proposition 5.4, which is valid for any 2-player symmetric game, establishes an equivalence between absorbing states of the BEPA1 dynamics and strict Nash equilibria. Proposition 5.5 below shows that, in SOC games, all absorbing states of the process without noise retain positive mass in the small noise limit.

Proposition 5.5: In any SOC game played in a population context, the set of *stochastically stable states in the small noise limit* of the process nBEPA1 is the set of pure states e_i corresponding to strategies i such that (i, i) is a strict Nash equilibrium, i.e., $\lim_{\epsilon \to 0} \mu^{N,\epsilon}(x) > 0$ if and only if x is a pure state e_i such that strategy profile (i, i) is a strict Nash equilibrium.

Proof: This proposition can be proved using well-known results derived by Young [37, Appendix]. Here we use a more recent and compact version of these results provided by Sandholm [4, Theorem 12.A.5]. Let $q \geq 1$ be the number of pure states e_i in the SOC game such that strategy profile (i, i) is a strict Nash equilibrium. Let us call these states *strict Nash states*. First, note that in the process without noise nBEPA1$^{\epsilon=0}$, the q strict Nash states are the only absorbing states. This can be proved using the same arguments as in the proof of Proposition 5.4, noting that here we do not need the population size N to be greater than 2 because, even if $N = 2$, in SOC games an absorbing state cannot have exactly two different strategies in its support. This is so because in SOC games there cannot be two different strategies $i \neq j$ such that $\mathrm{BR}(i) = \{j\}$ and $\mathrm{BR}(j) = \{i\}$. Secondly, note that in SOC games it is possible to reach at least one of the q strict Nash (absorbing) states from any state, so there are no other closed communicating classes. Therefore, these q states are the only recurrent classes of nBEPA1$^{\epsilon=0}$, so it is sufficient to look at the cost of moving between them. Now consider the process with $\epsilon > 0$. The cost of a transition from state a to state b is the minimum number of experimentations (or noise events) needed to reach state b from state a (see a formal definition in [4, p. 522]). Note that the cost of moving from any strict Nash state to any other strict Nash state is exactly 1, since (a) we need one experimentation to abandon the strict Nash state at the origin, and (b) for every strategy i such that (i, i) is a strict Nash equilibrium, it is possible to go from any state with at least one agent using i to the strict Nash state where every agent is using i via a path of cost equal to 0.[8] Therefore, all strict Nash states have a tree rooted on them with the minimum cost $(q - 1)$. By [4, Theorem 12.A.5], this implies that the set of *stochastically stable states in the small noise limit* is the set of strict Nash states. ■

We include Proposition 5.5 here to provide a complete picture of the nBEPA1 dynamics in SOC games but, in our opinion, the relevance of the small noise limit for engineering problems is generally low. The small noise limit is relevant in situations in which the noise level is so small that an escape from a pure state by way of a single experimentation is even less likely than a whole journey from the vicinity of the optimal pure state e_s to an inefficient pure state – a journey against the flow of the mean dynamic.

Thus, the dynamics described by the small noise limit are dynamics where the system spends most of the time at the pure states, and only rarely there is an experimentation that may move the stochastic process from one pure state to another. This often requires infinitesimally small levels of noise, already for populations with more than a handful of agents, and – consequently – the waiting times needed to approach the limiting distribution are typically extremely long. Moreover, in the type of problem we are considering here (SOC games), even in the small noise limit, nearly all the mass is concentrated on the optimal state if the population has more than a dozen agents.

To illustrate this point, we focus on a 2-strategy SOC game (1), since in games with two strategies Sandholm [4] provides analytical formulas for both the stationary distribution $\mu^{N,\epsilon}$ [4, section 11.2.1] and the average hitting time of any state from any other state [4, Example 11.A.5].

Fig. 4 shows the fraction of time that the system spends at inefficient state e_1 for various population sizes N. For $N = 10$, the long-run fraction is already below 2%, and this value decreases exponentially as the size of the population increases. For $N = 50$ the fraction is 1.36×10^{-11}, and for $N = 100$ it is 6.62×10^{-23}. Fixing N, these values decrease even more or stabilize as the noise level decreases, so for lower noise levels we can only expect similar or even lower time fractions (see fig. 5).

To make matters worse for the relevance of the small noise limit, the average hitting time of the inefficient state (in ticks, or revisions per agent, from initial state $x = (0.9, 0.1)$)[9] is very high already for low population sizes, and it increases exponentially both as the population size increases (see fig. 4) and as the noise level decreases (see fig. 5). As an example, for $N = 50$, $\mu^{N=50,\epsilon=10^{-3}}(e_1) = 1.36 \times 10^{-11}$ and it takes on average 7.60×10^{12} revisions per agent to reach the inefficient state e_1 for the first time when departing from state

[8] This is so because it is possible that every agent who is not using strategy i revises its strategy and tests all strategies against an i-player, thus adopting strategy i.

