Applied Mathematics and Computation 269 (2015) 863–884

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Computing equilibria in discounted dynamic games

Andriy Burkov, Brahim Chaib-draa*

Computer Science Department, Laval University, PQ, Canada

ARTICLE INFO

Keywords:
Game theory
Repeated games
Stochastic games
Markov chain games
Subgame-perfect equilibria
Automaton

ABSTRACT

Game theory (GT) is an essential formal tool for interacting entities; however computing equilibria in GT is a hard problem. When the same game can be played repeatedly over time, the problem becomes even more complicated. The existence of multiple game states makes the problem of computing equilibria in such games extremely difficult. In this paper, we approach this problem by first proposing a method to compute a nonempty subset of approximate (up to any precision) subgame-perfect equilibria in repeated games. We then demonstrate how to extend this method to approximate *all* subgame-perfect equilibria in a repeated game, and also to solve more complex games, such as Markov chain games and stochastic games. We observe that in stochastic games, our algorithm requires additional strong assumptions to become tractable, while in repeated and Markov chain games it allows approximating *all* subgame-perfect equilibria reasonably fast and under considerably weaker assumptions than previous methods.

1. Introduction

Repeated interactions are studied in Biology, Economics, Ecology, Social Sciences, Computer Science and many other domains. The theory of repeated games allows us to model such interactions by considering a group of agents (or players – both are interchangeable in this paper) evolving in a strategic interaction over and over. Notice that in repeated games with a *complete information*, the data of the strategic interaction is fixed over time and is known by all the players (this is the case of this paper). Usually, to solve a multi-player game means finding a particular strategy for each player, such that the collection of players' strategies forms an *equilibrium*. Just like in a single-agent case, each agent's strategy has to be preferred by that player over any other strategy, assuming the other players' strategies and the environment characteristics remain constant.

Dynamic games, such as repeated games [1,2], or stochastic games [3,4], involve multiple stages of interactions between players. However when these games are played by players having bounded rationality and do not share all the other severe assumptions sustained by the classical theory game, it is evolutionary game theory (EGT) [2,4,5] which used as theory of dynamic adaptation and learning. EGT is generally used for evolving populations (as for instance in [6]) and in this sense it focuses more on the dynamics of strategy change as influenced not solely by the quality induced by competing strategies, but also by the frequency with which those strategies are found in the population [7]. EGT uses an equilibrium refinement of the Nash equilibrium, called *evolutionarily stable strategy* (ESS) which is "evolutionary" stable: once it is adopted by a population in a given environment, it cannot be invaded by any alternative strategy, initially rare. Many refinements to the ESS have been proposed, among them, one can cite a recent approach based on beliefs [8]. Evolution and promotion of cooperation have been extensively studied in the context of EGT. Thus, Wang et al. studied optimal interdependence between network nodes for the evolution of cooperation [9,10].

[*] Corresponding author. Tel.: +1 4186562131x3226; fax: +1 4186562324.
E-mail address: chaib@ift.ulaval.ca (B. Chaib-draa).

http://dx.doi.org/10.1016/j.amc.2015.07.068

$$\begin{array}{cc|c|c|}
 & & \multicolumn{2}{c}{\text{Player 2}} \\
 & & C & D \\ \cline{3-4}
\text{Player 1} & C & 2,\ \ 2 & -1,\ \ 3 \\ \cline{3-4}
 & D & 3,\ -1 & 0,\ \ 0 \\ \cline{3-4}
\end{array}$$

Fig. 1. The payoff matrix of prisoner's dilemma (PD).

Other researchers have studied the promotion of cooperation either by generating random variables that determine the social diversity of players engaging in the prisoner's dilemma game [4,11], by introducing coevolutionary rules [12], or by inducing appropriate payoff aspirations in a small-world networked game [13].

When dynamic games are being played by rational players, that is, selfish payoff maximizers, (and this paper is devoted to this sort of games) the collection of equilibrium strategies should possess the property of subgame perfection. In *subgame perfect Nash equilibrium* (SPNE), each agent's strategy has to be preferred by that player over any other strategy *in every possible situation* and not only in situations that can arise when every other player follows its respective equilibrium strategy. In other words, a SPNE is an equilibrium such that players' strategies constitute a Nash equilibrium in every subgame of the original game. Thus, a SPNE can be elaborated by *backward induction*, starting from the leaves players' actions and pulling up until the original game. Such backward induction eliminates *noncredible threats* which are tolerated by the Nash equilibria.

Computing SPNE is a difficult problem even for very simple forms of games, such as stage-games [14]. Often, these games and more generally dynamic games possess an infinity of equilibrium solutions and the set of solutions is usually much wider and very difficult to identify. Several successful attempts have been made to algorithmically identify a subset of, or all solutions in dynamic games. However, these attempts still have a number of important limitations. For example, the algorithm of [15] aims at finding a subset of pure action subgame-perfect equilibria (which do not always exist) in a repeated game. The algorithm of [16] aims at efficiently solving repeated games with no discounting but the assumption of no discounting makes trivial the task of identifying the set of solutions. On the other hand, the approach of [17,18] aims at computing correlated equilibria (not necessarily subgame-perfect) in stochastic games, which imposes additional restricting assumptions on the model, such as a presence of a constant and unlimited communication between players, or a mediating third party.

In this paper, we approach the problem of computing subgame-perfect equilibria in dynamic games with a complete information by first proposing a method to compute a nonempty subset of approximate subgame-perfect equilibria in any repeated game. We then demonstrate how to extend this method for approximating *all* equilibria in a repeated game, and solving more complex games, such as Markov chain games and stochastic games.

2. Example and motivation

A discussion of an example allows us to show the significance of repeated games and brings up the questions that we can address relative to these games. Probably the most famous example of a repeated game is prisoner's dilemma (PD), whose payoff matrix is shown in Fig. 1.

In this game, there are two players, called Player 1 and Player 2. At each stage of the repeated game, each player has a choice between two actions: C (for cooperation) and D (for defection, i.e., non-cooperation). When the two players simultaneously perform their actions, the resulting pair of actions induces a numerical payoff given by the payoff matrix. For example, if Player 1 plays action C and Player 2 plays action D, then the payoffs they obtain are respectively -1 and 3. The game then passes to the next stage, where it can be played again by the same pair of players.

As we can see, in one-shot interaction, the only outcome consistent with game theory prediction is (D, D) since each player is better off playing D whatever the other player does. On the other hand, if the game is repeated and the players value sufficiently future payoffs relative to the present ones, and if past actions are known, then (C, C) is an acceptable outcome for which no one wants to deviate. The reason is that if each player plays C as long as the other one has played C in the past, there is *an incentive* for both players to always play C. Indeed, the short-term outcome that can be obtained by playing D is more than offset by the future losses induced by always playing D at all future stages, that the opponent player can adopt as strategy of reprisal [19].

The utility function of the agent playing a repeated game is usually a non-decreasing function of accumulated payoffs. Game theory assumes that the goal of each player is to play rationally, i.e., to maximize its utility function. When the *a priori* information about all players' strategies and their real strategic preferences coincide, we refer to *equilibria*. A pair of "Tit-For-Tat" (TFT) strategies is a well-known example of equilibrium in the repeated prisoner's dilemma. TFT for Player i, where $i \in \{1, 2\}$, consists of starting the repeated PD game by playing action C. Then, Player i is supposed to play the same action as the very recent action played by its opponent. As explained earlier, a pair of TFT strategies constitutes an equilibrium in the repeated prisoner's dilemma if players are sufficiently patient, that is, they repeat the game with an utility function defined as a discounted sum of accumulated payoffs and the discount factor–γ–is close to 1.

Strategies such as TFT are called *non-stationary*, because they depend on the history of the repeated game. Therefore, an equilibrium strategy profile given by a pair of TFT strategies is a *non-stationary equilibrium* strategy profile. A stationary strategy, in turn, is a strategy that does not depend on the history of the repeated game. It is easy to verify that in the repeated prisoner's dilemma, a stationary equilibrium strategy profile is a pair of strategies that prescribe, to each player, to play action D at every stage of the repeated game. Indeed, if one player is supposed to always play D, it is rational, for the other player, to always play

D as well. Evidently, strategy profiles like TFT strategies can often result, for each player, in a higher utility than the utility of any equilibrium in stationary strategies.

An algorithmic construction of non-stationary equilibrium strategy profiles in an arbitrary repeated game is challenging. Indeed, in spite of high interest for the subject in both economics and computer science, there exist only a few algorithms for solving repeated games by constructing equilibrium strategy profiles. For the case, where the utility function of a player is given by the average of accumulated payoffs, Littman and Stone [16] proposed a simple and efficient algorithm that constructs equilibrium strategies in two-player repeated games.[1] On the other hand, in repeated games with discounting the problem of computing equilibria is much more complicated. To deal with this case, Judd et al. [15] proposed an algorithmic approach for computing equilibria, but their approach is only limited to equilibria in pure strategies, a small fraction of all equilibria [21]. Furthermore, the approach of [15] has several other important limitations, which we discuss further in Sections 4.2 and 7.2.

In this paper, we present a novel algorithmic approach to the problem of computing equilibria in repeated games with discounting. Our algorithmic approach is more general than that of [16] because it allows for an arbitrary discount factor, and is free of four principal limitations of [15] algorithm, as explained in Section 4.2.

Why should one compute the set of equilibria in repeated games? Because as explained earlier, repeated game framework generally adds realism to the analysis of many issues. Until now however, all efforts for computing equilibria for these games have been limited due mainly to technical difficulties. Part of these difficulties has been addressed by the Folk theorem since it stipulates: there is some discount factor γ above which any feasible and individually rational payoff vector is an equilibrium. Unfortunately, this is not enough and for many issues the question that we need to address is: *Given* a discount factor γ, what is the set of equilibria? To answer this question, we approach the problem of computing subgame-perfect equilibria (SPE) in dynamic games by first proposing a method (via an algorithm called ASPEQ) to compute a nonempty subset of approximate SPE in any repeated game. We prove that this algorithm is guaranteed to terminate in finite time and returns a nonempty set of solutions satisfying the required precision of approximation. We then extend our method for approximating *all* subgame-perfect equilibria in repeated game. Unfortunately we do not have a formal proof that this extension covers all these equilibria.

The principal difference with the previous work on computing equilibria in games is that this is the first attempt to approximately compute *all kinds of equilibria* in repeated and Markov chain games with discounted average payoffs, including pure and mixed action equilibria, stationary and non-stationary equilibria. For stochastic games however, our algorithm requires additional assumptions to become tractable.

Once the set of equilibria has been determined, it remains to construct a corresponding SPE strategy profile sustaining a given equilibrium. To this end, we propose a procedure which determines the strategy profiles as finite automata.

3. Definitions

Stage-games: A *stage-game* is a tuple $(N, \{A_i\}_{i\in N}, \{r_i\}_{i\in N})$, where N is the set of individual players ($|N| \equiv n$). Player $i \in N$ has a finite set A_i of *(pure) actions*. Each player i chooses a certain action $a_i \in A_i$; the resulting vector $a \equiv (a_i)_{i\in N}$ forms an *action profile* that belongs to the set of action profiles $A \equiv \times_{i\in N} A_i$. The action profile is then executed and the corresponding stage-game *outcome* is realized. A player specific *payoff function* r_i specifies player i's payoffs for different game outcomes. A bijection is typically assumed between the set of action profiles and the set of game outcomes. In this case, a player's payoff function is defined as the mapping $r_i : A \mapsto \mathbb{R}$.

Given $a \in A$, $r(a) \equiv (r_i(a))_{i\in N}$ is called a *payoff profile*. A *mixed action* α_i of player i is a probability distribution over its actions, i.e., $\alpha_i \in \Delta(A_i)$. A *mixed action profile* is a vector $\alpha \equiv (\alpha_i)_{i\in N}$. We denote by $\alpha_i^{a_i}$ and α^a respectively the probability of playing action a_i by player i and the probability that the outcome a will be realized by α, i.e., $\alpha^a \equiv \prod_i \alpha_i^{a_i}$. Payoff functions can be extended to mixed action profiles by considering expectations.

Dynamic games (as shown in Fig. 2), such as repeated, Markov chain and stochastic games, extend stage-games.

Repeated games: A repeated game is a dynamic game in which the same stage-game is played in *periods* (or *stages*) $t = 0, 1, 2, \ldots$. When the number of game periods is not known in advance and can be infinite, the repeated game is called *infinite*. This is the scope of the present paper.

The set of the repeated game *histories up to period t* is given by $H^t \equiv \times_t A$. The set of *all possible histories* is given by $H \equiv \bigcup_{t=0}^{\infty} H^t$. A *(mixed) strategy of player i* is a mapping σ_i: $H \mapsto \Delta(A_i)$. A *strategy profile* is a vector $\sigma \equiv (\sigma_i)_{i\in N}$. We denote by Σ_i the set of strategies of player i, and by $\Sigma \equiv \times_{i\in N}\Sigma_i$ the set of strategy profiles.

A *subgame* is a dynamic game that continues after a certain history. For a pair (σ, h), the *subgame strategy profile induced by h* is denoted as $\sigma|_h$.

An *outcome path* in the repeated game is a possibly infinite stream $\vec{a} \equiv (a^0, a^1, \ldots)$ of action profiles. Let σ be a strategy profile, the *discounted average payoff of σ for player i* is defined as

$$u_i(\sigma) \equiv (1-\gamma)\sum_{t=0}^{\infty} \gamma^t r_i(a^t), \tag{1}$$

where $\gamma \in [0, 1)$ is the *discount factor*. We define the *payoff profile induced by σ* as $u(\sigma) \equiv (u_i(\sigma))_{i\in N}$.

[1] It has recently been demonstrated by Borgs et al. [20] that for the repeated games with more than two players, no efficient algorithm for computing Nash equilibria can exist.

