Efficient evolutionary dynamics with extensive–form games

Nicola Gatti and Fabio Panozzo and Marcello Restelli
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
{ngatti, panozzo, restelli}@elet.polimi.it

Abstract
Evolutionary game theory combines game theory and dynamical systems and is customarily adopted to describe evolutionary dynamics in multi–agent systems. In particular, it has been proven to be a successful tool to describe multi–agent learning dynamics. To the best of our knowledge, we provide in this paper the first replicator dynamics applicable to the sequence form of an extensive–form game, allowing an exponential reduction of time and space w.r.t. the currently adopted replicator dynamics for normal form. Furthermore, our replicator dynamics is realization equivalent to the standard replicator dynamics for normal form. We prove our results for both discrete–time and continuous–time cases. Finally, we extend standard tools to study the stability of a strategy profile to our replicator dynamics.

Introduction
Game theory provides the most elegant tools to model strategic interaction situations among rational agents. These situations are customarily modeled as games (FT91) in which the mechanism describes the rules and strategies describe the behavior of the agents. Furthermore, game theory provides a number of solution concepts. The central one is Nash equilibrium. Game theory assumes agents to be rational and describes “static” equilibrium states. Evolutionary game theory (Cre03) drops the assumption of rationality and assumes agents to be adaptive in the attempt to describe dynamics of evolving populations. Interestingly, there are strict relations between game theory solution concepts and evolutionary game theory steady states, e.g., Nash equilibria are steady states. Evolutionary game theory is commonly adopted to study economic evolving populations (CNP07) and artificial multi–agent systems, e.g., for describing multi–agent learning dynamics (THV06, TP07, PTL08) and as heuristics in algorithms (KMT11). In this paper, we develop efficient techniques for evolutionary dynamics with extensive–form games.

Extensive–form games are a very important class of games. They provide a richer representation than strategic–form games, the sequential structure of decision–making being described explicitly and each agent being allowed to be free to change her mind as events unfold. The study of extensive–form games is carried out by translating the game by means of tabular representations (SLB08). The most common is the normal form. Its advantage is that all the techniques applicable to strategic–form games can be adopted also with this representation. However, the size of normal form grows exponentially with the size of the game tree, thus being impractical. The agent form is an alternative representation whose size is linear in the size of the game tree, but it makes, even with two agents, each agent’s best–response problem highly non–linear. To circumvent these issues, sequence form was proposed (vS96). This form is linear in the size of the game tree and does not introduce non–linearities in the best–response problem. On the other hand, standard techniques for strategic–form games cannot be adopted with such representation, e.g. (LH64), thus requiring alternative ad hoc techniques, e.g. (Lem78). In addition, sequence form is more expressive than normal form. For instance, working with sequence form it is possible to find Nash–equilibrium refinements for extensive–form games—perfection based Nash equilibria and sequential equilibrium (MS10, GI11)—while it is not possible with normal form.

To the best of our knowledge, there is no result dealing with the adoption of evolutionary game theory tools with sequence form for the study of extensive–form games, all the known results working with the normal form (Cre03). In this paper, we originally explore this topic, providing the following main contributions.

• We show that the standard replicator dynamics for normal form cannot be adopted with the sequence form, the strategies produced by replication being not well–defined sequence–form strategies.
• We design an ad hoc version of the discrete–time replicator dynamics for sequence form and we show that it is sound, the strategies produced by replication being well–defined sequence–form strategies.
• We show that our replicator dynamics is realization equivalent to the standard discrete–time replicator dynamics for normal form and therefore that the two replicator dynamics evolve in the same way.
• We extend our discrete–time replicator dynamics to the continuous–time case, showing that the same properties are satisfied and extending standard tools to study the stability of the strategies to our replicator.
Extensive–form game definition. A perfect–information extensive–form game (PFT91) is a tuple (N, A, V, T, i, ρ, χ, u), where: N is the set of agents (i ∈ N denotes a generic agent), A is the set of actions (A_i ⊆ A denotes the set of actions of agent i and a ∈ A denotes a generic action), V is the set of decision nodes (V_i ⊆ V denotes the set of decision nodes of i), T is the set of terminal nodes (w ∈ V ∪ T denotes a generic node and w_0 is root node), i : V → N returns the agent that acts at a given decision node, ρ : V → ϕ(A) returns the actions available to agent i(w) at w, χ : V × A → V ∪ T assigns the next (decision or terminal) node to each pair (w, a) where a is available at w, and u = (u_1, ..., u_1(N)) is the set of agents’ utility functions u_i : T → R. Games with imperfect information extend those with perfect information, allowing one to capture situations in which some agents cannot observe some actions undertaken by other agents. We denote by V_i,h the h–th information set of agent i. An information set is a set of decision nodes such that when an agent plays at one of such nodes she cannot distinguish the node in which she is playing. For the sake of simplicity, we assume that every information set has a different index h, thus we can univocally identify an information set by h. Furthermore, since the available actions at all nodes w belonging to the same information set h are the same, with abuse of notation, we write ρ(h) in place of ρ(w) with w ∈ V_i,h. An imperfect–information game is a tuple (N, A, V, T, i, ρ, χ, u, H) where (N, A, V, T, i, ρ, χ, u) is a perfect–information game and H = (H_1, ..., H[N]) induces a partition V_i = U_h∈H V_i,h such that for all w, w’ ∈ V_i,h we have ρ(w) = ρ(w’). We focus on games with perfect recall where each agent recalls all the own previous actions and the ones of the opponents (PFT91).

