

An Introduction to *ABED*: Agent-Based Simulation of Evolutionary Game Dynamics*

Luis R. Izquierdo[†], Segismundo S. Izquierdo[‡], and William H. Sandholm[§]

February 1, 2018

Abstract

ABED is free and open-source software for simulating evolutionary game dynamics in finite populations. We explain how *ABED* can be used to simulate a wide range of dynamics considered in the literature and many novel dynamics. In doing so, we introduce a general model of revisions for dynamic evolutionary models, one that decomposes strategy updates into selection of candidate strategies, payoff determination, and choice among candidates. Using examples, we explore ways in which simulations can complement theory in increasing our understanding of strategic interactions in finite populations.

1. Introduction

This paper introduces *ABED*, a free and open-source software tool for simulating evolutionary game dynamics in finite populations. *ABED* is simple to use, but powerful enough to execute complex Monte Carlo experiments. It implements a wide range of specifications of the rules agents follow when updating their strategies, allowing it to simulate versions of a substantial proportion of the dynamic evolutionary models appearing

*Financial support from NSF Grants SES-1458992 and SES-1728853, U.S. Army Research Office Grant MSN201957, grants PRX15/00362 and PRX16/00048 awarded by the Spanish “Ministerio de Educación, Cultura y Deporte”, and Spanish Ministry of Science and Innovation’s project ECO2017-83147-C2-2-P (MINECO/AEI/FEDER, UE) is gratefully acknowledged.

[†]Department of Civil Engineering, Edificio la Milanera, Calle de Villadiego, Universidad de Burgos, 09001 Spain. e-mail: lrizquierdo@ubu.es; website: www.luis.izqui.org.

[‡]Department of Industrial Organization, Paseo del Cauce 59, Universidad de Valladolid, 47011 Spain. e-mail: segis@eis.uva.es; website: www.segis.izqui.org.

[§]Department of Economics, University of Wisconsin, 1180 Observatory Drive, Madison, WI 53706, USA. e-mail: whs@ssc.wisc.edu; website: www.ssc.wisc.edu/~whs.

in the literature, as well as many models that have yet to be analyzed (see [Table 1](#)). Nearly all parameters in *ABED* can be modified while the program is running, making it easy to examine the consequences of changing assumptions about how agents make decisions, and facilitating the exploration of novel models.

To best relate *ABED* to existing formal models, we provide a general specification of stochastic evolutionary game dynamics for finite populations. It is distinguished by a decomposition of decisions by revising agents into three stages: selection of candidate strategies, determination of their payoffs, and choice among the candidate strategies. This decomposition offers a unified framework for understanding a range of models that might otherwise not be easy to compare directly.

The paper also examines some roles for agent-based simulations in evolutionary game theory. Simulations are sometimes viewed with a skeptical eye in game and economic theory. They can be criticized on the grounds that any simulation run concerns a specific choice of parameter values, and that observing a certain outcome in a series of simulation runs does not imply that this outcome must always occur. But while simulations cannot prove theorems, they serve a variety of purposes that theorists should value. As theorem statements often invoke limits, simulations can be used to evaluate robustness to realistic choices of parameters, serving a similar role as small-sample robustness checks in econometrics. Simulations can help us interpret abstractly defined solution concepts by allowing us to flesh out the implicit assumptions that underlie them. Finally, simulations are a powerful exploratory tool, one that can suggest new models and results for formal analysis. While one may object to some of the grander claims about the role of simulations in social science, such objections do not exclude many other ways that theory and simulation can serve complementary purposes. In what follows we aim to illustrate certain aspects of this symbiosis.

1.1 Exact analysis, approximation, and simulation of population dynamics

The canonical model in evolutionary game theory is one in which a finite population of agents are recurrently matched to play a normal form game.¹ Each agent is occasionally given the opportunity to revise his (pure) strategy, basing his choice on the information about payoffs and current aggregate behavior he has obtained. Formulating such a model requires one to specify

- (i) the number of agents N in the population,
- (ii) the n -strategy normal form game the agents are recurrently matched to play,

¹For a historical overview, see [Sandholm \(2010b, Ch. 1\)](#).

I: Imitative protocols

		matching	
		complete matching	limited matching
decision method	pairwise difference	replicator ^a	
	best	imitate the best ^b	imitate the best realization ^c
	logit	imitative logit ^d	—
	positive proportional	Maynard Smith replicator ^e	—

II: Direct protocols

		matching / sampling		
		complete matching	limited matching	
			single sample	multiple samples
decision method	pairwise difference	Smith ^f	—	—
	best	best response ^g	sample best response ^h	best experienced payoff ⁱ
	logit	logit ^j	sample logit ^k	—
	positive proportional	—	—	—

^aTaylor and Jonker (1978); Helbing (1992); Schlag (1998); also see Weibull (1995); Hofbauer (1995a); Binmore et al. (1995).

^bHofbauer (1995a); Vega-Redondo (1997).

^cIzquierdo and Izquierdo (2013).

^dWeibull (1995).

^eMaynard Smith (1982); Weibull (1995); Fudenberg et al. (2006); Fudenberg and Imhof (2008); Sandholm (2010b).

^fSmith (1984); Sandholm (2010a).

^gGilboa and Matsui (1991); Hofbauer (1995b); Kandori and Rob (1995).

^hSandholm (2001); Oyama et al. (2015).

ⁱSethi (2000); Sandholm et al. (2017).

^jBlume (1997); Fudenberg and Levine (1998); Hofbauer and Sandholm (2007).

^kKreindler and Young (2013).

Table 1: Classes of revision protocols in ABED and principal instances of their mean dynamics.

- (iii) the rule describing how revision opportunities are assigned to the agents, and
- (iv) the protocol according to which agents revise their strategies when opportunities to do so arise.

Of these items, the assignment rule (iii) and especially the revision protocol (iv) are particular to the evolutionary approach. In economic modeling, a revision protocol captures the information agents possess when revising, how that information is obtained, and the way that agents use the information to select a new strategy.² In biological contexts, revision is often interpreted as a death and birth event rather than as a conscious decision, and assignment rules and revision protocols are specified accordingly.³

The state of the resulting evolutionary process can be described in two ways. One can define the state to be the profile of strategies currently chosen by each of the N agents. Alternatively, one can specify the state to be the population distribution of the strategies chosen by these agents. As the name indicates, agent-based simulations start from the former specification, while formal analyses generally use the more parsimonious specification in terms of distributions over strategies, or *population states*.

In the latter approach, the analysis of the evolutionary process focuses on the behavior of a Markov chain $\{X_t^N\}$ on the finite set of population states, a grid \mathcal{X}^N of mesh $\frac{1}{N}$ in the simplex $X \in \mathbb{R}^n$ of probability distributions over the n pure strategies. Under certain assumptions about the game and revision procedure, it is possible to analyze this Markov chain directly.⁴ More typically, the Markov chain $\{X_t^N\}$ is not susceptible to direct analysis.

This difficulty can be circumvented by studying limiting versions of the Markov chain $\{X_t^N\}$. Here we are most interested in the large population limit. To understand the behavior of the process $\{X_t^N\}$ over some fixed time horizon $[0, T]$, we take the population size N to infinity to obtain limiting processes that are easier to analyze than the original Markov chain, and the analysis is then performed on the limiting processes so obtained. Because the limiting processes run in continuous-time on the continuous state space X , they can be studied using methods from calculus and analysis.

Focusing on these limiting processes is formally justified by suitable approximation results.⁵ The most important of these is the finite-horizon deterministic approximation theorem (Benaïm and Weibull (2003), Roth and Sandholm (2013)). The theorem shows that over any fixed time horizon, as the population size N approaches infinity, the sample paths of the Markov chains $\{X_t^N\}$ converge to solutions of a deterministic *mean dynamic* defined

²See Weibull (1995), Björnerstedt and Weibull (1996), and Sandholm (2010b, 2015).

³See, e.g., Nowak et al. (2004) and Fudenberg et al. (2006).

⁴For instance, Monderer and Shapley (1996) and Blume (1997) on better response and logit dynamics in potential games, and Kandori and Rob (1995) on best response dynamics in supermodular games.

⁵We present these results in more detail in Appendices B.2 and B.3.

by the expected motion of $\{X_t^N\}$ from each state. This result formally links the wide array of deterministic evolutionary dynamics studied in the literature and the revision protocols that generate them (Table 1). To understand the behavior of the Markov chains $\{X_t^N\}$ near rest points x^* of the mean dynamic (in other words, to describe “equilibrium” behavior in the population), one can instead appeal to a diffusion approximation (Sandholm (2003)). This result shows that a “local behavior process”, obtained by magnifying deviations of $\{X_t^N\}$ from x^* by a factor of \sqrt{N} , is approximated by the solution to a stochastic differential equation whose linear drift coefficient and constant diffusion coefficient are determined from the transition probabilities of $\{X_t^N\}$ near x^* .⁶

Of course, the population sizes relevant in applications are finite, and not always very large. Thus as a practical matter, one should ask to what extent results that rely on large population limits describe behavior in populations of moderate or small size. These questions are generally quite difficult to answer analytically. Simulations provide a simple way of evaluating the robustness of formal analyses whose tractability relies on sending parameter values to extremes.

Mathematical tractability shapes not only the forms that questions about evolutionary models take, but also the specific assumptions that the models employ. As an example, most models of evolution for normal form games make the analytically convenient assumption that revising agents are able to evaluate strategies’ expected payoffs. But in applications where agents from large populations engage in random matches, knowledge of expected payoffs would need to come from a complete matching (so that expected payoffs are just realized payoffs), from knowledge of the current population state (so that expected payoffs could be calculated), or from a central source. While these possibilities describe some applications, in others it seems likely that payoff information is more limited, with evaluations of strategies based on information obtained from samples, on direct but limited experiences, or from observations of others’ experiences.⁷ In some cases, this additional level of detail can push models beyond the limits of mathematical tractability. One can use simulations to assess whether assumptions imposed for simplicity are robust

⁶Other limiting analyses are used to understand the behavior of $\{X_t^N\}$ over longer time spans. *Stochastic stability analyses* consider behavior in the infinite horizon limit by means of the stationary distribution of $\{X_t^N\}$. Tractability is gained by considering the limit as the noise level in agents’ decisions vanishes (see Foster and Young (1990); Kandori et al. (1993); Young (1993, 1998); Ellison (2000)), or as the population size grows large (see Binmore et al. (1995); Binmore and Samuelson (1997)), or both; see Sandholm (2010b, chapters 11 and 12) for an overview of this literature. Transitions between stable rest points in these limits are analyzed using methods from *large deviations theory*; see the previous references and Sandholm and Staudigl (2016) for the case of small noise limits, and Sandholm and Staudigl (2018) for the large population limit. Finally, *weak selection analyses* introduce a scaling parameter to study biological interactions with very small differences in fitness, so that the Markov process $\{X_t^N\}$ comes to resemble a perturbed random walk (see Nowak et al. (2004); Wu et al. (2010); Sample and Allen (2017)).

⁷Early work emphasizing this point includes Schlag (1998, 1999), Sethi (2000), and Sandholm (2001).

to more realistic alternatives, and, in cases where they are not, to look for alternative models and candidate theorems that seem likely to yield to formal analysis.⁸

1.2 Brief description of *ABED* and comparisons to other software

*ABED*⁹ is free and open-source software for running agent-based simulations in the finite-population evolutionary game framework described at the start of [Section 1.1](#). Its simple graphical user interface ([Figure 1](#)) allows one to implement a wide range of the revision protocols studied in the literature, including all of those appearing in [Table 1](#), and to fine-tune many other aspects of the evolutionary process. At present, *ABED* consists of two separate computer programs: *ABED-1pop*, for single populations of agents matched to play symmetric two-player games; and *ABED-2pop* for pairs of populations of agents matched to play (possibly asymmetric) two-player games.

ABED runs in *NetLogo* (Wilensky (1999)), a well-documented free and open-source platform for designing agent-based simulations, for conducting automatic explorations of the parameter space, and for carrying out statistical analyses of the results. In the present context, the term “agent-based” means that each agent in an *ABED* simulation is represented in the code as a distinct object. This approach makes it relatively easy to modify *ABED*’s code to incorporate heterogeneity in agents’ preferences, locations, and interaction patterns.¹⁰

There are other useful software packages available for analyzing and simulating evolutionary game dynamics. *Dynamo* (Sandholm et al. (2012); Franchetti and Sandholm (2013)) is a suite of easy-to-use *Mathematica* notebooks for generating phase diagrams, vector fields, and other graphics related to mean dynamics from evolutionary game theory. *BIRDS* (Dzonsons and Weibull (2013)) is a finite-population agent-based simulator that implements several revision protocols. *PDToolbox* (Barreto (2014)) is a set of functions coded in *Matlab* for analyzing both mean dynamics and finite-population agent-based models derived from built-in revision protocols. To start with basic comparisons, *ABED*, unlike *Dynamo* but like *BIRDS* and *PDToolbox*, simulates finite-population dynamics. Also, unlike *Dynamo* and *PDToolbox* but like *BIRDS*, *ABED* can be run without proprietary software. Turning to details, *ABED* is especially flexible in the specification of revision protocols, and incorporates revisions based on both complete and limited matching. Sim-

⁸Working in the reverse direction, Izquierdo et al. (2013) show different ways in which mathematical analysis and computer simulation can be usefully combined to produce a better understanding of the dynamics of agent-based evolutionary models.

⁹Downloadable from <https://luis-r-izquierdo.github.io/abed/>.

¹⁰For more on implementing, extending, and analyzing agent-based evolutionary models using *NetLogo*, see Izquierdo et al. (2018).

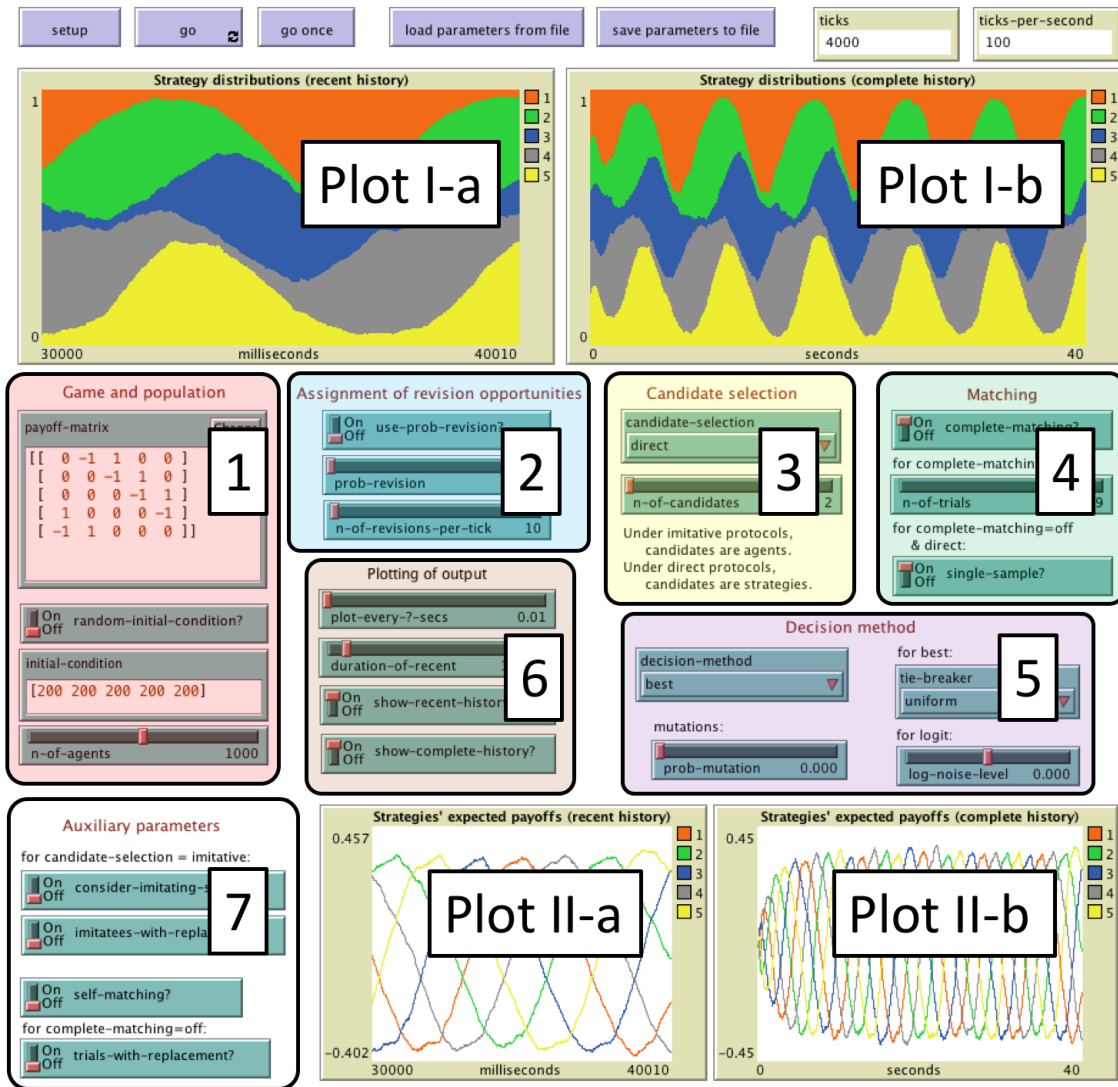


Figure 1: Interface of ABED-1pop

ulations in *ABED* can be inspected and parameters modified during runtime, making it easy to explore the effects of different modeling assumptions on population outcomes. Finally, *ABED* is exceptionally user-friendly, with a simple graphical user interface, and with extensive documentation and video tutorials available online.