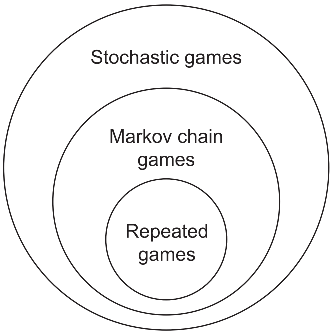


Fig. 2. Dynamic game models.

The strategy profile σ is a *(Nash) equilibrium* if, for each player i and its strategies $\sigma'_i \in \Sigma_i$,

$$u_i(\sigma) \geq u_i(\sigma'_i, \sigma_{-i}), \text{ where } \sigma \equiv (\sigma_i, \sigma_{-i}).$$

A strategy profile σ is a *subgame-perfect Nash equilibrium* (SPNE) in the repeated game if for all histories $h \in H$, the subgame strategy profile $\sigma|_h$ is a Nash equilibrium in the subgame.

Finally, a pair of strategies (σ_1, σ_2) satisfies the *one-stage deviation condition* if neither player can increase her payoff by deviating (unilaterally) from such strategy in any single stage and returning to the specified strategy thereafter. In these conditions: *a pair of strategies is a SPNE for a discounted game if and only if it satisfies the one-stage deviation condition* (proof of this can be found in [22]).

Finally, one should notice that there are a cluster of results, under the common name of *folk theorems*, characterizing the set of payoff profiles of Nash and subgame-perfect equilibria in a repeated game [1,23]. In order to illustrate the common principle behind these theorems, we present and give the complete proofs only for certain of them.

Theorem 1 (Nash folk theorem)**.** *Let $v \in F^{\dagger+}$ be a payoff profile in a repeated game. For all $\epsilon > 0$, there exists a discount factor $\underline{\gamma} \in (0, 1)$ and a payoff profile v', for which $\forall i \in N$, $|v'_i - v_i| < \epsilon$, such that for any $\gamma \in (\underline{\gamma}, 1)$, v' is a payoff profile of a certain Nash equilibrium.*

Proof. The proof of Theorem 1 for the case when $v \in F^{\dagger+p} \subseteq F^{\dagger+}$ is given in [23, p. 145]. For the general case, i.e. when $v \in F^{\dagger+}$, a similar proof can be obtained. □

A payoff profile v is said to be an *interior* feasible payoff profile if v is an interior point of $F^\dagger$, i.e., $v \in \text{int}F^\dagger$.

Theorem 2 (Perfect folk theorem)**.** *Let v be an interior feasible and strictly pure individually rational payoff profile in a repeated game. For all $\epsilon > 0$, there exists a discount factor $\underline{\gamma} \in (0, 1)$ and a payoff profile v' for which, $\forall i \in N$, $|v'_i - v_i| < \epsilon$, such that for any $\gamma \in (\underline{\gamma}, 1)$, v' is a payoff profile of a certain subgame-perfect equilibrium.*

Proof. For the proof of this theorem formulated in a slightly different form, see [24]. □

Strategy profile automata: The strategies for artificial agents should usually have a finite representation. Let $M \equiv (Q, q^0, f, \tau)$ be an *automaton implementation of a strategy profile σ*. It consists of a set of states Q, with the initial state $q^0 \in Q$; of a profile of decision functions $f \equiv (f_i)_{i \in N}$, where f_i: $Q \mapsto \Delta(A_i)$ is the decision function of player i; and of a transition function $\tau: Q \times A \mapsto Q$, which identifies the next state of the automaton given the current state and the action profile.

If $|M|$, the number of states of automaton M, is finite, M is called a *finite automaton*. Any SPNE can be approximated by a finite automaton as proven by Kalai and Stanford [25]. For an *approximation factor* $\epsilon > 0$, a strategy profile σ is an *ϵ-equilibrium*, if $\forall i \in N$ and $\forall \sigma'_i \in \Sigma_i$, $u_i^\gamma(\sigma) \geq u_i^\gamma(\sigma'_i, \sigma_{-i}) - \epsilon$, where $\sigma \equiv (\sigma_i, \sigma_{-i})$. In this case, a strategy profile σ is a *subgame-perfect ϵ-Nash equilibrium* (SPϵNE) in a repeated game, if $\forall h \in H$, $\sigma|_h$ is an ϵ-equilibrium in the subgame induced by h.

Markov chain and stochastic games: If we consider a stage-game as a state of an environment, then Markov chain games [26] and stochastic games [3] extend repeated games to multi-state environments.

A repeated game transforms into a Markov chain game if there is a set S and a transition function T: $S \times S \mapsto [0, 1]$ such that each state $s \in S$ is a certain stage-game, and $T(s, s')$ is a probability that the players will play stage-game s' after playing stage-game s. The payoff function is augmented with the state space, i.e., $r_i : S \times A \mapsto \mathbb{R}$.

Stochastic games is a more general class of dynamic games. They, in turn, extend Markov chain games by allowing players to control inter-state transitions. The transition function in stochastic games is defined as T: $S \times A \times S \mapsto [0, 1]$. The relations between different dynamic game models are shown in Fig. 2.

4. Nash reversion, continuation promise and self-generation

To compute SPE equilibria, we build upon the idea of self-generating sets, idea which refers to Nash reversion.

		Player 2	
		C	D
Player 1	C	$r(C,C)$	$r(C,D)$
	D	$r(D,C)$	$r(D,D)$

Fig. 3. A generic stage-game.

		Player 2	
		C	D
Player 1	C	$(1-\gamma)r(C,C)+\gamma u(\sigma\|_{h^t\cdot(C,C)})$	$(1-\gamma)r(C,D)+\gamma u(\sigma\|_{h^t\cdot(C,D)})$
	D	$(1-\gamma)r(D,C)+\gamma u(\sigma\|_{h^t\cdot(D,C)})$	$(1-\gamma)r(D,D)+\gamma u(\sigma\|_{h^t\cdot(D,D)})$

Fig. 4. An augmented game for the generic stage-game from Fig. 3.

4.1. Nash reversion

Similar to the GT strategy profile, any Nash reversion based strategy profile σ, in the repeated prisoner's dilemma, prescribes to start by playing a certain collaborative sequence of action profiles and then if either player deviates from the collaborative sequence, the other player reverts to permanently playing a certain stage-game Nash equilibrium. Hence, stage-game Nash equilibria are viewed as a punishment that *supports* the payoff profile corresponding to the collaborative sequence.

Let us develop this argument more formally, and let us limit ourselves to pure strategies. Given a strategy profile σ one can rewrite Eq. (1) as follows:

$$\begin{aligned} u_i(\sigma) &\equiv (1-\gamma)\sum_{t=0}^{\infty}\gamma^t r_i(a^t(\sigma)) \\ &= (1-\gamma)r_i(a^0(\sigma)) + \gamma\left[\sum_{t=1}^{\infty}\gamma^t r_i(a^t(\sigma))\right] \\ &= (1-\gamma)r_i(a^0(\sigma)) + \gamma u_i(\sigma|_{a^1(\sigma)}), \end{aligned}$$

where $a^t(\sigma)$ denotes the action profile suggested by strategy profile σ at period t of the repeated game.

Let $v_i(a_i, \sigma|_{h^t})$ denotes player i's long-term payoff for playing action a_i after history h^t, given the strategy profile σ. Let $\bar{a} \equiv (\bar{a}_i, \bar{a}_{-i})$ be the action profile prescribed by strategy σ after the history h^t, i.e., $\bar{a} \equiv \sigma(h^t) \equiv \sigma|_{h^t}(\varnothing)$. For all $a_i \in A_i$ one can write,

$$v_i(a_i, \sigma|_{h^t}) = (1-\gamma)r_i(a_i, \bar{a}_{-i}) + \gamma u_i(\sigma|_{h^{t+1}}), \tag{2}$$

where $h^{t+1} \equiv h^t \cdot a$ is a concatenation of the history h^t and the action profile $a \equiv (a_i, \bar{a}_{-i})$, and $u_i(\sigma|_{h^{t+1}})$ represents the so-called *continuation promise* of the strategy σ after the history h^{t+1} and it is defined as follows:

The continuation promise of the strategy profile σ to player i, $u_i(\sigma|_{h^{t+1}})$, is the utility of the strategy profile σ to player i if the history at the next period is h^{t+1}.

At each iteration of the repeated game, player i has a choice between different actions $a_i \in A_i$, each promising to that player a particular long-term payoff v_i. Consequently, the strategic choice at each iteration of the repeated game can be represented as a certain matrix game whose payoffs are equal to the original stage-game payoffs augmented by the corresponding continuation promises. Let us call such matrix game an *augmented game*.

For instance, let the stage-game of a repeated game be as shown in Fig. 3. Given a strategy profile σ and a history h^t, the augmented game corresponding to this stage-game is shown in Fig. 4. We can now reformulate the definition of subgame-perfect equilibrium by saying that a strategy profile σ is a subgame-perfect equilibrium if and only if it induces a stage-game Nash equilibrium in augmented games after any history. Thus, more formally,

A **Nash reversion** based strategy profile is such that the players execute a certain infinite sequence of action profiles unless one agent deviates; and following the deviation, a certain stage-game Nash equilibrium is played at every subsequent period.

Grim trigger (GT) is an example of a Nash reversion strategy. Consider the repeated prisoner's dilemma from Fig. 1. Let the strategy profile σ be a profile of two GT strategies. Consider a history h^t in which all players played C at each iteration. Now, each player has to take a decision whether to play C, as prescribed by the strategy, or to play D instead. In particular, we have: $u(\sigma|_{h^t\cdot(C,C)}) = (2,2)$.

Because, in the case of deviation, the unique stage-game Nash equilibrium (D, D), whose payoff profile is $(0, 0)$, should be played and in this case we have:

$$u(\sigma|_{h^t\cdot(D,C)}) = u(\sigma|_{h^t\cdot(C,D)}) = u(\sigma|_{h^t\cdot(D,D)}) = (0,0).$$

The augmented game corresponding to this situation is shown in Fig. 5.

Now suppose that Player 1 has deviated, then according to Eq. (2), her payoff is $v_i(D, C) = 3(1-\gamma) + 0$ which should respect $2 \geq 3(1-\gamma)$ so that (C, C) should be the "good equilibrium" and (D, D) the "bad equilibrium" used as drastic punishment. In these conditions, $2 \geq 3(1-\gamma)$ leads to $\gamma \geq 1/3$. Thus, when $\gamma \geq 1/3$, players have at their disposition the good and the bad

		Player 2	
		C	D
Player 1	C	$2, 2$	$-(1-\gamma), 3(1-\gamma)$
	D	$3(1-\gamma), -(1-\gamma)$	$0, 0$

Fig. 5. An augmented game for prisoner's dilemma from Fig. 1.

		Player 2	
		C	D
Player 1	C	$2, \ 2$	$-1, \ 3$
	D	$3, -1$	$0, \ 0$

Fig. 6. The payoff matrix of prisoner's dilemma.

equilibria. As they are rational, they should choose the good equilibrium because any deviation from that will lead to the bad one.

In prisoner's dilemma (see Fig. 6), it is sufficient to study Nash reversion based strategies, such as grim trigger. This is due to the fact that, in this game, the only stage-game equilibrium payoff profile coincides with the minmax profile for both players. Therefore, the strategy profile that prescribes playing (D, D) after any history is the *most severe subgame-perfect equilibrium punishment* available in this game. In other words, any playoff profile that can be supported by any subgame-perfect equilibrium continuation promise, can be supported by Nash reversion. However, not all games have such a property. In many games, Nash reversion is not the most severe subgame-perfect equilibrium punishment, and the set of Nash reversion based equilibria is either empty or excludes some equilibrium payoff profiles.

4.2. Self-generation

Let U denotes the set of subgame-perfect equilibrium payoffs that we want to identify. Recall Eq. (2): after history h^t, in order to make part of a subgame-perfect equilibrium strategy, action $a_i \in A_i$ has to be supported by a continuation promise $u_i(\sigma|_{h^{t+1}})$. By the definition of subgame-perfection, this must hold after any history. Therefore, if a_i makes part of a subgame-perfect equilibrium σ, then $u_i(\sigma|_{h^{t+1}})$ has to belong to U as well as $v_i(a_i, \sigma|_{h^t})$. This self-referential property of subgame-perfect equilibrium suggests a way by which one can identify the set U. The key to finding U is a construction of *self-generating sets* [27,28], that we now consider.

Let α_i^* be the best response of player i to the mixed action profile α_{-i}. Define the map B on a set $W \subset \mathbb{R}^n$ as

$$B(W) \equiv \bigcup_{(\alpha,w)\in\times_{i\in N}\Delta(A_i)\times W} (1-\gamma)r(\alpha)+\gamma w \tag{3}$$

where $\alpha \equiv (\alpha_i, \alpha_{-i})$ and w is the continuation promise that verifies, for all $i \in N$:

$$[(1-\gamma)r_i(\alpha_i, \alpha_{-i}) + \gamma w_i] \geq [(1-\gamma)r_i(\alpha_i^*, \alpha_{-i}) + \gamma \underline{w}_i], \tag{4}$$

and $\underline{w}_i \equiv \inf_{w\in W} w_i$, that is the i's minimum possible continuation value in W.

Equation (3) guarantees each player better tomorrow's payoffs to compensate for possible today's losses induced by a given strategy. Equation (4) guarantees player i a sufficient punishment imposed by the others if player i deviates from the given strategy.

Clearly, a value v is in $B(W)$ if there is some action profile $\alpha \equiv (\alpha_i, \alpha_{-i})$, and a continuation promise, $w \in W$, such that α is the value for players $(i, -i)$ of playing α today and receiving the continuation promise w tomorrow, and, for each i, she will choose to play α_i leading to $r_i(\alpha \equiv (\alpha_i, \alpha_{-i}))$, because she believes that to do otherwise will yield her the worst possible continuation payoff $\underline{w}_i$.