[9] For $N < 10$, the initial state is $(\lfloor 0.9N \rfloor / N, 1 - \lfloor 0.9N \rfloor / N)$.

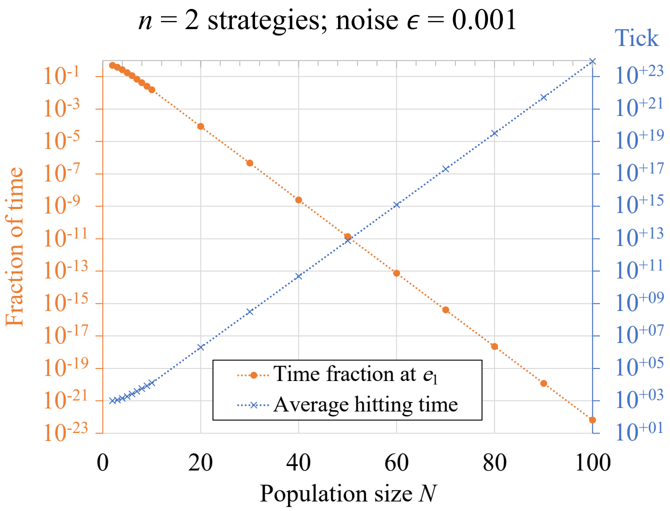


Fig. 4: Long-run fraction of time $\mu^{N,\epsilon=10^{-3}}(e_1)$ spent at inefficient state e_1 (in orange) and average hitting time of e_1 from state $x = (0.9, 0.1)$ (in blue), for the nBEPA1 protocol in the 2-strategy SOC game with payoff matrix (1), for various population sizes.

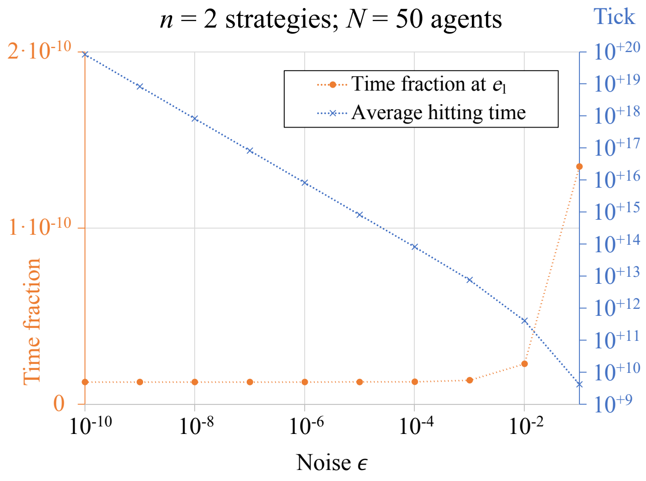


Fig. 5: Long-run fraction of time $\mu^{N=50,\epsilon}(e_1)$ spent at inefficient state e_1 (in orange) and average hitting time of e_1 from state $x = (0.9, 0.1)$ (in blue), for the nBEPA1 protocol in the 2-strategy SOC game with payoff matrix (1) played in a population with $N = 50$ agents, for various noise levels.

$x = (0.9, 0.1)$.

To sum up, the relevance of the small noise limit in our problem is very low.

F. How extreme do population sizes and noise levels need to be?

The analytical results presented above require low levels of noise and sufficiently large populations. To get an order of magnitude of how small the noise and how large the population must be for analytical results to be useful, we analyze SOC game (1) with $n = 2$ strategies and $\epsilon = 10^{-3}$.

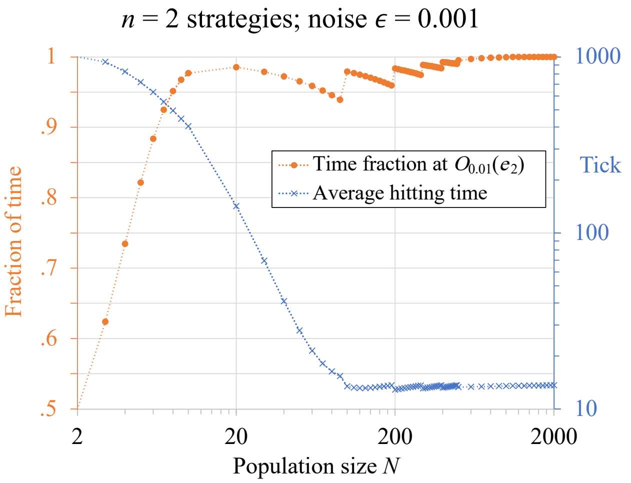


Fig. 6: Long-run fraction of time $\mu^{N,\epsilon=10^{-3}}(O_{0.01}(e_2))$ spent in the neighborhood $O_{0.01}(e_2)$ (in orange) and average hitting time of the same neighborhood from state $x = (0.9, 0.1)$ (in blue), for the nBEPA1$^{\epsilon=10^{-3}}$ protocol in the 2-strategy SOC game with payoff matrix (1), for various population sizes.