1.3 Outline

The paper proceeds as follows. [Section 2](#) provides a detailed description of *ABED*, focusing for simplicity on *ABED*-1pop. [Section 2.1](#) explains *ABED*'s parameters, including a thorough account of those used to specify revision protocols. [Section 2.2](#) shows how *ABED* can be parameterized to model a wide range of revision protocols studied in the literature, plus many others yet to be analyzed. The next two sections deal with *ABED*'s output: [Section 2.3](#) describes *ABED*'s plots, and [Section 2.4](#) indicates how to conduct Monte Carlo automatic explorations of *ABED*'s parameter space. In [Section 3](#) we present a variety of examples that both show *ABED*'s capabilities and illustrate the roles for simulation in evolutionary game theory. Some concluding remarks are offered in [Section 4](#).

The Appendices contain a variety of details omitted from the main text. Appendix [A](#) provides further information on *ABED*'s parameters. Appendix [B](#) provides a formal framework for describing finite-population stochastic evolutionary processes that covers all those implemented in *ABED*, reviews the deterministic and diffusion approximations, and presents a number of examples. Finally, Appendix [C](#) provides detailed instructions to run computational experiments with *ABED*.

2. Description of *ABED*

ABED's interface ([Figure 1](#)) contains buttons for running the simulation and for loading/saving parameter files, monitors that show the passage of time, interface elements organized into blocks for choosing simulation parameters, and plots that present simulation results.

The top row of the interface contains five blue buttons. The [setup](#) button initializes the model, the [go](#) button makes the model run indefinitely (until the same button is clicked again), and the [go once](#) button runs the model for only one period (or “tick” in *NetLogo* parlance). The two remaining buttons are used to load and save parameter files.

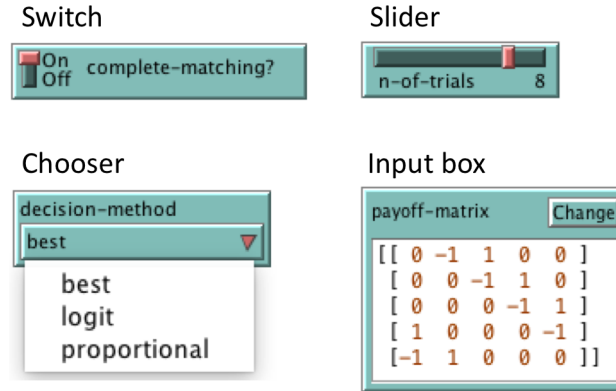


Figure 2: *NetLogo* interface elements used in *ABED* to input parameter values

2.1 *ABED*'s parameters

In this section, we explain the parameters of *ABED*. In the course of doing so, we explain how simulations in *ABED* are executed, and we describe the range of models that *ABED* is able to enact. It is worth reiterating that nearly all parameter values can be modified while *ABED* is running, with immediate effects on how the simulation proceeds.¹¹

Input in *ABED* is entered using *switches* (for binary parameters), *choosers* (dropdown menus for parameters with multiple named values), *sliders* (for numerical parameters), and an *input box* (to enter the payoff matrix). Each is illustrated in Figure 2. In what follows we identify an interface item with the parameter it controls.

The main parameters that specify a simulation (Blocks 1–5) appear in the middle part of *ABED*'s interface (Figure 3). Block 6 contains parameters that control the plotting of output, and Block 7 includes a few secondary parameters that are explained in Appendix A.1.

2.1.1 The game, the population size, and the initial state

Block 1 contains the parameters used to specify the game, the population size, and the initial state.

The symmetric normal form game is entered into the payoff-matrix input box in the form of a square matrix, with spaces between payoff entries in each row, a pair of square brackets containing each row, and an additional pair of brackets containing all of the rows. *ABED* deduces the number of strategies (henceforth *n-of-strategies*) from the size of this square matrix. How agents are matched to play this game is defined by parameters from later blocks.

¹¹The payoff matrix and the initial conditions are the only two exceptions.

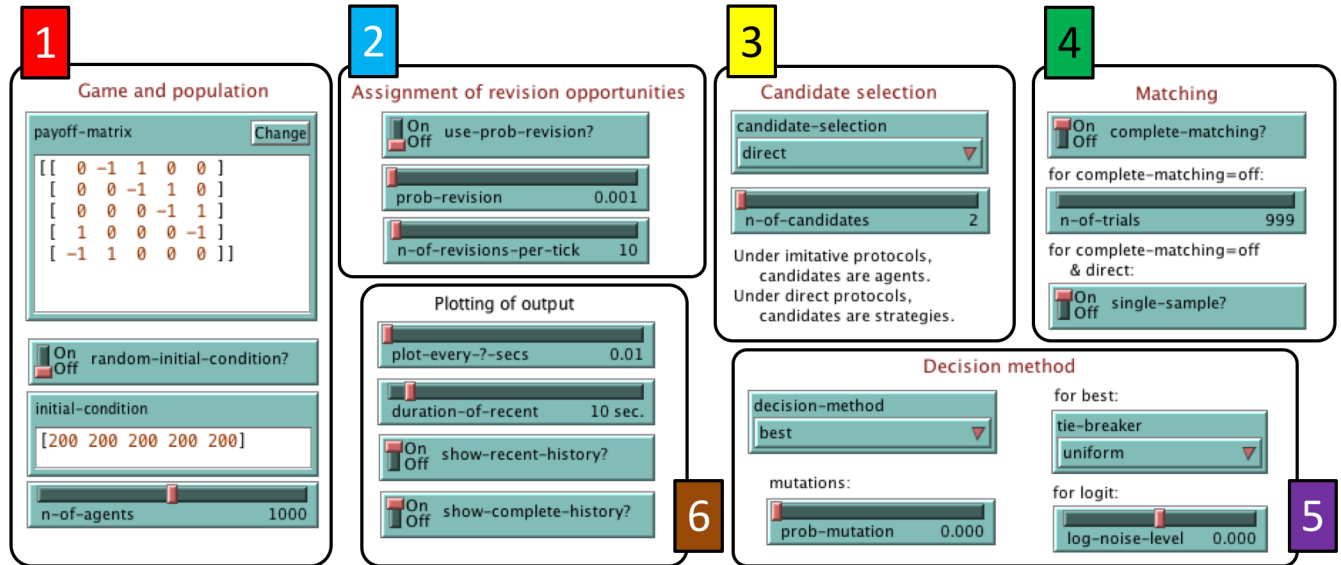


Figure 3: Main parameters in ABED-1pop.

The initial population state of the simulation is set as follows:

- If the switch `random-initial-condition?` is *on*, then the number of agents in the population is specified using the slider `n-of-agents`, and each of these agents is assigned one of the `n-of-strategies` strategies randomly.
- If `random-initial-condition?` is *off*, then the initial condition is read from the `initial-condition` input box. The input here is a list of `n-of-strategies` nonnegative integers, separated by spaces and enclosed by square brackets (e.g. `[200 200 200 200 200]`). The i -th element of the list `initial-condition` specifies the number of agents using strategy i at the beginning of the simulation. In this case the value of `n-of-agents` is automatically set to the sum of the elements in `initial-condition`.

As with the other parameters, the value of `n-of-agents` can be changed while the simulation is running. If the slider is moved to the left, randomly chosen agents are removed from the population; if it is moved to the right, randomly chosen agents are cloned. Thus on average, the proportions of agents playing each strategy remain the same, although the actual effect on these proportions is stochastic.

2.1.2 Assignment of revision opportunities to agents

The assignment of revision opportunities to agents during each period is controlled by the options in Block 2.

- If the switch `use-prob-revision?` is *on*, then assignments are stochastic and independent, with the probability that an agent is assigned an opportunity being specified using the slider `prob-revision`.
- If instead `use-prob-revision?` is *off*, then a fixed number of opportunities are allocated among the `n-of-agents` agents at random. This number is specified using the slider `n-of-revisions-per-tick`.

2.1.3 Revision protocols

ABED allows the user considerable flexibility in specifying the agents' revision protocols. Here revision protocols are defined by three main elements: how candidate strategies are selected, how agents evaluate candidate strategies, and how the latter information is mapped to the probabilities of choosing each candidate strategy. These elements are defined using the parameters from Blocks 3–5 of the interface, along with a few auxiliary parameters placed in Block 7 (see Appendix A.1) whose default values are usually satisfactory.

Selection of candidate strategies

The parameters in Block 3 determine how agents select and evaluate candidate strategies. The chooser `candidate-selection` specifies which of the two basic options is applied.

- *Direct protocols.* If the `candidate-selection` is set to *direct*, then the revising agent's additional candidate strategies—those besides his current strategy—are a random subset of the other strategies, with a strategy's current share having no bearing on whether it is chosen as a candidate. In this case, the slider `n-of-candidates` determines the total number of strategies the revising agent considers. The revising agent's current strategy is always part of the set of candidates. To evaluate a strategy's performance, an agent plays it a number of times against a sample of opponents, as described in the next section.
- *Imitative protocols.* If `candidate-selection` is set to *imitative*, then an agent decides which strategies besides his current one to include as candidates by choosing other agents from the population at random. Specifically, the revising agent will compile a multiset of `n-of-candidates` agents to copy the strategy of one of them. The revising agent is always part of this multiset of agents. The selection of the other (`n-of-candidates` - 1) agents is conducted randomly, so popular strategies in the population are more likely to be observed than less popular ones. All agents in the

set of candidates play the game against a sample of opponents (as described below), and the revising agent observes the total payoffs obtained by each. In the end, the revising agent has a list of *n-of-candidates* candidate strategies with associated payoffs, where the same candidate strategy may appear several times on the list.

Matching and payoffs

The matching process, which determines the total payoff an agent obtains when playing or testing a strategy is governed by the three parameters in Block 4:

- If the switch *complete-matching?* is *on*, then an agent playing a strategy is matched against all other agents.
- If *complete-matching?* is *off*, then the agent plays his current strategy against a random sample of opponents. The size of this sample is specified using the slider *n-of-trials*.¹² If in addition the protocol is *direct*, the switch *single-sample?* determines how many samples of opponents the agent faces during testing:
 - If *single-sample?* is *on*, the revising agent draws a single sample of *n-of-trials* opponents and tests each of his candidate strategies against this sample.
 - If *single-sample?* is *off*, the agent tests each of his candidate strategies against distinct, independent samples of *n-of-trials* opponents. Thus, different strategies are tested against potentially different opponents.

Decision method

The procedures described in the previous two sections leave a revising agent with an assignment of a payoff to each strategy in a multiset of *n-of-candidates* elements. How the agent responds to this information—the probabilities with which he winds up switching to each strategy—is specified in Block 5. The main parameter here is the *decision-method*:

- If the *decision-method* is *best*, then the agent chooses a strategy whose payoff was highest. The chooser *tie-breaker* provides various options for handling multiple maxima that are described in Appendix A.2. One option, *random walk*, has agents randomize among optimal strategies with probabilities determined by an auxiliary random walk, allowing the population to wander around a component of Nash equilibria (see Section 3.5 and Appendix B.4).

¹²Under an imitative protocol, during any period in which multiple agents revise, a single sequence of trials is conducted for each agent who is either revising or a candidate for imitation, and this sequence determines the average payoff recorded for the agent during the period. Payoffs an agent obtains when he is participates in another agent's sequence are not counted toward his own average payoff.

- If the decision-method is *logit*, then the choice probabilities are determined by applying the logit choice rule to the collection of *average* payoffs. The logit noise level is specified using the slider `log-noise-level` (see details in Appendix B.1, eq. 11).
- If the decision-method is *positive-proportional*, then the agent chooses a strategy randomly with probabilities proportional to *average* payoffs. Thus, when using this decision method, the payoffs specified in `payoff-matrix` should be non-negative.
- If the decision-method is *pairwise-difference*, the agent is required to be considering exactly two strategies, his current one and a single (random) alternative.¹³ The agent switches to the alternative strategy only if that strategy yielded a higher average payoff than his current strategy, and in this case he switches with probability proportional to the payoff difference. These payoff differences are converted into probabilities by dividing by the largest payoff difference that is possible in the game.

Finally, the parameter `prob-mutation` specifies the probability with which a revising agent opts out of the process described above, instead choosing a strategy uniformly at random from the `n-of-strategies` available strategies.

2.2 Markov chains, mean dynamics, and diffusion approximations

A simulation run in *ABED* is a sample path of the Markov chain whose transition law is induced by the game, the population size, how revision opportunities are assigned to agents, and the revision protocol. Focusing on the main qualitative options that determine the revision protocol, Table 1 distinguishes 20 trios of choices for candidate selection, matching, and decision method, 8 imitative and 12 direct. Of these 20 trios of main options, there are 12 (to our knowledge) that have been studied in the literature, either in finite population models with explicitly stated revision protocols, or in infinite-population models in the form of mean dynamics. Table 1 reports the names of these mean dynamics and provides references to papers introducing either a mean dynamic or a finite population process corresponding to the trio in question.¹⁴

¹³Thus, `n-of-candidates` is automatically set to 2.

¹⁴We should emphasize that in most cases, the trios by themselves only partially describe the revision protocol. The choices of number of opponents to observe, alternate strategies to test, trials playing a strategy, noise parameters, and other parameters are needed to complete the specification; describe the combined with the various other parameter choices to create a large variety of revision protocols. The mean dynamics named in Table 1 sometimes restrict these other parameters as well, so even the trios that have been studied may only have been studied in some instances.

Appendix B provides a formal framework for describing such finite-population Markov processes of the sort simulated by *ABED*. This framework is based on a decomposition of revision protocols into the three stages described above: selection of candidate strategies, matching and determination of payoffs, and choice among the candidate strategies. It also shows how mean dynamic and local diffusion approximations are constructed from the transition probabilities of the Markov chain, and constructs the former for the latter in a variety of examples. In addition to providing a structured and broadly applicable framework for defining finite-population game dynamics, this presentation demonstrates how the mean dynamics appearing in the interior of Table 1 are derived from the ingredients of revision protocols that label the rows and columns.

2.3 *ABED* plots and monitors

The results of an *ABED* simulation run are presented in two pairs of plots. The upper pair (Plots I-a and I-b in Figure 1) show the time path of the population state, and the lower pair (Plots II-a and II-b in Figure 1) show the time path of each strategy’s expected payoff. The first plot in each pair (labeled “recent history”) shows only recent data from the simulation, while the second (labeled “complete history”) shows all data starting from the beginning of the simulation. The data represented in any plot can be easily exported to a *csv* file at any time, simply by right-clicking on the plot and selecting “Export” on the contextual menu.