Let's consider an example of self-generation adapted from [29] and let limit ourselves to pure strategies with perfect monitoring. Consider the prisoner's dilemma again with *perfect monitoring* of Fig. 1.

In this game with public monitoring, there are the following public outcomes: $\{(C, C), (C, D), (D, C), (D, D)\}$.

Claim: If $\gamma \geq 1/3$, the set $W = \{(0, 0), (2, 2)\}$ is self generating.

Proof. In fact, we want to show that $(0, 0) \in B(W)$, and $(2, 2) \in B(W)$ for $\gamma \geq 1/3$. Let us consider $(0, 0)$ first. We can see that strategy profile (D, D) and its corresponding rewards $(0, 0)$ satisfy Eqs. (3) and (4) for *any* γ and the minimum of $W : \underline{w} = (0, 0)$ (notice that $\underline{w}$ is also the *minmax* of the game). Indeed,

$$0 = (1-\gamma)r_i(D, D) + \gamma w_i(D, D)$$

and for $a_i^* = D$,

$$0 \geq (1-\gamma)r_i(a_i^*, D) + \gamma \underline{w}_i.$$

Similarly for $w = (2, 2)$, we can show that (C, C) and its corresponding reward $(2, 2)$ satisfy Eqs. (3) and (4) for $\gamma \geq 1/3$ and the minimum of $W : \underline{w} = (0, 0)$. Indeed, for all $a \neq (C, C)$, then :

$$2 = (1 - \gamma) r_i(C, C) + \gamma w_i(C, C)$$

and for $a_i^* = C$, if $\gamma \geq 1/3$

$$2 \geq (1 - \gamma) r_i(a_i^*, C) + \gamma \underline{w}_i.$$

So $W \subset B(W)$ for $\gamma \geq 1/3$, meaning that $W = \{(0, 0), (2, 2)\}$ is self generating. □

Cronshaw and Luenberger[30] show that the largest fixed-point of the mapping $B(W)$ is U. In fact, any numerical implementation of $B(W)$ requires an efficient representation of the set W in a machine. Judd et al. [15] used convex sets for approximating both W and $B(W)$ as an intersection of a finite number of hyperplanes. With this in hand, each application of a $B(W)$ is reduced to solving a convex optimization problem. The algorithm of [15] permits computing subgame-perfect equilibrium payoff profiles in a given repeated game for a given discount factor. However, it has several significant limitations. The main limitations of the algorithm of [15] are as follows:

1. it is capable of computing only pure action subgame-perfect equilibria, and not mixed action SPE;
2. it assumes the existence of at least one pure action stage-game equilibrium, so that it can guarantee a non-empty result;
3. it cannot find SPE strategy profiles implementable by finite automata; and
4. it assumes sets W and $B(W)$ to be convex.

In the next section, we first present our algorithm for approximating SPE, which is free of the above four principal limitations of [15]. We then complete it with a procedure for extracting SPE strategy profiles represented in the form of finite automata.

To approximate the set of SPE payoff profiles U, we start by a set W that entirely contains U, and then we iteratively eliminate all points $w \in W$, for which we cannot find any pair $((\alpha_i, \alpha_{-i}), w') \in \Delta(A_1) \times \ldots \times \Delta(A_n) \times W$, such that for all $i \in N$

$$\begin{cases} w_i = (1 - \gamma) r(\alpha_i, \alpha_{-i}) + \gamma w_i', \text{ and} \\ w_i \geq (1 - \gamma) r_i(\alpha_i^*, \alpha_{-i}) + \gamma \underline{w}_i. \end{cases} \tag{5}$$

For all $i \in N$ and for all $a_i \in A_i$, Eq. (5) can be rewritten as:

$$\begin{cases} w_i(a_i) = (1 - \gamma) \sum_{a_{-i}} \alpha_{-i}(a_{-i}) r_i(a_i, a_{-i}) + \gamma w_i'(a_i), \text{ and} \\ w_i(a_i) \geq (1 - \gamma) \sum_{a_{-i}} \alpha_{-i}(a_{-i}) r_i(a_i^*, a_{-i}) + \gamma \underline{w}_i. \end{cases} \tag{6}$$

Prior to applying the algorithm for solving a given game, the values of the game matrix have to be adjusted to contain positive values only. This can be done by adding a sufficiently large positive constant to all values in the payoff matrix. This trick does not change the set of equilibria of the game, but allows simplifying the definition of the algorithm.

5. Solving repeated games

We approach the problem of computing subgame-perfect equilibria in repeated games by first proposing an algorithm (called ASPEQ) to compute a nonempty subset of approximate subgame-perfect equilibria in any repeated game. We then demonstrate how to extend this algorithm (i) for approximating all subgame-perfect equilibria in repeated game and (ii) solve Markov and stochastic games.

5.1. *The main procedure of* ASPEQ

Algorithm 1 outlines the basic structure of our approximate subgame-perfect equilibrium ASPEQ, a computation technique for approximating U. It starts with an initial set W entirely containing the set of SPE payoff profiles U. Set W, in turn, is represented by a union of disjoint hypercubes belonging to a set C (an example will follow). Each hypercube $c \in C$ is identified by its origin (its first coordinate) $o^c \in \mathbb{R}^n$ and by the hypercube side length l. Initially, C contains only one hypercube c, whose origin o^c is set to be a vector $(\underline{r})_{i \in N}$; the side length l is set to be $l = \bar{r} - \underline{r}$, where $\underline{r} \equiv \min_{a, i} r_i(a)$, and $\bar{r} \equiv \max_{a, i} r_i(a)$.

Algorithm 1 refines the set of payoff profiles W by an iterative process similar to value iteration [31]. Each iteration of Algorithm 1 (the loop, line 3) consists of verifying, for each hypercube c, whether it has to be eliminated from set C. To do so, the CubeSupported procedure verifies, by searching for an appropriate continuation promise w' in set W, whether c contains any point w that satisfies the conditions of Eq. (5). If CubeSupported returns False, the hypercube c is entirely withdrawn from set C. If, by the end of an iteration, no hypercube was withdrawn, each remaining hypercube is split into 2^n disjoint hypercubes with side $l/2$ (procedure SplitCubes in line 18). The process continues until, for each remaining hypercube, a stopping criterion is satisfied (procedure CubeCompleted in line 14).

Let us take an example and let us assume that in the repeated prisoner's dilemma from Fig. 1, the value of the discount factor is such that the set of subgame perfect equilibrium payoff profiles is given by the gray region, shown in Fig. 7a. We want to algorithmically identify this gray region. Thus, we split the *whole set of payoff profiles* into a set of hypercubes C, as shown in

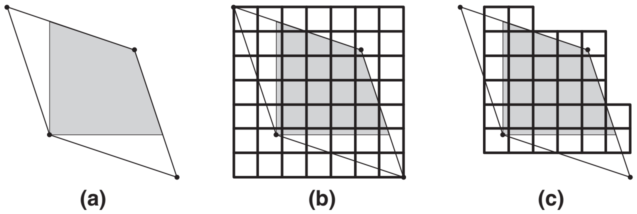


Fig. 7. An example of SPE set refinement using hypercubes.

Fig. 7b, where each cell of the grid reflects a certain two-dimensional hypercube $c \in C$. Then we verify each hypercube using the CubeSupported procedure, and we withdraw those hypercubes for which CubeSupported returns False. The resulting set of hypercubes is shown in Fig. 7c. Then, ASPEQ splits each remaining hypercube into smaller hypercubes, and the process continues.

We now define in detail two procedures used in Algorithm 1, namely CubeSupported (line 8) and CubeCompleted (line 14).

5.2. The CubeSupported procedure

In order to verify hypercube c, CubeSupported has to find an $\alpha \in \Delta(A_1) \times \ldots \times \Delta(A_n)$, a w inside c, and a $w' \in W$. However, we cannot test each tuple (α, w, w') one by one, because α, w and w' are not discrete values. We circumvent this difficulty by first defining a special mixed integer program (MIP). We then let the solver decide which actions are to be included into the mixed action support of each player, and what probability has to be assigned to those actions. Note that player i is only willing to randomize according to a suggested mixture α_i over its pure actions if it is indifferent to the pure actions in support of that mixture. The trick is to specify different continuation promises for different actions in the support such that the expected payoff of each pure action remains bounded by the dimensions of the hypercube, which we verify.

Algorithm 2 contains a definition of procedure CubeSupported for Algorithm 1. If it finds a solution, the procedure returns a mixed action profile α and the corresponding continuation promise payoffs for each action in support of α_i. Otherwise, the procedure returns False. The indifference of player i between the actions in support of α_i is (approximately) secured by the constraint (4) of the MIP in line 2 of Algorithm 2.

In an optimal solution of the MIP, any binary indicator variable $y_i^{a_i}$ can only be equal to 1 if a_i is in support of α_i. Therefore, each $w_i(a_i)$ is either bounded by the dimensions of the hypercube, if $a_i \in A_i^{\alpha_i}$, or, otherwise, is below the origin of the hypercube.

Note that the MIP in Algorithm 2 is only linear in the case of two players. For more than two players, the problem becomes non-linear because α_{-i} is now given by a product of decision variables α_j, for all $j \in N \backslash \{i\}$. Such optimization problems are known to be very difficult to solve [32]. In our experiments, we only solved linear problems and to this end, we used CPLEX [33] together with OptimJ [34].

Algorithm 1 The basic structure of ASPEQ.

```
Input: $r$, a payoff matrix; $\gamma$, $\epsilon$, the parameters.
 1: Let $l \equiv \bar{r} - \underline{r}$ and$o^c \equiv (\underline{r})_{i \in N}$;
 2: Set $C \leftarrow \{(o^c, l)\}$;
 3: loop
 4:   Set AllCubesCompleted ← True;
 5:   Set NoCubeWithdrawn ← True;
 6:   for each $c \equiv (o^c, l) \in C$ do
 7:     Let $\underline{w}_i \equiv \min_{c \in C} o_i^c$; set $\underline{w} \leftarrow (\underline{w}_i)_{i \in N}$;
 8:     if CubeSupported$(c, C, \underline{w})$ is False then
 9:       Set $C \leftarrow C \backslash \{c\}$;
10:       if $C = \varnothing$ then
11:         return False;
12:       Set NoCubeWithdrawn ← False;
13:     else
14:       if CubeCompleted$(c)$ is False then
15:         Set AllCubesCompleted ← False;
16:   if NoCubeWithdrawn is True then
17:     if AllCubesCompleted is False then
18:       Set $C \leftarrow$ SplitCubes$(C)$;
19:     else
20:       return $C$.
```

Algorithm 2 The CubeSupported procedure.

```
Input: $c \equiv (o^c, l)$, a hypercube; $C$, a set of hypercubes; $\underline{w}$ a vector of payoffs.
for each $\tilde{c} \equiv (o^{\tilde{c}}, l) \in C$ do
    Solve the following Mixed Integer Program (MIP):
    Decision variables: $w_i(a_i) \in \mathbb{R}$, $w'_i(a_i) \in \mathbb{R}$, $y_i^{a_i} \in \{0, 1\}$, $\alpha_i(a_i) \in [0, 1]$ forall $i \in \{1, 2\}$ and for all $a_i \in A_i$.
    Objective function: min $f \equiv \sum_i \sum_{a_i} y_i^{a_i}$.
    Subject to constraints:
      (1)  $\forall i : \sum_{a_i} \alpha_i(a_i) = 1$;
      For all $i$ and for all $a_i \in A_i$:
      (2)  $\alpha_i(a_i) \le y_i^{a_i}$,
      (3)  $w_i(a_i) = (1-\gamma) \sum_{a_{-i}} \alpha_{-i}(a_{-i}) r_i(a_i, a_{-i}) + \gamma w'_i(a_i)$,
      (4)  $o_i^c y_i^{a_i} \le w_i(a_i) \le l y_i^{a_i} + o_i^c$,
      (5)  $\underline{w}_i(1 - y_i^{a_i}) + o_i^{\tilde{c}} y_i^{a_i} \le w'_i(a_i) \le (\underline{w}_i + l) y_i^{a_i} + (o_i^{\tilde{c}} + l) y_i^{a_i}$.
    if a solution is found then return $w'_i(a_i)$ and $\alpha_i(a_i)$ for all $i \in \{1, 2\}$ and for all $a_i \in A_i$.
return False
```

5.3. Computing strategies and the stopping criterion

Algorithm 1 , returns a set of hypercubes C such that the union of these hypercubes gives W, a set that contains subgame-perfect equilibrium payoff profiles. Each hypercube $w \in W$ groups strategy profiles that induce similar payoff profiles. Consequently, we view hypercubes as states of an automaton. We developed a procedure ConstructAutomaton(C, v) that takes a set of hypercubes C and a payoff profile v as input and returns an automaton that approximately induces v as the payoff profile of an approximate equilibrium. To do so, the procedure first identifies the hypercube $c \in C$ that contains the point v. This hypercube defines the initial state of the automaton. Then, the procedure uses the solution of the CubeSupported MIP for the hypercube c in order to identify the decision and the transition functions in the automaton state c. The hypercubes containing the continuation promises for hypercube c are then treated in a similar way until the automaton is completely defined.

Algorithm 3 gives a formal definition of the ConstructAutomaton(C, v) procedure. The algorithm starts with an empty set of states Q. Then it puts N punishment states into this set, one for each player (lines 3–4). Here, the punishment state for player i is the automaton state, which is based on the hypercube that contains a payoff profile v, such that $v_i = \underline{w}_i$.