Fig. 6 shows the time fraction spent at the neighborhood $O_{0.01}(e_2)$ of the optimal state and the average hitting time from state $x = (0.9, 0.1)$ to the same neighborhood, for noise $\epsilon = 10^{-3}$ and different population sizes N.[10]

It is remarkable that for populations as small as $N = 10$ agents, the time fraction $\mu^{N=10,\epsilon=10^{-3}}(O_{0.01}(e_2))$ is already greater than 97%. From then onward, the time fraction never falls below 93% and the average hitting time decreases quickly to values around 14 ticks. For any $N \geq 100$, $\mu^{N,\epsilon=10^{-3}}(O_{0.01}(e_2)) > 0.95$ and the average hitting time is less than 14 ticks, i.e. it takes on average less than 14 revisions per agent to reach the neighborhood $O_{0.01}(e_2)$, and the process spends more than 95% of the time in that neighborhood.

G. Summary of analytical results

Propositions 5.2 and 5.3 together characterize the dynamics of the nBEPA1 stochastic process $\{X_t^{N,\epsilon}\}$ in any game G^{PC} played in a population context, with positive noise level $(\epsilon > 0)$ and sufficiently large population size N.

They state that, for a large enough population size N and a low enough noise level ϵ, starting from any initial state, the stochastic process $\{X_t^{N,\epsilon}\}$ will approach a neighborhood of the optimal pure state e_s and stay in that neighborhood a fraction of time as high as desired. Moreover, we can make this neighborhood as small as we like by making the noise level ϵ sufficiently small.

An exploration of the 2-strategy SOC game with payoff matrix (1) suggests that neither noise levels have to be excessively low nor populations particularly large for the process to

[10] The sudden changes in the pattern at $N = 100, 200, 300...$ are due to the fact that at these population sizes, the neighborhood $(O_{0.01}(e_2))$ includes one more state for the first time. As an example, at $N \leq 99$ the only state in $O_{0.01}(e_2)$ is the state where every agent is choosing strategy 2, but at $N = 100$, the state where every agent except for one is choosing strategy 2 is also included in $O_{0.01}(e_2)$.

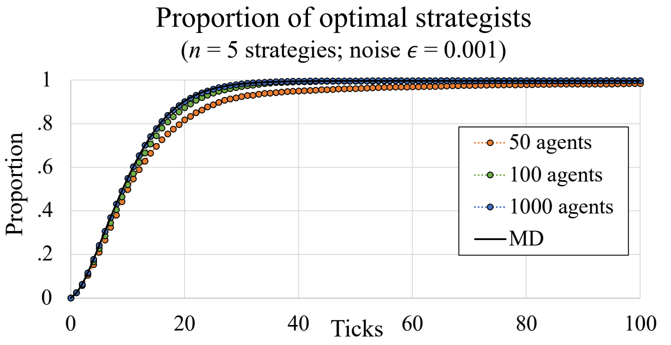


Fig. 7: Time series of the average proportion of agents that are using the optimal strategy $s = 5$ in a SOC game (1) with $n = 5$ strategies, with noise level $\epsilon = 10^{-3}$, starting from initial state $x = (0.9, 0.1, 0, 0, 0)$, for different number of agents. The dots represent the average over 1000 simulation runs and the line labeled MD shows the solution trajectory of the Mean Dynamic (3). All standard errors are below 0.01.

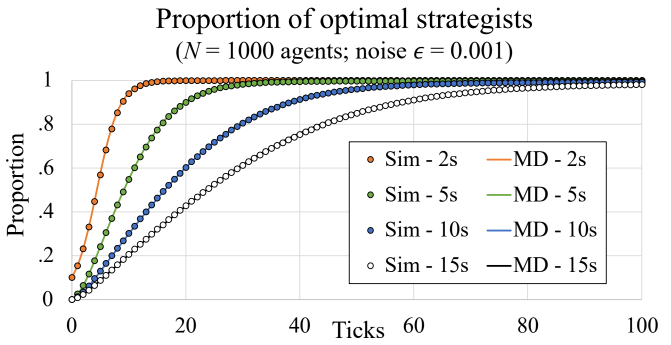


Fig. 8: Time series of the average proportion of agents that are using optimal strategy n in a SOC game (1) with n strategies played in a population with $N = 1000$ agents, starting from an initial state where 90% of agents are using strategy 1 and 10% are using strategy 2, with noise level $\epsilon = 10^{-3}$. The dots represent the average over 1000 simulation runs and the lines labeled MD show the solution trajectory of the Mean Dynamic (3). All standard errors are below 0.01.

quickly reach a small neighborhood around the optimal state and spend most of the time there.