Figure 1: Example of two–agent perfect–information extensive–form game, x.y denote the y–th node of agent x.

(Reduced) Normal form (CNM44). It is a tabular representation in which each normal–form action, called plan and denoted by p ∈ P_i where P_i is the set of plans of agent i, specifies one action a per information set. We denote by π_(agent) a normal–form strategy of agent i and by π_(agent)(p) the probability associated with plan p. The number of plans (and therefore the size of the normal form) is exponential in the size of the game tree. The reduced normal form is obtained from the normal form by deleting replicated strategies (V938). Although reduced normal form can be much smaller than normal form, it is exponential in the size of the game tree.

Example 1 The reduced normal form of the game in Fig. 1 and a pair of normal–form strategies are:

| agent 1 | agent 2 |
|---------|---------|
| L_1     | 2.4     |
| R_1     | 1       |
| L_3     | 3.1     |
| R_3     | 3.4     |

Replicator dynamics. The standard discrete–time replicator equation with two agents is (Cre03):

\[ \pi_1(p, t + 1) = \pi_1(p, t) \cdot \frac{\sigma_1^T \cdot U_1 \cdot \pi_2(t)}{\pi_1^T(t) \cdot U_1 \cdot \pi_2(t)} \]  
\[ \pi_2(p, t + 1) = \pi_2(p, t) \cdot \frac{\sigma_2^T \cdot U_2 \cdot \pi_0(p, t)}{\pi_1^T(t) \cdot U_2 \cdot \pi_2(t)} \]  

which is an equation for the population dynamics of agents with fixed strategies π_1 and π_2. It states that the proportion of population of agent 1 with strategy p_1 increases if and only if \( \sigma_1^T \cdot U_1 \cdot \pi_2(t) > \pi_1(p, t) \cdot \pi_1^T(t) \cdot U_1 \cdot \pi_2(t) \).
while the continuous–time one is
\[\pi_1(p) = \pi_1(p) \cdot [(e_p - \pi_1)^T \cdot U_1 \cdot \pi_2] \tag{3}\]
\[\pi_2(p) = \pi_2(p) \cdot [(\pi_1^T \cdot U_2 \cdot (e_p - \pi_2)] \tag{4}\]

where \(e_p\) is the vector in which the \(p\)-th component is “1” and the others are “0”.

**Discrete–time replicator dynamics for sequence–form representation**

Initially, we show that the standard discrete–time replicator dynamics for normal form cannot be directly applied when sequence form is adopted. Standard replicator dynamics applied to the sequence form is easily obtained by considering each sequence as a plan \(p\) and thus substituting \(e_q\) to \(e_p\) in (1)–(2) where \(e_q\) is zero for all the components \(q\) such that \(q' \neq q\) and one for the component \(q'\) such that \(q' = q\).

**Proposition 3** The replicator (7–2) does not satisfy the sequence–form constraints.

*Proof.* The proof is by counterexample. Consider \(x_1(t)\) and \(x_2(t)\) equal to the strategies used in Example 2. At time \(t + 1\) the strategy profile generated by (1)–(2) is:

\[x_1^T(t+1) = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \end{bmatrix} \quad x_2^T(t+1) = \begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}\]

that does not satisfy the sequence–form constraints, e.g., \(x_1(q_{1}, t + 1) \neq 1\) for all \(i\).

The critical issue behind the failure of the standard replicator dynamics lies in the definition of vector \(e_q\). Now we describe how the standard discrete–time replicator dynamics can be modified to be applied to the sequence form. In our variation, we substitute \(e_q\) with an opportune vector \(g_q\) that depends on the strategy \(x_1(t)\) and it is generated as described in Algorithm 1 obtaining:

\[x_1(q, t + 1) = x_1(q, t) \cdot \frac{g_q^T(x_1(t) - U_1 \cdot x_2(t))}{x_1^T(t) - U_1 \cdot x_2(t)} \tag{5}\]

\[x_2(q, t + 1) = x_2(q, t) \cdot \frac{g_q^T(x_2(t) - U_2 \cdot g_q(x_2(t)))}{x_2^T(t) - U_2 \cdot g_q(x_2(t))} \tag{6}\]

The basic idea behind the construction of vector \(g_q\) is:

• assigning “1” to the probability of all the sequences contained in \(q\).

• normalizing the probability of the sequences extending the contained in \(q\).

• assigning “0” to the probability of all the other sequences.