The passage of time in the simulation is tracked in the two monitors in the upper right corner of *ABED*’s interface. The first, labeled `ticks`, tallies the number of periods (“ticks” in *NetLogo*) that have elapsed in the current simulation run. The second, labeled `ticks-per-second`, displays the number of periods that elapse per unit of clock time (here taken to be seconds). This number is computed from the parameters concerning the assignment of revision opportunities from Block 2 of the interface (see Section 2.1.2), with the presumption that each agent expects to receive one revision opportunity per unit of clock time.¹⁵ The conversion of ticks to seconds determines the horizontal scaling of the plots: in Plots I-a and II-a, time is measured in milliseconds, and in Plots I-b and II-b it is measured in seconds.

Block 6 of the interface contains the parameters that control plotting. The sliders `plot-every-?-secs` and `duration-of-recent` determine how often plots are updated and how many seconds worth of data are shown in Plots I-a and II-a. The switches `show-recent-history?` and `show-complete-history?` determine whether the left-hand

¹⁵Thus if `use-prob-revision?` is *on*, then `ticks-per-second` = 1 / `prob-revision`, while if `use-prob-revision?` is *off*, then `ticks-per-second` = `n-of-agents` / `n-of-revisions-per-tick`.

and right-hand plots are *on* or *off*. Naturally, turning plotting off speeds up the simulation.

2.4 Using *ABED* for Monte Carlo experiments

Besides running individual simulations using the interface, it is also possible to conduct several runs for different combinations of parameter values in a systematic fashion. This automatic exploration of the parameter space can be easily conducted using *BehaviorSpace* (Wilensky and Shargel, 2002), a software tool integrated within *NetLogo* which greatly facilitates running a model many times, systematically varying the desired parameter values, and keeping a record of the results of each run. Thus, together with *BehaviorSpace*, one can use *ABED* not only for exploratory or illustrative purposes, but also to conduct statistical analyses of various revision protocols for any symmetric 2-player game in finite populations. We explain how to perform Monte Carlo experiments in *ABED* using *BehaviorSpace* in Appendix C.

3. Examples

We now illustrate *ABED*'s features through a series of examples. Files containing the parameter specifications for each example are included with *ABED*'s release and can be easily loaded by clicking on the button labeled [load parameters from file](#).

3.1 “Equilibrium” behavior in finite-population dynamics

(Parameter file: `hawk-dove.csv`.)

As explained above, deterministic evolutionary dynamics are often derived by first defining finite-population stochastic models and then taking large population limits. A rest point x^* of the resulting mean dynamics is a state at which the *expected* motion of the finite-population stochastic processes equal (or converge to) zero. If multiple strategies are used at x^* , and if agents base their decisions on information obtained from limited numbers of matches, then behavior at x^* is not at rest in the finite-population processes. Instead, there is variation in the use of each strategy, with agents switching among the strategies in the support of x^* . Applying results of Stroock and Varadhan (1979), Sandholm (2003) shows that if choice probabilities depend smoothly on the population state, then as the population size N grows large, rescaling deviations from x^* by \sqrt{N} allows one to approximate the population's behavior near x^* by a diffusion with linear drift and constant diffusion coefficient. When local variation does not cause the state to move away

from x^* ,¹⁶ this diffusion converges to a stationary distribution. In this case, the behavior of the original finite-population process when N is sufficiently large, is approximated by a multivariate normal distribution that is centered at x^* , and whose covariance matrix is of order $\frac{1}{N}$.

Using *ABED*, we can determine we can illustrate this random variation near “equilibrium points”, and can determine how well predictions for the diffusion limit work for relatively small populations. We focus on a simple model in which in each discrete time period, agents are randomly matched once to play a Hawk-Dove game, and one agent is randomly chosen to revise her strategy. The revision protocol is simple: the revising agent looks at another (randomly selected) individual and adopts her strategy if and only if she got a payoff higher than the revising agent. Since agents are matched just once, the payoff comparison is very simple: when a Hawk player and Dove player each participate in a match, the Hawk player will have the higher payoff if and only if he encountered a Dove player. It is easy to verify that the mean dynamic of this process at all interior states moves in the direction of the rest point x^* at which the population is evenly split between Hawk and Dove. As we show in [Example B.5](#), the diffusion approximation predicts that for large N and over long time spans, fluctuations of the proportions of agents playing Hawk and Dove around state x^* should be approximately normally distributed with variance $\frac{1}{4N}$.

To specify this model in *ABED*, we choose the following parameter values:¹⁷

Game, population size and initial state		
Game	Initial state	Population size
payoff-matrix = $\begin{bmatrix} 0 & 3 \\ 1 & 2 \end{bmatrix}$	random-initial-condition? = <i>off</i> initial-condition = [10 10]	(n-of-agents \leftarrow 20)

Assignment of revision opportunities	
use-prob-revision? = <i>off</i>	n-of-revisions-per-tick = 1

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>imitative</i> n-of-candidates = 2 consider-imitating-self? = <i>off</i>	complete-matching? = <i>off</i> n-of-trials = 1 self-matching? = <i>off</i>	decision-method = <i>best</i> tie-breaker = <i>stick-uniform</i> prob-mutation = 0

¹⁶When all strategies are used at x^* , this is true when x^* is linearly stable under the mean dynamic. More generally, one need only consider stability under the mean dynamic with respect to directions corresponding to switches among strategies in the support of x^* , since near x^* there is no stochastic variation in the use of strategies outside the support of x^* . See [Sandholm \(2003\)](#) for details.

¹⁷We use symbol \leftarrow to denote parameter assignments that are automatically set as a consequence of other parameter values. We use gray text to denote these automatic parameter settings and also the setting of auxiliary parameters—explained in [Appendix A.1](#)—which have little influence on the dynamics of the model in most cases.

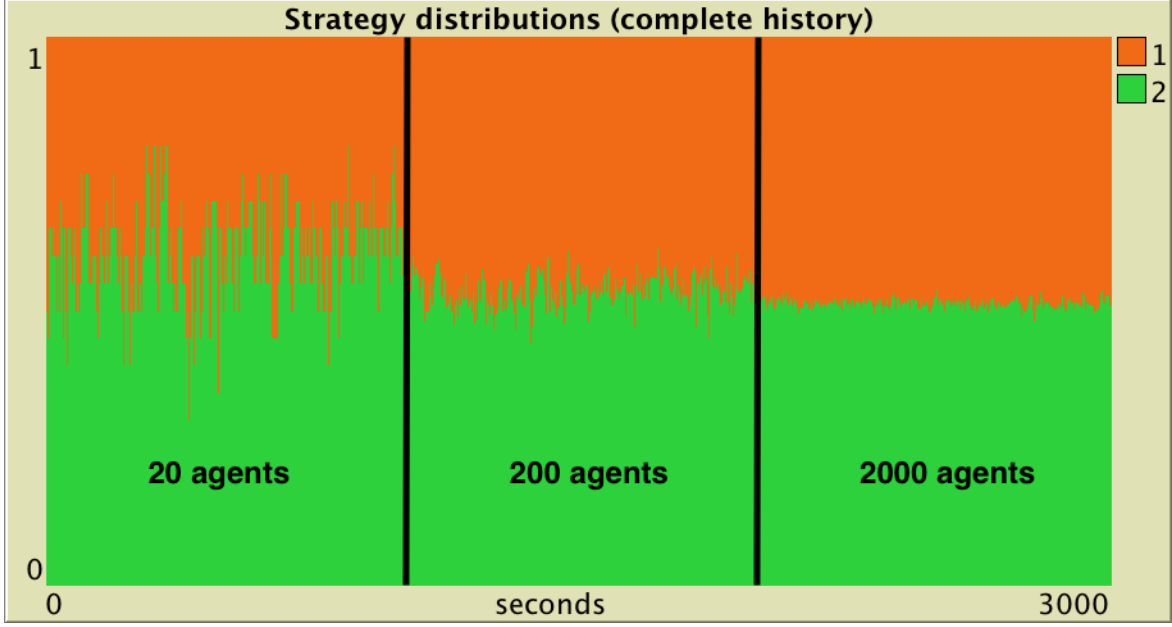


Figure 4: Time series of the strategies' shares in the Hawk-Dove game, where the number of agents is changed from 20 to 200, and then to 2000.

Initially, there are 10 agents playing Hawk and 10 agents playing Dove. This state coincides with a rest point of the mean dynamic, since the expected change in the strategy distribution is zero.¹⁸

Figure 4 shows a simulation run initially set up with the 20 agents. At time $t = 1000$, we set `n-of-agents = 200` (i.e., we increase the number of agents by a factor of 10) and, after 1000 more units of clock time in the simulation, we set `n-of-agents = 2000` (i.e., a factor of 10, again).¹⁹

The following table compares the standard deviations in the use of Hawk and Dove in the simulation to predictions based on the stationary distribution of the local diffusion approximation.

Population size	Empirical SD in the simulation	SD from local diffusion approximation
20	.1131	$\frac{1}{2\sqrt{20}} \approx .1118$
200	.0367	$\frac{1}{2\sqrt{200}} \approx .03536$
2000	.0110	$\frac{1}{2\sqrt{2000}} \approx .01118$

¹⁸The dynamics of this model are analyzed in detail by Izquierdo and Izquierdo (2013, case 5.1), who also provide convergence results of this revision protocol to its mean dynamics in general n -player games.

¹⁹Recall that a unit of clock time is defined in such a way that each agent expects to receive one revision opportunity per unit of clock time (Section 2.3). In this particular example, since `n-of-revisions-per-tick = 1`, one unit of clock time corresponds to `n-of-agents` ticks.

Although the diffusion approximation assumes that the number of agents is large, and the stationary distribution is only guaranteed to describe the empirical distribution of play over long time spans, in this example the theoretical predictions turn out to be accurate for small populations and over moderate time spans.

3.2 Imitation and direct selection in Rock-Paper-Scissors

(Parameter file: `rock-paper-scissors.csv`.)

The fact that parameter values in *ABED* can be modified at runtime makes it particularly easy to explore the impact of changes in modeling assumptions. As an example, we explore the difference between imitative and direct protocols based on pairwise comparisons in the Rock-Paper-Scissors game. We assume here that matching is complete (`complete-matching? = on`), ensuring that agents' experienced payoffs coincide with their expected payoffs. Thus agents' choices are based on exact information about the performances of the strategies they compare, as traditionally assumed in models of deterministic game dynamics.²⁰

The full parameter settings are as follows:

Game, population size and initial state		
Game	Initial state	Population size
$\text{payoff-matrix} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$	<code>random-initial-condition? = off</code> <code>initial-condition = [800 100 100]</code>	(<code>n-of-agents</code> \leftarrow 1000)

Assignment of revision opportunities	
<code>use-prob-revision? = off</code>	<code>n-of-revisions-per-tick = 1</code>

Revision protocol		
Candidate selection	Matching	Decision method
<code>candidate-selection = imitative</code> (<code>n-of-candidates</code> \leftarrow 2) ^a <code>consider-imitating-self? = off</code>	<code>complete-matching? = on</code> <code>self-matching? = on</code> (<code>n-of-trials</code> \leftarrow 1000) ^b	<code>decision-method =</code> <code>= pairwise-difference</code> <code>prob-mutation = 10⁻³</code>

^aValue automatically set because `decision-method = pairwise-difference` and `candidate-selection = imitative`

^bValue automatically set because `complete-matching? = on` and `self-matching? = on`

²⁰Alternatively, one can interpret this setting as describing a “playing the field” model, in which agents' payoffs are determined directly (and linearly) from the proportions of agents playing each strategy, rather than by way of a matching process.

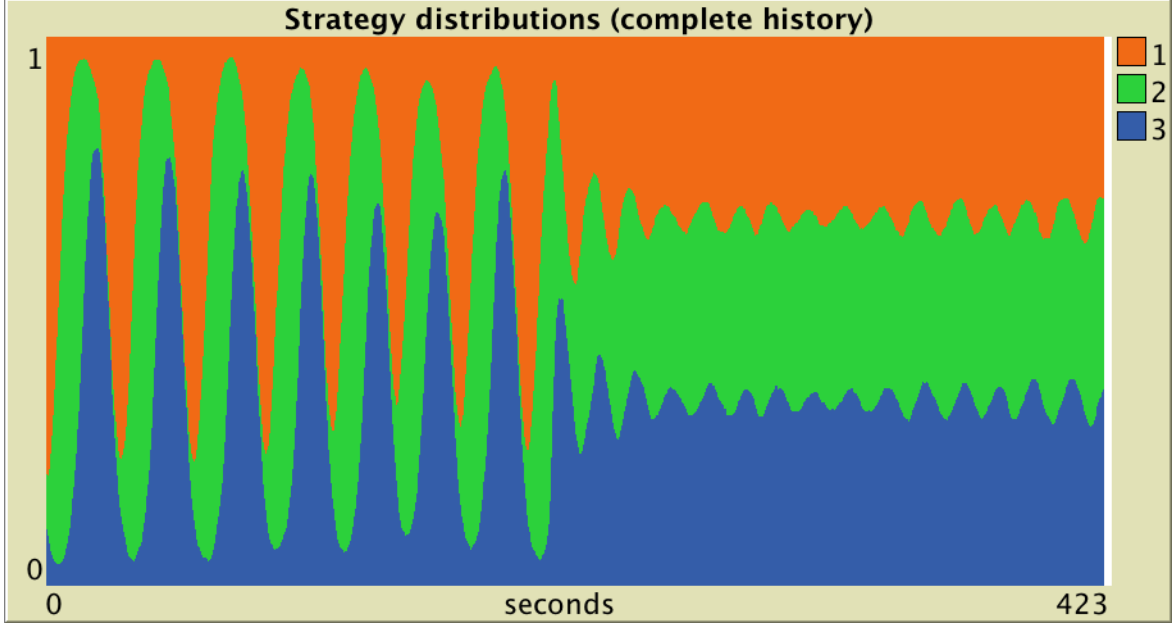


Figure 5: Time series of the strategies' shares in Rock-Paper-Scissors, where candidate-selection is changed from *imitative* to *direct* in the middle of the run.

At the beginning of the simulation, we set `candidate-selection = imitative`, but this setting to *direct* about halfway through the run.

The evolution of the strategies' population shares and payoffs are shown in Figures 5 and 6. While candidate selection is imitative, the population follows rather severe cycles among the three strategies. When candidate selection is switched to direct, the cycles quickly become much smaller in magnitude, but persist indefinitely.

As indicated in Table 1, these parameter settings correspond to changing the mean dynamics from the replicator dynamic (if `candidate-selection = imitative`) to the Smith dynamic (if `candidate-selection = direct`). Under the former mean dynamic, the unique Nash equilibrium at $x^* = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is surrounded by a continuum of closed orbits.²¹ Thus the first portion of Figure 5 exhibits cyclical behavior, but the size of the cycles varies randomly over time. Under the Smith dynamic, the Nash equilibrium is a global attractor, although the equilibrium is approached extremely slowly once the state moves fairly close to x^* .²² This agrees with the second portion of Figure 5, in which the state quickly approaches a region surrounding the equilibrium, and then cycles around the equilibrium at varying small distances.

²¹See Hofbauer et al. (1979) and Zeeman (1980).

²²See Hofbauer and Sandholm (2009). For phase diagrams of the two mean dynamics, see Sandholm (2010b, Figure 5.3) or Sandholm (2015, Figure 13.6).