VI. Simulation results

In this section we conduct several simulation experiments to gain a deeper understanding about the speed of convergence of the nBEPA1 process, and on its possible dependence on the number of agents, the number of strategies, the level of noise and on the way agents are scheduled to revise their strategies. We also evaluate the performance of the nBEPA1 algorithm in different types of networks where agents cannot interact with all the other agents, but only with a small subset of the population, i.e. their neighbors in the network.

A. Simulation details

Simulations have been conducted in NetLogo [38], an open-source modeling environment designed for coding and running agent-based simulations. All figures and simulation results reported in this paper can be easily replicated using open-source software freely available under GNU GPL – see Appendix I.

Simulations run in discrete time-steps called ticks. Unless stated otherwise, we use the *asynchronous random independent updating scheme* [39], where in every tick we repeat the following procedure as many times as agents there are: “Take one agent at random and give it the opportunity to revise its strategy.” Thus, the number of revisions that take place in a simulation equals the number of agents times the number of ticks, and every agent is expected to receive exactly one revision opportunity in every tick.

All simulations are conducted on the SOC game with payoff matrix (1). Nonetheless, note that any transformation of the payoff matrix that preserves the relative ordering of payoffs would lead to the same dynamics. Finally, the initial condition for every simulation is the state where 90% of the population are using strategy 1 and 10% are using strategy 2.

B. Different number of agents

In this section we present simulation results of the nBEPA1$^{\epsilon=10^{-3}}$ dynamics for the SOC game with payoff matrix (1) for different population sizes N. Fig. 7 shows the results of 1000 simulation runs of the SOC game (1) with $n = 5$ strategies (where strategy 5 is the optimal), for three different population sizes. Fig. 7 shows that the nBEPA1 dynamics approach the optimal state very quickly (in less than 50 ticks), and that the mean dynamic provides a very good approximation of the transient nBEPA1 dynamics already for 100 agents (and even better for 1000 agents, as one would expect).[11] This also highlights the fact that the time until convergence (in ticks, or number of revisions per agent) is not affected significantly by the number of agents (since the mean dynamic has been derived using the limit as the population size goes to infinity). As for the long-run behavior, by tick 100, the average proportion of 5-strategists is already greater than 98% for $N = 50$, and greater than 99.5% for both $N = 100$ and $N = 1000$.

C. Different number of strategies

Fig. 8 shows a similar experiment where we consider different numbers of strategies n. It is clear that the Mean Dynamic (MD) provides an outstanding approximation for all the different numbers of strategies considered here. As one would expect, the greater the number of strategies, the slower the convergence towards the (small) neighborhood around the optimal state. However, it is remarkable that the vicinity of the optimal state is reached already within 100 ticks even for games with 15 different strategies.

[11] For $N = 50$, in some simulation runs, agents coordinate first on inefficient pure state e_1, and it takes some time to get an experimentation event that allows the population to escape e_1 and coordinate on the optimal state e_5. This explains why the average proportion of optimal strategists across simulation runs is lower for $N = 50$ than for larger populations. For smaller population sizes, this effect is more acute, especially with initial conditions near or at the boundary.

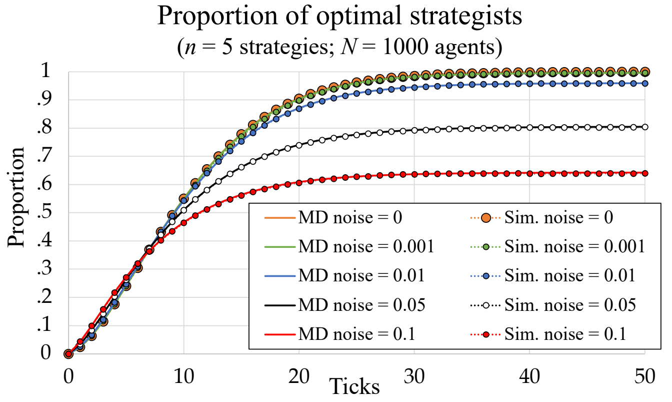


Fig. 9: Time series of the average proportion of agents that are using the optimal strategy $s = 5$ in a SOC game (1) with $n = 5$ strategies played in a population with $N = 1000$ agents, starting from initial state $x = (0.9, 0.1, 0, 0, 0)$, for different levels of noise ϵ. The dots represent the average over 1000 simulation runs and the lines labeled MD show the solution trajectory of the corresponding Mean Dynamic (3). All standard errors are below 0.01.