We describe the generation of vector \(g_q(x_1(t))\), for clarity we use as running example the generation of \(g_{R_1R_2}(x_1(t))\) related to Example 2:

• all the components of \(g_q(x_1(t))\) are initialized equal to “0”, e.g.,

\[g_{R_1R_2}(x_1(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\]

• if sequence \(q\) is played, the algorithm assigns:

    - “1” to all the components \(g_q(q', x_1(t))\) of \(g_q(x_1(t))\) where \(q' \subseteq q\) (i.e., \(q'\) is a subsequence of \(q\), e.g.,

\[g_{R_1R_2}(x_1(t)) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}\]

\[g_q(q', x_1(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\]

- all the other components are initialized equal to “0”,

- if sequence \(q\) is not played, the algorithm assigns: the output strategy profile \(x_1(t+1)\) of replicator dynamics (5–6) satisfies sequence–form constraints.

**Theorem 4** Given a well–defined sequence–form strategy profile \((x_1(t), x_2(t))\), the output strategy profile \((x_1(t + 1), x_2(t + 1))\) of replicator dynamics (5–6) satisfies sequence–form constraints.

*Proof.* The constraints forced by sequence form are:

• \(x_1(q_i, t) = 1\) for every \(i\),

• \(x_1(q_i, t) = \sum_{a \in \rho(q)\omega} x_1(q_i a, t)\) for every sequence \(q\), action \(a\), node \(\omega\) such that \(w = h(q)\), and for every agent \(i\).

Assume, by hypothesis of the theorem, that the above constraints are satisfied at \(t\), we need to prove that constraints

\[x_1(q_i, t + 1) = 1\]

\[x_1(q_i, t + 1) = \sum_{a \in \rho(q)\omega} x_1(q_i a, t + 1)\]

are satisfied. Constraint (7) always holds because \(g_{R_1R_2}(x_1(t)) = x_1(t)\). We rewrite constraints (8) as

\[x_1(q_i, t) \cdot \frac{g_q^T(x_1(t) - U_1 \cdot x_2(t))}{x_1^T(t) - U_1 \cdot x_2(t)} = \sum_{a \in \rho(q)\omega} x_1(q_i a, t) \cdot \frac{g_q^T(x_i(t) - U_1 \cdot x_2(t))}{x^T_1(t) - U_1 \cdot x_2(t)}\]

Conditions (8) hold if the following condition holds

\[x_1(q_i, t) \cdot g_q^T(x_1(t)) = \sum_{a \in \rho(q)\omega} (x_1(q_i a, t) \cdot g_q^T(x_1(t)))\]

Notice that condition (9) is a vector of equalities, one per sequence \(q\). Condition (10) is trivially satisfied for components \(q'\) such that \(q_0(q', x_1(t)) = 0\). To prove the condition for all the other components, we introduce two lemmas.

**Algorithm 1** generate \(g_q(x_1(t))\)

1: \(g_q(x_1(t)) = 0\)
2: if \(x_1(q, t) = 0\) then
3: \(g_q(q', x_1(t)) = 1\)
4: for \(q' \in Q_i\) s.t. \(q' \subseteq q\) do
5: \(g_q(q', x_1(t)) = \frac{1}{x_1(q', t)}\)
6: for \(q'' \in Q_i\) s.t. \(q'' \cap q = q'\) and \(q'' / q' = a\) do \(q' \neq h\) do
7: return \(g_q(x_1(t))\)
Lemma 5 Constraint (10) holds for all components $g_q(q', x(t))$ of $g_q(x(t))$ such that $q' \leq q$.

Proof. By construction, $g_q(q', x(t)) = 1$ for every $q' \leq q$. For every extension $q|a$ of $q$, we have that $q' \leq q \subset q|a$. For this reason $g_{q|a}(q', x(t)) = 1$. Thus

$$x_i(q, t) - g_{q|a}(q', x(t)) = \sum_{a \in P \cap p(q)} (x_i(q|a), t) - g_{q|a}(q', x(t))$$

and therefore conditions (11) hold.

Lemma 6 Constraint (10) holds for all components $g_q(q'', x(t))$ of $g_q(x(t))$ where $q'' \leq q$ with $q' = q'' \cap q$ and sequence $q'' = q'|a'_1| \ldots | a'_n | q'' + h$. Thus, $g_{q''}(q'', x(t)) = \sum_{p \in P \cap q''} (x_i(q|a'), t)$, in the latter there exists only one action $a'$ such that $g_{q|a'}(q', x(t)) = \sum_{p \in P \cap q''} (x_i(q|a'), t)$, while for the other actions $a'$ the value of $g_{q|a'}(q', x(t))$ is zero. Hence, we can have two cases: if $q|a \not\subset q''$, then

$$x_i(q, t) - g_{q|a'}(q', x(t)) = \sum_{a \in P \cap q''} (x_i(q|a), t) - g_{q|a'}(q', x(t))$$

that holds by hypothesis, otherwise if $q|a \subset q''$, then

$$x_i(q, t) - g_{q|a'}(q', x(t)) = \sum_{a \in P \cap q''} (x_i(q|a), t) - g_{q|a'}(q', x(t))$$

and therefore conditions (11) hold.