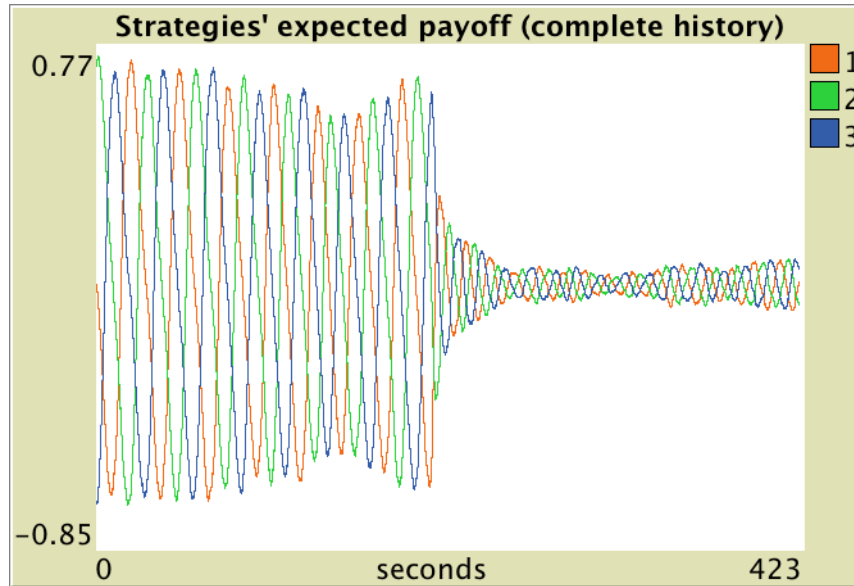


Figure 6: Time series of the strategies’ expected payoffs in Rock-Paper-Scissors, where candidate-selection is changed from *imitative* to *direct* in the middle of the run.

3.3 Mutations and the hypercycle system

(Parameter file: `hypercycle-5.csv`.)

The hypercycle system is a differential equation model introduced by Eigen and Schuster (1979) to model cyclical catalysis in mixtures of polynucleotides. A basic aim in introducing this system was to determine conditions under which a variety of different large organic molecules could coexist in perpetuity, providing the “primordial soup” that allows for the inception of life (see Eigen et al. (1981) and Hofbauer and Sigmund (1988)).

The hypercycle system can be represented as an instance of the replicator dynamic for a game with a simple cyclical payoff structure—specifically, one in which each strategy yields a payoff of 1 against the preceding strategy (modulo n) and a payoff of 0 against the others.²³ Schuster et al. (1978) and Hofbauer et al. (1981) show that when $n \leq 4$, all interior solutions converge to the game’s unique Nash equilibrium x^* , which puts equal mass on each strategy. But when $n \geq 5$, both this Nash equilibrium and the boundary of the simplex are repelling. In the latter case, Hofbauer et al. (1991) prove the existence of a stable periodic orbit. Thus as long as the number of molecules in the catalytic chain is not too small, perpetual cyclical catalysis is possible.

Here we use *ABED* to simulate a finite-population imitative process whose mean dynamic is the hypercycle system with $n = 5$. We use the following parameter settings:

²³This observation is from Schuster and Sigmund (1983), the source of the term “replicator dynamic”.

Game, population size and initial state	
Game	Initial state
payoff-matrix =	random-initial-condition? = <i>off</i>
	initial-condition = [200 200 200 200 200]
	Population size
	(n-of-agents \leftarrow 1000)

Assignment of revision opportunities	
use-prob-revision? = <i>off</i>	n-of-revisions-per-tick = 10

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>imitative</i> (n-of-candidates \leftarrow 2) ^a consider-imitating-self? = <i>off</i>	complete-matching? = <i>on</i> self-matching? = <i>off</i> (n-of-trials \leftarrow 999) ^b	decision-method = = <i>pairwise-difference</i> prob-mutation = 10^{-3}

^aValue automatically set because decision-method = *pairwise-difference* and candidate-selection = *imitative*

^bValue automatically set because complete-matching? = *on* and self-matching? = *off*

The choice of a positive mutation rate (prob-mutation = 10^{-3}) serves two roles here. First, since the population size is finite, each strategy has a significant chance of becoming extinct at the trough of the cycle, and since candidate-selection = *imitative*, a strategy that became extinct would remain permanently so without mutations. Thus including mutations permits extinct strategies to be resurrected, allowing cyclical reinforcement to proceed.²⁴ Second, mutations are needed to leave the initial state where the population is equally split among the five strategies. At this (repelling) initial state no switches between strategies occur through the basic revision process, but a single mutation may be sufficient to drive the process away.

The mean dynamic corresponding to these parameter settings is the replicator dynamic (Examples B.1 and B.2) plus a slight inward perturbation toward due to the presence of mutations. A typical simulation run for a population of size 1000 is shown in Figure 7. Initially, the only changes in strategy are due to mutations. Once this random variation leads to an above-average number of agents playing strategy i , a slight payoff advantage accrues to strategy $i + 1$, so that a revising agent who observes an opponent playing strategy $i + 1$ will imitate this opponent with positive probability. Thus the number of

²⁴In contrast, no strategy initially in use can become extinct in finite time under any continuous mean dynamic.

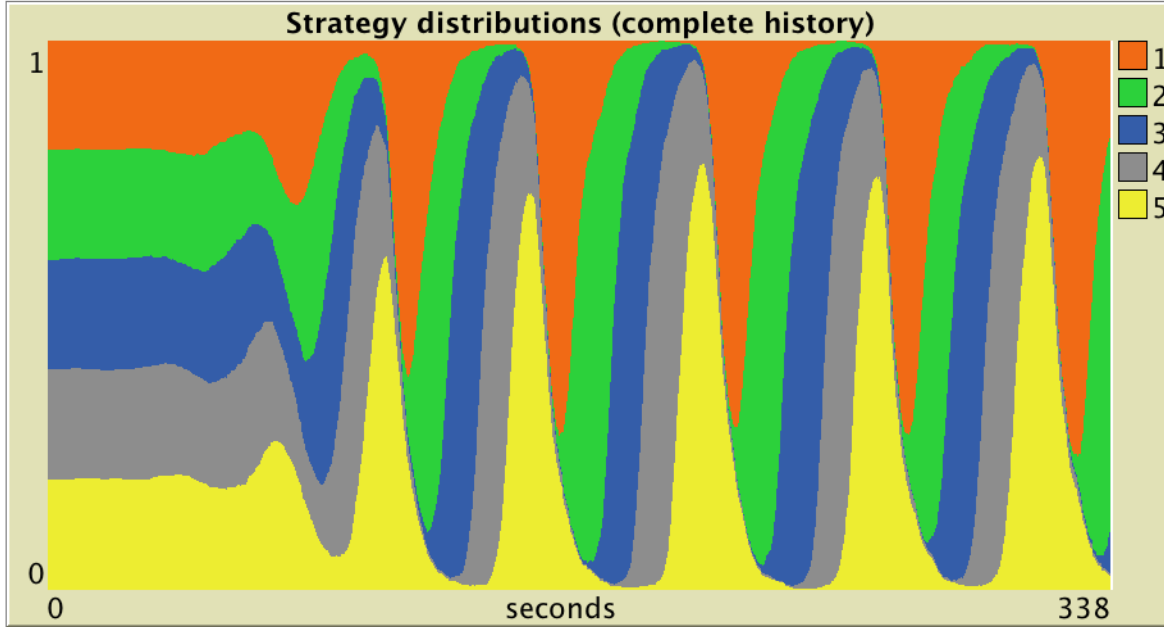


Figure 7: Time series of the strategies' shares in a 5-strategy hypercycle game.

agents playing strategy $i + 1$ grows, shifting the payoff advantage to strategy $i + 2$. The figure shows that these initially small fluctuations grow in magnitude, until each strategy is played by approximately 80% of the population at its peak.

3.4 “Chaos” in finite-population game dynamics

(Parameter file: `chaos.csv`.)

Solutions to smooth differential equations on the unit interval must converge to a limit point; in two dimensions, convergence to closed orbits or chains of rest points and connecting orbits are also possible. In higher dimensions, limiting behavior can be far more complex, with limiting sets taking the form of chaotic attractors. The work of Arneodo et al. (1980) (on chaos in Lotka-Volterra systems) and Hofbauer (1981) (on the equivalence between Lotka-Volterra and replicator systems) established that chaotic dynamics are possible under the replicator dynamics in the simplest feasible case of four strategy games.²⁵

To illustrate chaotic behavior using *ABED*, we consider the following specification based on a game due to Arneodo et al. (1980) and Skyrms (1992):

²⁵See also Schnabl et al. (1991), Skyrms (1992), and Sato et al. (2002).

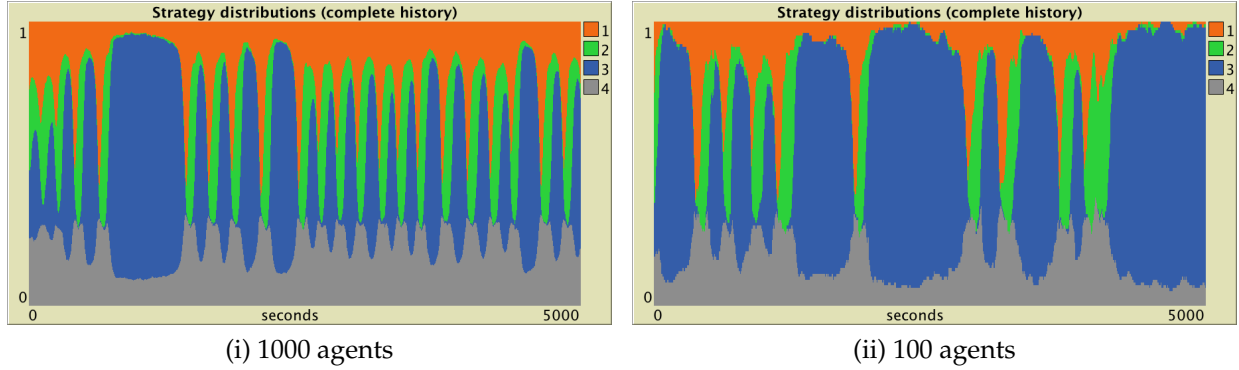


Figure 8: Time series of the strategies' shares in a game for which the replicator dynamic is chaotic.

Game, population size and initial state	
Game	Population size
payoff-matrix = $\begin{bmatrix} 0 & -12 & 0 & -22 \\ 20 & 0 & 0 & -10 \\ -21 & -4 & 0 & 35 \\ 10 & -2 & 2 & 0 \end{bmatrix}$	n-of-agents = 1000
	Initial state
	random-initial-condition? = on

Assignment of revision opportunities	
use-prob-revision? = on	prob-revision = 0.01

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>imitative</i> (n-of-candidates $\leftarrow 2$) ^a consider-imitating-self? = <i>off</i>	complete-matching? = <i>on</i> self-matching? = <i>on</i> (n-of-trials $\leftarrow 1000$) ^b	decision-method = = <i>pairwise-difference</i> prob-mutation = 10^{-3}

^aValue automatically set because decision-method = *pairwise-difference* and candidate-selection = *imitative*

^bValue automatically set because complete-matching? = *on* and self-matching? = *on*

To mimic the replicator dynamic in ABED-1pop, we again set candidate-selection = *imitative*, decision-method = *pairwise-difference* and complete-matching? = *on* (see Examples B.1 and B.2). To avoid the extinction of strategies, we also add a little bit of noise, i.e. prob-mutation = 10^{-3} . Figure 8 shows a representative run of this setting with 1000 agents and another run with 100 agents.

Strictly speaking, the stochastic finite-population dynamics implemented in ABED cannot be chaotic since they are Markov chains on a finite state space. Nonetheless, Figure 8(i) illustrates that the irregular fluctuations characteristic of continuous chaotic

dynamics persist in finite population settings, even in populations as small as 100 agents (Figure 8(ii)).

3.5 Random-walk tie breaking and setwise stability

(Parameter file: `random-walk-tie-breaking.csv`.)

In evolutionary game models in which agents optimize exactly, one must specify what they do when choosing among multiple optimal strategies. Most stochastic models posit that the evolutionary process is Markov with respect to the population state. But when the population is large, large numbers arguments then imply that the proportions of revising agents choosing each of the optimal strategies are fixed, resulting in motion in a nearly deterministic direction. In other words, the usual Markov assumption combined with large population sizes create strong implicit restrictions on adjustments at states with multiple best responses.

These restrictions rule out the possibility that the population state wanders around a component of Nash equilibria. This is important because wandering around equilibrium components plays an key role in evolutionary models of signalling and cheap talk (Kim and Sobel (1995)). It also underpins generalizations of the ESS concept of Maynard Smith and Price (1973) to sets of Nash equilibrium states, as defined by Thomas (1985) and Balkenborg and Schlag (2001). van Veelen (2012) characterizes the sets of states that are evolutionarily stable as defined by these authors as those that are *robust against indirect invasions* (RAII). Roughly speaking, the latter definition requires a set of states to be mutually accessible through sequences of neutral mutations, but not to be susceptible to mutations that lead out of the set and that outperform the incumbents in the post-entry state.²⁶

To allow for wandering within a Nash equilibrium component, *ABED* includes the option of random-walk tie breaking when `decision-method = best`. As explained in Appendix B.4, this option initiates an auxiliary random walk that runs on a discrete set of interior population states and whose stationary distribution is uniform on this set. The value of this auxiliary variable determines the relative probabilities with which currently optimal strategies are chosen. The resulting persistence in tie-breaking probabilities allows the population state to explore Nash equilibrium components, and in particular to find equilibrium states from which escape from the component is possible.

As an illustration, we consider the following game of van Veelen (2012, Ex. 4):

²⁶For an appealing application to repeated games, see García and van Veelen (2016).

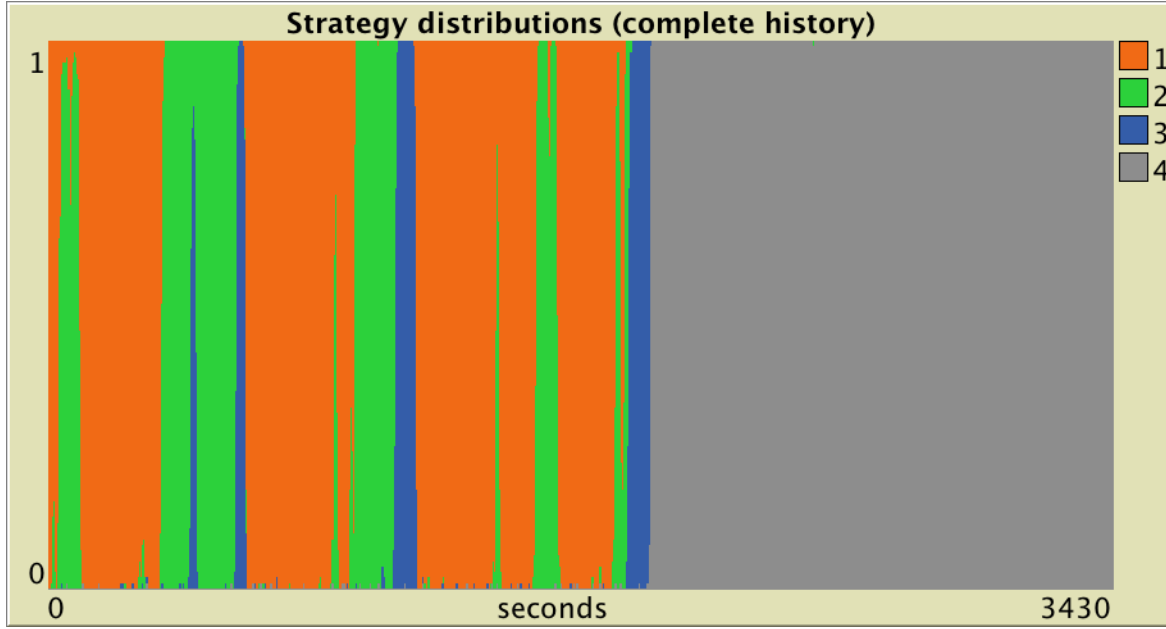


Figure 9: Wandering out of a Nash component in van Veelen's game.

$$(1) \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

In this game, any state in which only strategies 1 and 2 are used are neutrally stable (Maynard Smith (1982)), meaning that mutants playing alternate best responses do not perform strictly better in the post-entry population. If the population reaches the state at which only population 2 is used, then strategy 3 can invade as a neutral mutant. If the population then wanders among states at which only strategies 2 and 3 are used, it may reach the state at which only strategy 3 is used. There strategy 4 is also a best response, and a mutation introducing this strategy can quickly lead to its taking over the population. Indeed, only the state at which only strategy 4 is used is robust against equilibrium entrants, implying that the set consisting of this state is the unique evolutionarily stable set.