D. Different levels of noise

In this section we explore the impact of noise. Fig. 9 corresponds to SOC game (1) with $n = 5$ strategies, for different levels of noise. As with the other simulation results in populations with $N = 1000$ agents, the mean dynamic provides an outstanding approximation for the transient and long run dynamics of the stochastic process. For low levels of noise, i.e. $\epsilon \leq 0.01$, the populations quickly approach the vicinity of the optimal state and spend most of the time around there. With higher levels of noise, agents experiment quite often – note that the expected number of experimentations per tick is $N\epsilon$ – and this increases the probability of miscoordinations. Still, we can see in fig. 9 that even with $\epsilon = 0.05$ (i.e. 50 agents are expected to experiment in every tick), on average more than 80% of the population are using the optimal strategy.

E. Robustness to different updating schemes

In this section we explore the robustness of our analytical results to changes in the way agents are scheduled to revise their strategies. In particular, we considered the following updating schemes [39]:

- *Asynchronous random independent*. This is the baseline scheme, where in every tick we repeat the following procedure as many times as agents there are: “Take one agent at random and give it the opportunity to revise its strategy.”
- *Asynchronous random order*. In every tick, we give all agents the opportunity to revise their strategy sequentially in a random order.
- *Synchronous*. In every tick, all agents revise their strategy at the same time (i.e. synchronously).

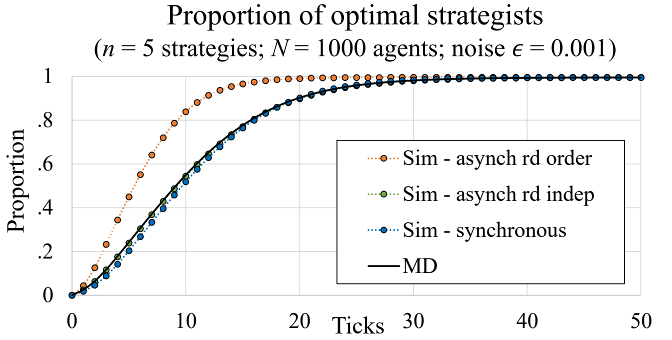


Fig. 10: Time series of the average proportion of agents that are using the optimal strategy $s = 5$ in a SOC game (1) with $n = 5$ strategies played in a population of $N = 1000$ agents, starting from initial state $x = (0.9, 0.1, 0, 0, 0)$, with noise level $\epsilon = 10^{-3}$, for different updating schemes. The dots represent the average over 1000 simulation runs and the line labeled MD shows the solution trajectory of the Mean Dynamic (3). All standard errors are below 0.01.

Fig. 10 shows that, at least for the SOC game (1) with $n = 5$ strategies, our results are robust to these different updating schemes. The synchronous scheme gives results that are in line with the baseline updating scheme (i.e. asynchronous random independent), while the asynchronous random order scheme boosts the speed of convergence to the vicinity of the optimal state e_s, at least in the SOC game (1) with $n = 5$.

F. Robustness to different networks of interaction

Finally, we test the nBEPA1$^{\epsilon=10^{-3}}$ algorithm in different types of networks. In particular, we consider the following topologies: *Ring*, *Barabási–Albert preferential attachment* [40], *Watts-Strogatz small world* with different rewiring probabilities and average degrees [41], and *Complete* (which corresponds to the baseline situation). Fig. 11 shows the time required to enter the set $O_{0.01}(e_s) \equiv \{x \in \Delta S : x_s \geq 0.99\}$ in SOC game (1) with $n = 5$ strategies, for different topologies, including the neighborhood size for each topology as a percentage of the total number of agents.

It is clear that the time required to approach the vicinity of the optimal state e_5 decreases as the network average degree (i.e. the average neighborhood size in the network) increases. This makes intuitive sense – as information flows more quickly through larger neighborhoods – and has also been observed in similar contexts with different algorithms (see e.g. [20], [21], [42]). What is not so obvious is that a neighborhood size of only 5% seems to be enough to reach the vicinity of e_5 as quickly as in the complete network (which has a neighborhood size of 100%), regardless of the topology.

For neighborhood sizes lower than 5%, topology does play a role even when fixing the neighborhood size. This observation is clear when we compare the *Ring* topology and the *Preferential Attachment* topology, both with average neighborhood sizes of 0.2%. The *Ring* topology is much more regular than the *Preferential Attachment* and the time required to reach the vicinity of e_5 is significantly higher. The hypothesis that information flows more slowly through regular topologies makes intuitive sense and is also borne out in *Small*