The transition and the decision function for any state that has just been put into Q remain undefined. From the set Q of automaton states, Algorithm 3 then iteratively picks some state q, for which the transition and the decision function have not yet been defined (line 8). Then the procedure CubeSupported is applied to state q, and a mixed action profile α and continuation payoff profiles $w(a)$ for all $a \in \times_{i \in N} A_i^{\alpha_i}$ are obtained. This mixed action profile α will be played by the players when automaton enters into state q during game play (line 10). For each $w(a)$, a hypercube $c \in C$, which $w(a)$ belongs to, is then identified. Those hypercubes are also put into the set of states Q (line 12), and the transition function for state q is finally defined (lines 13 and 15). Algorithm 3 terminates when, for all $q \in Q$, the transition function and the decision function have been defined.

The values of the flags NoCubeWithdrawn and AllCubesCompleted of Algorithm 1 determine whether ASPEQ should stop and return the solution. At the end of each iteration of Algorithm 1, the flag AllCubesCompleted is only True if, for each

Algorithm 3 The ConstructAutomaton procedure.

```
Input: $C$, a set of hypercube, such that $W$ is their union; $\tilde{v} \in W$, a payoff profile.
 1: Find a hypercube $c \in C$, whose $\tilde{v}$ belongs to; set $Q \leftarrow \{c\}$ and $q^0 \leftarrow c$;
 2: for each player $i$ do
 3:     Find $\underline{w}^i = \min_{w \in W} w_i$ and a hypercube $c^i \in C$, whose $\underline{w}^i$ belongs to;
 4:     Set $Q \leftarrow Q \cup \{c^i\}$;
 5:     Set $f \leftarrow \varnothing \mapsto \times_i \Delta(A_i)$;
 6:     Set $\tau \leftarrow \varnothing \mapsto C$.
 7: loop
 8:     Pick a hypercube $q \in Q$, for which $f(q)$ is not defined, or return $M \equiv (Q, q^0, f, \tau)$ if there are no such hypercubes.
 9:     Apply the procedure CubeSupported$(q)$ and obtain a (mixed) action profile $\alpha$ and continuation payoff profiles $w(a)$ for all
        $a \in \times_i A_i^{\alpha_i}$.
10:     Define $f(q) \equiv \alpha$.
11:     for each $a \in \times_i A_i^{\alpha_i}$ do
12:         Find a hypercube $c \in C$, whose $w(a)$ belongs to, set $Q \leftarrow Q \cup \{c\}$;
13:         Define $\tau(q, a) \equiv c$.
14:     for each $i$ and each $a^i \in (A \backslash A_i^{\alpha_i}) \times_{j \in N \setminus \{i\}} A_i^{\alpha_i}$ do
15:         Define $\tau(q, a^i) \equiv c^i$.
```

hypercube c that remains in C, the procedure CubeCompleted(c) returns True. The latter procedure verifies, by dynamic programming, the following two conditions:

1. The automaton M^c that starts in the state c induces an SPϵE.
2. The payoff profile induced by M^c is ϵ-close to o^c.

Both conditions can be verified by using the standard value iteration algorithm [31]. To do this, the deviating agent i, $\forall i \in N$, has to be considered as the only decision maker (optimizer). The remaining agents' strategy profile $\sigma_{-i}^{M^c}$ can then be viewed as the decision maker's stationary environment, represented as a Markov decision process (MDP). To construct automaton M^c, the procedure ConstructAutomaton(c) is used.

By computing the optimal strategy for player i in that MDP, as well as player i's respective payoff, one can then compare this payoff with the payoff of following the proposed automaton strategy profile, $u_i(M^c)$. If the difference is less than or equal to ϵ, then the first condition is verified. If, in turn, $u_i(M^c)$ is ϵ-close to o^c, then the second one is verified too, and the hypercube c is consequently marked as completed.

5.4. Theoretical analysis of ASPEQ and ConstructAutomaton

Theorem 3 presents the main theoretical result obtained for our ASPEQ algorithm.

Theorem 3 (The main theorem)**.** *For any repeated game, discount factor $\gamma \in [0, 1)$ and approximation factor ϵ,*

(i) ASPEQ *(i.e., Algorithm 1 with* CubeSupported *given by Algorithm 2) terminates in finite time,*
(ii) *the set of hypercubes C, at any moment, contains at least one hypercube, and*
(iii) *for any input $\tilde{v} \in W$, the* ConstructAutomaton *procedure terminates in finite time and returns a finite automaton M that satisfies:*
  (a) *the strategy profile σ^M implemented by M induces the payoff profile v, s.t. $\tilde{v}_i - v_i \leq \epsilon$, $\forall i \in N$, and*
  (b) *the maximum payoff g_i that each player i can achieve by unilaterally deviating from σ^M is such that $g_i - v_i \leq \epsilon$.*

Theorem 3 shows that our algorithm is guaranteed to terminate in finite time and returns a nonempty set of solutions satisfying the required precision of approximation. The proof of Theorem 3 relies on the six following supporting lemmas.

Lemma 1. *At any point of execution of Algorithm 1a, where* CubeSupported *is given by Algorithm 1b, C contains at least one hypercube.*

Proof. According to [35], any stage-game has at least one equilibrium. Let v be a payoff profile of a certain Nash equilibrium in the stage-game. For the hypercube c that contains v, the procedure CubeSupported will always return True, because, for any γ, v satisfies the two conditions of Eq. (3), with $w = w' = v$ and α being a mixed action profile that induces v. Therefore, c will never be withdrawn. □

Lemma 2. *An iteration of Algorithm 1a, such that* NoCubeWithdrawn *is* True, *will be reached in finite time.*

Proof. Because the number of hypercubes is finite, the procedure CubeSupported will terminate in finite time. For a constant l, set C is finite and contains at most $\lceil(\bar{r} - \underline{r})/l\rceil$ elements. Therefore, and by Lemma 1, after a finite time, there will be an iteration of Algorithm 1a, such that for all $c \in C$, CubeSupported(c) returns True. □

Lemma 3. *Let C be a nonempty set of hypercubes at the end of a certain iteration of Algorithm 1a, such that* NoCubeWithdrawn *is* True. *For all $c \in C$, Algorithm 3 terminates in finite time and returns a complete finite automaton.*

Proof. By observing the definition of Algorithm 3, the proof follows from the fact that the number of hypercubes and, therefore, the possible number of the automaton states is finite. Furthermore, the automaton is complete because the fact that NoCubeWithdrawn is True implies that for each hypercube $c \in C$, there is a mixed action α and a continuation payoff profile w belonging to a certain hypercube $c' \in C$. Consequently, for each state q of the automaton, the functions $f(q)$ and $\tau(q)$ will be defined. □

Lemma 4. *Let C be the set of hypercubes at the end of a certain iteration of Algorithm 1a, such that* NoCubeWithdrawn *is* True. *Let l be the current value of the hypercube side length. For every $c \in C$, the strategy profile σ^M, implemented by the automaton M that starts in c, induces the payoff profile $v \equiv u^{\gamma}(\sigma^M)$, such that $o_i^c - v_i \leq \frac{\gamma l}{1-\gamma}$, $\forall i \in N$.*

Proof. When player i follows the strategy prescribed by the automaton constructed by Algorithm 3 with CubeSupported given by Algorithm 1b, this process can be reflected by an *equilibrium graph* like the one shown in Fig. 8.

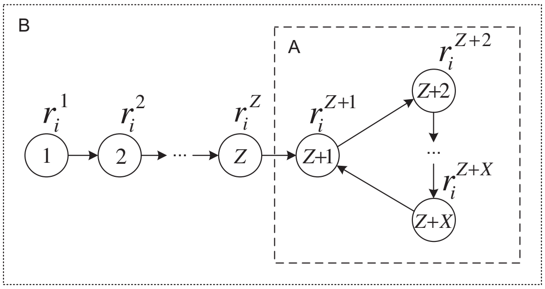


Fig. 8. Equilibrium graph for player *i*. The graph represents the initial state followed by a non-cyclic sequence of states (nodes 1 to *Z*) followed by a cycle of *X* states (nodes $Z+1$ to $Z+X$). The labels over the nodes are the immediate expected payoffs collected by player *i* in the corresponding states.

Because for all hypercubes *c* behind the states of the automaton, CubeSupported returns True, we have:

$$\begin{array}{ll}
(1.1) & o_i^1 \le (1-\gamma)r_i^1 + \gamma w_i^1 \le o_i^1 + l,\\
(1.2) & o_i^2 \le w_i^1 \le o_i^2 + l,\\
(2.1) & o_i^2 \le (1-\gamma)r_i^2 + \gamma w_i^2 \le o_i^2 + l,\\
(2.2) & o_i^3 \le w_i^2 \le o_i^3 + l,\\
\ldots & \\
(Z.1) & o_i^Z \le (1-\gamma)r_i^Z + \gamma w_i^Z \le o_i^Z + l,\\
(Z.2) & o_i^{Z+1} \le w_i^Z \le o_i^{Z+1} + l,\\
(Z+1.1) & o_i^{Z+1} \le (1-\gamma)r_i^{Z+1} + \gamma w_i^{Z+1} \le o_i^{Z+1} + l,\\
(Z+1.2) & o_i^{Z+2} \le w_i^{Z+1} \le o_i^{Z+2} + l,\\
\ldots & \\
(Z+X.1) & o_i^{Z+X} \le (1-\gamma)r_i^{Z+X} + \gamma w_i^{Z+X} \le o_i^{Z+X} + l,\\
(Z+X.2) & o_i^{Z+1} \le w_i^{Z+X} \le o_i^{Z+1} + l,
\end{array} \tag{7}$$

where o_i^q, r_i^q and w_i^q stand respectively for (*i*) the payoff of player *i* in the origin of the hypercube behind the state *q*, (*ii*) the immediate expected payoff of player *i* for playing according to $f_i(q)$ or for deviating inside the support of $f_i(q)$, and (*iii*) the continuation promise payoff of player *i* for playing according to the equilibrium strategy profile in state *q*.

The following development uses only the inequalities of Eq. (7) one by one. It starts with inequality (Z+1):

$$\begin{aligned}
o_i^{Z+1} &\le (1-\gamma)r_i^{Z+1} + \gamma w_i^{Z+1}\\
&\quad \langle \text{By inequality (Z+1.2)}\rangle\\
&\le (1-\gamma)r_i^{Z+1} + \gamma(o_i^{Z+2} + l)\\
&\quad \langle \text{By inequality (Z+2.1)}\rangle\\
&\le (1-\gamma)r_i^{Z+1} + \gamma\big((1-\gamma)r_i^{Z+2} + \gamma w_i^{Z+2}\big) + \gamma l\\
&\quad \langle \text{By inequality (Z+2.2)}\rangle\\
&\le (1-\gamma)r_i^{Z+1} + \gamma(1-\gamma)r_i^{Z+2} + \gamma^2 o_i^{Z+3} + \gamma l\\
&\quad \ldots
\end{aligned} \tag{8}$$

$$\begin{aligned}
&\langle \text{By inequality (Z+X.2)}\rangle\\
&\le (1-\gamma)\sum_{x=1}^{X}\gamma^{x-1}r_i^{Z+x} + \gamma^X o_i^{Z+1} + \gamma\sum_{x=1}^{X}\gamma^{x-1}l.
\end{aligned} \tag{9}$$

Denote by g_i^A the long-term expected non-normalized payoff for player *i* for passing through cycle *A* of the equilibrium graph infinitely often.

$$g_i^A = \sum_{x=1}^{X}\gamma^{x-1}r_i^{Z+x} + \gamma^X g_i^A \tag{10}$$

$$= \frac{\sum_{z=1}^{X}\gamma^{x-1}r_i^{Z+x}}{1-\gamma^X}. \tag{11}$$

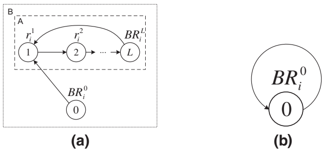


Fig. 9. Deviation graphs for player *i*. (a) A generic deviation graph for player *i*. The graph represents the initial deviation state (node 0) followed by a transition into the punishment state (node 1) followed by a number of in-equilibrium (or, otherwise, inside-the-support deviation) states (nodes 1 to $L-1$) followed by the subsequent out-of-the-support deviation state (node *L*). (b) A particular, one state deviation graph, where the only deviation state is the punishment state for player *i*. The labels over the nodes are the immediate expected payoffs collected by player *i* in the corresponding states.