To simulate this process of indirect invasions, we run *ABED* in game *A* using the following parameter values:

Game, population size and initial state		
Game	Initial state	
payoff-matrix =	[[1 1 1 1]	random-initial-condition? = <i>off</i>
	[1 1 1 1]	initial-condition = [100 0 0 0]
	[0 1 1 1]	Population size
	[0 0 1 2]]	(n-of-agents \leftarrow 100)

Assignment of revision opportunities	
use-prob-revision? = <i>off</i>	n-of-revisions-per-tick = 1

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>imitative</i>	complete-matching? = <i>on</i>	decision-method = <i>best</i>
n-of-candidates = 2	self-matching? = <i>off</i>	tie-breaker = <i>random-walk</i>
consider-imitating-self? = <i>off</i>	(n-of-trials \leftarrow 99) ^a	prob-mutation = 10 ⁻³

^aValue automatically set because complete-matching? = *on* and self-matching? = *off*

Under the revision protocol specified above, a revising agent observes a single randomly-chosen opponent (since candidate-selection = *imitative* and n-of-candidates = 2). The agent adopts whichever strategy generates a greater expected payoff, resolving ties using the *random-walk* tie-breaker. Because the protocol is imitative, strategies that go extinct can only be reintroduced via mutations, allowing the stepping-stone path described in the previous paragraph to be traversed. The simulation run presented in Figure 9 shows that this happens within a reasonable time span.

The sequence of events required to go from all strategy 1 to all strategy 4 is also possible with other tie-breaking rules. But without persistence in tie-breaking probabilities, the sequence is unlikely to be observed in a reasonable amount of time unless the population size is small (see Figure 10).

3.6 Frequency-dependent Moran processes

The frequency-dependent Moran process (Nowak et al. (2004); Taylor et al. (2004)) is a finite-population stochastic process named for Moran (1958, 1962) and commonly studied in the biology literature.²⁷ During each time step of this process, one individual is randomly selected to be cloned, with probability proportional to payoffs—which are computed assuming that every individual interacts with every other individual. The

²⁷See Nowak (2006) for references.

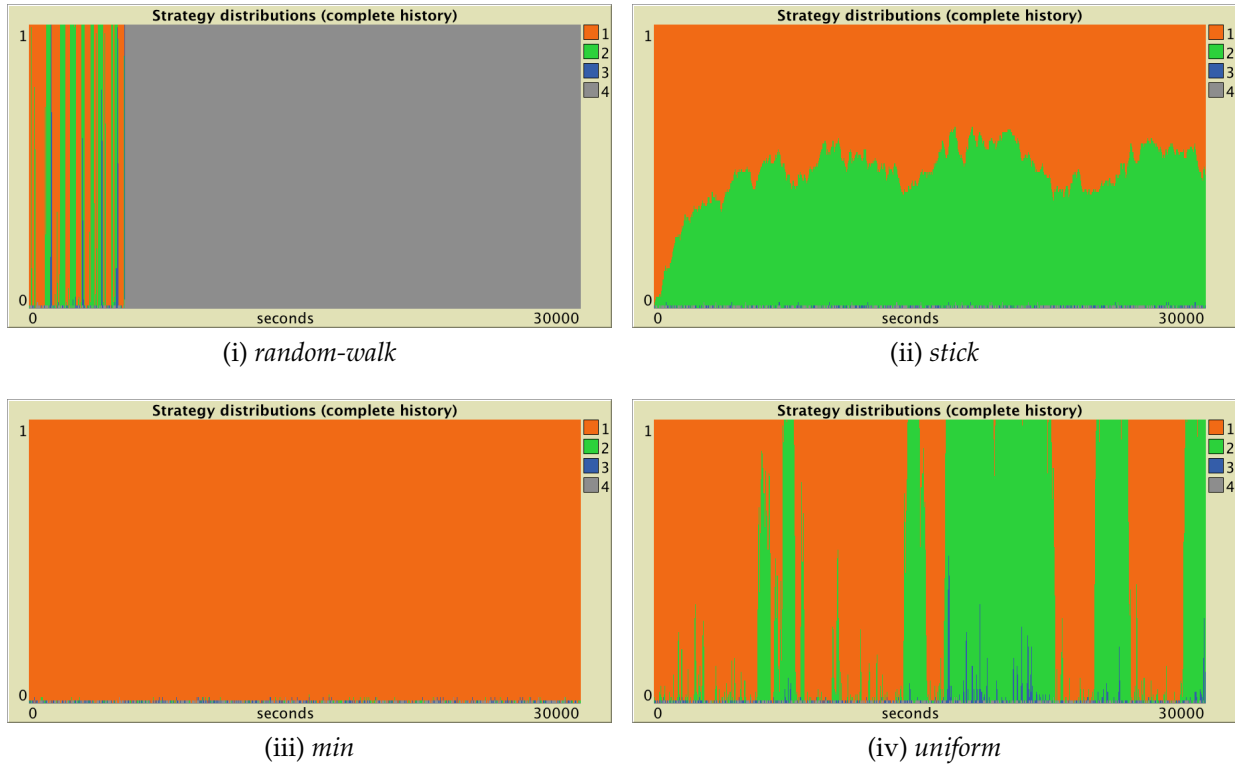


Figure 10: Evolution of the population state in van Veelen's game under four tie-breaking rules (see Appendix A.2).

newborn replaces another randomly selected individual, thus keeping the population size constant.

Such processes can be modeled in ABED-1pop by setting the following parameter values:

Assignment of revision opportunities		
use-prob-revision? = <i>off</i>		n-of-revisions-per-tick = 1

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>imitative</i> n-of-candidates = n-of-agents imitatees-with-replacement? = <i>off</i> consider-imitating-self? = <i>off</i>	complete-matching? = <i>on</i> self-matching? = <i>off</i>	decision-method = = <i>positive-proportional</i>

The mean dynamic of the frequency-dependent Moran process is the Maynard Smith replicator dynamics (Maynard Smith (1982)), which in one-population settings differs from the classical replicator dynamics only by a speed factor (Sandholm, 2010b, p. 156 and 175).²⁸ However, most analyses of the frequency-dependent Moran process do not focus on its mean dynamic (i.e. the limit of the process when $N \rightarrow \infty$ over finite time spans), but on its infinite-horizon dynamics (i.e. the limit of the process when $t \rightarrow \infty$, for N fixed), and they also include some noise (i.e. $\eta > 0$). The inclusion of noise η makes the underlying Markov chain irreducible and aperiodic, which implies that its unique stationary distribution has full support (i.e., the probability of finding the process in any particular state in the long run is strictly positive). Because of this, a full characterization of the stationary distribution would be cumbersome, so most papers focus on the extreme cases where $\eta \rightarrow 0$ and/or $N \rightarrow \infty$ (after taking $t \rightarrow \infty$), which tend to lead to distributions concentrated on only a few homogeneous states (or even just one state) where the whole population is using the same strategy (Binmore and Samuelson (1997), Fudenberg et al. (2006), and Fudenberg and Imhof (2006, 2008)).

Since most analyses of Moran processes focus on extreme cases where one or more limits are taken, they do not offer insights about the dynamics of any particular finite system parameterized with concrete values.²⁹ ABED can be used to (numerically) characterize these dynamics, and thus be a great complement to mathematical analyses.

²⁸In multi-population games, the two dynamics will generally have different stability properties, since the speed factor varies across populations.

²⁹For an exception, see Imhof et al. (2005). Games analyzed in this paper can be simulated in ABED-1pop using parameter files Imhof-Fudenberg-Nowak-PNAS-2005-Fig1.csv and Imhof-Fudenberg-Nowak-PNAS-2005-Fig2.csv, included in the release.

3.7 Exploratory analysis: Cooperative play in Centipede

(Parameter file: centipede-10-nodes-sampleBR-min.csv.)

To illustrate how stochastic evolutionary dynamics in two-player asymmetric games can be easily modeled with ABED-2pop, we consider the Centipede game of Rosenthal (1981). In Rosenthal's Centipede game there are 10 nodes. Player 1 decides at odd nodes whilst player 2 decides at even nodes. Players' choices at each node are called Down (i.e. stop the game) and Right (i.e. continue). The payoffs obtained by each player at every possible outcome are shown in Figure 11. For each player, strategy i is the plan to continue for $i - 1$ decision nodes and stop at his i th decision node. Strategy 6 for both players is the plan to always continue (i.e. never stop).

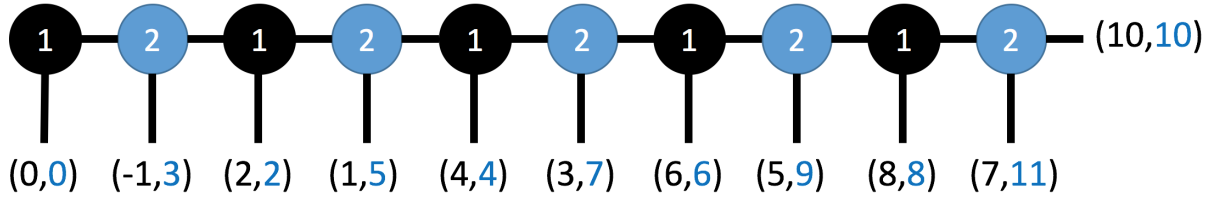


Figure 11: The Centipede game (Rosenthal (1981)).

Here we explore the 1-sample best response protocol (Sandholm (2001)) and the 1-sample best experienced payoff protocol (Sandholm et al. (2017)), breaking ties in favor of the least cooperative strategy in both cases. The only difference between the two revision protocols comes from the value of single-sample? (see Table 1). To model the former protocol, we set the following parameter values:

Game, population size and initial state																			
Game								Population size											
payoff-matrix =	[[[0	0]	[0	0]	[0	0]	[0	0]	[0	0]]		
		[-1	3]	[2	2]	[2	2]	[2	2]	[2	2]]		
		[-1	3]	[1	5]	[4	4]	[4	4]	[4	4]]		
		[-1	3]	[1	5]	[3	7]	[6	6]	[6	6]]		
		[-1	3]	[1	5]	[3	7]	[5	9]	[8	8]]		
		[-1	3]	[1	5]	[3	7]	[5	9]	[7	11]	[10	10]
								pop-1-n-of-agents = 500											
								pop-2-n-of-agents = 500											
								Initial state											
								random-initial-condition? = on											

Assignment of revision opportunities	
use-prob-revision? = off	n-of-revisions-per-tick = 50

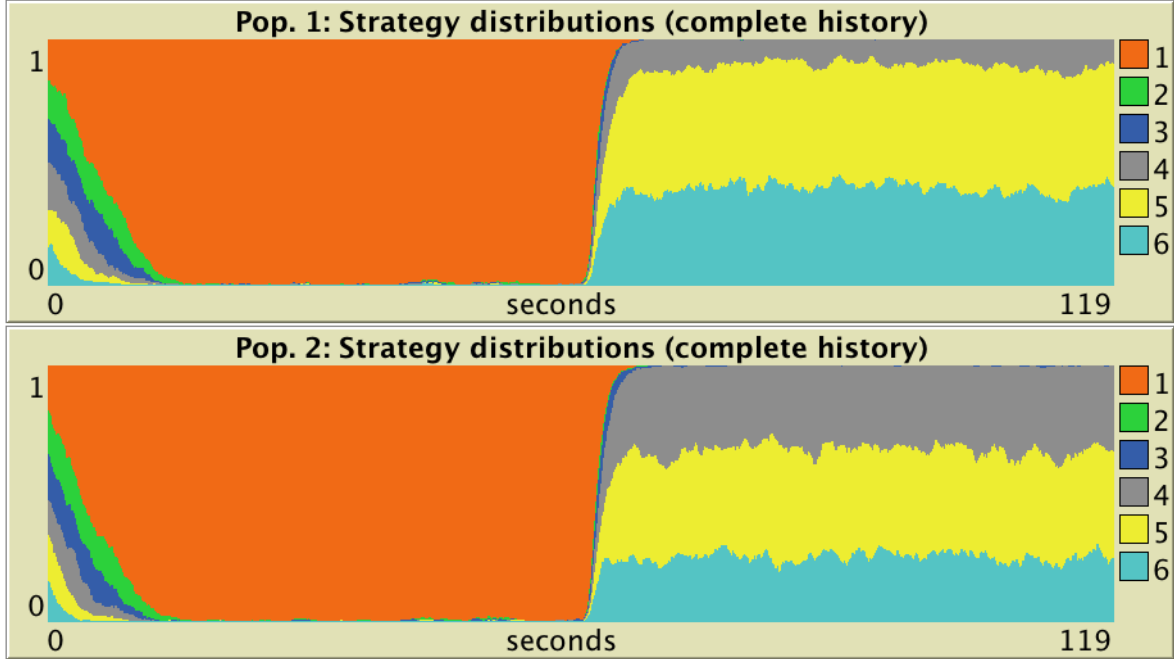


Figure 12: Time series of the strategies' shares in a 10-node Centipede game, where `single-sample?` is changed from *On* to *Off* at the middle of the run.

Revision protocol		
Candidate selection	Matching	Decision method
candidate-selection = <i>direct</i>	complete-matching? = <i>off</i>	decision-method = <i>best</i>
pop-1-n-of-candidates = 6	n-of-trials = 1	tie-breaker = <i>min</i>
pop-2-n-of-candidates = 6	single-sample? = <i>on</i>	prob-mutation = 10^{-3}

Figures 12 and 13 show the time series of strategies' shares and payoffs in a representative run of ABED-2pop where `single-sample?` is initially set to *On*, and it is later changed to *Off* at the middle of the simulation run. It is striking that parameter `single-sample?` can have such a dramatic effect on the level of cooperation in the Centipede game. Thus, this final example is a clear illustration of the fact that the type of strategies that are likely to emerge and be sustained in finite evolutionary contexts can be strongly dependent on assumptions that have not traditionally received much attention.

4. Concluding remarks

Evolutionary game theory provides a powerful set of analytical tools for studying behavior in populations of myopic, strategically interacting agents, tools which have been deployed in wide-ranging applications.³⁰ As with any theory, progress in evolutionary

³⁰For a thorough survey of recent applications, see Newton (2018).

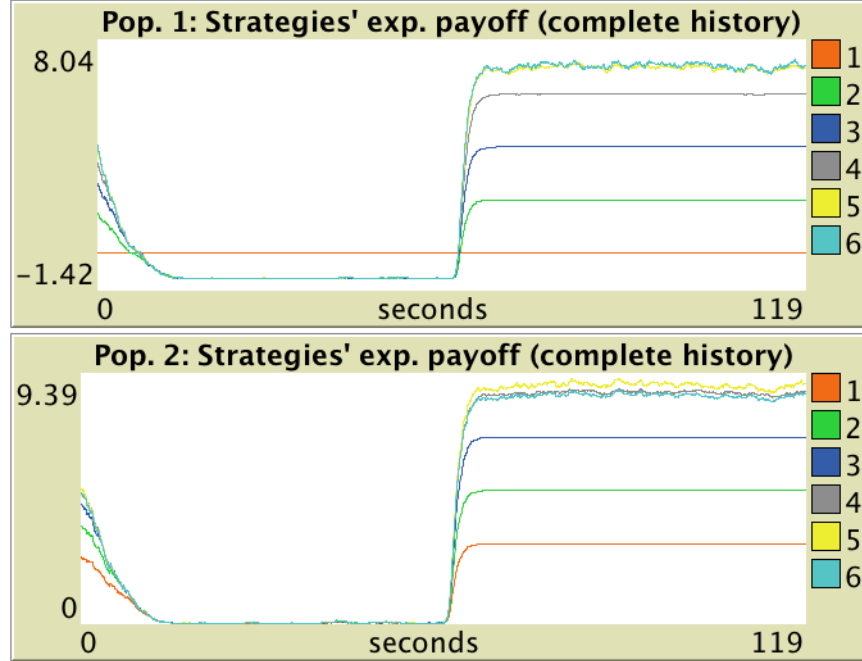


Figure 13: Time series of the strategies' expected payoffs in a 10-node Centipede game, where *single-sample?* is changed from *on* to *off* at the middle of the run.

game theory has relied on assumptions and approximations that make its models analytically tractable. The examples presented above illustrate how *ABED* can be used to see how theoretical predictions stand up in settings where these assumptions are relaxed—for instance, when populations are not especially large, when matching of agents is incomplete, and when agents' information about strategies' performances comes from their own and others' direct experiences. By encompassing a great variety of specifications of revision processes within a single framework, *ABED* not only facilitates comparisons between different models from the literature, but also provides powerful tools for exploring the implications of new specifications of agent's choice processes.