From the application of Lemmas 5 and 6 it follows that condition (10) holds.

Replicator dynamics realization equivalence

There is a well-known relation, based on the concept of realization, between normal–form and sequence–form strategies. In order to exploit it, we introduce two results from [KMv99].

Definition 7 (Realization equivalent) Two strategies of an agent are realization equivalent if, for any fixed strategies of the other agents, both strategies define the same probabilities for reaching the nodes of the game tree.

Proposition 8 For an agent with perfect recall, any normal–form strategy is realization equivalent to a sequence–form strategy.

We recall in addition that each pure sequence–form strategy corresponds to a pure normal–form strategy in the reduced normal form (KMv99). We can show that the evolutionary dynamics of (5) (6) are realization equivalent to the evolutionary dynamics of the normal–form replicator dynamics and therefore that the two replicator dynamics evolve in the same way.

Initially, we introduce the following lemma that we will exploit to prove the main result.

Lemma 9 Given

• a reduced–normal–form strategy $\pi_i(t)$ of agent $i$,

that holds by hypothesis. Therefore the lemma is proved.

Rephrasing the wordings for the behavioral strategies, we can write

$$x_i(q, t) - g_{q|a'}(q', x(t)) = \sum_{a \in P \cap q''} (x_i(q|a'), t)$$

that can be easily rewrite as—for details (7)—

$$x_i(q, t) - g_{q|a'}(q', x(t)) = \sum_{a \in P \cap q''} (x_i(q|a'), t)$$

and therefore conditions (11) hold.

This completes the proof of the lemma.

Now we state the main result. It allows us to study the evolution of a strategy in a game directly in sequence form, instead of using the normal form, and it guarantees that the two dynamics (sequence and normal) are equivalent.
Theorem 10
• a normal–form strategy profile \((\pi_1(t), \pi_2(t))\) and its evolution \((\pi_1(t + 1), \pi_2(t + 1))\) according to \((\text{I})\)–\((\text{II})\).
• a sequence–form strategy profile \((x_1(t), x_2(t))\) and its evolution \((x_1(t + 1), x_2(t + 1))\) according to \((\text{I})\)–\((\text{II})\).

if \((\pi_1(t), \pi_2(t))\) and \((x_1(t), x_2(t))\) are realization equivalent, then also \((\pi_1(t + 1), \pi_2(t + 1))\) and \((x_1(t + 1), x_2(t + 1))\) are realization equivalent.

Proof. Assume, by hypothesis of the theorem, that \((x_1(t), x_2(t))\) is realization equivalent to \((\pi_1(t), \pi_2(t))\). Thus, according to \((\text{I})\)–\((\text{II})\), for every agent \(i\) it holds

\[ x_i(q|a, t) = \sum_{p \in P_{A|p}} \pi_i(p, t) \quad \forall a \in A_i. \]

We need to prove that the following conditions hold:

\[ x_i(q|a, t + 1) = \sum_{p \in P_{A|p}} \pi_i(p, t + 1) \quad \forall a \in A_i. \] \(\text{(12)}\)

By applying the definition of replicator dynamics, we can rewrite the conditions \((\text{12})\) as:

\[ x_i(q|a, t) \cdot \frac{\mathbf{g}_i(x_i(t)) \cdot U_i \cdot x_i(t)}{x_i(t)} = \sum_{p \in P_{A|p}} \left( \pi_i(p, t) \cdot \frac{\mathbf{g}_i(p) \cdot U_i \cdot \pi_i(t)}{\pi_i(p) \cdot U_i \cdot \pi_i(t)} \right) \quad \forall a \in A_i. \] \(\text{(13)}\)

Given that, by hypothesis, \(x_i^T(t) \cdot U_i \cdot x_i(t) = \pi_i^T(t) \cdot U_i \cdot \pi_i(t)\), we can rewrite conditions \((\text{13})\) as:

\[ x_i(q|a, t) \cdot \mathbf{g}_i(x_i(t)) = \sum_{p \in P_{A|p}} \left( \pi_i(p, t) \cdot \mathbf{g}_i(p) \cdot U_i \cdot \pi_i(t) \right) \quad \forall a \in A_i. \]

These conditions hold if and only if \(\sum_{p \in P_{A|p}} \pi_i(p, t) \cdot \mathbf{g}_i^T \) is realization equivalent to \(x_i(q|a, t) \cdot \mathbf{g}_i^T(x_i(t))\). By Lemma 9 this equivalence holds. \(\square\)

Continuous–time replicator dynamics for sequence–form representation

The sequence–form continuous–time replicator equation is

\[ \dot{x}_1(q, t) = x_1(q, t) \cdot \left( [\mathbf{g}_1(x_1(t)) - x_1(t)]^T \cdot U_1 \cdot x_2(t) \right) \]
\[ \dot{x}_2(q, t) = x_2(q, t) \cdot \left( [\mathbf{g}_2(x_2(t)) - x_2(t)]^T \cdot U_1 \cdot x_2(t) \right) \] \(\text{(14)}\), \(\text{(15)}\)

Theorem 11
Given a well–defined sequence–form strategy profile \((x_1(t), x_2(t))\), the output strategy profile \((x_1(t + \Delta t), x_2(t + \Delta t))\) of replicator dynamics \((\text{I})\)–\((\text{II})\) satisfies sequence–form constraints.