The property of the infinite sum of the geometric series allows us to write:

$$\sum_{x=1}^{X} \gamma^{x-1} l = \frac{(1-\gamma^X)l}{1-\gamma}. \tag{12}$$

From Eqs. (8)–(12) it follows that,

$$o^{Z+1} \geq (1-\gamma) g_i^A + \frac{\gamma l}{1-\gamma}. \tag{13}$$

Using inequalities (1.1 - Z.2) of Eq. (7), the following development is possible:

$$\begin{aligned}
& \quad \langle \text{By inequality (1.1)} \rangle \\
o_i^1 & \leq (1-\gamma) r_i^1 + \gamma w_i^1 \\
& \quad \langle \text{By inequality (1.2)} \rangle \\
& \leq (1-\gamma) r_i^1 + \gamma (o_i^2 + l) \\
& \quad \langle \text{By inequality (2.1)} \rangle \\
& \leq (1-\gamma) r_i^1 + \gamma \big((1-\gamma) r_i^2 + \gamma w_i^2 \big) + \gamma l \\
& \quad \langle \text{By inequality (2.2)} \rangle \\
& \leq (1-\gamma) r_i^1 + \gamma (1-\gamma) r_i^2 + \gamma^2 (o_i^3 + l) + \gamma l \\
& \quad \ldots
\end{aligned} \tag{14}$$

$$\begin{aligned}
& \langle \text{By inequality (Z.2)} \rangle \\
& \leq (1-\gamma) \sum_{z=1}^{Z} \gamma^{z-1} r_i^z + \gamma^Z o_i^{Z+1} + \gamma \sum_{z=1}^{Z} \gamma^{z-1} l.
\end{aligned} \tag{15}$$

From Eqs. (13) and (14) it follows that,

$$o_i^1 \leq (1-\gamma) \left(\sum_{z=1}^{Z} \gamma^{z-1} r_i^z + \gamma^Z g_i^A \right) + \frac{\gamma l}{1-\gamma}. \tag{16}$$

Denote by g_i^B the long-term expected (normalized) payoff for player *i* for passing through the equilibrium graph (graph *B* in Fig. 9a) infinitely often. Observe that,

$$g_i^B \equiv (1-\gamma) \left(\sum_{z=1}^{Z} \gamma^{z-1} r_i^z + \gamma^Z g_i^A \right).$$

Therefore,

$$o_i^0 - g_i^B \leq \frac{\gamma l}{1-\gamma},$$

which, using the initial Lemma formulation, is equivalent to writing

$$o_i^c - v_i \leq \frac{\gamma l}{1-\gamma}.$$

□

Lemma 5. *Let C be the set of hypercubes at the end of a certain iteration of Algorithm 1a, such that* NoCubeWithdrawn *is* True. *Let l be the current value of the hypercube side length. For every $c \in C$, the maximum payoff g_i that each player i can achieve by unilaterally deviating from the strategy profile σ^M implemented by an automaton M that starts in c and induces the payoff profile $v \equiv u^\gamma(\sigma^M)$ is such that $g_i - v_i \leq \frac{2l}{1-\gamma}$, $\forall i \in N$.*

Proof. To prove the lemma, one has to bound the maximum gain of a deviation that starts in an arbitrary state of an automaton. Consider two deviation graphs for player *i* depicted in Fig. 9.

A *deviation graph for player i* is a finite graph, which reflects the optimal behavior for player *i* assuming that the behavior of the other players is fixed and is given by an automaton returned by Algorithm 3. The nodes of the deviation graph correspond to the states of the automaton. The labels over the nodes are the immediate expected payoffs collected by player *i* in the corresponding states. A *generic deviation graph for player i* (Fig. 9a) is a deviation graph that has one cyclic and one non-cyclic part. In the cyclic part (subgraph *A*), player *i* follows the equilibrium strategy or deviations take place inside the support of the prescribed mixed actions (nodes 1 to $L-1$, with node 1 corresponding to the punishment state[2] for player *i*). In the last node of the cyclic part (node *L*), an out-of-the-support deviation takes place. The non-cyclic part of the generic deviation graph contains a single node corresponding to the state where the initial out-of-the-support deviation of player *i* from the SPE strategy profile occurs. If the state of the initial deviation of player *i* is itself the punishment state for player *i*, then the deviation graph will look as shown in Fig. 9b.

The present proof only considers the generic deviation graph (Fig. 9a); the proof for the particular cases, like those of Fig. 9b, can be obtained by analogy, and because they bring the same result, we omit them here. Consider first the subgraph *A* of the generic deviation graph. Because for all hypercubes *c* behind the states of the automaton, CubeSupported returns True, we have:

$$\begin{array}{ll}
(1.0) & o_i^1 \le \underline{w}_i \le o_i^1 + l, \\
(1.1) & o_i^1 \le (1-\gamma) r_i^1 + \gamma w_i^1 \le o_i^1 + l, \\
(1.2) & o_i^2 \le w_i^1 \le o_i^2 + l, \\
(2.1) & o_i^2 \le (1-\gamma) r_i^2 + \gamma w_i^2 \le o_i^2 + l, \\
(2.2) & o_i^3 \le w_i^2 \le o_i^3 + l, \\
\ldots & \\
(\mathrm{Z}.1) & o_i^Z \le (1-\gamma) r_i^Z + \gamma w_i^Z \le o_i^Z + l, \\
(\mathrm{Z}.2) & (1-\gamma) r_i^Z + \gamma w_i^Z - (1-\gamma) BR_i^Z - \gamma \underline{w}_i \ge 0,
\end{array} \tag{17}$$

where o_i^q, r_i^q and w_i^q stand respectively for (*i*) the payoff of player *i* in the origin of the hypercube behind state *q*, (*ii*) the immediate expected payoff of player *i* for playing according to $f_i(q)$ or for deviating inside the support of $f_i(q)$, and (*iii*) the continuation promise payoff of player *i* for playing according to the equilibrium strategy profile in state *q*.

The following development only uses the inequalities of Eq. (17) one by one. It starts with inequality (1.1):

$$\begin{aligned}
o_i^1 &\ge (1-\gamma) r_i^1 + \gamma w_i^1 - l \\
&\quad \langle \text{By inequality (1.2)} \rangle \\
&\ge (1-\gamma) r_i^1 + \gamma o_i^2 - l \\
&\quad \langle \text{By inequality (2.1)} \rangle \\
&\ge (1-\gamma) r_i^1 + \gamma \big((1-\gamma) r_i^2 + \gamma w_i^2 - l \big) - l \\
&\quad \langle \text{By inequality (2.2)} \rangle \\
&\ge (1-\gamma) r_i^1 + \gamma (1-\gamma) r_i^2 + \gamma^2 o_i^3 - \gamma l - l \\
&\quad \ldots \\
&\quad \langle \text{By inequalities (3.1) to (Z-1.1)} \rangle \\
&\ge (1-\gamma) \sum_{z=1}^{Z-1} \gamma^{z-1} r^z + \gamma^{Z-1} o^Z - \gamma \sum_{z=1}^{Z-1} \gamma^{z-1} l \\
&\quad \langle \text{By inequality (Z.1)} \rangle \\
&\ge (1-\gamma) \sum_{z=1}^{Z-1} \gamma^{z-1} r^z + \gamma^{Z-1} \big((1-\gamma) r_i^Z + \gamma w_i^Z - l \big) - \gamma \sum_{z=1}^{Z-1} \gamma^{z-1} l \\
&\quad \langle \text{By inequality (Z.2)} \rangle \\
&\ge (1-\gamma) \sum_{z=1}^{Z-1} \gamma^{z-1} r^z + \gamma^{Z-1} \big((1-\gamma) BR_i^Z + \gamma \underline{w}_i - l \big) - \gamma \sum_{z=1}^{Z-1} \gamma^{z-1} l \\
&\quad \langle \text{By inequality (1.0)} \rangle \\
&\ge (1-\gamma) \left(\sum_{z=1}^{Z-1} \gamma^{z-1} r^z + \gamma^{Z-1} BR_i^Z \right) + \gamma^Z o_i^1 - \gamma \sum_{z=1}^{Z} \gamma^{z-1} l.
\end{aligned} \tag{18}$$

[2] The punishment state for player *i* is the automaton state, which is based on the hypercube that contains a payoff profile *v*, such that $v_i = \underline{w}_i$.

Denote by g_i^A the long-term expected non-normalized payoff of player i for passing through cycle A of the generic deviation graph infinitely often:

$$g_i^A = \sum_{z=1}^{Z-1} \gamma^{z-1} r_i^z + \gamma^{Z-1} BR_i^Z + \gamma^Z g_i^A \tag{19}$$

$$= \frac{\sum_{z=1}^{Z-1} \gamma^{z-1} r_i^z + \gamma^{Z-1} BR_i^Z}{1-\gamma^Z}. \tag{20}$$

The property of the infinite sum of the geometric series allows us to write:

$$\sum_{z=1}^{Z} \gamma^{z-1} l = \frac{(1-\gamma^Z)l}{1-\gamma}. \tag{21}$$

From Eqs. (18)–(21) it follows that

$$o^1 \geq (1-\gamma) g_i^A - \frac{\gamma l}{1-\gamma}. \tag{22}$$

Furthermore, we have,

$$\begin{array}{ll} (1.1) & (1-\gamma) r_i^0 + \gamma w_i^0 - (1-\gamma) BR_i^0 - \gamma \underline{w}_i \geq 0, \\ (1.2) & o_i^0 \leq (1-\gamma) r_i^0 + \gamma w_i^0 \leq o_i^0 + l. \end{array} \tag{23}$$

The following development is possible:

$$\begin{aligned} & \langle \text{By inequality (1.1) of Eq. (23)} \rangle \\ (1-\gamma) r_i^0 + \gamma w_i^0 & \geq (1-\gamma) BR_i^0 + \gamma \underline{w}_i \\ & \langle \text{By inequality (1.2) of Eq. (23)} \rangle \\ o_i^0 & \geq (1-\gamma) BR_i^0 + \gamma \underline{w}_i - l \\ & \langle \text{By inequality (1.0) of Eq. (17)} \rangle \\ & \geq (1-\gamma) BR_i^0 + \gamma o_i^1 + \gamma l - l \end{aligned} \tag{24}$$

From Eqs. (22) and (24) it follows that

$$o_i^0 \geq (1-\gamma) BR_i^0 + \gamma \left((1-\gamma) g_i^A - \frac{\gamma l}{1-\gamma} \right) + \gamma l - l. \tag{25}$$

Denote by g_i^B the long-term expected (normalized) payoff of player i for following the generic deviation graph (graph B in Fig. 9a). Observe that,

$$g_i^B \equiv (1-\gamma) BR_i^0 + \gamma (1-\gamma) g_i^A.$$

Therefore,

$$g_i^B - o_i^0 \leq \frac{\gamma^2 l}{1-\gamma} - \gamma l + l.$$

Finally, by Lemma 4, starting from the state that corresponds to the node 0 of the generic deviation graph, the payoff profile v, induced by the automaton, satisfies: $o_i^0 \leq \frac{\gamma l}{1-\gamma} + v_i$. Therefore,

$$\begin{aligned} g_i^B - v_i & \leq \frac{\gamma^2 l}{1-\gamma} + \frac{\gamma l}{1-\gamma} - \gamma l + l \\ & \leq \frac{2l}{1-\gamma}. \end{aligned}$$

□

Lemma 6. *Let* CubeSupported *be given by Algorithm 1b. Algorithm 1a terminates in finite time.*

Proof. The hypercube side length l is reduced by half every time no hypercube has been withdrawn by the end of an iteration of Algorithm 1a. Therefore, and by Lemma 2, any given value of l will be reached after a finite time. By Lemmas 4 and 5, Algorithm 1a, in the worst case, terminates when l becomes lower than or equal to $\frac{\epsilon(1-\gamma)}{2}$. □

By combining the above Lemmas 1–6 we obtain the proof of our main theorem: point (i) of the statement follows from Lemma 6; point (ii) follows from Lemma 1; and point (iii) follows from Lemmas 3–5.

The assumption that all continuation payoff profiles for hypercube c are contained within a certain hypercube $\tilde{c} \in C$ (Algorithm 1b) makes the optimization problem linear and, therefore, easier to solve. However, this restricts the set of equilibria that can be approximated using this technique. Furthermore, examining prospective continuation hypercubes one by one

Algorithm 4 CubeSupported with public correlation.

```
Input: $c \equiv (o^c, l)$, a hypercube; $C$, a set of hypercubes.
1: $P \leftarrow$ GetHalfplanes(C);
2: Solve the following linear MIP:
   Decision variables: $w_i(a_i) \in \mathbb{R}, w'_i(a_i) \in \mathbb{R}, y_i^{a_i} \in \{0,1\}, \alpha_i^{a_i} \in [0,1]$ for all $i \in \{1,2\}$ and for all $a_i \in A_i$; $z^{a_1,a_2} \in \{0,1\}$ for all pairs $(a_1,a_2) \in A_1 \times A_2$;
   Objective function: min $f \equiv \sum_{(a_1,a_2)\in A_1\times A_2} z^{a_1,a_2}$;
   Subject to constraints:
       (1) $\sum_{a_i} \alpha_i^{a_i} = 1, \forall i \in \{1,2\}$;
       For all $i \in \{1,2\}$ and for all $a_i \in A_i$:
       (2) $\alpha_i^{a_i} \leq y_i^{a_i}$,
       (3) $w_i(a_i) = (1-\gamma)\sum_{a_{-i}} \alpha_{-i}(a_{-i}) r_i(a_i, a_{-i})$
           $+\gamma w'_i(a_i)$,
       (4) $o_i^c y_i^{a_i} \leq w_i(a_i) \leq l y_i^{a_i} + o_i^c$,
       (5) $\underline{w}_i - \underline{w}_i y_i^{a_i} \leq w'_i(a_i) \leq (\underline{w}_i + l)$
           $-(\underline{w}_i + l) y_i^{a_i} + \bar{r} y_i^{a_i}$;
       $\forall a_1 \in A_1$ and $\forall a_2 \in A_2$:
       (6) $y_1^{a_1} + y_2^{a_2} \leq z^{a_1,a_2} + 1$;
       $\forall p \equiv (\phi^p, \psi^p, \lambda^p) \in P$ and $\forall (a_1,a_2) \in A_1 \times A_2$:
       (7) $\phi^p w'_1(a_1) + \psi^p w'_2(a_2) \leq \lambda^p z^{a_1,a_2}$
           $+M - M z^{a_1,a_2}$.
3: if a solution is found then return $w'_i(a_i)$ and $\alpha_i^{a_i}$ for all $i \in \{1,2\}$ and for all $a_i \in A_i$.
4: return False
```

is computational time-consuming: the worst-case complexity of one iteration of Algorithm 1a is $O(|C|^2)$, assuming that solving one MIP takes a unit time. We now present two extensions of our algorithm that first allow us to entirely approximately U^γ in repeated games and then to solve more complex dynamic games.

6. Extensions

6.1. Approximating all repeated game equilibria

By assuming the set of continuation promises to be convex, one can guarantee that *all* realizable SPNE payoff profiles will be preserved in W. A convexification of the set of continuation payoff profiles can be done in different ways, one of which is public correlation. In practice, this implies the existence of a certain random signal observable by all players after each repeated game iteration [1].