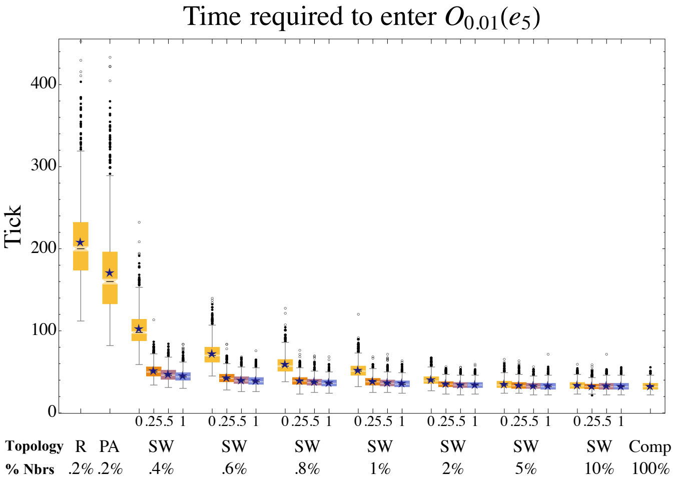


Fig. 11: Distributions of the time required to enter the set $O_{0.01}(e_5) \equiv \{x \in \Delta S : x_5 \geq 0.99\}$ (in ticks) in a SOC game (1) with $n = 5$ strategies played in a population of $N = 1000$ agents, with noise level $\epsilon = 10^{-3}$, starting from an initial state where 90% of agents are using strategy 1 and 10% of agents are using strategy 2, for different network topologies: R: *Ring*, PA: *Preferential attachment*, SW: *Small World* with different average degrees (i.e. neighborhood sizes) and rewiring probabilities (i.e. 0, .25, .5, and 1), and Comp: *Complete*. Each distribution has been compiled running 1000 simulation runs; a new network was generated at the beginning of each simulation run.

World networks. In these networks, the rewiring probability determines how regular the network is. A rewiring probability of zero produces regular networks and a rewiring probability of 1 leads to networks similar to Erdős–Rényi random graphs. In Fig. 11 it is clear that – *ceteris paribus* – the higher the rewiring probability, the lower the time required to approach the vicinity of the optimal state e_5.

Finally, it is striking that *Small World* networks with rewiring probabilities higher than 0.25 and neighborhood sizes greater than just 2% reach the vicinity of the optimal state approximately as quickly as the complete network.

VII. Conclusions

In this paper we have shown that a noisy version of the *Best Experienced Payoff* protocol – named nBEPA1 – can be used to make large populations of agents quickly coordinate on the optimal state in *Single-Optimum Coordination* (SOC) games. The algorithm is completely decentralized, very fast, and scalable both in the number of agents and in the number of strategies.

In terms of methodology, the main value of this paper is that it provides a complete picture of the nBEPA1 dynamics in SOC games, including a formal and computational analysis of both the transient and the asymptotic behavior of the stochastic process. We also provide formal results on both the large population double limit and the small noise limit, and discuss their relevance for engineering problems.

The main limitation of this work is its limited domain of application. Our main results (i.e. Propositions 5.2 and 5.3) are valid in games that satisfy the two Payoff Conditions 3.1. A numerical exploration of the nBEPA1 dynamics in different games relaxing Payoff Conditions 3.1, and the stability results in [13], suggest that Payoff Conditions 3.1 cannot be relaxed to a great extent while preserving global convergence of the algorithm to the vicinity of the optimal state.

Appendix I
Software to Replicate All Figures and Simulation Results

All software created for this paper is open-source and has been released under GNU General Public License.

- Figures 1-3 have been created using EvoDyn-3s [35].
- Figures 4-6 have been created using a *Mathematica* notebook freely available at https://github.com/luis-r-izquierdo/nBEPA1.
- Simulation results reported in figures 7-10 can be replicated using a purpose-built NetLogo model freely available at https://luis-r-izquierdo.github.io/nbepa1-socg/, or with the more general software ABED-1pop, using parameter file *noisy-bep-all-1-single-optimum-coordination-game-5s.csv* as a baseline.
- Simulation results reported in figure 11 can be replicated using a purpose-built NetLogo model freely available at https://luis-r-izquierdo.github.io/nbepa1-socg-nw/.

References

[1] N. Quijano, C. Ocampo-Martinez, J. Barreiro-Gomez, G. Obando, A. Pantoja, and E. Mojica-Nava, "The Role of Population Games and Evolutionary Dynamics in Distributed Control Systems: The Advantages of Evolutionary Game Theory," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 70–97, 2017. doi: 10.1109/MCS.2016.2621479

[2] J. R. Marden and J. S. Shamma, "Game Theory and Distributed Control," in *Handbook of Game Theory with Economic Applications*, H. P. Young and S. Zamir, Eds. Amsterdam: Elsevier, 2015, vol. 4, ch. 16, pp. 861–899.