Proof. The constraints forced by sequence form are:

• \(x_i(q, t) = 1\) for every \(i\),
• \(x_i(q, t) = \sum_{a \in A(w)} x_i(q|a, t)\) for every sequence \(q\),

Assume, by hypothesis of the theorem, that constraints are satisfied at a given time point \(t\), we need to prove that constraints

\[ x_i(q_0, t + \Delta t) = 1 \] \(\text{(16)}\)
\[ x_i(q, t + \Delta t) = \sum_{a \in A(w)} x_i(q|a, t + \Delta t) \] \(\text{(17)}\)

are satisfied. Constraint \((\text{16})\) always holds because \(x_i(q_0, t) = x_1(t)\). We rewrite constraints \((\text{17})\) as

\[ x_i(q, t) \cdot [\mathbf{g}_i(x_i(t)) - x_i(t)]^T \cdot U_1 \cdot x_i(t) = \sum_{a \in A(w)} \left( x_i(q|a, t) \cdot [\mathbf{g}_i(x_i(t)) - x_i(t)]^T \cdot U_1 \cdot x_i(t) \right) \] \(\text{(18)}\)

Conditions \((\text{18})\) hold if the following conditions hold

\[ x_i(q, t) \cdot \mathbf{g}_i^T(x_i(t)) = \sum_{a \in A(w)} \left( x_i(q|a, t) \cdot \mathbf{g}_i^T(x_i(t)) \right) \] \(\text{(19)}\)

Notice that condition \((\text{19})\) is a vector of equalities. The above condition is trivially satisfied for components \(q_i\) such that \(q_i(q^*, x_i(t)) = 0\). From the application of Lemmas 5 and 6 the condition \((\text{19})\) holds also for all the other components. \(\square\)

Theorem 12
• a normal–form strategy profile \((\pi_1(t), \pi_2(t))\) and its evolution \((\pi_1(t + \Delta t), \pi_2(t + \Delta t))\) according to \((\text{I})\)–\((\text{II})\).
• a sequence–form strategy profile \((x_1(t), x_2(t))\) and its evolution \((x_1(t + \Delta t), x_2(t + \Delta t))\) according to \((\text{I})\)–\((\text{II})\).

if \((\pi_1(t), \pi_2(t))\) and \((x_1(t), x_2(t))\) are realization equivalent, then also \((\pi_1(t + \Delta t), \pi_2(t + \Delta t))\) and \((x_1(t + \Delta t), x_2(t + \Delta t))\) are realization equivalent.

Proof. Assume, by hypothesis of the theorem, that \((x_1(t), x_2(t))\) is realization equivalent to \((\pi_1(t), \pi_2(t))\). Thus, according to \((\text{I})\)–\((\text{II})\), for every agent \(i\) it holds

\[ x_i(q|a, t + \Delta t) = \sum_{p \in P_{A|p}} \pi_i(p, t + \Delta t) \quad \forall a \in A_i. \] \(\text{(20)}\)

By applying the definition of replicator dynamics, we can rewrite the conditions \((\text{20})\) as:

\[ x_i(q|a, t) \cdot \left( [\mathbf{g}_i(x_i(t)) - x_i(t)]^T \cdot U_1 \cdot x_i(t) \right) = \sum_{p \in P_{A|p}} \left( \pi_i(p, t) \cdot \mathbf{g}_i(p) \cdot U_1 \cdot \pi_i(t) \right) \quad \forall a \in A_i. \] \(\text{(21)}\)

Given that, by hypothesis, \(x_i^T(t) \cdot U_1 \cdot x_i(t) = \pi_i^T(t) \cdot U_1 \cdot \pi_i(t)\), we can rewrite conditions \((\text{21})\) as:

\[ x_i(q|a, t) \cdot \mathbf{g}_i^T(x_i(t)) = \sum_{p \in P_{A|p}} \left( \pi_i(p, t) \cdot \mathbf{g}_i(p) \right) \cdot U_1 \cdot \pi_i(t) \quad \forall a \in A_i. \]

These conditions hold if and only if \(\sum_{p \in P_{A|p}} \pi_i(p, t) \cdot \mathbf{e}_p^T \) is realization equivalent to \(x_i(q|a, t) \cdot \mathbf{g}_i^T(x_i(t))\). By Lemma 9 this equivalence holds. \(\square\)