Because *ABED* is thoroughly documented and runs in the user-friendly *NetLogo* platform, it is not overly difficult to add new parameters, new protocols, or more elaborate alterations (including agents heterogeneity) by directly modifying its code. Future work will substantially expand *ABED*'s capabilities, and we gladly accept implementation requests directed towards this end.

Appendix

A. Further details on *ABED* parameters

A.1 Auxiliary parameters

The bottom left corner of *ABED*'s interface (Block 7 in [Figure 1](#)) contains four auxiliary parameters that are only of practical relevance if *n-of-agents* is small. The default setting of these parameters is *off*.

- *self-matching?*: If this switch is *off*, agents are never matched to play the game against themselves. Turning this switch *on* allows self matches.
- *trials-with-replacement?*: If *complete-matching?* is *off*, this switch determines whether the sample(s) of opponents against whom an agent is matched is drawn with or without replacement.
- *imitatees-with-replacement?*: If *candidate-selection* is *imitative* and this switch is *off*, the set of (*n-of-candidates* – 1) agents that the revising agent considers imitating is drawn without replacement. If it is *on*, draws are with replacement.
- *consider-imitating-self?*: If *candidate-selection* is *imitative*, turning this switch *on* leads the revising agent to consider himself as a possible imitator.³¹

A.2 Tie-breakers for decision-method *best*

The tie-breakers available when the decision-method is *best* are as follows:

- *uniform*: randomize uniformly among the highest-payoff strategies.
- *stick-uniform*: stick with the current strategy if it provides the highest payoff; otherwise randomize uniformly among the highest-payoff strategies.
- *min*: choose the highest-payoff strategy with the smallest index number (under the natural ordering).
- *stick-min*: stick with the current strategy if it is among the highest-payoff ones; otherwise use the *min* tie-breaker.

³¹The *n-of-candidates* agents whose strategies are candidates always includes the revising agent himself, exactly once if *consider-imitating-self?* is *off*, and possibly more than once if it is *on*.

- *random-walk*: randomize among the highest-payoff strategies with probabilities that are determined by an auxiliary random walk. See Appendix B.4 for details.

B. A framework for finite-population evolutionary game dynamics

In this appendix, we present a framework for specifying finite-population evolutionary game dynamics and use it to formally describe the Markov chains simulated in *ABED*. The main novelty is a general definition of revision protocols that decomposes them into three stages: candidate selection, matching and determination of payoffs, and choices of new strategies (Section B.1). The description of Markov chains covers a range of parameter choices wide enough to allow all of the combinations presented in Table 1, but for simplicity certain parameter choices are assumed fixed. We also review results on deterministic (Section B.2) and diffusion (Section B.3) approximations for these Markov chains, and exhibits examples of each. Finally, Section B.4 presents the random-walk tie-breaker introduced in Example 3.5.

B.1 The Markov chain

When the agents in the population play a game with strategy set $S = \{1, \dots, n\}$, strategy distributions are elements of the simplex $X = \{x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\}$, whose i th vertex is denoted e_i . If the population is of size N (= n-of-agents), the population state is an element of the grid $\mathcal{X}^N = \{x \in X : Nx_i \in \mathbb{Z} \text{ for all } i\}$.

We define the Markov chain as a discrete-time process on the set of population states \mathcal{X}^N .³² We assume during each period of duration $\frac{1}{N}$, exactly one agent receives a revision opportunity (i.e., use-prob-revision? = off and n-of-revisions-per-tick = 1). The transition law of the Markov chain $\{X_t^N\}$ is then described by

$$(2) \quad \mathbb{P}\left(X_{k+1/N}^N = x + \frac{1}{N}(e_j - e_i) \mid X_k^N = x\right) = x_i \rho_{ij}^A(x).$$

For an i player to switch to strategy j during the next period, the player who receives the next revision opportunity must be an i player, which happens with probability x_i , and he must switch to strategy j , which happens with probability $\rho_{ij}^A(x)$. The *revision protocol* ρ captures both the information that agents have when revising due to matching, and the decision rule they employ. Choice probabilities under the revision protocol depend on the game $A \in \mathbb{R}^{n \times n}$ (= payoff-matrix) being played and the population state x . In general

³²This description does not correspond to the agent-based specification of the process within *ABED*'s code.

choice probabilities in *ABED* can depend directly on the population size N , but in what follows we will focus on parameter values that preclude such dependence.

We first provide a general formulation (3) of revision protocols, and then explain each component of (3) in detail.

Let c (= n-of-candidates) denote the number of observations about strategies' performances that a revising agent obtains.³³ The total information a revising agent obtains is described by a *record* (s, π) , a $c \times 2$ matrix whose first column s is the *strategy record* and whose second column π is the *payoff record*. The h th row of the record is a pair (s_h, π_h) that describes the strategy played and the payoff obtained in the revising agent's h th observation.

Revision protocols implemented in *ABED* can be expressed in the form

$$(3) \quad \rho_{ij}^A(x) = \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi).$$

The summand in (3) expresses the probability that a revising strategy i player obtains record (s, π) and switches to strategy j . Summing this over all payoff records yields the probability that a revising strategy i player switches to j .

The summand in (3) is the product of three terms. The first, $p_i(s | x)$, describes candidate selection. It is the probability that a revising agent playing strategy i obtains strategy record s . This probability will depend on the population state x when the protocol is imitative, but not when it is direct.

The second term, $q(\pi | s, x, A)$ describes matching and sampling. It is the probability that a revising agent with strategy record s obtains corresponding payoff record π . This probability depends on the population state x , and the payoff matrix A .

The third term, $\sigma_{ij}(s, \pi)$, describes the decision method. It is the probability that a revising strategy i player with record (s, π) chooses strategy j . This probability only depends on x and A by way of the record (s, π) .³⁴

Candidate selection and strategy records

In constructing a strategy record $s = \{s_h\}_{h=1}^c$ for a revising agent playing strategy i , we let $s_1 = i$ be the revising agent's strategy, and view subsequent strategies as being added sequentially.³⁵

³³Here a "performance" is an average payoff generated by the matching procedure in place.

³⁴A minor exception is noted below.

³⁵The ordering of the pairs (s_h, π_h) is irrelevant in subsequent parts of the protocol. One could alternatively express revision protocols directly in terms of classes of records that are equivalent up to permutation. Then instead of there being distinct summands for equivalent records (see Examples B.3 and B.4), larger combinatorial terms would appear in the function p . On balance the approach we follow here is notationally

Under a *direct protocol*, strategies s_2, \dots, s_c are drawn without replacement from $S \setminus \{i\}$. Thus the probability of any strategy record with $s_1 = i$ and all strategies distinct is the same, and equal to the inverse of the number of such records:

$$(4) \quad p_i(s|x) = \frac{(n-c)!}{(n-1)!}.$$

Thus when $c = 2$, the $n - 1$ feasible strategy records are of the form (i, j) with $j \neq i$, and each has probability $\frac{1}{n-1}$. When $c = n$, the $(n - 1)!$ feasible strategy records start with i and contain all n strategies, and each has probability $\frac{1}{(n-1)!}$.

How an *imitative* protocol and the population state x determine the probabilities of strategy lists depends on the values of the auxiliary parameters *imitatees-with-replacement?* and *consider-imitating-self?*. If both are *on*, then the probability of a strategy record s with $s_1 = i$ is

$$(5) \quad p_i(s|x) = \prod_{h=2}^c x_{s_h}.$$

If both parameters are *off* and $c = N$, then every strategy record $s \in S^N$ with $s_1 = i$ and with Nx_j entries equal to j for all $j \in S$ has equal probability:

$$(6) \quad p_i(s|x) = \frac{(Nx_i - 1)! \prod_{k \neq i} (Nx_k)!}{(N - 1)!}.$$

Matching and sampling parameters and payoff records

Given a strategy record $s = \{s_h\}_{h=1}^c$, a population state x , and the payoff matrix A , the matching and sampling parameters determine the probabilities of each payoff record $\pi = \{\pi_h\}_{h=1}^c$. To rule out finite-population effects we assume here that *self-matching?* is *on*.

If matching is complete, then each agent gets the expected payoff of the strategy he plays:

$$(7) \quad q(((Ax)_{s_1}, \dots, (Ax)_{s_c}) | s, x, A) = 1.$$

For cases where matching is not complete, we assume for simplicity that *n-of-trials* = 1. If the protocol is *imitative*, or if it is *direct* and *single-sample?* is *off*, then the payoff associated with each strategy in the record is determined by a distinct random match. If

simpler.

we assume for convenience that elements within each row of A are distinct,³⁶ then

$$(8) \quad q((A_{s_1 t_1}, \dots, A_{s_c t_c}) | s, x, A) = \prod_{h=1}^c x_{t_h}.$$

If instead the protocol is direct and single-sample? is *on*, then all strategies are evaluated by taking expectations with respect to the empirical distribution of the single sample. The case with `n-of-trials = 1` is very simple:³⁷ if A has distinct rows, then

$$(9) \quad q((A_{s_1 k}, \dots, A_{s_c k}) | s, x, A) = x_k.$$

Decision methods

Each decision method defines a collection of functions σ_{ij} , where $\sigma_{ij}(s, \pi)$ is the probability that a revising i player with record (s, π) chooses strategy j . Under `decision-method = best`,

$$(10) \quad \text{if } \left[\pi_h = \max_{h'} \pi_{h'} \Rightarrow s_h = j \right], \text{ then } \sigma_{ij}(s, \pi) = 1.$$

If the antecedent condition fails (i.e., if the maximum payoff is achieved by more than one strategy in the record), then choice probabilities are determined by the value of `tie-breaker`.³⁸

Under `decision-method = logit`,

$$(11) \quad \sigma_{ij}(s, \pi) = \frac{\sum_{h: s_h=j} \exp(\eta^{-1} \pi_h)}{\sum_{k \in S} \sum_{h: s_h=k} \exp(\eta^{-1} \pi_h)},$$

where the noise level η is set using *ABED* parameter `log-noise-level = log10 η`. Similarly, `decision-method = positive-proportional` is defined for positive payoff records π by

$$(12) \quad \sigma_{ij}(s, \pi) = \frac{\sum_{h: s_h=j} \pi_h}{\sum_{k \in S} \sum_{h: s_h=k} \pi_h}.$$

If `decision-method = pairwise-difference`, then $c = 2$, and $s_1 = i$ is the revising agent's strategy. If $s_2 = i$, then $\sigma_{ii}(s, \pi) = 1$. Otherwise,

³⁶If rows may have repeated elements, we must account for the fact that the payoff value obtained by an agent playing strategy j does not uniquely determine the strategy of his match partner. See [Example B.2](#).

³⁷For an example with larger numbers of trials, see Oyama et al. (2015).

³⁸See [Appendix A.2](#). Some of these condition on the revising agent's strategy i .

$$(13) \quad \sigma_{ij}((i, \pi_1), (j, \pi_2)) = \frac{[\pi_2 - \pi_1]_+}{\bar{\Delta}A} \text{ and } \sigma_{ii}((i, \pi_1), (j, \pi_2)) = 1 - \frac{[\pi_2 - \pi_1]_+}{\bar{\Delta}A}.$$

where the constant $\bar{\Delta}A$ is the maximum difference between entries of the payoff matrix A .

B.2 Finite-horizon deterministic approximation: the mean dynamic

The mean dynamic of the Markov chain described by (2) is

$$(14) \quad \dot{x}_i = V_i^{\rho, A}(x) = \sum_{j \in S} x_j \rho_{ji}^A(x) - x_i.$$

Results of Benaïm and Weibull (2003) (for Lipschitz continuous dynamics) and Roth and Sandholm (2013) (for differential inclusions) imply that over finite time spans, sample paths are very likely to be closely approximated by solutions to (14) if the population size N is large enough. Next, we present four revision protocols and their mean dynamic.

Example B.1. The replicator dynamic from complete matching. Suppose candidate selection is imitative as in (5) with $c = 2$, matching is complete (7), and the decision method is *pairwise-difference* (13). To derive the mean dynamics for protocols using *pairwise-difference*, it is convenient to express the formula (14) for the mean dynamic without the diagonal terms ρ_{ii} of the revision protocol:

$$(15) \quad \dot{x}_i = V_i^{\rho, A}(x) = \sum_{j \neq i} (x_j \rho_{ji}^A(x) - x_i \rho_{ij}^A(x)).$$

The off-diagonal terms of the revision protocol are

$$\begin{aligned} \rho_{ij}^A(x) &= \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi) \\ &= x_j \times 1 \times \sigma_{ij}((i, (Ax)_i), (j, (Ax)_j)) \\ &= x_j \frac{[(Ax)_j - (Ax)_i]_+}{\bar{\Delta}A}, \end{aligned}$$

where the second equality uses the fact that the only strategy record that allows a revising i player to switch to j is $(s_1, s_2) = (i, j)$. Using (15), we compute the mean dynamic as

$$\begin{aligned} \dot{x}_i &= \sum_{j \neq i} (x_j \rho_{ji}^A(x) - x_i \rho_{ij}^A(x)) \\ &= \sum_{j \neq i} \left(x_j \left(x_i \frac{[(Ax)_i - (Ax)_j]_+}{\bar{\Delta}A} \right) - x_i \left(x_j \frac{[(Ax)_j - (Ax)_i]_+}{\bar{\Delta}A} \right) \right) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\bar{\Delta}A} x_i \sum_{j \neq i} x_j ((Ax)_i - (Ax)_j) \\
&= \frac{1}{\bar{\Delta}A} x_i \sum_{j \in S} x_j ((Ax)_i - (Ax)_j) \\
&= \frac{1}{\bar{\Delta}A} x_i ((Ax)_i - x'Ax).
\end{aligned}$$

This is the replicator dynamic, sped up by a constant factor of $\frac{1}{\bar{\Delta}A}$. ♦

Example B.2. The replicator dynamic from single matches. Repeat the setup from [Example B.1](#), but with complete matching replaced by limited matching with `n-of-trials` = 1, as in [\(8\)](#). Then for $j \neq i$,

$$\begin{aligned}
\rho_{ij}^A(x) &= \sum_{(s,\pi)} p_i(s|x) q(\pi|s,x,A) \sigma_{ij}(s,\pi) \\
&= \sum_{k \in S} \sum_{\ell \in S} x_j \times x_k x_\ell \times \sigma_{ij}((i, A_{ik}), (j, A_{j\ell})) \\
&= x_j \sum_{k \in S} \sum_{\ell \in S} x_k x_\ell \frac{[A_{j\ell} - A_{ik}]_+}{\bar{\Delta}A},
\end{aligned}$$

where the sums are over the strategies of the match partners of the revising i player and the j player that he observes. The mean dynamic is

$$\begin{aligned}
\dot{x}_i &= \sum_{j \neq i} (x_j \rho_{ji}^A(x) - x_i \rho_{ij}^A(x)) \\
&= \sum_{j \neq i} \left(x_j \left(x_i \sum_{k \in S} \sum_{\ell \in S} x_k x_\ell \frac{[A_{ik} - A_{j\ell}]_+}{\bar{\Delta}A} \right) - x_i \left(x_j \sum_{k \in S} \sum_{\ell \in S} x_k x_\ell \frac{[A_{j\ell} - A_{ik}]_+}{\bar{\Delta}A} \right) \right) \\
&= \frac{1}{\bar{\Delta}A} x_i \sum_{j \neq i} x_j \sum_{k \in S} \sum_{\ell \in S} x_k x_\ell (A_{ik} - A_{j\ell}) \\
&= \frac{1}{\bar{\Delta}A} x_i \sum_{j \neq i} x_j ((Ax)_i - (Ax)_j) \\
&= \frac{1}{\bar{\Delta}A} x_i ((Ax)_i - x'Ax).
\end{aligned}$$