Algorithm 4 contains the definition of the CubeSupported procedure that convexifies the set of continuation promises. The definition is given for two players. The procedure first identifies co$W$, the smallest convex set containing all hypercubes of set C (procedure GetHalfplanes). This convex set is represented as a set P of half-planes. Each element $p \in P$ is a vector $p \equiv (\phi^p, \psi^p, \lambda^p)$, s.t. the inequality $\phi^p x + \psi^p y \leq \lambda^p$ identifies a half-plane in a two-dimensional space. The intersection of these half-planes gives co$W$. Set P can be found using Graham scan [36].

The procedure CubeSupported defined in Algorithm 4 convexifies set W and searches for continuation promises for the hypercube c inside co$W$. New indicator variables, z^{a_1,a_2}, for all pairs $(a_1, a_2) \in A_1 \times A_2$, are introduced. The new constraint (6), jointly with the modified objective function, verify that $z^{a_1,a_2} = 1$, if and only if $a_1 \in A_1^{\alpha_1}$ and $a_2 \in A_2^{\alpha_2}$. Constraint (7) verifies that $(w_1(a_1), w_2(a_2))$, the continuation promise payoff profile, belongs to co$W$ if and only if $(a_1, a_2) \in A_1^{\alpha_1} \times A_2^{\alpha_2}$. Note that in the constraint (7), M stands for a sufficiently large number. This is a standard trick for relaxing any constraint.

6.2. Solving Markov chain games

Markov chain games can be solved similar to repeated games. We first assign a separate set $W(s)$ to each state $s \in S$ of the environment. In each state, we start by a set $W(s)$ that entirely contains $U^\gamma(s)$, the set of SPNE that starts in state s of the environment. We then do a value iteration-like process: we update the states of the environment one by one. In each state s, we iteratively eliminate all points $w(s) \in W(s)$, for which we cannot find any vector $(\alpha, (w(s'))_{s'\in S}) \in \times_{i\in N}\Delta(A_i) \times_{s\in S} W(s)$, such that

$$\begin{cases} w(s) = (1-\gamma) r(s,\alpha) + \gamma \sum_{s'\in S} T(s,s') w(s') \\ w_i(s) - (1-\gamma) r_i(s, BR_i(s,\alpha), \alpha_{-i}) - \gamma \sum_{s'\in S} T(s,s') \underline{w}_i(s') \geq 0, \ \forall i. \end{cases} \tag{26}$$

	R	P	S
R	$0, 0$	$-1, 1$	$1, -1$
P	$1, -1$	$0, 0$	$-1, 1$
S	$-1, 1$	$1, -1$	$0, 0$

(a)

	O	F
O	$1, 2$	$0, 0$
F	$0, 0$	$2, 1$

(b)

	A	B
A	$1, 0$	$2, 1$
B	$0, 3$	$3, 2$

(c)

Fig. 10. Four payoff matrices: (a) Rock, Paper, Scissors, (b) Battle of the Sexes, and (c) Game with no pure action Nash equilibrium in stage-game.

The remaining part of the algorithm for Markov chain games is similar to that for repeated games. It is straightforward to adopt both formulations of the CubeSupported procedure (with and without public correlation) to Markov chain games.

6.3. Solving stochastic games

Stochastic games add an additional difficulty to the problem. We can modify the conditions of Eq. (26) to reflect a more general transition function:

$$\begin{cases} w(s) = (1-\gamma)r(s,\alpha) + \gamma \sum_{a\in A}\sum_{s'\in S} T(s,a,s')\alpha^a w(s') \\ w_i(s) - (1-\gamma)r_i(s,\hat{\alpha}(s,i)) - \gamma \sum_{a\in A}\sum_{s'\in S} T(s,a,s')\hat{\alpha}(s,i)^a \underline{w}_i(s') \geq 0,\ \forall i, \end{cases} \tag{27}$$

where $\hat{\alpha}(s,i) \equiv (BR_i(s,\alpha), \alpha_{-i})$. However, the problem here is complicated by the presence of the nonlinear terms $\alpha^a w(s)$ and $\hat{\alpha}(s,i)^a \underline{w}_i(s)$. The quadratic optimization problem defined in this form has a non-convex objective and cannot be solved using standard optimization techniques.

We can keep the problem linear if we adopt a modified CubeSupported procedure given by Algorithm 1b. If fact, we can assume that $w(s)$ is not a decision variable and is given by the extreme points of the hypercube $\tilde{c}(s) \in C(s)$. As l, the hypercube side length, tends to 0, the error induced by this assumption vanishes.

The main drawback of this approach is its low scalability. The worst-case complexity of verifying one hypercube grows quadratically with the number of hypercubes and is exponential in the number of stochastic game states. So, the solution of even very small stochastic games is intractable for more refined approximations. However, if we make a stronger assumption of existence of a third-party mediator, like, for example, in [17,18], our algorithm can be transformed into the one for finding subgame-perfect correlated equilibria. The problem of finding correlated equilibria is greatly simplified by the fact that the mediator is supposed to be capable of sampling player's actions from a unique distribution over action-profiles. The latter property makes the optimization problem for verifying hypercubes linear, and we obtain the results similar to these obtained by Dermed and Isbell [17], but also having the property of subgame-perfection.

7. Experimental results

7.1. Experiments

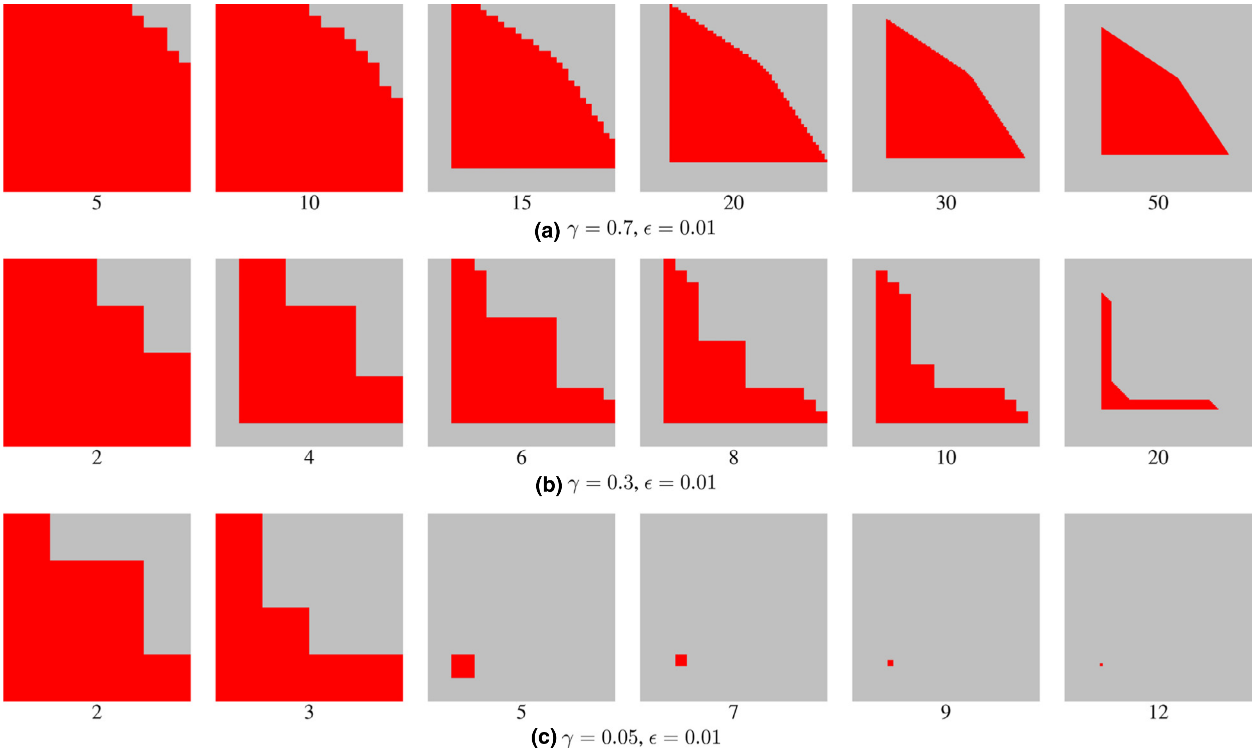
The repeated games we studied were prisoner's dilemma (Fig. 1), Rock, Paper, Scissors (Fig. 10a), Battle of the Sexes (Fig. 10b), and a game with no stage-game pure action equilibrium (Fig. 10c). For these games, equilibrium properties are known or can readily be verified analytically.

The graphs in Fig. 11 reflect, for three different values of the discount factor, the evolution of the set of SPE payoff profiles in the repeated prisoner's dilemma, computed by ASPEQ assuming public correlation. Here and below, the vertical and the horizontal axes of each graph correspond respectively to the payoffs of the first and second players. The upper and lower limits of each axis are given respectively by $\bar{r}$ and $\underline{r}$ (i.e., the maximum and the minimum immediate payoffs that can be obtained in the repeated game by any player). The numbers under the graphs reflect the iterations of ASPEQ. The red (darker) regions on a graph reflect the hypercubes that remain in set C by the end of the corresponding iteration.

One can see in Fig. 11a that when γ is sufficiently close to 1, the algorithm maintains a set that converges toward set $F^{\dagger *}$ of feasible and individually rational payoffs, the largest possible set of SPE payoff profiles in any repeated game. On the other hand, in Fig. 11c one can see that when γ is close to 0 the set of SPE payoff profiles converges, as expected, toward the point (0, 0) that corresponds to stationary Nash equilibrium: a strategy profile that prescribes playing D at every repeated game period.

Rock, paper, scissors (RPS) is a symmetric zero-sum game. In the repeated RPS game, point (0, 0) is the only possible SPE payoff profile, regardless of the discount factor. This payoff profile can be realized by a stationary strategy profile prescribing that each player sample actions from the uniform distribution. The graphs in Fig. 12 certify the correctness of ASPEQ in this case.

Battle of the Sexes (BotS) is the game that has two pure action stage-game equilibria, (O, O) and (F, F), with payoff profiles respectively (1, 2) and (2, 1). The game also has one mixed action stage-game equilibrium with payoff profile (2/3, 2/3). When γ is sufficiently close to 0, the set of SPE payoff profiles computed by ASPEQ converges toward these three points (Fig. 13b), which is the expected behavior. As γ grows, the set of SPE payoff profiles becomes larger (Fig. 13a). We also ascertained that when the value of γ becomes sufficiently close to 1, the set of SPE payoff profiles converges toward $F^{\dagger *}$ and eventually includes point



(a) $\gamma = 0.7, \epsilon = 0.01$

(b) $\gamma = 0.3, \epsilon = 0.01$

(c) $\gamma = 0.05, \epsilon = 0.01$

Fig. 11. The evolution of the set of SPE payoff profiles computed by ASPEQ with public correlation in the repeated prisoner's dilemma. The numbers under the graphs reflect the algorithm's iterations. The red (darker) regions denote the hypercubes that remain in the set *C* by the end of the corresponding iteration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

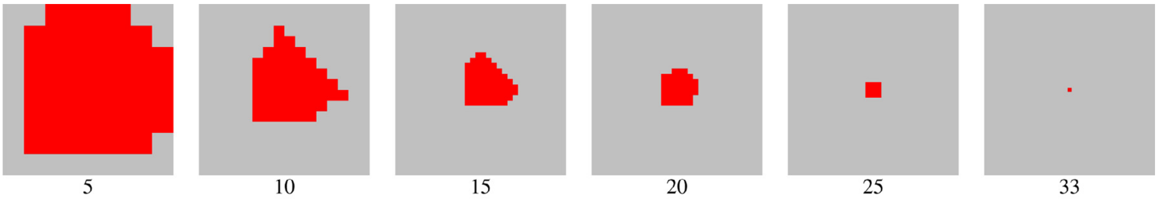
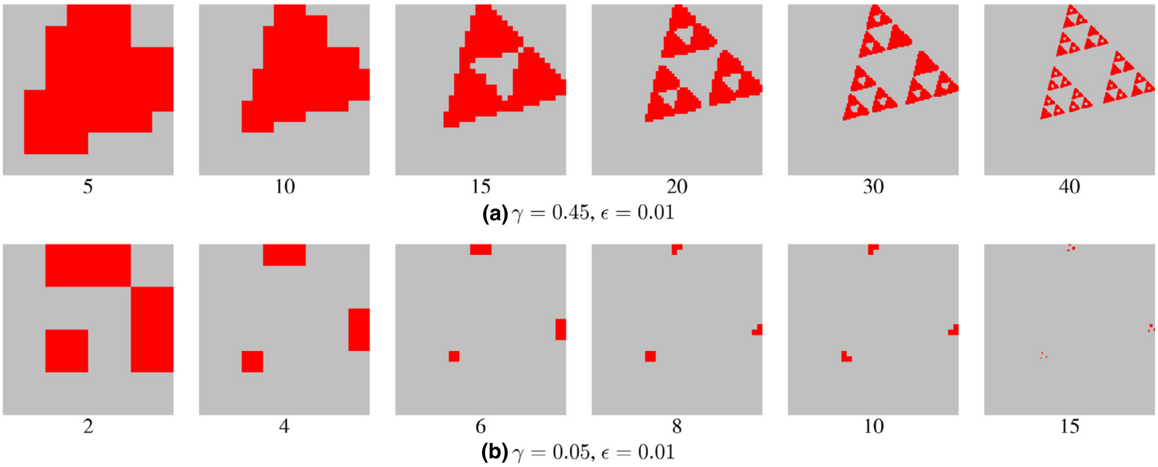


Fig. 12. The evolution of the set of SPE payoff profiles computed by ASPEQ in the repeated Rock, Paper, Scissors with $\gamma = 0.7$ and $\epsilon = 0.01$.



(a) $\gamma = 0.45, \epsilon = 0.01$

(b) $\gamma = 0.05, \epsilon = 0.01$

Fig. 13. The evolution of the set of SPE payoff profiles computed by ASPEQ without public correlation in the repeated Battle of the Sexes.