Analyzing the stability of a strategy profile

We focus on characterizing a strategy profile in terms of evolutionary stability. When the continuous–time replicator dynamics for normal–form is adopted, evolutionary stability can be analyzed by studying the eigenvalues of the Jacobian in that point \((\text{AP92})\)—non–positiveness of the eigenvalues is a necessary condition for asymptotical stability, while strict negativity of the eigenvalues is sufficient. The Jacobian is

\[ J = \begin{bmatrix} \frac{\partial \dot{x}_1(q_1, t)}{\partial x_1(q_1, t)} & \frac{\partial \dot{x}_1(q_2, t)}{\partial x_1(q_2, t)} \\ \frac{\partial \dot{x}_2(q_1, t)}{\partial x_2(q_1, t)} & \frac{\partial \dot{x}_2(q_2, t)}{\partial x_2(q_2, t)} \end{bmatrix} \quad \forall q_1, q_2 \in Q_1, \]

\[ q_1, q_2 \in Q_2. \]
In order to study the Jacobian of our replicator dynamics, we need to complete the definition of $g_q(x_i(t))$. Indeed, we observe that some components of $g_q(x_i(t))$ are left arbitrary by Algorithm 1. Exactly, some $q_i'$ that are related to $q$ with $x_i(q', t) = 0$. While it is not necessary to assign values to such components during the evolution of the replicator dynamics, it is necessary when we study the Jacobian. The rationale follows. If $x_i(q', t) = 0$, then it will remain zero even after $t$. Instead, if, after the dynamics converged to a point, such a point has $x_i(q') = 0$ for some $q'$, it might be the case that along the dynamics it holds $x_i(q') \neq 0$. Thus, in order to define these components of $g_q(x_i(t))$, we need to reason backward, assigning the values that they would have in the case such sequence would be played with a probability that goes to zero. In absence of degeneracy, Algorithm 2 addresses this issue assigning a value of “1” to a sequence $q''$ if it is the (unique, the game being non-degenerate) best response among the sequences extending $q'$ and “0” otherwise, because at the convergence the agents play only the best response sequences. Notice that, in this case, $g_q(x_i(t), x_{-i}(t))$ depends on both agents’ strategies.

**Algorithm 2** generate $g_q(x_i(t), x_{-i}(t))$

1. $g_q(x_i(t), x_{-i}(t)) = 0$
2. for $q' \in Q_i, s.t. q' \in q$ do
3. $g_q(q', x_i(t), x_{-i}(t)) = 1$
4. for $q'' \in Q_i, s.t. q'' \cap q = q'$ and $q''' = q'' \cap \{a \in \rho(h), q \neq h\}$ do
5. if $x_i(q', t) = 0$ then
6. $g_q(q'', x_i(t), x_{-i}(t)) = x_i(q', t)$
7. else if $q'' = \arg\max_{q'' \in \rho(h)} \mathbb{E}[U_i(q'' \cdot x_{-i}(t))]$ then
8. $g_q(q'', x_i(t), x_{-i}(t)) = 1$
9. return $g_q(x_i(t), x_{-i}(t))$

Given the above complete definition of $g_q$, we can observe that all the components of $g_q(x_i(t), x_{-i}(t))$ generated by Algorithm 2 are differentiable, being “0” or “1” or $x_i(q', t)$. Therefore, we can derive the Jacobian as:

$$\frac{\partial x_i(q', t)}{\partial x_i(q'_1, t)} = \begin{cases} (g_q(x_i(t), x_{-i}(t)) - x_i(t))^T \cdot U_i \cdot x_{-i}(t) + x_i(q'_1, t) & \text{if } i = j \\ -x_i(q'_1, t) \cdot \left( (g_q(x_i(t), x_{-i}(t)) - x_i(t))^T \cdot U_i \cdot x_{-i}(t) - e_i \right) & \text{if } i \neq j \end{cases}$$

$$\frac{\partial x_i(q', t)}{\partial x_i(q'_2, t)} = x_i(q'_1, t) \cdot \left( (g_q(x_i(t), x_{-i}(t)) - x_i(t))^T \cdot U_i \cdot x_{-i}(t) - e_i \right)$$

$$\frac{\partial x_i(q', t)}{\partial x_i(q'_k, t)} = x_i(q'_k, t) \cdot \left( (g_q(x_i(t), x_{-i}(t)) - x_i(t))^T \cdot U_i \cdot x_{-i}(t) - e_i \right)$$

$$\frac{\partial x_i(q', t)}{\partial x_i(q'_1, t)} = \begin{cases} x_i(t)^T \cdot U_i \cdot (g_q(x_i(t), x_{-i}(t)) - x_i(t)) + x_i(q'_1, t) & \text{if } k = i \\ x_i(t)^T \cdot U_i \cdot \left( (g_q(x_i(t), x_{-i}(t)) - x_i(t))^T \cdot U_i \cdot x_{-i}(t) - e_i \right) & \text{if } k \neq i \end{cases}$$

With degenerate games, given a opponent’s strategy profile $x_{-i}(t)$ and a sequence $q \in Q_i$ such that $x_i(q, t) = 0$, we can have multiple best responses. Consider, e.g., the game in Example 2 with $x_1^T(t) = [1 1 0 0 0 0 0 0]$, $x_2^T(t) = [1 1 0]$ and compute $g_{L_1}(x_1(t), x_2(t))$: both sequences $R_1L_2$ and $R_1R_2$ are best responses to $x_2(t)$.