Thus the mean dynamic is again the replicator dynamic, as in the previous example. ♦

Example B.3. The Maynard Smith replicator dynamic. Suppose candidate selection is imitative with $c = N$ and without replacement, as in [\(6\)](#), that matching is complete [\(7\)](#), and that the decision method is *positive-proportional* [\(12\)](#) (with the entries of A assumed to

be positive). Letting $S^N(x, i)$ denote the set of strategy records $s \in S^N$ with $s_1 = i$ and $\#\{h: s_h = j\} = Nx_j$ for all $j \in S$, we compute the revision protocol as

$$\begin{aligned}\rho_{ij}^A(x) &= \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi) \\ &= \sum_{s \in S^N(x, i)} \frac{(Nx_i - 1)! \prod_{k \neq i} (Nx_k)!}{(N - 1)!} \times 1 \times \frac{\sum_{h: s_h = j} \pi_h}{\sum_{k \in S} \sum_{h: s_h = k} \pi_h}. \\ &= \frac{x_j (Ax)_j}{\sum_{k \in S} x_k (Ax)_k},\end{aligned}$$

since $p_i(s | x)$ is the number of strategy records $s \in S^N$ with $s_1 = i$ and empirical distribution x . Using (14), we compute the mean dynamic:

$$\begin{aligned}\dot{x}_i &= \sum_{j \in S} x_j \rho_{ji}^A(x) - x_i \\ &= \frac{x_i (Ax)_i}{\sum_{k \in S} x_k (Ax)_k} - x_i \\ &= \frac{x_i ((Ax)_i - x' Ax)}{x' Ax}.\end{aligned}$$

This is the Maynard Smith replicator dynamic. ♦

Example B.4. Best experienced payoff dynamics. Suppose candidate selection is direct with $c = n$ as in (4), that matching is based on limited multiple samples with `n-of-trials` = 1, as in (8), and that the decision method is *best*, as in (10). To derive the revision protocol, we introduce the following notations: Let $S^n(i)$ denote the set of strategy records $s \in S^n$ that are a permutation of the elements of $S = \{1, \dots, n\}$ with $s_1 = i$. For $s, t \in S^n$ and payoff matrix A , define the payoff record $\tilde{\pi}(s, t, A) \in \mathbb{R}^n$ by $\tilde{\pi}_h(s, t, A) = A_{s_h t_h}$. Define $s^\circ \in S^n$ by $s_h^\circ = h$ for all h . And let $S_*^n(j, A) = \{t \in S^n: A_{jt_j} > A_{kt_k} \text{ for all } k \in S\}$.

If all entries of A are distinct we can derive the revision protocol as follows:

$$\begin{aligned}\rho_{ij}^A(x) &= \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi) \\ &= \sum_{s \in S^n(i)} \sum_{t \in S^n} \frac{1}{(n - 1)!} \times \prod_{h=1}^n x_{t_h} \times \sigma_{ij}(s, \tilde{\pi}(s, t, A)) \\ &= \sum_{t \in S^n} \prod_{h=1}^n x_{t_h} \sigma_{ij}(s^\circ, \tilde{\pi}(s^\circ, t, A))\end{aligned}$$

$$= \sum_{t \in S_x^n(j, A)} \prod_{h=1}^n x_{t_h}.$$

The third equality uses the invariance of σ_{ij} to the arrangement of the pairs in (s, π) and the fact that the sum is taken over all t in S^n . The mean dynamic that this protocol generates,

$$\dot{x}_i = \sum_{t \in S_x^n(i, A)} \prod_{h=1}^n x_{t_h} - x_i,$$

is an instance of a best experienced payoff dynamic. ♦

B.3 Diffusion approximation near equilibria

Suppose that x^* is a rest point of the mean dynamic (14): $V^{\rho, A}(x^*) = \mathbf{0}$. We define the *local behavior process* near x^* , denoted $\{Z_t^N\}_{t \geq 0}$, by

$$(16) \quad Z_t^N = \sqrt{N}(\bar{X}_t^N - x^*)$$

where $\{\bar{X}_t^N\}_{t \geq 0}$, defined by $\bar{X}_t^N = X_{\lfloor Nt \rfloor / N}^N$ is the piecewise constant interpolation of $\{X_t^N\}_{t \in \{0, 1/N, \dots\}}$, and where $X_0^N \in \mathcal{X}^N$ converges to x^* at rate $o(1/\sqrt{N})$. Let $DV_i^{\rho, A}(x^*)$ denote the Jacobian of $V^{\rho, A}$ evaluated at x^* , and define the local covariance matrix $\Sigma^* \in \mathbb{R}^{n \times n}$ by

$$(17) \quad \Sigma_{ij}^* = \begin{cases} -\left(x_i^* \rho_{ij}^A(x^*) + x_j^* \rho_{ji}^A(x^*)\right) & \text{if } i \neq j, \\ \sum_{k \neq i} \left(x_i^* \rho_{ik}^A(x^*) + x_k^* \rho_{ki}^A(x^*)\right) & \text{if } i = j. \end{cases}$$

Now consider the stochastic differential equation

$$(18) \quad dZ_t = DV_i^{\rho, A}(x^*)Z_t dt + \sqrt{\Sigma^*} dB_t,$$

where $\sqrt{\Sigma^*}$ is the symmetric positive semidefinite square root of Σ^* . Applying results of Stroock and Varadhan (1979), Sandholm (2003) shows that over finite time spans, the local behavior process (16) converges in distribution (with respect to the Skorokhod topology) to the solution to (18) with initial condition $\mathbf{0}$ as the population size N grows large.³⁹

The following example uses the diffusion approximation to describe “equilibrium” behavior in a Hawk-Dove game under the imitative protocol from Section 3.1.

³⁹The result in Sandholm (2003) is stated for sequences of continuous-time processes, but also holds for sequences of discrete-time processes as defined here. Compare Durrett (1996, Sec. 8.7).

Example B.5. Local diffusion approximation for “imitate the best realization” in Hawk-Dove.

Suppose that agents use imitative candidate selection (5) with $c = 2$, that matching is limited matching with `n-of-trials` = 1, as in (8), and that the decision method is *best*, as in (10). In addition, suppose that agents play a Hawk-Dove game with strategy set $S = \{H, D\}$ and a payoff matrix $\mathcal{A} \in \mathbb{R}^{S \times S}$ with $\mathcal{A}_{HD} > \mathcal{A}_{DD} > \mathcal{A}_{DH} > \mathcal{A}_{HH}$. Then in a record containing both strategies, the payoff associated with H is higher if and only if it is \mathcal{A}_{HD} . Thus for a revising H player, (10) can be summarized as follows:

$$\text{in game } \mathcal{A}, \sigma_{HD}((H, \pi_H), (D, \pi_D)) = 1_{\pi_H = \mathcal{A}_{HH}}.$$

Thus

$$\rho_{HD}^{\mathcal{A}}(x) = \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi) = \sum_{\ell \in S} x_D \times x_H x_\ell \times 1 = x_D x_H,$$

and similarly,

$$\rho_{DH}^{\mathcal{A}}(x) = \sum_{(s, \pi)} p_i(s | x) q(\pi | s, x, A) \sigma_{ij}(s, \pi) = \sum_{k \in S} x_H \times x_k x_D \times 1 = x_H x_D.$$

It is convenient to change the state variable from $x = (x_H, x_D) \in X$ to $y \in [0, 1]$ by letting $y = x_H = 1 - x_D$. The mean dynamic (15) generated by ρ and \mathcal{A} is then

$$(19) \quad \dot{y} = V^{\rho, \mathcal{A}}(y) = (1 - y) \rho_{DH}^{\mathcal{A}}(y) - y \rho_{HD}^{\mathcal{A}}(y) = y(1 - y)(1 - 2y).$$

The rest points of (19) are 0, 1, and $y^* = \frac{1}{2}$.

To describe the diffusion approximation at interior rest point $y^* = \frac{1}{2}$, let $\{Y_t^N\}_{t \in \{0, 1/N, \dots\}}$ be the Markov chain (2) expressed in terms of the new state variable, let $\{\bar{Y}_t^N\}_{t \geq 0}$ be its piecewise constant interpolation, and let

$$(20) \quad Z_t^N = \sqrt{N}(\bar{Y}_t^N - y^*)$$

be the local behavior process near y^* . To use approximation (18), we calculate the derivative

$$DV^{\rho, \mathcal{A}}(y^*) = 6y^2 - 6y - 1 \Big|_{y=y^*} = -\frac{1}{2}$$

and the local covariance (17):

$$\Sigma^* = (1 - y^*) \rho_{DH}^{\mathcal{A}}(y^*) + y^* \rho_{HD}^{\mathcal{A}}(y^*) = y^*(1 - y^*) = \frac{1}{4}.$$

Then (20) is approximated in distribution by the solution to

$$(21) \quad dZ_t = DV_i^{\rho, A}(x^*)Z_t dt + \sqrt{\Sigma^*} dB_t = -\frac{1}{2}Z_t dt + \frac{1}{2}dB_t.$$

Now write $\mu^* = DV_i^{\rho, A}(x^*) = -\frac{1}{2}$. Applying Ito's lemma shows that the (weak) solution to (21) with $Z_0 = 0$ is the zero-mean Gaussian process

$$Z_t = \int_0^t \sqrt{\Sigma^*} \exp(\mu^*(t-s)) dB_s = \int_0^t \frac{1}{2} \exp\left(-\frac{1}{2}(t-s)\right) dB_s,$$

whose variance at time T is

$$\text{Var}(Z_T) = \int_0^T \Sigma^* \exp(2\mu^*t) dt = \frac{\Sigma^*}{2\mu^*}(\exp(2\mu^*T) - 1) = \frac{1}{4}(1 - e^{-T})$$

(see Karatzas and Shreve (1991)). Thus the limiting distribution of $\{Z_t\}$ is *normal* $(0, \frac{1}{4})$.

Using (20) to return to the scale of the original process, the diffusion approximation (21) estimates the variance of the fluctuations of $\{Y_t\}$ about y^* as $(\frac{1}{\sqrt{N}})^2 \cdot \frac{1}{4} = \frac{1}{4N}$, and the standard deviation of these fluctuations as $\frac{1}{2\sqrt{N}}$. ♦

B.4 Random-walk tie-breaking

Here we describe the *random-walk* tie-breaking option for decision method *best* (10). This option makes use of an auxiliary Markov chain $\{Y_t^N\}_{t \in [0, 1/N, \dots]}$ that runs on the state space $\mathcal{X}_+^N = \{y \in \mathcal{X}^N : y_i > 0 \text{ for all } i \in S\}$ according to the following transition law.⁴⁰

$$\mathbb{P}\left(Y_{k+1/N}^N = y + \frac{1}{N}(e_j - e_i) \mid Y_k^N = y\right) = y_i \frac{Ny_j + 1}{N + n - 1}.$$

Sandholm (2010b, Example 11.4.3) shows that the stationary distribution of this irreducible Markov chain is the uniform distribution on \mathcal{X}_+^N .

Returning to the original model, let (s, π) be the record of a revising agent, and let $m_j(s, \pi) = \#\{h : s_h = j \text{ and } \pi_h = \max_{h'} \pi_{h'}\}$ be the number of pairs in the record comprised of strategy j and the highest payoff in the record. Let y denote the current state of the

⁴⁰To interpret this process, imagine that there is an auxiliary population of $N+n$ agents choosing strategies in S . For each strategy in S , the population has one agent who always plays that strategy. The process $\{Y_t^N\}$ describes the behavior of the remaining N agents. In each period, one of these agents is chosen at random; this agent then chooses one of the remaining $N+n-1$ agents at random and switches to that agent's strategy.

auxiliary process. Then decision method *best* with *random-walk* tie-breaking is defined by

$$(22) \quad \sigma_{ij}(s, \pi, y) = \frac{m_j(s, \pi) y_j}{\sum_{k \in S} m_k(s, \pi) y_k}.$$

Thus strategy j is chosen with probability proportional both to the number of times in the record that it is associated with the highest payoff and to the j th component of the auxiliary Markov chain.⁴¹

C. Monte Carlo simulations in *ABED* using *BehaviorSpace*

In this appendix, we explain how to run Monte Carlo experiments in *ABED* using *BehaviorSpace* (Wilensky and Shargel, 2002).⁴²

To create a computational experiment in *ABED*, we go to *Tools > BehaviorSpace* in the NetLogo top menu, and then click on *New*. This will open a window like the one shown in Figure 14. The first field in the window allows us to specify a name for the experiment, while the second, larger field is where we specify the parameter values that we wish to explore. Initially, this second field contains the names of all parameters in *ABED* with their current value, surrounded by square brackets, e.g. ["n-of-agents" 20]. To explore several values of one parameter, we just add the values on the appropriate list, e.g. ["n-of-agents" 20 200 2000]. There is also the possibility of defining these values with a loop, using the syntax ["name-of-parameter" [start increment end]]. (Note the additional brackets.) For instance, writing ["n-of-agents" [100 10 200]] would run simulations with 100, 110, 120, ... , 200 agents.

If more than one parameter is taken to vary, then *BehaviorSpace* will consider all possible combinations of the specified parameter values. For each combination, *BehaviorSpace* will run the number of simulations specified in the field labeled "Repetitions".

The next field contains the information that we wish to extract from each simulation run, written in *NetLogo* language. For instance, to obtain the proportion of players using strategy 1, we should write:

⁴¹Note that (22) agrees with (10) when the antecedent of (10) is satisfied (i.e., when tie-breaking is unnecessary).

⁴²There is also the possibility of linking *NetLogo* with advanced software tools like *R* (R Core Team, 2018), *Mathematica* (Wolfram Research, Inc., 2018) or *Matlab* (MathWorks, Inc., 2018). Specifically, using an *R* package called *RNetLogo* (Thiele (2014); Thiele et al. (2014, 2012a,b)), it is possible in *R* to run and control *NetLogo* models, execute *NetLogo* commands, and obtain any information from a *NetLogo* model. The so-called *Mathematica link* (Bakshy and Wilensky, 2007) does a similar job for *Mathematica* and it comes bundled as part of the latest *NetLogo* releases. Conversely, one can also call *R* and *Matlab* commands from within *NetLogo* using *R-Extension* (Thiele and Grimm, 2010) and *MatNet* (Biggs and Papin, 2013) respectively.

Experiment

Experiment name
hawk-dove

Vary variables as follows (note brackets and quotation marks):

```
["payoff-matrix" "[0 3]\n [1 2]"]
["n-of-agents" 20 200 2000]
["random-initial-condition?" true]
["initial-condition" "[10 10]"]
["use-prob-revision?" true]
["prob-revision" 0.1]
["n-of-revisions-per-tick" 1]
["candidate-selection" "imitative"]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions
10

run each combination this many times

☒ Run combinations in sequential order

For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
count players with [strategy = 1] / n-of-agents
```

one reporter per line; you may not split a reporter
across multiple lines

☒ Measure runs at every step

if unchecked, runs are measured only when they are over

Setup commands:

startup

Go commands:

go

▶ Stop condition:

the run stops if this reporter becomes true

▶ Final commands:

run at the end of each run

Time limit
10000

stop after this many steps (0 = no limit)

Cancel
OK

Figure 14: *BehaviorSpace* window used to set up a computational experiment.

count players with [strategy = 1] / n-of-agents

The information detailed in this field will be gathered either at every tick, or only at the very end of each simulation, depending on whether the box labeled “Measure runs at every step” is ticked or not.

The field labeled “Setup commands” includes the names of the functions that will be run at the beginning of each simulation run. This field should include the name `startup` for *ABED*. Similarly, the field labeled “Go commands” gathers the names of the functions that will be repeatedly executed. This field should be filled in with the name `go` for *ABED*. One can also run specific functions at the end of each simulation run—for instance, exporting plots to files—by using the field “Final commands”.

Each simulation will be run a number of ticks specified in the field labeled “Time limit”. It is possible to define other termination conditions, but this requires writing code in *NetLogo*.