[3] ——, "Game-theoretic learning in distributed control," in *Handbook of Dynamic Game Theory*, T. Başar and G. Zaccour, Eds. Cham: Springer International Publishing, 2017, pp. 511–546. ISBN 978-3-319-44374-4

[4] W. H. Sandholm, *Population Games and Evolutionary Dynamics*. The MIT Press, 2010. ISBN 9780262195874

[5] N. Li and J. R. Marden, "Designing Games for Distributed Optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, apr 2013. doi: 10.1109/JSTSP.2013.2246511

[6] J. R. Marden and A. Wierman, "Distributed welfare games," *Operations Research*, vol. 61, no. 1, pp. 155–168, 2013. doi: 10.1287/opre.1120.1137

[7] Y. Wang, D. Cheng, and X. Liu, "Matrix expression of Shapley values and its application to distributed resource allocation," *Science China Information Sciences*, vol. 62, no. 2, p. 22201, 2019. doi: 10.1007/s11432-018-9414-5

[8] D. Monderer and L. S. Shapley, "Potential Games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996. doi: 10.1006/game.1996.0044

[9] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007. ISBN 9780511800481

[10] J. R. Marden, H. P. Young, and L. Y. Pao, "Achieving Pareto optimality through distributed learning," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2753–2770, 2014. doi: 10.1137/110850694

[11] J. R. Marden, S. D. Ruben, and L. Y. Pao, "A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1207–1214, 2013. doi: 10.1109/TCST.2013.2257780

[12] W. H. Sandholm, S. S. Izquierdo, and L. R. Izquierdo, "Best experienced payoff dynamics and cooperation in the centipede game," *Theoretical Economics*, vol. 14, no. 4, pp. 1347–1385, 2019. doi: 10.3982/TE3565

[13] W. H. Sandholm, S. S. Izquierdo, and L. Izquierdo, "Stability for best experienced payoff dynamics," *Journal of Economic Theory*, vol. 185, p. 104957, 2020. doi: 10.1016/J.JET.2019.104957

[14] S. Hart and A. Mas-Colell, "Uncoupled dynamics do not lead to Nash equilibrium," *American Economic Review*, vol. 93, no. 5, pp. 1830–1836, 2003. doi: 10.1257/000282803322655581

[15] Y. Babichenko, "Completely uncoupled dynamics and Nash equilibria," *Games and Economic Behavior*, vol. 76, no. 1, pp. 1–14, sep 2012. doi: 10.1016/J.GEB.2012.06.004

[16] D. P. Foster and H. P. Young, "Regret testing: learning to play nash equilibrium without knowing you have an opponent," *Theoretical Economics*, vol. 1, no. 3, pp. 341–367, 2006.

[17] Y. Shoham and M. Tennenholtz, "On the emergence of social conventions: modeling, analysis, and simulations," *Artificial Intelligence*, vol. 94, no. 1, pp. 139–166, 1997. doi: 10.1016/S0004-3702(97)00028-3

[18] S. S. Izquierdo and L. R. Izquierdo, "Stochastic approximation to understand simple simulation models," *Journal of Statistical Physics*, vol. 151, no. 1, pp. 254–276, 2013. doi: 10.1007/s10955-012-0654-z

[19] G. Loginov, "Ordinal imitative dynamics," *International Journal of Game Theory*, 2021. doi: 10.1007/s00182-021-00797-7

[20] M. Mihaylov, K. Tuyls, and A. Nowé, "A decentralized approach for convention emergence in multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 28, no. 5, pp. 749–778, 2014. doi: 10.1007/s10458-013-9240-2

[21] D. Villatoro, S. Sen, and J. Sabater-Mir, "Exploring the dimensions of convention emergence in multiagent systems," *Advances in Complex Systems*, vol. 14, no. 02, pp. 201–227, 2011. doi: 10.1142/S0219525911003013

[22] G. E. Kreindler and H. P. Young, "Fast convergence in evolutionary equilibrium selection," *Games and Economic Behavior*, vol. 80, pp. 39–67, 2013. doi: 10.1016/j.geb.2013.02.004

[23] H. P. Young, *Individual Strategy and Social Structure*. Princeton: Princeton University Press, 1998.

[24] ——, "The dynamics of social innovation," *Proceedings of the National Academy of Sciences*, vol. 108, no. supplement_4, pp. 21 285–21 291, 2011. doi: 10.1073/pnas.1100973108

[25] M. Belloc and S. Bowles, "The persistence of inferior cultural-institutional conventions," *The American Economic Review*, vol. 103, no. 3, pp. 93–98, 05 2013. doi: 10.1257/aer.103.3.93

[26] J. Newton and S. D. Angus, "Coalitions, tipping points and the speed of evolution," *Journal of Economic Theory*, vol. 157, pp. 172–187, 2015. doi: 10.1016/j.jet.2015.01.003

[27] M. Kandori and R. Rob, "Evolution of equilibria in the long run: A general theory and applications," *Journal of Economic Theory*, vol. 65, no. 2, pp. 383–414, 1995. doi: 10.1006/jeth.1995.1014