Reasoning backward, we have different vectors $g_q(x_i, x_{-i})$ for different dynamics. More precisely, we can partition the strategy space around $(x_i, x_{-i})$, associating a different best response with a different subspace and therefore with a different $g_q(x_i, x_{-i})$. Thus, in principle, in order to study the stability of a strategy profile, we would need to compute and analyze all the (potentially combinatorial) Jacobians. However, we can show that all these Jacobians are the same and therefore, even in the degenerate case, we can safely study the Jacobian by using a $g_q(x_i, x_{-i})$ as generated by Algorithm 2 except, if there are multiple best responses, Step 7–8 assign “1” only to one, randomly chosen, best response.

**Theorem 13** Given

- a specific sequence $q \in Q_i$ such that $x_i(q, t) = 0$,
- a sequence–form strategy $x_{-i}(t)$,
- a sequence $q' \subseteq q$,
- the number of sequences $q''$ such that $q'' \cap q = q'$ and $q''' = q'' \cap \{a \in \rho(h), q \neq h\}$ and that are best responses to $x_{-i}(t)$ larger than one,
- the eigenvalues of the Jacobian are independent from which sequence $q''$ is chosen as best–response.

**Conclusions and future works**

In this paper we developed efficient evolutionary game theory techniques to deal with extensive–form games. We designed, to the best of our knowledge, the first replicator dynamics applicable with the sequence form of an extensive–form game, allowing an exponential reduction of time and space w.r.t. the standard (normal–form) replicator dynamics. Our replicator dynamics is realization equivalent w.r.t. the standard one and therefore these two replicator dynamics are equivalent in the same way. We show the equivalence for both the discrete and continuous time cases. Finally, we discuss how standard tools from dynamical systems for the study of the stability of strategies can be adopted with our continuous–time replicator dynamics.

In future, we intend to explore the following problems: extending the results on multi–agent learning when sequence form is adopted taking into account also Nash refinements for extensive–form games (we recall, while this is possible with sequence form, it is not with the normal form); extending our results to other forms of dynamics, e.g., best response dynamics, imitation dynamics, smoothed best replies, the Brown–von Neumann–Nash dynamics; comparing the expressivity and the effectiveness of replicator dynamics when applied to the three representation forms.
Appendix

Relation between normal–form/behavioral/sequence–form strategies

We briefly review how realization equivalent strategies can be derived according to (VS96).

Given a behavioral strategy \( \sigma_i \), we can derive the (realization) equivalent normal–form strategy and sequence–form strategy as follow

\[
\pi_i(p) = \prod_{a \in p \in P} \sigma_i(a) \quad \forall p \in P_i
\]
\[
x_i(q) = \prod_{a \in q \in Q_i} \sigma_i(a) \quad \forall q \in Q_i
\]

Given a normal–form strategy \( \pi_i \), we can derive the (realization) equivalent behavioral strategy:

\[
\sigma_i(a) = \sum_{p \in \pi_a \in \pi} \pi_i(p)
\]

Given a normal–form strategy \( \pi_i \), in reduced normal form, we can derive the (realization) equivalent sequence–form strategy:

\[
x_i(qa) = \sum_{p \in \pi_a \in \pi} \pi_i(p)
\]

We denote by \( q(a) \) the sequence whose last action is \( a \). We state the following lemma that we use to prove a main result.

Lemma 14 Given:

• a normal–form strategy \( \pi_i \) in reduced normal form,
• its equivalent behavioral strategy \( \sigma_i \),
• a subset of actions \( \{a_1, \ldots, a_m\} \subseteq A_i \),

it holds

\[
\sum_{p \in P \subset A_1 \cdots A_m} \pi_i(p) = \prod_{a \in q \in Q} \sigma_i(a)
\]

Proof. Suppose that \( p = a_1, \ldots, a_m \). By (22) we know that

\[
\pi_i(p) = \sigma_i(a_1) \cdots \sigma_i(a_n)
\]

For all plan of actions \( p \in P \) where \( \{a_1, \ldots, a_m\} \subseteq p \), given an action \( a \) such that \( a \notin \{a_1, \ldots, a_m\} \), we can have two possibilities

1. \( a \in \bigcup_{j=1}^{m} q(a_j) \), in this case the action \( a \) is present in every plan of actions \( p \), being always present \( \{a_1, \ldots, a_m\} \); thus

\[
\sum_{p \in P \subset A_1 \cdots A_m} \pi_i(p') = \sigma_i(a) \prod_{j=1}^{m} \sigma_i(a_j) \sum_{p \in P \subset A_1 \cdots A_m} (\sigma_i(a_{m+2}) \cdots \sigma_i(a_n))
\]

2. \( a \notin \bigcup_{j=1}^{m} q(a_j) \), in this case there is a subset \( P' \subseteq P \) such that there is exactly a \( p \in P' \) for each action \( a' \in \rho(h) \), where \( a \in \rho(h) \).