Once the experiment is set up, it is initiated by clicking on “Run”. As the experiment proceeds, *NetLogo* will save all the requested data on a *csv* file, with name and location to be chosen by the user.

Finally, it is worth mentioning that computational experiments set up with *BehaviorSpace* can be run from the command line, i.e. without having to open *NetLogo*’s graphical user interface. This feature is particularly useful for launching large-scale experiments in computer clusters.

References

- Arneodo, A., Coulet, P., and Tresser, C. (1980). Occurrence of strange attractors in three-dimensional Volterra equations. *Physics Letters*, 79A:259–263.
- Bakshy, E. and Wilensky, U. (2007). Netlogo-mathematica link. Software. <http://ccl.northwestern.edu/netlogo/mathematica.html>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- Balkenborg, D. and Schlag, K. H. (2001). Evolutionarily stable sets. *International Journal of Game Theory*, 29:571–595.
- Barreto, C. (2014). Population dynamics toolbox (pdtoolbox). Software. https://github.com/carlobar/PDToolbox_matlab.
- Benaïm, M. and Weibull, J. W. (2003). Deterministic approximation of stochastic evolution in games. *Econometrica*, 71:873–903.

- Biggs, M. B. and Papin, J. A. (2013). Novel multiscale modeling tool applied to pseudomonas aeruginosa biofilm formation. *PLOS ONE*, 8(10).
- Binmore, K. and Samuelson, L. (1997). Muddling through: Noisy equilibrium selection. *Journal of Economic Theory*, 74:235–265.
- Binmore, K., Samuelson, L., and Vaughan, R. (1995). Musical chairs: Modeling noisy evolution. *Games and Economic Behavior*, 11:1–35. Erratum, 21 (1997), 325.
- Björnerstedt, J. and Weibull, J. W. (1996). Nash equilibrium and evolution by imitation. In Arrow, K. J. et al., editors, *The Rational Foundations of Economic Behavior*, pages 155–181. St. Martin’s Press, New York.
- Blume, L. E. (1997). Population games. In Arthur, W. B., Durlauf, S. N., and Lane, D. A., editors, *The Economy as an Evolving Complex System II*, pages 425–460. Addison-Wesley, Reading, MA.
- Durrett, R. (1996). *Stochastic Calculus: A Practical Introduction*. CRC Press, Boca Raton, FL.
- Dzonsons, K. and Weibull, J. (2013). Birds. game theory simulator. Software. <http://www.kcons.eu/birds>.
- Eigen, M., Gardiner, W., Schuster, P., and Winkler-Oswatitsch, R. (1981). The origin of genetic information. *Scientific American*, 244(April):88–118.
- Eigen, M. and Schuster, P. (1979). *The Hypercycle: A Principle of Natural Self-Organization*. Springer, Berlin.
- Ellison, G. (2000). Basins of attraction, long run equilibria, and the speed of step-by-step evolution. *Review of Economic Studies*, 67:17–45.
- Foster, D. P. and Young, H. P. (1990). Stochastic evolutionary game dynamics. *Theoretical Population Biology*, 38:219–232. Corrigendum, 51 (1997), 77-78.
- Franchetti, F. and Sandholm, W. H. (2013). An introduction to *Dynamo*: Diagrams for evolutionary game dynamics. *Biological Theory*, 8(2):167–178.
- Fudenberg, D. and Imhof, L. A. (2006). Imitation processes with small mutations. *Journal of Economic Theory*, 131:251–262.
- Fudenberg, D. and Imhof, L. A. (2008). Monotone imitation dynamics in large populations. *Journal of Economic Theory*, 140:229–245.
- Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games*. MIT Press, Cambridge.
- Fudenberg, D., Nowak, M. A., Taylor, C., and Imhof, L. A. (2006). Evolutionary game dynamics in finite populations with strong selection and weak mutation. *Theoretical Population Biology*, 70:352–363.

- García, J. and van Veelen, M. (2016). In and out of equilibrium I: Evolution of strategies in repeated games with discounting. *Journal of Economic Theory*, 161:161 – 189.
- Gilboa, I. and Matsui, A. (1991). Social stability and equilibrium. *Econometrica*, 59:859–867.
- Helbing, D. (1992). A mathematical model for behavioral changes by pair interactions. In Haag, G., Mueller, U., and Troitzsch, K. G., editors, *Economic Evolution and Demographic Change: Formal Models in Social Sciences*, pages 330–348. Springer, Berlin.
- Hofbauer, J. (1981). On the occurrence of limit cycles in the Volterra-Lotka equation. *Nonlinear Analysis*, 5:1003–1007.
- Hofbauer, J. (1995a). Imitation dynamics for games. Unpublished manuscript, University of Vienna.
- Hofbauer, J. (1995b). Stability for the best response dynamics. Unpublished manuscript, University of Vienna.
- Hofbauer, J., Mallet-Paret, J., and Smith, H. L. (1991). Stable periodic solutions for the hypercycle system. *Journal of Dynamics and Differential Equations*, 3:423–436.
- Hofbauer, J. and Sandholm, W. H. (2007). Evolution in games with randomly disturbed payoffs. *Journal of Economic Theory*, 132:47–69.
- Hofbauer, J. and Sandholm, W. H. (2009). Stable games and their dynamics. *Journal of Economic Theory*, 144:1665–1693.
- Hofbauer, J., Schuster, P., and Sigmund, K. (1979). A note on evolutionarily stable strategies and game dynamics. *Journal of Theoretical Biology*, 81:609–612.
- Hofbauer, J., Schuster, P., and Sigmund, K. (1981). Competition and cooperation in catalytic self-replication. *Journal of Mathematical Biology*, 11:155–168.
- Hofbauer, J. and Sigmund, K. (1988). *Theory of Evolution and Dynamical Systems*. Cambridge University Press, Cambridge.
- Imhof, L. A., Fudenberg, D., and Nowak, M. A. (2005). Evolutionary cycles of cooperation and defection. *Proceedings of the National Academy of Sciences*, 31:10797–10800.
- Izquierdo, L. R., Izquierdo, S. S., Galán, J. M., and Santos, J. I. (2013). Combining mathematical and simulation approaches to understand the dynamics of computer models. In Edmonds, B. and Meyer, R., editors, *Simulating Social Complexity: A Handbook*, chapter 11, pages 235–271. Springer Berlin Heidelberg.
- Izquierdo, L. R., Izquierdo, S. S., and Sandholm, W. H. (2018). *Agent-Based Evolutionary Game Dynamics*. University of Wisconsin Pressbooks.
- Izquierdo, S. S. and Izquierdo, L. R. (2013). Stochastic approximation to understand simple simulation models. *Journal of Statistical Physics*, 151(1):254–276.

- Kandori, M., Mailath, G. J., and Rob, R. (1993). Learning, mutation, and long run equilibria in games. *Econometrica*, 61:29–56.
- Kandori, M. and Rob, R. (1995). Evolution of equilibria in the long run: A general theory and applications. *Journal of Economic Theory*, 65:383–414.
- Karatzas, I. and Shreve, S. E. (1991). *Brownian Motion and Stochastic Calculus*. Springer, New York, second edition.
- Kim, Y.-G. and Sobel, J. (1995). An evolutionary approach to pre-play communication. *Econometrica*, 63:1181–1193.
- Kreindler, G. E. and Young, H. P. (2013). Fast convergence in evolutionary equilibrium selection. *Games and Economic Behavior*, 80:39–67.
- MathWorks, Inc. (2018). Matlab. Software. <https://mathworks.com>. Natick, Massachusetts.
- Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge.
- Maynard Smith, J. and Price, G. R. (1973). The logic of animal conflict. *Nature*, 246:15–18.
- Monderer, D. and Shapley, L. S. (1996). Potential games. *Games and Economic Behavior*, 14:124–143.
- Moran, P. A. P. (1958). Random processes in genetics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 54(1):60–71.
- Moran, P. A. P. (1962). *The Statistical Processes of Evolutionary Theory*. Clarendon Press, Oxford.
- Newton, J. (2018). Evolutionary game theory: a renaissance. Unpublished manuscript.
- Nowak, M. A. (2006). *Evolutionary Dynamics: Exploring the Equations of Life*. Belknap/Harvard, Cambridge.
- Nowak, M. A., Sasaki, A., Taylor, C., and Fudenberg, D. (2004). Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428:646–650.
- Oyama, D., Sandholm, W. H., and Tercieux, O. (2015). Sampling best response dynamics and deterministic equilibrium selection. *Theoretical Economics*, 10:243–281.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org>.
- Rosenthal, R. W. (1981). Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economic Theory*, 25:92–100.

- Roth, G. and Sandholm, W. H. (2013). Stochastic approximations with constant step size and differential inclusions. *SIAM Journal on Control and Optimization*, 51:525–555.
- Sample, C. and Allen, B. (2017). The limits of weak selection and large population size in evolutionary game theory. *Journal of Mathematical Biology*, 75(5):1285–1317.
- Sandholm, W. H. (2001). Almost global convergence to p -dominant equilibrium. *International Journal of Game Theory*, 30:107–116.
- Sandholm, W. H. (2003). Evolution and equilibrium under inexact information. *Games and Economic Behavior*, 44:343–378.
- Sandholm, W. H. (2010a). Pairwise comparison dynamics and evolutionary foundations for Nash equilibrium. *Games*, 1:3–17.
- Sandholm, W. H. (2010b). *Population Games and Evolutionary Dynamics*. MIT Press, Cambridge.
- Sandholm, W. H. (2015). Population games and deterministic evolutionary dynamics. In Young, H. P. and Zamir, S., editors, *Handbook of Game Theory with Economic Applications*, volume 4, chapter 13, pages 703–778. Elsevier, Amsterdam.
- Sandholm, W. H., Dokumacı, E., and Franchetti, F. (2012). Dynamo: Diagrams for evolutionary game dynamics. Software. <http://www.ssc.wisc.edu/~whs/dynamo>.
- Sandholm, W. H., Izquierdo, S. S., and Izquierdo, L. R. (2017). Best experienced payoff dynamics and cooperation in the Centipede game. Unpublished manuscript, University of Wisconsin, Universidad de Valladolid, and Universidad de Burgos.
- Sandholm, W. H. and Staudigl, M. (2016). Large deviations and stochastic stability in the small noise double limit. *Theoretical Economics*, 11:279–355.
- Sandholm, W. H. and Staudigl, M. (2018). Sample path large deviations for stochastic evolutionary game dynamics. *Mathematics of Operations Research*, forthcoming.
- Sato, Y., Akiyama, E., and Farmer, J. D. (2002). Chaos in learning a simple two-person game. *Proceedings of the National Academy of Sciences*, 99:4748–4751.
- Schlag, K. H. (1998). Why imitate, and if so, how? A boundedly rational approach to multi-armed bandits. *Journal of Economic Theory*, 78:130–156.
- Schlag, K. H. (1999). Which one should I imitate? *Journal of Mathematical Economics*, 31:493–522.
- Schnabl, W., Stadler, P. F., Forst, C., and Schuster, P. (1991). Full characterization of a strange attractor: Chaotic dynamics in low-dimensional replicator systems. *Physica D*, 48:65–90.

- Schuster, P. and Sigmund, K. (1983). Replicator dynamics. *Journal of Theoretical Biology*, 100:533–538.
- Schuster, P., Sigmund, K., and Wolff, R. (1978). Dynamical systems under constant organization I: Topological analysis of a family of nonlinear differential equations—a model for catalytic hypercycles. *Bulleting of Mathematical Biology*, 40:743–769.
- Sethi, R. (2000). Stability of equilibria in games with procedurally rational players. *Games and Economic Behavior*, 32:85–104.
- Skyrms, B. (1992). Chaos in game dynamics. *Journal of Logic, Language, and Information*, 1:111–130.
- Smith, M. J. (1984). The stability of a dynamic model of traffic assignment—an application of a method of Lyapunov. *Transportation Science*, 18:245–252.
- Stroock, D. W. and Varadhan, S. R. S. (1979). *Multidimensional Diffusion Processes*. Springer, New York.
- Taylor, C., Fudenberg, D., Sasaki, A., and Nowak, M. A. (2004). Evolutionary game dynamics in finite populations. *Bulletin of Mathematical Biology*, 66:1621–1644.
- Taylor, P. D. and Jonker, L. (1978). Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156.
- Thiele, J. C. (2014). R marries NetLogo: Introduction to the RNetLogo package. *Journal of Statistical Software*, 58(2):1–41.
- Thiele, J. C. and Grimm, V. (2010). Netlogo meets r: Linking agent-based models with a toolbox for their analysis. *Environmental Modelling & Software*, 25(8):972 – 974.
- Thiele, J. C., Kurth, W., and Grimm, V. (2012a). Agent-based modelling: Tools for linking netlogo and R. *Journal of Artificial Societies and Social Simulation*, 15(3):8.
- Thiele, J. C., Kurth, W., and Grimm, V. (2012b). RNetLogo: An R package for running and exploring individual-based models implemented in NetLogo. *Methods in Ecology and Evolution*, 3(3):480–483.
- Thiele, J. C., Kurth, W., and Grimm, V. (2014). Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using netlogo and R. *Journal of Artificial Societies and Social Simulation*, 17(3):11.
- Thomas, B. (1985). On evolutionarily stable sets. *Journal of Mathematical Biology*, 22:105–115.
- van Veelen, M. (2012). Robustness against indirect invasions. *Games and Economic Behavior*, 74:382–393.
- Vega-Redondo, F. (1997). The evolution of Walrasian behavior. *Econometrica*, 65:375–384.

- Weibull, J. W. (1995). *Evolutionary Game Theory*. MIT Press, Cambridge.
- Wilensky, U. (1999). Netlogo. Software. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- Wilensky, U. and Shargel, B. (2002). Behaviorspace. Software. <http://ccl.northwestern.edu/netlogo/behaviorspace.html>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- Wolfram Research, Inc. (2018). Mathematica. Software. <https://www.wolfram.com>. Champaign, Illinois.
- Wu, B., Altrock, P. M., Wang, L., and Traulsen, A. (2010). Universality of weak selection. *Phys. Rev. E*, 82:046106.
- Young, H. P. (1993). The evolution of conventions. *Econometrica*, 61:57–84.
- Young, H. P. (1998). *Individual Strategy and Social Structure*. Princeton University Press, Princeton.
- Zeeman, E. C. (1980). Population dynamics from game theory. In Nitecki, Z. and Robinson, C., editors, *Global Theory of Dynamical Systems (Evanston, 1979)*, number 819 in Lecture Notes in Mathematics, pages 472–497, Berlin. Springer.

List of parameters in ABED-1pop

candidate-selection, 11, 16, 18–21, 23, 26, 28, 30, 32
complete-matching?, 12, 16, 18, 21, 23, 26, 28, 30, 32
consider-imitating-self?, 16, 18, 21, 23, 26, 28, 32, 35

decision-method, 12, 13, 16, 18, 21, 23, 24, 26, 28, 30, 32, 36
duration-of-recent, 14

imitatees-with-replacement?, 28, 32, 35
initial-condition, 10, 16, 18, 21, 26

log-noise-level, 13, 36

n-of-agents, 10, 11, 14, 16–18, 21, 23, 26, 28, 32, 33, 43
n-of-candidates, 11–13, 16, 18, 21, 23, 26, 28, 32, 34
n-of-revisions-per-tick, 11, 14, 16–18, 21, 26, 28, 29, 33
n-of-trials, 12, 16, 18, 21, 23, 26, 30, 35, 36, 38, 39, 41

payoff-matrix, 9, 13, 16, 18, 21, 23, 26, 29, 33
plot-every-?-secs, 14
prob-mutation, 13, 16, 18, 21, 23, 26, 30
prob-revision, 11, 14, 23

random-initial-condition?, 10, 16, 18, 21, 23, 26, 29

self-matching?, 16, 18, 21, 23, 26, 28, 32, 35
show-complete-history?, 14
show-recent-history?, 14
single-sample?, 12, 29–31, 35, 36

tie-breaker, 12, 16, 26, 30, 32, 36
trials-with-replacement?, 32

use-prob-revision?, 11, 14, 16, 18, 21, 23, 26, 28, 29, 33