[28] G. E. Kreindler and H. P. Young, "Rapid innovation diffusion in social networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 3, pp. 10 881–10 888, 2014. doi: 10.1073/pnas.1400842111

[29] M. J. Osborne and A. Rubinstein, "Games with Procedurally Rational Players," *American Economic Review*, vol. 88, no. 4, pp. 834–847, 1998. [Online]. Available: http://www.jstor.org/stable/117008

[30] R. Sethi, "Stability of Equilibria in Games with Procedurally Rational Players," *Games and Economic Behavior*, vol. 32, no. 1, pp. 85–104, 2000. doi: 10.1006/GAME.1999.0753

[31] ——, "Stable sampling in repeated games," *Journal of Economic Theory*, vol. 197, p. 105343, 2021. doi: 10.1016/j.jet.2021.105343

[32] S. Arigapudi, Y. Heller, and I. Milchtaich, "Instability of defection in the prisoner's dilemma under best experienced payoff dynamics," *Journal of Economic Theory*, vol. 197, p. 105174, 2021. doi: 10.1016/j.jet.2020.105174

[33] S. S. Izquierdo and L. R. Izquierdo, "'Test two, choose the better' leads to high cooperation in the centipede game," *Journal of Dynamics and Games*, 2021. doi: 10.3934/jdg.2021018

[34] M. Benaïm and J. W. Weibull, "Deterministic Approximation of Stochastic Evolution in Games," *Econometrica*, vol. 71, no. 3, pp. 873–903, 2003. doi: 10.1111/1468-0262.00429

[35] L. R. Izquierdo, S. S. Izquierdo, and W. H. Sandholm, "EvoDyn-3s: A Mathematica computable document to analyze evolutionary dynamics in 3-strategy games," *SoftwareX*, vol. 7, pp. 226–233, 2018. doi: 10.1016/j.softx.2018.07.006

[36] M. Benaïm and J. W. Weibull, "Mean-field approximation of stochastic population processes in games." 2009. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00435515/document

[37] H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, no. 1, pp. 57–84, 1993. doi: 10.2307/2951778

[38] U. Wilensky, "Netlogo. Software. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL," 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[39] D. Cornforth, D. G. Green, and D. Newth, "Ordered asynchronous processes in multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 204, no. 1-2, pp. 70–82, 2005. doi: 10.1016/j.physd.2005.04.005

[40] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999. doi: 10.1126/science.286.5439.509

[41] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998. doi: 10.1038/30918

[42] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Rev. Mod. Phys.*, vol. 81, pp. 591–646, 2009. doi: 10.1103/RevModPhys.81.591

Luis R. Izquierdo was born in Madrid, Spain, in 1977. He received a M.Sc. degree in industrial engineering from the University of Valladolid (Spain) in 2002, a B.Sc. in business and management from the Open University of Catalunya (Spain) in 2003, and the Ph.D. degree in game theory and social simulation from Manchester Metropolitan University (UK) in 2008.

From 2002 to 2006 he worked as an Agent-Based Modeler at the Macaulay Institute (UK). Since then, he has been lecturing at the University of Burgos (Spain), where he became a Full Professor in 2019. During this time, he has also visited various prestigious research institutions all around the globe. He is an Associate Editor of the *Journal of Artificial Societies and Social Simulation* and, together with Segismundo S. Izquierdo and Prof. William H. Sandholm, he has designed, implemented and released more than 30 open-source computational programs (see https://luis-r-izquierdo.github.io). His research interests include evolutionary game theory and the analysis of complex systems.

Segismundo S. Izquierdo was born in Santander, Spain, in 1972. He received a M.Sc. degree in industrial engineering from the University of Valladolid (Spain) in 1998, and, after some years working as a business consultant at PricewaterhouseCoopers, he obtained a PhD in System Identification in 2005.

He is a Full Professor at Universidad de Valladolid, and his main area of research is evolutionary game theory. He has been awarded prestigious international grants, such as a Fulbright fellowship in 2021, and has been a visiting researcher at different centers, such as the University of Wisconsin-Madison (USA), the European University Institute (Italy) or the Stockholm School of Economics (Sweden).

Javier Rodríguez was born in León, Spain, in 1978. He received the B.S. degree in physics from the University of Salamanca (Spain) in 2007, the M.S. degree in instrumentation in physics from the University of Valladolid (Spain) in 2011 and the Ph.D. degree in remote sensing and machine learning from the University of Valladolid (Spain) in 2014.

During his Ph.D., he visited the Max Planck Institute for the Physics of Complex Systems (Dresden, Germany; 2013). He currently works as a Data Scientist in Telefónica I+D. His research interests include complex systems, and machine learning.