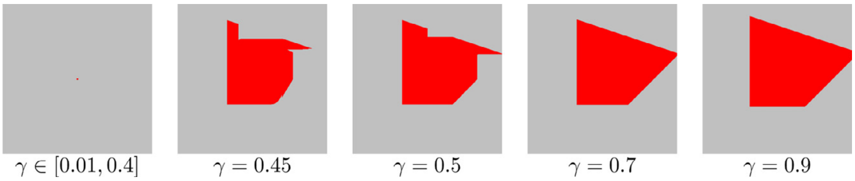

$\gamma \in [0.01, 0.4]$ $\gamma = 0.45$ $\gamma = 0.5$ $\gamma = 0.7$ $\gamma = 0.9$

Fig. 14. The sets of SPE payoff profiles computed in the repeated game from Fig. 10d with $\epsilon = 0.01$ for different values of the discount factor.

Table 1
The performance of ASPEQ in the repeated Battle of the Sexes for different values of the approximation factor ϵ.

ϵ	l	Iterations	Time
0.025	0.008	55	1750
0.050	0.016	41	770
0.100	0.031	28	165
0.200	0.063	19	55
0.300	0.125	10	19
0.500	0.250	5	15

Table 2
Comparison ASPEQ with Judd et al. approach [15].

	[15]	ASPEQ & ConstructAutomata
Pure and mixed action SPE	Only pure	Yes
Stationary and non-stationary EQ	No	Yes
Existence of pure one stage-game EQ	Required	Non-required
Assume B and $B(W)$ convex	Yes	No
For games that do not possess any pure action stationary equilibrium	No solution	Return a set of SPE
Constructs a strategy profile as a finite automaton from an ϵ-SPE	No	Yes

(3/2, 3/2). The latter point is interesting in that it maximizes the Nash product [37]. Such equilibrium points are often preferred over other Pareto-efficient equilibrium points because they optimize social welfare [16].

Another experiment was conducted with the game that does not possess any pure action stationary equilibrium (Fig. 10(c)). In such games, for lower discount factors, the algorithm of [15] (capable of computing only pure action payoff profiles) is incapable of returning any solution. On the other hand, ASPEQ without public correlation does return a nonempty SPE set for the whole range of values of the discount factor (Fig. 14).

Finally, the numbers in Table 1 demonstrate how different values of the approximation factor ϵ impact the performance of ASPEQ in terms of (i) the number of iterations until convergence, and (ii) the time spent by the algorithm computing a set of solutions. The game chosen for this experiment was the repeated Battle of the Sexes from Fig. 10c.

7.2. Comparison with other computing methods

As we said earlier, to the best of our knowledge, the only other work on computing subgame perfect-equilibria for the discounted repeated games is the [15] work. However and conversely to this approach, our algorithm neither makes an assumption about the existence of a pure action stage-game equilibrium, nor about the convexity of the set of SPE payoff profiles. It is capable of computing pure action SPE as well as mixed action SPE in any repeated game for any given value of the discount factor. Furthermore, for a given SPE payoff profile, it constructs an SPE strategy profile, and returns it in the form of a finite automaton. Table 2 compares the two approaches.

Notice that our algorithm finds more solutions than of [15] and consequently we did not find it opportune to compare their performances in terms of space and time.

8. Applications

There are many applications where repeated games of perfect monitoring have been used to address questions relative to economic behavior [1], as well as to sciences and engineering. This section gives a general overview of how our ASPEQ algorithm and constructing automata procedure can be used in relation to some of these applications.

		Firm 2		
		L	M	H
G_d Firm 1	L	$10, 10$	$3, 15$	$0, 7$
	M	$15, 3$	$7, 7$	$-4, 5$
	H	$7, 0$	$5, -4$	$-15, -15$

Fig. 15. Payoff matrix of both firms in the Duopoly.

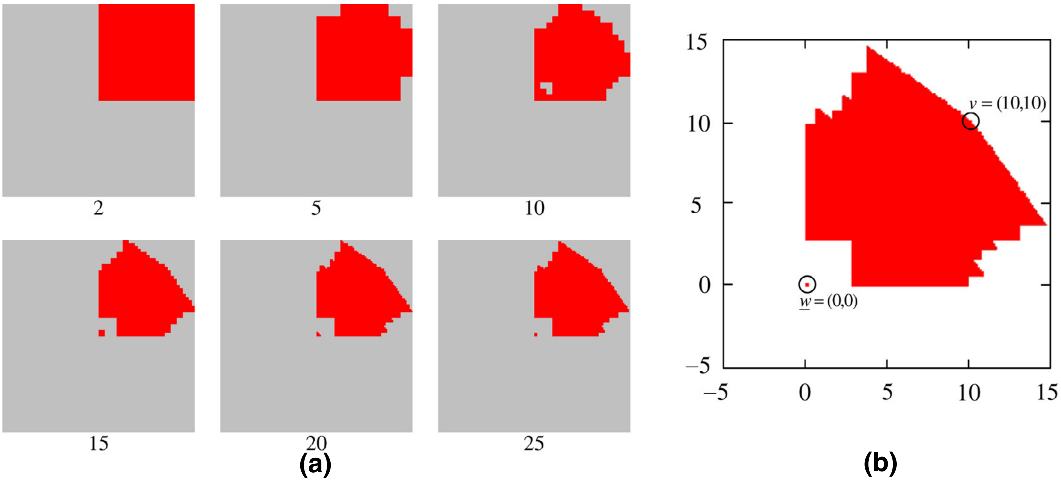


Fig. 16. SPE payoff profiles in the repeated Duopoly game computed by ASPEQ limited to pure action strategies with $\gamma = 0.6$ and $\epsilon = 0.01$. (a) The evolution of the set of SPE payoff profiles through different algorithm's iterations. (b) Abreu's optimal penal code solution is contained within the set of SPE payoff profiles returned by our algorithm.

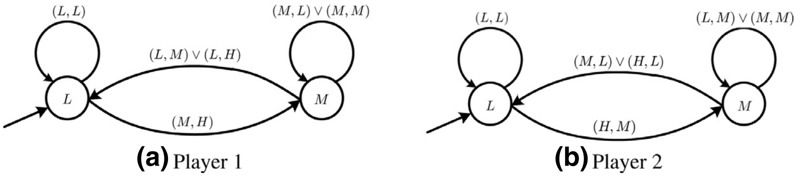


Fig. 17. A finite automaton (Computed via The ConstructAutomaton Algorithm 3) in the Duopoly game from Fig. 15.

8.1. Collusion between interacting agents

Consider the Duopoly game, introduced by Abreu [38] and presented in Fig. 15. It reflects an illustration of repeated interactions between two firms. The G_d stage game may be seen as a, symmetric discrete, quantity-setting duopoly game in which Firm-1 and Firm-2 may have “Low”, “Medium” or “High” output level (of production). As we can see G_d has an unique Nash equilibrium (M, M).

Let us now repeat the stage game G_d and denote by $G_d^{\infty}(\gamma)$ the infinitely repeated game, where γ is the discount factor. In this case, the payoff profile $v = (10, 10)$ corresponding to the pure action profile (L, L) is clearly the only Pareto efficient payoff profile in this repeated game, yielding the equal (symmetric) per player payoffs. Let us suppose that we want to determine the set of all subgame-perfect equilibria, for a given discount factor γ, and construct an automaton profile inducing a subgame-perfect equilibrium strategy profile supporting the “collusive payoff profile” $v = (10, 10)$. One possibility is to resort to penal codes as [38] did, but penal codes are generally limited [39]. Let us now first check how ASPEQ can deal with the problem of duopoly $G_d^{\infty}(\gamma)$ as it has been proposed by Abreu [38].

Our ASPEQ algorithm and the constructing automata procedure are appropriate here. First, our ASPEQ limited to considering only pure strategies preserves the point $(10, 10)$ in the set of SPE payoff profiles (see Fig. 16a;b). Abreu et al. [38] showed that this point can only make part of the set of SPE payoff profiles if $\gamma > 4/7$. In our experiments, we observed that for $\epsilon = 0.01$, the point $v = (10, 10)$ indeed remains in the set of SPE payoff profiles when $4/7 < \gamma < 1$. Moreover, the payoff profile $w = (0, 0)$ of the optimal penal code does also remain there (Fig. 16a;b).

ASPEQ also returns an automaton (Fig. 17) that induces a strategy profile that generates the payoff profile $(10, 10)$. Interestingly, this automaton induces a strategy profile, which is equivalent to the optimal penal code based strategy profile proposed by Abreu [38]. To the best of our knowledge, this is the first time that optimal penal code based equilibrium strategy profiles, which so far were only proven to exist (in the general case), were algorithmically computed.

		Player 2	
		C	D
Player 1	C	$1, \quad 1$	$-c, \quad b$
	D	$b, \quad -c$	$0, \quad 0$

Fig. 18. The payoff matrix of a parameterized prisoner's dilemma.

8.2. Bargaining and repeated games with negotiations

On the Pareto frontier, one can find generally many equilibria, and these equilibria are characterized by the fact that the gain of one is necessary a loss for others. In this context, how do players decide which Pareto-optimal equilibrium to select? It seems that *negotiation* is often used to enforce cooperation and efficient outcomes. Starting for these considerations, the question which raises is: can Bargaining be used to select an equilibria in repeated games?

Unfortunately the answer to this question is 'no' as demonstrated by Busch and Wen [40]. Indeed, these authors have shown that results, as induced by Folk theorem, persist even if the players have the opportunity to agree on an efficient equilibria. To show that, they introduced a bargaining game where, in each period, two players bargain (in Rubinstein's alternating-offers procedure) over the distribution of a period surplus which is commonly known. If an offer is rejected and before the game moves to the next period, the players engage in a repeated disagreement game to determine their outcomes for that period. This specific disagreement game, called "negotiation game", (i) has a Pareto frontier which is contained in the bargaining frontier and (ii) admits also a large number of subgame equilibria. More specifically, Busch and Wen [40] show that, *provided the players are sufficient patient, the negotiation game in general has a continuum of equilibria which will involve delay in agreement and inefficiency*.

Recently [41,42] have contributed to enrich the issue of how bargaining can be used to select (efficient) equilibria in repeated games. Their work departs from the usual rationality paradigm and introduces the notion of *complexity* into the negotiation game. With this notion, the equilibrium strategies supporting inefficient outcomes are too complex to implement and consequently they are unnecessarily to consider. According to [41], bargaining extended to the player's preferences for less complex strategies (even at the margin) select only *efficient* outcomes in the repeated game when the payers are sufficiently patient. Notice that an efficient outcome or efficient equilibrium is an equilibrium that maximizes the sum of players' utilities, disregarding the monetary payments that agents are required to make [43].

In repeated games played by automata, the number of states of the machine is often used as a measure of complexity [44–47]. This is because Kalai and Stanford [25] have shown that the counting-state measure of complexity is equivalent to looking at the number of continuation strategies that the strategy induces at different histories of the game. Starting from this, Lee and Sabourian [41,42] have extended the complexity so that one can include transaction costs where players should pay a small participation cost in each period of the negotiation game. If at least one player foregoes the payment, there is no bargaining and one should proceed to the disagreement costs. Lee and Sabourian [41] came up to the following result:

> "When each player has a preference for less complex strategies (at the margin), only *efficient equilibria* arise in complete information models of bargaining/negotiation without transaction costs while, perpetual disagreement, and inefficiency, are the only possible features of an equilibrium outcome with arbitrary small transaction costs".

As we can see if we adapt our ASPEQ algorithm and the ConstructAutomaton procedure so that they can take in consideration the preferences of players for less complex strategies, one can then use them for determining the "efficient equilibria" in complete information models of bargaining/negotiation without transactions costs. We left this for future work.

8.3. Emergence and maintenance of cooperation

The emergence and maintenance of cooperation between selfish players is an essential question not only in biology, economics, and social sciences, but also in computer sciences and physics. As we have shown in this paper the discount factor plays a crucial role in determining the set of possible supergame payoffs and consequently on the cooperation between agents. If we refer to the PD illustrated in Fig. 1, we denote by 'COOPERATION' the symmetric Pareto optimal outcome (C, C) and all other outcomes (except (D, D) forever) involve some form of 'cooperation' (as adopted by Stahl [48]).

If the players adopt a rate of time preference r and there is an exogenous termination probability β, then the discount factor $\gamma = (1 - \beta)/(1 + r)$ can serve as a parameter for ASPEQ so that we can see how the set of SPE (and then 'COOPERATION' as well as 'cooperation') evolves in function of γ (i.e, in function of β and r). By doing so we have a clear picture for intermediate values of the discount factors, that is, values between $\gamma \to 0$ and $\underline{\gamma}$ (see "Repeated Games" of Section 3).

A complete parametrization as proposed by Stahl [48] can also be done. In this case, the classic one-shot PD is specified by two parameters a and b so that we can cover the family of PDs (see Fig. 18).

As it is a PD, we need $b > 1$ and $c > 0$, so that we can have D as a dominant strategy. In addition, (C, C) should be strictly Pareto-efficient and consequently $b < C + 2$.

Now the PD and consequently the cooperation is parameterized by (a, b, γ) and we can compute the set of SPE equilibria, using ASPEQ, and see the shape and the different regions of this set in function of these parameters. By doing so, we can see how cooperation emerges and keeps up in function of all these parameters.

In practice, especially in online social network settings, cooperation is often aided by the introduction of explicit *reputation* mechanisms. These mechanisms offer an effective way of ensuring a *level of trust* which is become essential in electronic commerce. They are generally based on the repeated interactions where one remembers misconduct, cheating, etc. and then changes her terms of business accordingly in the future. Therefore and according to what we have seen on discounted repeated games in this paper, the expected gains due to future transactions in which the player has a higher reputation should offset the loss incurred by not cheating in the present [49].