By definition of behavioral strategy we know that

\[
\sum_{a \in \rho(h)} \sigma_i(a) = 1 \quad \forall h \in H
\]

Thus

\[
\sum_{p \in P \subset A_1 \cdots A_m} \pi_i(p) = \prod_{a \in q \in Q} \sigma_i(a)
\]

Thus, we can write

\[
\sum_{p \in P \subset A_1 \cdots A_m} \pi_i(p) = \sum_{a \in \bigcup_{j=1}^{m} q(a_j)} \prod_{a \in q \in Q} \sigma_i(a)
\]

and, by Point 2, the other actions sum to “1”

\[
\sum_{p \in P \subset A_1 \cdots A_m} \pi_i(p) = \prod_{a \in \bigcup_{j=1}^{m} q(a_j)} \sigma_i(a)
\]

This completes the proof of the lemma. \( \square \)

Proof of Theorem 13

Proof. For each sequence \( q'' \) that is a best–response to \( x_{-i}(t) \), we can have different vectors \( g_q(x_i(t), x_{-i}(t)) \). Suppose to take two different vectors \( g_q(x_i(t), x_{-i}(t)) \) and \( g_q'(x_i(t), x_{-i}(t)) \). To prove the equality of the two Jacobians we have to prove that each term is the same. All the terms multiplied by \( x_i(q, t) = 0 \) can be discarded, being equal to zero. For this reason the only term different from 0 in the Jacobian is \( \frac{\partial x_i(q, t)}{\partial x_i(q, t)} \), thus we have to prove

\[
(g_q(x_i(t), x_{-i}(t)) - x_{-i}(t)) \cdot U_i = x_{-i}(t)
\]

We can rewrite the equality (28) as

\[
\frac{\partial x_i}{\partial x_i(q, t)}(x_i(t), x_{-i}(t)) \cdot U_i = x_{-i}(t)
\]

that always holds because \( g_q(x_i(t), x_{-i}(t)) \) and \( g_q'(x_i(t), x_{-i}(t)) \), even if they differ for some components, provide the same expected utility by definition of best response. Even if an agent randomizes over multiple best responses, the theorem holds for the same reason. \( \square \)
References

[AP92] D.K. Arrowsmith and C.M. Place. *Dynamical Systems*. Chapman & Hall, London, 1992.

[CNP07] K. Cai, J. Niu, and S. Parsons. Using evolutionary game–theory to analyse the performance of trading strategies in a continuous double auction market. In *ALAMAS*, pages 44–59, 2007.

[Cre03] R. Cressman. *Evolutionary dynamics and extensive form games*. MIT Press, Cambridge, Mass., 2003.

[FT91] D. Fudenberg and J. Tirole. *Game Theory*. 1991.

[GI11] N. Gatti and C. Iuliano. Computing an extensive–form perfect equilibrium in two–player games. In *AAAI*, pages 669–674, August 7–11 2011.

[KMT11] C. Kiekintveld, J. Marecki, and M. Tambe. Approximation methods for infinite bayesian stackelberg games: modeling distributional payoff uncertainty. In *AA–MAS*, pages 1005–1012, 2011.

[KMv96] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two–person games. *GAME ECON BEHAV*, 14(2):220–246, 1996.

[Kuh50] H. W. Kuhn. Extensive games. *Proc. Natl. Acad. Sci. U. S. A.*, 36:570–576, 1950.

[Lem78] C. Lemke. Some pivot schemes for the linear complementarity problem. *Mathematical Programming Study*, 7:15–35, 1978.

[LH64] C. E. Lemke and J. J. T. Howson. Equilibrium points of bimatrix games. *SIAM J APPL MATH*, 12(2):413–423, 1964.

[MS10] P. B. Miltersen and T. B. Sørensen. Computing a quasi–perfect equilibrium of a two–player game. *ECON THEOR*, 42(1):175–192, 2010.

[PTL08] L. Panait, K. Tuyls, and S. Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *J MACH LEARN RES*, 9:423–457, 2008.

[Sel75] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *INT J GAME THEORY*, 4(1):25–55, 1975.

[SLB08] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. 2008.

[THV06] K. Tuyls, P. J. Hoen, and B. Vanschoenwinkel. An evolutionary dynamical analysis of multi–agent learning in iterated games. *AUTON AGENT MULTI–AG*, 12(1):115–153, 2006.

[TP07] K. Tuyls and S. Parsons. What evolutionary game theory tells us about multiagent learning. *ARTIF INTELL*, 171(7):406–416, 2007.

[VJ98] D. Vermeulen and M. Jansen. The reduced form of a game. *EUR J OPER RES*, 106(1):204–211, 1998.

[vNM44] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

[vS96] B. von Stengel. Efficient computation of behavior strategies. *GAME ECON BEHAV*, 14(2):220–246, 1996.