However, most of the mechanisms, elaborated in this context, suffer from a variety of drawbacks as: barrier-to-enter, issues with illicitely feedback and difficulties ensuring honest reports [50]. Some of these issues can be circumvented through the use of mechanism design as proposed by Jurca and Faltings [49]. According to these authors an efficient algorithm that allows to find the SPE set will certainly help to improve the upper bound on the number of times the provider delivers low quality in any equilibrium. Our ASPEQ might be such algorithm, in future work, we will check this carefully.

8.4. Predicting the behavior of artificial and human agents

Another practical application of the Algorithms presented in this paper, consists of using them *to predict the behavior of artificial and human players* in repeated games. Recently, several authors presented their results in this direction. Engle-Warnick et al. [51] proposed modeling human play in a repeated game using finite automata. They first handcrafted a set of candidate automata for a given game and then derived the most likely automata using a Markov chain Monte Carlo (MCMC) technique. Our approximate SPE computation approach would permit to automatically generate the candidate automata based only on the game matrix and a predicted discount factor of the player. More recently, [52], in turn, explored existing models suitable for practical prediction of human play in stage-games.

9. Conclusion and future work

In this paper, we addressed the problem of computing subgame-perfect equilibria (SPE) in discounted repeated games. We first discussed the existing approaches for solving this problem, and pointed out their principal limitations. We then presented our novel algorithm, called ASPEQ, for approximating, up to any precision, a set of subgame-perfect equilibria in repeated games with discounting.

Our ASPEQ algorithm gradually refines the set of SPE payoff profiles in the repeated game, by partitioning the set of payoff profiles into a set of adjacent hypercubes. The process starts with a single hypercube that contains all realizable payoff profiles. Then the initial hypercube is gradually partitioned into a set of smaller hypercubes, while those hypercubes that do not contain equilibrium points are withdrawn. Whether a given hypercube can contain an equilibrium point is verified by an appropriate mixed integer program.

We validated ASPEQ both theoretically and experimentally. The theoretical analysis led to a conclusion that our algorithm terminates in finite time and always returns a non-empty subset of approximate SPE payoff profiles in any repeated game with discounting. We then proposed an extension of ASPEQ which is based on the notion of public correlation, for computing all approximate subgame-perfect equilibria in a repeated game. We also described a procedure for extracting SPE strategy profiles represented in the form of finite automata. From the experimental validation, in turn, we concluded that the extended ASPEQ scales well in the number of players' actions and dynamic game states.

In this paper, we adopted an usual assumption that the discount factor, γ, is the same for all players. However, ASPEQ can readily be modified to incorporate player specific discount factors. Furthermore, for simplicity of presentation, we assumed that the hypercube side length, l, is the same for all players. This is also not a strict requirement; it is straightforward to generalize our algorithm and theoretical results to the case of player specific hypercube side lengths.

The extended version of ASPEQ, based on the CubeSupportedPC procedure for verifying hypercubes, assumes the presence of a source of a commonly observed random signal (public correlating device) *or* communication between players. A question would be why not aiming, in this case, at computing a richer set of *subgame perfect correlated equilibrium* (SPCE) payoff profiles [53]. Indeed, several algorithms for computing correlated equilibria have recently been proposed [17,54]. Indeed, ASPEQ can be transformed into an algorithm for approximating the set of SPCE payoff profiles. In this case, the mathematical programming problem for the CubeSupported procedure will be even simpler than that for SPE. This is due to the fact that for computing a correlated equilibrium, one has to find a *unique* probability distribution over action profiles, and not a profile of probability distributions whose superposition enforces equilibrium.

In our approach, the equilibria computed using the basic formulation of ASPEQ neither require a mediator nor a communication. Furthermore, the assumption of public correlation, adopted in order to implement the CubeSupportedPC procedure, only requires the presence of a source of a (constant) uniformly distributed signal that has to be observed by all players only at certain repeated game stages.

We believe that the present paper raises a lot of interesting questions. In particular, the way of subdividing the space of payoff profiles into a set of hypercubes results in a rapidly increasing complexity of each iteration of our algorithm. We only know that the time of execution of our algorithm is always finite, but for lower approximation factors and higher discount factors, this time could be unacceptably long, especially if agents are supposed to solve dynamic games in real-time. There is a need of finding a way to withdraw non-equilibrium segments of the initial set of payoff profiles, by keeping the complexity of each iteration of the algorithm constant, or at least polynomial in a certain subset of input parameters.

Finally, one more interesting problem is the computation of subgame-perfect equilibria in the games of imperfect monitoring. This is the setting where the players do not directly observe the actions played by the opponents after each repeated game stage. What each player observes is a certain stochastic signal reflecting the played action profile. Fudenberg et al. [55–58] characterized equilibria in such settings with different assumptions about the form of this stochastic signal. However, to the best of our knowledge, there are no algorithmic computational methods capable of finding such equilibria.

References

[1] G. Mailath, L. Samuelson, Repeated Games and Reputations: Long-Run Relationships, Oxford University Press, USA, 2006.
[2] G. Szabó, G. Fath, Evolutionary games on graphs, Phys. Rep. 446 (4) (2007) 97–216.
[3] J. Mertens, A. Neyman, Stochastic games, Int. J. Game Theory 10 (2) (1981) 53–66.
[4] M. Perc, Coherence resonance in a spatial prisoner's dilemma game, New J. Phys. 8 (2) (2006) 22.
[5] M. Perc, J. Gómez-Gardeñes, A. Szolnoki, L.M. Floría, Y. Moreno, Evolutionary dynamics of group interactions on structured populations: a review, J. R. Soc. Interface 10 (80) (2013) 20120997.
[6] T. Wang, K. Huang, Z. Wang, X. Zheng, Impact of small groups with heterogeneous preference on behavioral evolution in population evacuation, PloS ONE 10 (3) (2015) e0121949.
[7] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
[8] X. Deng, Z. Wang, Q. Liu, Y. Deng, S. Mahadevan, A belief-based evolutionarily stable strategy, J. Theor. Biol. 361 (2014) 81–86.
[9] Z. Wang, A. Szolnoki, M. Perc, Optimal interdependence between networks for the evolution of cooperation, Sci. Rep. 3 (2013).
[10] Z. Wang, A. Szolnoki, M. Perc, Self-organization towards optimally interdependent networks by means of coevolution, New J. Phys. 16 (3) (2014) 033041.
[11] M. Perc, A. Szolnoki, Social diversity and promotion of cooperation in the spatial prisoner's dilemma game, Phys. Rev. E 77 (1) (2008) 011904.
[12] M. Perc, A. Szolnoki, Coevolutionary games mini review, BioSystems 99 (2) (2010) 109–125.
[13] X. Chen, L. Wang, Promotion of cooperation induced by appropriate payoff aspirations in a small-world networked game, Phys. Rev. E 77 (1) (2008) 017103.
[14] C. Daskalakis, P. Goldberg, C. Papadimitriou, The complexity of computing a Nash equilibrium, in: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC'06), ACM, New York, NY, 2006, pp. 71–78.
[15] K. Judd, S. Yeltekin, J. Conklin, Computing supergame equilibria, Econometrica 71 (4) (2003) 1239–1254.
[16] M. Littman, P. Stone, A polynomial-time Nash equilibrium algorithm for repeated games, Decis. Support Syst. 39 (1) (2005) 55–66.
[17] L.M. Dermed, C. Isbell, Solving stochastic games, in: Advances in Neural Information Processing Systems 22 (NIPS'09), 2009, pp. 1186–1194.
[18] L.M. Dermed, K. Narayan, C. Isbell, L. Weiss, Quick polytope approximation of all correlated equilibria in stochastic games, in: Proceedings of the Twenty-Sixth National Conference on Artificial Intelligence ((AAAI),'2011), 2015.
[19] O. Gossner, T. Tomala, Repeated games, in: Encyclopedia of Complexity and Systems Science, Springer, 2015.
[20] C. Borgs, J. Chayes, N. Immorlica, A. Kalai, V. Mirrokni, C. Papadimitriou, The myth of the folk theorem, in: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08), ACM, New York, NY, USA, 2008, pp. 365–372.
[21] G. Gottlob, G. Greco, F. Scarcello, Pure Nash equilibria: hard and easy games, J. Artif. Intell. Res. 24 (2005) 357–406.
[22] J. Webb, Game Theory: Decisions, Interaction and Evolution, Springer Verlag, 2007.
[23] M. Osborne, A. Rubinstein, A Course in Game Theory, MIT press, 1999.
[24] E. Ben-Porath, B. Peleg, On the folk theorem and finite automata, Center for Research in Mathematical Economics and Game Theory, Hebrew University, Res. Mem 77 (1987).
[25] E. Kalai, W. Stanford, Finite rationality and interpersonal complexity in repeated games, Econometrica 56 (2) (1988) 397–410.
[26] J. Renault, et al., The value of Markov chain games with lack of information on one side, Math. Oper. Res. 31 (3) (2006) 490–512.
[27] D. Abreu, D. Pearce, E. Stacchetti, Toward a theory of discounted repeated games with imperfect monitoring, Econometrica (1990) 1041–1063.
[28] M. Cronshaw, Algorithms for finding repeated game equilibria, Comput. Econ. 10 (2) (1997) 139–168.
[29] J. Levin, Repeated games I: perfect monitoring (2006) web.stanford.edu/~jdlevin/Econ%20286/Repeated%20Games%20I.pdf.
[30] M. Cronshaw, D. Luenberger, Strongly symmetric subgame perfect equilibria in infinitely repeated games with perfect monitoring and discounting, Games Econ. Behav. 6 (2) (1994) 220–237.
[31] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, first, John Wiley & Sons, Inc., New York, NY, USA, 1994.
[32] A. Saxena, P. Bonami, J. Lee, Disjunctive cuts for non-convex mixed integer quadratically constrained programs, Lect. Notes Comput. Sci. 5035 (2008) 17.
[33] IBM, Corp., IBM ILOG CPLEX Callable Library Version 12.1 C API Reference Manual, 2009, (http://www-01.ibm.com/software/integration/optimization/cplex/).
[34] ATEJI, OptimJ – a Java language extension for optimization, 2009, (http://www.ateji.com/optimj.html).
[35] J. Nash, Equilibrium points in n-person games, in: Proceedings of the National Academy of the USA, 36 (1), 1950.
[36] R. Graham, An efficient algorithm for determining the convex hull of a finite planar set, Inform. Proc. Lett. 1 (4) (1972) 132–133.
[37] J. Nash, The bargaining problem, Econometrica 18 (2) (1950) 155–162.
[38] D. Abreu, On the theory of infinitely repeated games with discounting, Econometrica 56 (2) (1988) 383–396.
[39] A. Burkov, Leveraging repeated games for solving complex multiagent decision problems, Dept CS and Software Eng., Laval University, 2010 Ph.D. thesis.
[40] L. Busch, Q. Wen, Perfect equilibria in a negotiation model, Econometrica: J. Economet. Soc. (1995) 545–565.
[41] J. Lee, H. Sabourian, Complexity and efficiency, Cambridge Working Paper in Economics No. 0419 (2005).
[42] J. Lee, H. Sabourian, Coase theorem, complexity and transaction costs, J. Econ. Theory 135 (1) (2007) 214–235.
[43] Y. Shoham, K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, New York, 2009.
[44] D. Abreu, A. Rubinstein, The structure of Nash equilibrium in repeated games with finite automata, Econometrica 56 (6) (1988) 1259–1281.
[45] A. Rubinstein, Finite automata play the repeated prisoner's dilemma, J. Econ. Theory 39 (1) (1986) 83–96.
[46] M. Piccione, A. Rubinstein, Finite automata play a repeated extensive game, J. Econ. Theory 61 (1993) 160–163.
[47] M. Piccione, Finite automata equilibria with discounting, J. Econ. Theory 56 (1) (1992) 180–193.
[48] D.O. Stahl, The graph of prisoners' dilemma supergame payoffs as a function of the discount factor, Game Econ. Behav. 3 (1991) 368–384.
[49] R. Jurca, B. Faltings, Obtaining reliable feedback for sanctioning reputation mechanisms, J. AI Res. (JAIR) 29 (2007) 391–419.
[50] P. Resnick, R. Zeckhauser, R. Friedman, K. Kuwabara, Reputation systems, Commun. ACM 43 (12) (2000) 45–48.
[51] J. Engle-Warnick, W.J. McCausland, J.H. Miller, The ghost in the machine: inferring machine-based strategies from observed behavior, Cahiers de recherche, Universite de Montreal, Departement de sciences economiques, 2004.
[52] J.R. Wright, K. Leyton-Brown, Beyond equilibrium: Predicting human behavior in normal-form games, in: Twenty-Fourth Conference of the Association for the Advancement of Artificial Intelligence (AAAI-10), 2010, pp. 901–907.
[53] R. Aumann, Correlated equilibrium as an expression of Bayesian rationality, Econometrica 55 (1) (1987) 1–18.
[54] C. Murray, G. Gordon, Multi-robots negotiation: approximating the set of sugame perfect equilibria in general sum-stochastic games, in: Advances in Neural Information Processing Systems (NIPS'07), 19, MIT Press, 2007.
[55] D. Fudenberg, D. Levine, An approximate folk theorem with imperfect private information, J. Econ. Theory 54 (1) (1991) 26–47.
[56] D. Fudenberg, D. Levine, E. Maskin, The folk theorem with imperfect public information, Econometrica 62 (5) (1994) 997–1039.
[57] J. Hrner, W. Olszewski, The folk theorem for games with private almost-perfect monitoring, Econometrica 74 (6) (2006) 1499–1544.
[58] O. Compte, Communication in repeated games with imperfect private monitoring, Econometrica 66 (3) (1998) 597–626.